

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第5636097号
(P5636097)

(45) 発行日 平成26年12月3日(2014.12.3)

(24) 登録日 平成26年10月24日(2014.10.24)

(51) Int. Cl.		F I	
G06F 12/10	(2006.01)	G06F 12/10	505Z
G06F 13/28	(2006.01)	G06F 12/10	505B
G06F 13/14	(2006.01)	G06F 12/10	555
		G06F 13/28	310M
		G06F 13/14	320A

請求項の数 12 (全 41 頁)

(21) 出願番号	特願2013-515734 (P2013-515734)	(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2013-537658 (P2013-537658A)		INTERNATIONAL BUSINESS MACHINES CORPORATION
(43) 公表日	平成25年10月3日 (2013.10.3)		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/EP2010/067043	(74) 代理人	100108501
(87) 国際公開番号	W02011/160722		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2日 (2013.8.2)		弁理士 太佐 種一
(31) 優先権主張番号	12/821, 170	(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】 入力/出力アドレスをメモリ・アドレスに変換するための方法、コンピュータ・システム、およびコンピュータ・プログラム

(57) 【特許請求の範囲】

【請求項1】

コンピューティング環境においてアドレスを変換するための方法であって、
変換されることになるアドレスをアダプタから取得することであって、前記アドレスは複数のビットを含み、前記複数のビットはビットの第1の部分およびビットの第2の部分を含む、取得すること、

許可されたアドレスの領域を示すアドレス領域値を受信すること、

少なくとも前記ビットの第1の部分および前記受信したアドレス領域値を使用して前記アドレスを妥当性検査することであって、前記ビットの第1の部分が前記アドレス領域値によって示される、0でない前記アドレスの領域内にある旨の決定を条件に処理の続行を許可する、前記検査すること、および、

前記アドレスを、前記コンピューティング環境のメモリにアクセスする際に使用可能なメモリ・アドレスに変換することであって、前記変換することが、前記ビットの第1の部分を無視し、前記変換を実行するためのアドレス変換テーブルの1つまたは複数のレベルからアドレス情報を取得するために前記ビットの第2の部分を使用する、変換すること、を含む、方法。

【請求項2】

前記ビットの第1の部分は前記アドレスの高位ビットを備え、前記ビットの第2の部分は前記アドレスの低位ビットを備え、前記低位ビットが、前記メモリ・アドレスを含む割り当てられたアドレス・スペースのサイズに基づいて決定される、請求項1に記載の方法

【請求項 3】

アドレス変換テーブルのレベルの数は、前記メモリ・アドレスを含む割り当てられたアドレス・スペースのサイズ、前記変換で使用されることになる1つまたは複数のアドレス変換テーブルのサイズ、および、前記メモリ・アドレスによってアクセスされるメモリの単位のサイズの少なくとも1つに基づくものである、請求項1に記載の方法。

【請求項 4】

前記変換することが、前記アドレスの変換に使用されることになるアドレス変換テーブルを選択することを含み、前記選択することは、前記変換で使用されるデバイス・テーブル・エントリ内でのポインタを使用する、請求項1に記載の方法。

10

【請求項 5】

前記方法が、前記デバイス・テーブル・エントリの位置を特定することをさらに含み、前記位置を特定することは、前記変換されることになるアドレスまたは前記アドレスの一部を含む要求を発行する前記アダプタの要求側識別子のうちの少なくとも1つを使用する、請求項4に記載の方法。

【請求項 6】

前記方法が、前記選択されたアドレス変換テーブルのフォーマットを決定することをさらに含み、前記決定することは、前記デバイス・テーブル・エントリのフォーマット・フィールドを使用する、請求項4に記載の方法。

【請求項 7】

前記アドレスを妥当性検査するために、前記アドレス領域が、前記デバイス・テーブル・エントリの基本フィールドおよび限界フィールドのうちの少なくとも1つの値によって画定される、請求項4に記載の方法。

20

【請求項 8】

前記アドレスを前記妥当性検査することが、
前記ビットの第1の部分がアドレス領域の対応するビットの領域内にある旨の決定に
応答して実行を許可することであって、前記アドレス領域の前記対応するビットがすべて0
ではない、許可すること、および
前記ビットの第1の部分がアドレス領域の対応するビットの領域内でない旨の決定に
応答して実行を許可しないことであって、前記アドレス領域の前記対応するビットがすべて
0ではない、許可しないこと、
を含む、請求項7に記載の方法。

30

【請求項 9】

前記アダプタが Peripheral Component Interconnect 機能を備える、請求項1に記載の方法。

【請求項 10】

コンピューティング環境においてアドレスを変換するためのコンピュータ・システムであって、

メモリと、

前記メモリと通信するプロセッサと、

変換されることになるアドレスをアダプタから取得するためのアドレス取得器であって、前記アドレスは複数のビットを含み、前記複数のビットはビットの第1の部分およびビットの第2の部分を含む、アドレス取得器と、

許可されたアドレスの領域を示すアドレス領域値を受信するための受信器と、

少なくとも前記ビットの第1の部分および前記受信したアドレス領域を使用して前記アドレスを妥当性検査するための妥当性検査器であって、前記ビットの第1の部分が前記アドレス領域値によって示される、0でない前記アドレスの領域内にある旨の決定を条件に処理の続行を許可する、前記妥当性検査器と、

40

前記アドレスを、前記コンピューティング環境のメモリにアクセスする際に使用可能なメモリ・アドレスに変換するための変換器であって、前記変換器が、前記ビットの第1の

50

部分を無視し、前記変換を実行するためのアドレス変換テーブルの1つまたは複数のレベルからアドレス情報を取得するために前記ビットの第2の部分を使用する、変換器と、を備える、コンピュータ・システム。

【請求項11】

請求項1から9のいずれか1項に記載の方法のすべてのステップをコンピュータに実行させるためのコンピュータ・プログラム。

【請求項12】

請求項1から9のいずれか1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るコンピュータ・システム。

【発明の詳細な説明】

10

【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アドレスの変換に関し、特に、入力/出力アドレスの、環境のシステム・メモリにアクセスする際に使用可能なメモリ・アドレスへの変換に関する。

【背景技術】

【0002】

システム・メモリは、読み取りおよび書き込みの要求によってアクセスされる。これらの要求は、中央処理ユニットならびにアダプタを含む、コンピューティング環境の様々な構成要素から発することができる。各要求は、システム・メモリにアクセスする際に使用されるアドレスを含む。しかしながらこのアドレスは、典型的には、システム・メモリ内の物理位置と1対1で対応している訳ではない。したがって、アドレス変換が実行される。

20

【0003】

アドレス変換は、システム・メモリにアクセスする際に直接使用することのできない形で提供されたアドレスを、システム・メモリ内の物理位置にアクセスする際に直接使用できる他の形に変換するために使用される。たとえば、中央処理ユニットによって提供される要求に含まれた仮想アドレスは、システム・メモリ内の実アドレスまたは絶対アドレスに変換される。他の例として、アダプタからの要求で提供される Peripheral Component Interconnect (PCI) アドレスは、システム・メモリ内の絶対アドレスに変換可能である。

30

【0004】

2008年7月10日付公告の、Greggによる「I/O Adapter LPAR Isolation In A Hypertransport Environment With Assigned Memory Space Indexing A TVT Via Unit Ids」という名称の米国公開第2008/0168208 A1号は、データ処理システム、およびシステム内の複数のI/Oアダプタを分離する方法について記載している。データ処理システムは、プロセッサのセット、ホスト・ブリッジ、ならびに、プロセッサのセットとホスト・ブリッジとを接続するシステム・バスも備えている。各I/OアダプタはそれぞれのIDを有し、I/OアダプタのIDのうちの1つまたは複数を含むコマンドを、ホスト・ブリッジに送信する。好ましい実施形態では、これらのIDはHyperTransport定義のユニットIDであり、I/Oアダプタによって発行されるコマンドは、I/OアダプタのユニットIDのうちの1つまたは複数を含むユニットIDを含む。ユニットIDはそれぞれ、固有の独立したシステム・メモリ・スペースを識別するように、TVTにインデックス付けするために使用される。

40

【0005】

2009年7月16日付公告の、Greiner等による「Dynamic Address Translation with Frame Management」という名称の米国公開第2009/0182966 A1号は、コンピュータ・システムの機械体

50

系に関して定義されたフレーム管理機能を開示している。一実施形態では、第1および第2の汎用レジスタを識別するフレーム管理命令が取得される。第1の汎用レジスタは、アクセス保護ビットおよびブロック・サイズ指示を備えたキー・フィールドを有するフレーム管理フィールドを含む。ブロック・サイズ指示が大きなブロックを示す場合、大きなデータ・ブロックのオペランド・アドレスが、第2の汎用レジスタから取得される。大きなデータ・ブロックは、複数の小さなブロックを有し、そのそれぞれが、複数のストレージ・キー・アクセス保護ビットを有する対応するストレージ・キーに関連付けられる。ブロック・サイズ指示が大きなブロックを示す場合、大きなブロック内の各小さなブロックの各対応するストレージ・キーのストレージ・キー・アクセス保護ビットには、キー・フィールドのアクセス保護ビットがセットされる。

10

【先行技術文献】

【特許文献】

【0006】

【特許文献1】米国公開第2008/0168208A1号

【特許文献2】米国公開第2009/0182966A1号

【特許文献3】米国特許証第6009261号

【特許文献4】米国特許証第5574873号

【特許文献5】米国特許証第5551013号

【特許文献6】米国特許証第6308255号

【特許文献7】米国特許証第6463582号

20

【特許文献8】米国特許証第5790825号

【発明の概要】

【発明が解決しようとする課題】

【0007】

コンピューティング環境においてアドレスを変換するためのコンピュータ・プログラム製品を提供することによって、従来技術の欠点が克服され、追加の利点が与えられる。

【課題を解決するための手段】

【0008】

コンピュータ・プログラム製品は、処理回路によって読み取り可能であり、方法を実行するために処理回路によって実行するための命令を格納する、コンピュータ読み取り可能ストレージ・メディアを含む。方法は、たとえば、変換されることになるアドレスをアダプタから取得することであって、アドレスは複数のビットを含み、複数のビットはビットの第1の部分およびビットの第2の部分を含む、取得すること、許可されたアドレスの領域を示すアドレス領域値を受信すること、少なくともビットの第1の部分および受信したアドレス領域を使用してアドレスを妥当性検査すること、ならびに、このアドレスを、コンピューティング環境のメモリにアクセスする際に使用可能なメモリ・アドレスに変換することであって、変換することは、ビットの第1の部分を無視し、変換を実行するためのアドレス変換テーブルの1つまたは複数のレベルからアドレス情報を取得するためにビットの第2の部分を使用する、変換すること、を含む。

30

【0009】

本明細書では、本発明の1つまたは複数の態様に関する方法およびシステムについても説明および主張する。

40

【0010】

追加の特徴および利点は、本発明の技法を通じて実現される。本明細書では、本発明の他の実施形態および態様について詳細に説明され、主張された発明の一部とみなされる。

【0011】

本発明の1つまたは複数の態様が特に注目され、明細書の終わりの特許請求の範囲で例として明確に主張される。本発明の前述および他の目的、特徴、および利点は、添付の図面に関連した以下の詳細な説明から明らかとなる。

【図面の簡単な説明】

50

【 0 0 1 2 】

【図 1】本発明の 1 つまたは複数の態様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】本発明の態様に従った、図 1 のシステム・メモリおよび I / O ハブのさらなる細部の一実施形態を示す図である。

【図 3】本発明の態様に従った、アダプタに関する DMA (ダイレクト・メモリ・アクセス) アドレス・スペースを登録するための論理の概要の一実施形態を示す図である。

【図 4】本発明の態様に従った、アダプタに関する DMA アドレス・スペースの登録の様々な細部の一実施形態を示す図である。

【図 5】本発明の態様に従った、DMA 動作を処理するための論理の一実施形態を示す図である。

10

【図 6】アドレスを変換するためおよびページにアクセスするために、アドレス変換テーブルにインデックス付けするためにアドレス全体が使用される場合に採用される、変換のレベルの一例を示す図である。

【図 7】本発明の態様に従った、アドレス変換テーブルにインデックス付けする際にアドレスの一部が無視される場合に採用される、変換のレベルの一例を示す図である。

【図 8】本発明の 1 つまたは複数の態様に従って使用可能な、様々な CPU DAT 適合フォーマットの例を示す図である。

【図 9】本発明の 1 つまたは複数の態様に従って使用可能な、様々な I / O 拡張アドレス変換フォーマットの例を示す図である。

20

【図 10】本発明の態様に従って使用される、PCI 機能制御修正命令の一実施形態を示す図である。

【図 11】本発明の態様に従った、図 10 の PCI 機能制御修正命令によって使用されるフィールドの一実施形態を示す図である。

【図 12】本発明の態様に従った、図 10 の PCI 機能制御修正命令によって使用される他のフィールドの一実施形態を示す図である。

【図 13】本発明の態様に従って使用される、機能情報ブロック (FIB) のコンテンツの一実施形態を示す図である。

【図 14】本発明の態様に従った、PCI 機能制御修正命令の論理の概要の一実施形態を示す図である。

30

【図 15】本発明の態様に従った、PCI 機能制御修正命令によって指定可能な、I / O アドレス変換パラメータ登録動作に関連付けられた論理の一実施形態を示す図である。

【図 16】本発明の態様に従った、PCI 機能制御修正命令によって指定可能な、I / O アドレス変換パラメータ登録解除動作に関連付けられた論理の一実施形態を示す図である。

【図 17】本発明の 1 つまたは複数の態様を組み込んだコンピュータ・プログラム製品の一実施形態を示す図である。

【図 18】本発明の 1 つまたは複数の態様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 19】本発明の 1 つまたは複数の態様を組み込んで使用するための、コンピュータ・システムの他の例を示す図である。

40

【図 20】本発明の 1 つまたは複数の態様を組み込んで使用するためのコンピュータ・ネットワークを備える、コンピュータ・システムの他の例を示す図である。

【図 21】本発明の 1 つまたは複数の態様を組み込んで使用するためのコンピュータ・システムの様々な要素の一実施形態を示す図である。

【図 22】本発明の 1 つまたは複数の態様を組み込んで使用するための、図 21 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 23】本発明の 1 つまたは複数の態様を組み込んで使用するための、図 21 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 24】本発明の 1 つまたは複数の態様を組み込んで使用するための、図 21 のコンピ

50

ユーザ・システムのロード／格納ユニットの一実施形態を示す図である。

【図25】本発明の1つまたは複数の態様を組み込んで使用するための、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図である。

【発明を実施するための形態】

【0013】

本発明の態様によれば、コンピューティング環境においてアドレスを変換するための機能が提供される。一例では、変換されるアドレスは、アダプタによって提供されるアドレス（本明細書では、入力／出力（I/O）アドレスと呼ばれる）であり、システム・メモリにアクセスする際に使用可能なアドレスに変換されるものである。アドレス変換を実行するために、いくつかのレベルのアドレス変換が使用され、このレベルの数は、たとえば、アダプタに割り当てられたメモリ・アドレス・スペースのサイズ、変換に使用される1つまたは複数のアドレス変換テーブルのサイズ、または、アクセスされることになるページ（またはメモリの他の単位）のサイズ、あるいはそれらすべてに基づくものである。

10

【0014】

変換されるアドレスは複数のビットを含み、一実施形態では、それらのビットの一部のみが、変換されたアドレスを取得するためにアドレス変換テーブルにインデックス付けするために使用される。他のビットは変換については無視される。たとえばアドレスは、（割り当てられたアドレス・スペースのサイズに基づいて）高位ビットおよび低位ビットを含む。この例では、ページ・テーブルを含むアドレス変換テーブルにインデックス付けするため、および、ページ自体にインデックス付けするために、低位ビットが使用される。高位ビットは、（値、すなわちゼロまたは非ゼロに関係なく）変換については無視され、アドレス変換テーブルにインデックス付けするためには使用されない。これにより、変換に使用されるアドレス変換テーブルのレベル数は減少する。（低位ビットおよび高位ビットの指示は、ビットがどのように番号付けされるかとは無関係である。）

20

【0015】

一例では、変換は、System z (R) サーバなどの企業クラス・サーバに最適な保護を提供しながら、実行される。例として、メモリにアクセスする際にはフル・アドレス（たとえば64ビット・アドレス全体）が使用されるが、変換にはアドレスの一部のみが使用される。変換については無視されるアドレスの他の部分は、変換に使用されるアドレスの少なくとも一部と共に、妥当性検査（たとえば領域チェック）に使用される。変換のためにアドレスの一部のみを使用することによって、アドレス変換ルックアップは最小限に抑えられる。これは、アドレス自体のサイズではなく、たとえばアダプタによるアクセスに登録されたDMAアドレス・スペースのサイズに基づいて、いくつかのレベルの変換を実行することによって達成される。

30

【0016】

本発明の1つまたは複数の態様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について、図1を参照しながら説明する。一例では、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・マシーニズ・コーポレーションによって提供されるSystem z (R) である。System z (R) は、インターナショナル・ビジネス・マシーニズ・コーポレーションによって提供されるz / Architecture (R) に基づくものである。z / Architecture (R) に関する詳細については、2009年2月、IBM (R) 発行の「z / Architecture Principles of Operation」と題するIBM出版物第SA22-7832-07号で説明されている。IBM (R)、System z (R)、およびz / Architecture (R) は、ニューヨーク州アーモンクのインターナショナル・ビジネス・マシーニズ・コーポレーションの登録商標である。本明細書で使用される他の名前は、インターナショナル・ビジネス・マシーニズ・コーポレーションまたは他の会社の登録商標、商標、または製品名とすることができる。

40

【0017】

一例では、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

50

システム・メモリ104(別名メイン・メモリ)に結合された、1つまたは複数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使用されるアドレスを含む、読み取りまたは書き込み要求を発行する。要求に含まれるアドレスは、通常、システム・メモリへのアクセスに直接使用することはできないため、システム・メモリへのアクセスに直接使用可能なアドレスに変換される。アドレスは、変換メカニズム(XLATE)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(DAT)を使用して、仮想アドレスから実または絶対アドレスへと変換される。

【0018】

変換済みアドレスを含む要求は、メモリ・コントローラ106によって受信される。一例では、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関する調停(arbitrate)およびメモリの一貫性の維持のために使用される。この調停はCPU102から受信される要求、ならびに1つまたは複数のアダプタ110から受信される要求に対して実行される。中央処理ユニットと同様に、アダプタは、システム・メモリにアクセスできるようにするための要求を発行する。

10

【0019】

一例では、アダプタ110は、Peripheral Component Interconnect(PCI)、あるいは1つまたは複数のPCI機能を含むPCI Express(PCIE)アダプタである。PCI機能は、システム・メモリへのアクセスを必要とする要求を発行する。要求は、1つまたは複数のスイッチ(たとえばPCIEスイッチ)114を介して入力/出力ハブ112(たとえばPCIハブ)へと経路指定される。一例では、入力/出力ハブは、1つまたは複数の状態機械を含むハードウェアからなる。

20

【0020】

本明細書で使用されるアダプタと言う用語は、任意のタイプのアダプタ(たとえば、ストレージ・アダプタ、ネットワーク・アダプタ、処理アダプタ、PCIアダプタ、暗号アダプタ、他のタイプの入力/出力アダプタなど)を含む。一実施形態では、アダプタは1つのアダプタ機能を含む。しかしながら、他の実施形態では、アダプタは複数のアダプタ機能を含むことができる。本発明の1つまたは複数の態様は、アダプタが1つのアダプタ機能または複数のアダプタ機能を含むかどうかにかかわらず、適用可能である。さらに、本明細書に提示された例では、特に注記されない限り、アダプタはアダプタ機能(たとえばPCI機能)と互換的に使用される。

30

【0021】

入力/出力ハブは、たとえば、スイッチからの要求を受信するルート・コンプレックス(root complex)116を含む。要求は、変換されることになる入力/出力アドレスを含むため、ルート・コンプレックスはアドレス変換および保護ユニット118にアドレスを提供する。このユニットは、以下でより詳細に説明するように、たとえばI/Oアドレスを、システム・メモリ104にアクセスするために直接使用可能なアドレスに変換する、ハードウェア・ユニットである。

【0022】

アダプタから開始された、変換済みアドレスを含む要求は、たとえばI/O-メモリ・バス(I/O-to-memory bus)120を介して、メモリ・コントローラ106へ提供される。メモリ・コントローラはその調停を実行し、要求を変換済みアドレスと共に適切な時点でシステム・メモリに転送する。

40

【0023】

システム・メモリおよび入力/出力ハブに関するさらなる詳細について、図2を参照しながら説明する。この実施形態では、メモリ・コントローラは示されていない。しかしながら、I/Oハブを直接、またはメモリ・コントローラを介して、システム・メモリに結合することができる。一例では、システム・メモリ104は1つまたは複数のアドレス・スペース200を含む。アドレス・スペースとは、特定のアダプタなどのコンピューティ

50

ング環境の特定の構成要素に割り当てられた、システム・メモリの特定部分である。一例では、アドレス・スペースには、アダプタによって開始されたダイレクト・メモリ・アクセス(DMA)によってアクセス可能であるため、本明細書の例では、アドレス・スペースはDMAアドレス・スペースと呼ばれる。しかしながら他の例では、アドレス・スペースにアクセスするためにダイレクト・メモリ・アクセスは使用されない。

【0024】

さらに、一例では、システム・メモリ104は、システム・メモリにアクセスする際に直接使用できないアドレスから直接使用可能なアドレスへと、アドレスを変換する際に使用される、アドレス変換テーブル202を含む。一実施形態では、DMAアドレス・スペースに割り当てられた1つまたは複数のアドレス変換テーブルが存在し、それらの1つまたは複数のアドレス変換テーブルは、たとえば、割り当て先のアドレス・スペースのサイズ、アドレス変換テーブル自体のサイズ、または、ページ(またはメモリの他の単位)のサイズ、あるいはそれらすべてに基づいて、構成される。

10

【0025】

一例では、アドレス変換テーブルの階層が存在する。たとえば図2に示されるように、IOATポインタ218(以下で説明)によってポイントされる第1レベル・テーブル202a(たとえばセグメント・テーブル)、および、第1レベル・テーブルのエントリ206aによってポイントされる第2の低レベル・テーブル202b(たとえばページ・テーブル)が存在する。特定の低レベル・テーブル202bを示す、特定のエントリ206aの位置を特定するため、テーブル202aにインデックス付けするために、受信アドレス204の1つまたは複数のビットが使用される。その後、そのテーブル内の特定のエントリ206bの位置を特定するために、アドレス204の1つまたは複数の他のビットが使用される。この例では、そのエントリは、正しいページの位置を特定するために使用されるアドレスを提供し、データ転送を実行するためにページ内の特定の位置208を特定するために、アドレス204内の追加のビットが使用される。すなわち、システム・メモリにアクセスするために直接使用可能なアドレスを提供するために、エントリ206b内のアドレス、および受信PCIアドレス204の選択されたビットが使用される。たとえば、直接使用可能なアドレスは、エントリ206b内のアドレスの高位ビット(たとえば、4Kページの例では、ビット63:12)と、受信PCIアドレスからの選択された低位ビット(たとえば、4Kページではビット11:0)との連結から、形成される。

20

30

【0026】

一例では、これは、特定のアダプタにDMAアドレス・スペースを割り当てるオペレーティング・システムである。この割り当ては、登録プロセスを介して実行され、これによって、そのアダプタに関するデバイス・テーブル・エントリ210が(たとえば、信頼できるソフトウェアを介して)初期化される。デバイス・テーブル・エントリは、I/Oハブ112内に配置されたデバイス・テーブル211内に配置される。たとえば、デバイス・テーブル211は、I/Oハブのアドレス変換および保護ユニット内に配置される。

【0027】

一例では、デバイス・テーブル・エントリ210は、以下のようないくつかのフィールドを含む。

40

フォーマット212:このフィールドは、たとえばアドレス変換・テーブルの上位レベル・テーブル(たとえばこの例では、第1レベル・テーブル)のアドレス変換フォーマット(レベルを含む)を含む、様々な情報を示すための複数のビットを含む。

ページ・サイズ213:このフィールドは、アクセスされることになるページ(またはメモリの他の単位)のサイズを示す。

PCIベース・アドレス214およびPCI限界216:これらの値は、DMAアドレス・スペースを定義し、受信したアドレス(たとえばPCIアドレス)が有効かどうかを検証するために使用される、領域を提供する。

IOAT(入力/出力アドレス変換)ポインタ218:このフィールドは、DMAアドレス・スペースに使用されるアドレス変換テーブルの最高レベルに対するポインタを含む

50

【0028】

他の実施形態では、DTEは、より多くの、より少ない、または、異なる情報を含むことができる。

【0029】

一実施形態では、特定の変換で使用されることになるデバイス・テーブル・エントリは、アダプタに関連付けられたPCI機能220（またはアドレスの一部、あるいはその両方）によって発行される要求内に配置される、要求側識別子（RID）を使用して配置される。要求側ID（たとえば、バス番号、デバイス番号、および機能番号を指定する、16ビット値）は、要求、ならびに、システム・メモリへのアクセスに使用されることになるI/Oアドレス（たとえば、64ビットPCIeアドレス）内に含まれる。RIDおよびI/Oアドレスを含む要求は、たとえば、インデックス値を提供するために使用される、たとえばスイッチ114を介して、コンテンツ・アドレス指定可能メモリ（CAM）230へと提供される。たとえば、CAMは複数のエントリを含み、各エントリはデバイス・テーブルへのインデックスに対応する。各CAMエントリは、RIDの値を含む。たとえば、受信したRIDが、CAM内のエントリに含まれる値と一致する場合、デバイス・テーブル・エントリの位置を特定するために、対応するデバイス・テーブル・インデックスが使用される。すなわち、CAMの出力は、デバイス・テーブル・エントリ210の位置を特定するために、デバイス・テーブル211へのインデック付けに使用される。一致しない場合、システム・メモリへのアクセスは実行されることなく、受信したパケットが廃棄される。（他の実施形態では、CAMまたは他のルックアップは不要であり、RIDはインデックスとして使用される。）

【0030】

その後、デバイス・テーブル・エントリ内のフィールドは、アドレスの妥当性検査およびアドレス変換テーブルの構成を保証するために使用される。たとえば、要求内の着信アドレスは、アドレスを提供した要求のRIDを使用して位置が特定されたデバイス・テーブル・エントリに格納された、PCIベース・アドレス214およびPCI限界216によって画定された、境界内にあることを保証するために、I/Oハブのハードウェア（たとえば、アドレス変換および保護ユニット）によってチェックされる。これによって、アドレスが、事前に登録された領域内にあること、および、それに対してアドレス変換テーブルが有効に構成されることが保証される。

【0031】

本発明の態様によれば、一実施形態において、I/Oアドレス（すなわち、I/Oサブシステムのアダプタまたは他の構成要素によって提供されるアドレス）を、システム・メモリ・アドレス（すなわち、システム・メモリへのアクセスに直接使用可能なアドレス）に変換するために、最初に、登録プロセスが実行される。この登録プロセスは、特定のアダプタまたはアダプタ機能などの、特定のアドレス・スペースを特定の要求側に登録する。この登録プロセスの概要の一例について、図3を参照しながら説明する。

【0032】

最初に、ステップ300で、システム・メモリに結合された中央処理ユニットのうちの1つ内で実行するオペレーティング・システムが、アダプタがアクセスするアドレス・スペースのサイズおよび位置を決定する。一例では、アドレス・スペースのサイズは、オペレーティング・システムによって設定されたPCIベース・アドレスおよびPCI限界によって決定される。オペレーティング・システムは、1つまたは複数の基準を使用してベースおよび限界を決定する。たとえば、オペレーティング・システムがPCIアドレスをCPU仮想アドレスに直接マッピングさせたい場合、ベースおよび限界はそのように設定される。他の例では、アダプタまたはオペレーティング・システム・イメージまたはその両方の間の追加の分離が望ましい場合、非重複および独立のアドレス・スペースを提供するように、使用されているアドレスが選択される。位置もオペレーティング・システムによって指定され、たとえばアダプタの特徴に基づく。

【 0 0 3 3 】

その後、ステップ 3 0 2 で、その D M A アドレス・スペースをカバーするために、1 つまたは複数のアドレス変換テーブルが作成される。例として、テーブルは C P U アドレス変換テーブルに適合可能であるか、または、入力 / 出力ハブによってサポートされる固有のフォーマットが提供可能である。一例では、作成は、テーブルを構築すること、およびテーブル・エントリ内に適切なアドレスを配置することを含む。例として、変換テーブルのうちの 1 つは 5 1 2 の 6 4 ビット・エントリを有する 4 K ページ・テーブルであり、各エントリは、割り当てられたアドレス・スペースに適合可能な 4 K ページ・アドレスを含む。

【 0 0 3 4 】

その後、図 4 を参照しながらより詳細に説明するように、ステップ 3 0 4 で、アダプタに対して D M A アドレス・スペースが登録される。この例では、アダプタについて 1 つの P C I 機能が存在し、したがってアダプタについて 1 つの要求側 I D が存在するものと想定される。この論理は、たとえば、オペレーティング・システム要求に回答して、システム・メモリに結合された中央処理ユニットによって実行される。

【 0 0 3 5 】

最初に、一実施形態では、ステップ 3 1 0 で、アダプタの要求側 I D に対応するように、使用可能なデバイス・テーブル・エントリが選択される。すなわち、デバイス・テーブル・エントリの位置を特定するために、要求側 I D が使用されることになる。

【 0 0 3 6 】

加えて、ステップ 3 1 2 で、P C I ベース・アドレスおよび P C I 限界がデバイス・テーブル・エントリに格納される。さらに、ステップ 3 1 4 で、最高レベル・アドレス変換テーブルのフォーマットも、ならびにステップ 3 1 6 で、最高レベル・アドレス変換テーブルをポイントするために使用される入力 / 出力アドレス変換 (I O A T) ポインタも、デバイス・テーブル・エントリ (たとえばフォーマット・フィールド) に格納される。これで登録プロセスが完了する。

【 0 0 3 7 】

登録の実行に回答して、D M A アドレス・スペースおよび対応するアドレス変換テーブル、ならびにデバイス・テーブル・エントリが使用できるようになる。システム・メモリにアクセスするために、アダプタなどの要求側によって発行される要求の処理に関する詳細について、図 5 を参照しながら説明する。以下で説明される処理は、I / O ハブによって実行される。一例では、これは、論理を実行するアドレス変換および保護ユニットである。一実施形態では、最初に、ステップ 4 0 0 で、入力 / 出力ハブで D M A 要求が受信される。たとえば P C I 機能は、たとえば P C I スイッチを介して P C I ハブに転送される要求を発行する。要求内の要求側 I D を使用して、ステップ 4 0 2 で、適切なデバイス・テーブル・エントリの位置が特定される。その後、照会 4 0 4 で、デバイス・テーブル・エントリが有効かどうか決定される。一例では、有効性は、エントリ自体の有効性ビットをチェックすることによって決定される。このビットは、たとえばオペレーティング・システムによる有効化機能要求の実行に回答して、設定される。有効化されている場合、ビットは、たとえば 1 (すなわち有効) に設定され、有効化されていない場合、ゼロ (すなわち無効) のままである。他の例では、登録プロセスが完了した時点でビットを設定することができる。

【 0 0 3 8 】

デバイス・テーブル・エントリが無効の場合、ステップ 4 0 5 でエラーが提示される。無効でない場合、照会 4 0 6 で、要求内に提供された P C I アドレスが、デバイス・テーブル・エントリに格納された P C I ベース・アドレスよりも小さいかどうか決定される。小さい場合、アドレスは有効領域外にあり、ステップ 4 0 7 でエラーが提供される。しかしながら、P C I アドレスがベース・アドレスよりも大きいかまたは等しい場合、照会 4 0 8 で、P C I アドレスがデバイス・テーブル・エントリ内の P C I 限界値よりも大きいかどうか、さらに決定される。P C I アドレスが限界よりも大きい場合、アドレスは有

10

20

30

40

50

効領域外にあるため、ステップ409で再度エラーが提示される。しかしながら、アドレスが有効範囲内にある場合、処理は続行される。

【0039】

一例では、ステップ410で、アドレス変換に使用されることになるアドレス内のPCIアドレス・ビットを決定するために、デバイス・テーブル・エントリ内に提供されるフォーマットが使用される。たとえば、フォーマットが、上位レベル・テーブルが4Kページを備えた第1レベル・テーブルであることを示す場合、第1レベル・テーブルにインデックス付けするためにアドレスのビット29:21が使用され、ページ・テーブルにインデックス付けするためにビット20:12が使用され、4Kページにインデックス付けするためにビット11:0が使用される。使用されるビットは、所与のサイズ・ページまたはテーブルにインデックス付けするためにどれだけのビットが必要であるかに依存する。たとえば、バイト・レベル・アドレス指定を伴う4Kページの場合、4096バイトのアドレス指定には12ビットが使用され、それぞれ8バイト、512エントリを伴う4Kページの場合、512エントリのアドレス指定には9ビットが使用される、などである。

10

【0040】

次に、ステップ412で、PCIハブは、適切なアドレス変換テーブル・エントリをフェッチする。たとえば、最初に、デバイス・テーブル・エントリのIOATポインタを使用して、最高レベル変換テーブルの位置が特定される。その後、そのテーブル内の特定のエントリの位置を特定するために、アドレスのビット(変換ではなく有効性に使用される高位ビットの後:たとえば上記の例ではビット29:21)が使用される。

20

【0041】

その後、照会414で、たとえば、位置が特定されたアドレス変換エントリが正しいフォーマットを有するかどうかに関して、デバイス・テーブル・エントリに提供されたフォーマットに基づいて決定される。たとえば、デバイス・テーブル・エントリ内のフォーマットが、アドレス変換エントリ内に示されるフォーマットと比較される。等しい場合、デバイス・テーブル・エントリに内のフォーマットは有効である。等しくない場合、ステップ415でエラーが提供され、そうでなければ、照会416で、これが処理される最後のテーブルであるかどうかの決定によって処理が続行される。すなわち、実または絶対アドレスを取得するために必要な他のアドレス変換テーブルが存在するかどうか、または、最低レベルのテーブル・エントリの位置が特定されたかどうかに関して、決定される。これは、提供されたフォーマットおよびすでに処理されているテーブルのサイズに基づいて決定される。最後のテーブルでない場合、ステップ412で処理が続行される。そうでなければ、I/Oハブはステップ418で、変換済みのアドレスでデータのフェッチまたは格納を実行可能にするための処理を続行する。一例では、I/Oハブは、変換済みアドレスをメモリ・コントローラに転送し、これは変換済みアドレスによって指定されたDMA位置でデータをフェッチまたは格納するために、アドレスを使用する。

30

【0042】

前述のように、本発明の態様によれば、変換のレベル数、したがってさらに変換を実行するために必要なフェッチ数が減少する。これは、たとえば、変換中にアドレスの高位ビットを無視すること、および、たとえば、アダプタに割り当てられたDMAアドレス・スペースのサイズに基づく、変換テーブルをトラバースするために低位ビットのみを使用することによって、達成される。以下の例で、部分アドレスの使用対フル・アドレスの使用について示される。

40

【0043】

最初に、図6を参照すると、アドレス変換/メモリ・アクセスでアドレス全体が使用される例が示されている。この従来技法では、ページ・テーブルを含む6レベルの変換テーブルが必要である。IOATポインタによって、最高レベル・テーブル(たとえばこの例では、第5レベル・テーブル)の開始がポイントされ、その後、テーブル内のエントリの位置を特定するために、PCIアドレスのビットが使用される。各変換テーブル・エントリは、低レベル変換テーブルの始まりまたはページをポイントする(たとえば、第5レベ

50

ル・テーブル内のエントリは第4レベル・テーブルの始まりをポイントする)。

【0044】

この例では、DMAアドレス・スペース(DMAAS)は6Mサイズであり、各テーブルは最大512の8バイト・エントリを有する4Kバイトである(アドレスのサイズに基づいて、128エントリを有する第5レベル・テーブルを除く)。たとえばアドレスは、64ビット: FFFF C000 0009 C600である。第5レベル・テーブルの始まりはI/OATポインタによってポイントされ、第4レベル・テーブルの始まりの位置を特定するために第5レベル・テーブルにインデックス付けするためにPCIアドレスのビット63:57が使用され、第3レベル・テーブルの始まりの位置を特定するために第4レベル・テーブルにインデックス付けするためにPCIアドレスのビット56:48が使用され、第2レベル・テーブルの始まりの位置を特定するために第3レベル・テーブルにインデックス付けするためにビット47:39が使用され、第1レベル・テーブルの始まりの位置を特定するために第2レベル・テーブルにインデックス付けするためにビット38:30が使用され、ページ・テーブルの始まりの位置を特定するために第1レベル・テーブルにインデックス付けするためにビット29:21が使用され、ページの始まりの位置を特定するためにページ・テーブルにインデックス付けするためにビット20:12が使用され、4Kページ内のエントリの位置を特定するためにビット11:0が使用される。

10

【0045】

これは、アドレス・スペースが同じサイズ(たとえば6M)であり、アドレスが同じであるが、変換技法が変換中にアドレス・ビットのいくつかを無視する、図7の例とは対照的である。この例では、アドレスのビット63:30は変換について無視される。I/OATポインタは、第1レベル・テーブルの始まりをポイントし、ページ・テーブルの始まりの位置を特定するために第1レベル・テーブルにインデックス付けするためにビットPCIアドレスのビット29:21が使用され、ページの始まりの位置を特定するために適切なページ・テーブルにインデックス付けするためにビット20:12が使用され、4Kページにインデックス付けするためにビット11:0が使用される。

20

【0046】

図に示されるように、第1レベル・テーブル500は3つのエントリ502を含み、それぞれが3つのページ・テーブル504のうちの1つにアドレスを提供する。したがって、必要なページ・テーブルの数、したがって他のレベルのテーブルの数は、たとえばDMAアドレス・スペースのサイズ、変換テーブルのサイズ、またはページのサイズ、あるいはそれらすべてに依存する。この例では、DMAアドレス・スペースは6Mであり、各ページ・テーブルは4Kで、512までのエントリを有する。したがって各ページ・テーブルは2Mまでのメモリをマッピングすることができる(4K%512エントリ)。したがって、6Mのアドレス・スペースには3つのページ・テーブルが必要である。第1レベル・テーブルは、各ページ・テーブルについて1つの、3つのエントリを保持することが可能であるため、この例では、他のレベルのアドレス変換テーブルは不要である。

30

【0047】

他の実施形態では、アドレス変換に使用される異なるフォーマットのアドレス変換テーブルが存在可能であり、フォーマット内に変形が存在可能である。したがって、特定のテーブルまたはページにインデックス付けするために使用されるビットに、変形が存在する。これらの例のいくつかについて、図8および図9を参照しながら説明する。

40

【0048】

たとえば、1つのフォーマットは、変換テーブルがCPU DAT変換に使用される変換テーブルに適合する、CPU DAT適合フォーマットである。様々なCPU DAT適合フォーマットが存在可能であり、その例について図8を参照しながら説明する。図に示されるように、例として、一方のCPU DAT適合フォーマットは4KページCPU DAT適合フォーマット550であり、他方は1MページCPU DAT適合フォーマット552である。示されたビット数は、そのページまたはテーブルにインデックス付け

50

するため（またはそうでなければ、そのページまたはテーブル内のエントリの位置を特定するため）に使用されるアドレス・ビット数である。たとえば、P C Iアドレスの12ビット554は、4 Kページ556内へのバイト・オフセットとして使用され、8ビット558はページ・テーブル560内へのインデックスとして使用され、11ビット562はセグメント・テーブル564内へのインデックスとして使用される、などである。指定されたアドレス変換テーブルの下で位置が特定されるのは、そのアドレス変換テーブルによってサポートされるアドレス・スペースの最大サイズである。たとえば、ページ・テーブル560は1 MのDMAアドレス・スペースをサポートし、セグメント・テーブル564は2 GのDMAアドレス・スペースをサポートする、などである。この図、ならびに図9では、K = キロバイト、M = メガバイト、G = ギガバイト、T = テラバイト、P = ペタバイト、およびE = エクサバイトである。

10

【0049】

図に示されるように、ページのサイズが大きくなるほど、変換テーブルのレベル数は減少する。たとえば4 Kページ556の場合、ページ・テーブルが必要であるが、1 Mページには必要ない。他の例および変形が可能である。

【0050】

他のアドレス変換フォーマットは、拡張アドレス変換テーブルが使用されるI/O拡張アドレス変換フォーマットである。I/O拡張アドレス変換フォーマットの様々な例が、図9に示される。たとえば、4 Kページを伴う4 Kアドレス変換テーブル570、4 Kページを伴う1 Mアドレス変換テーブル572、および1 Mページを伴う1 Mアドレス変換

20

【0051】

1つの特定の実装では、アダプタへのDMAアドレス・スペースの登録を実行するために、P C I機能制御修正(M P F C)命令と呼ばれる命令が使用される。たとえば、オペレーティング・システムは、どのアドレス変換フォーマットを使用したいかを決定し、そのフォーマットに対するアドレス変換テーブルを構築した後、命令のオペランドとしてそのフォーマットが含まれるM P F C命令を発行する。一例では、命令のフォーマットおよび他のオペランドが、命令のオペランドである機能情報ブロック(以下で説明)に含まれる。次にこの機能情報ブロックは、D T E、および一実施形態では、オプションで、アダプタの動作パラメータを含む機能テーブル・エントリ(F T E)を、更新するために使用される。

30

【0052】

この命令に関する細部の一実施形態および、特に登録プロセスについて、図10~図16を参照しながら説明する。図10を参照すると、P C I機能制御修正命令600は、たとえばP C I機能制御修正命令を示す命令コード602と、動作パラメータが確立されているアダプタ機能に関する様々な情報が含まれる位置を指定する第1のフィールド604と、P C I機能情報ブロック(F I B)のフェッチ元である位置を指定する第2のフィールド606と、を含む。フィールド1および2によって指定された位置のコンテンツについて、以下でさらに説明する。

40

【0053】

一実施形態では、フィールド1は様々な情報を含む汎用レジスタを指定する。図11に示されるように、レジスタのコンテンツは、たとえば、どの修正命令が実行されているかの代わりに、アダプタ機能のハンドルを識別する機能ハンドル610と、機能ハンドルによって指定されたアダプタ機能に関連付けられたシステム・メモリ内のアドレス・スペースを指定するアドレス・スペース612と、アダプタ機能に関して実行される動作を指定

50

する動作制御 6 1 4 と、事前定義されたコードによって命令が完了した場合に命令に関する状況を提供する場合 6 1 6 と、を含む。

【 0 0 5 4 】

－実施形態では、機能ハンドルは、たとえばハンドルが実行可能かどうかを示す実行可能インジケータと、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、機能テーブルにインデックス付けするために使用可能である）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、機能テーブル内の機能テーブル・エントリ（F T E）の位置を特定するために使用される。各機能テーブル・エントリは、動作パラメータまたはそのアダプタ機能に関連付けられた他の情報あるいはその両方を含む。一例では、機能テーブル・エントリは、以下を含む。

10

インスタンス番号：このフィールドは、機能テーブル・エントリに関連付けられたアダプタ機能ハンドルの特定のインスタンスを含む。

デバイス・テーブル・エントリ（D T E）インデックス 1 ... n：1 つまたは複数のデバイス・テーブル・インデックスが存在可能であり、各インデックスはデバイス・テーブル・エントリ（D T E）の位置を特定するためのデバイス・テーブルへのインデックスである。アダプタ機能につき 1 つまたは複数のデバイス・テーブル・エントリが存在し、各エントリは、アダプタ機能の要求（たとえば、DMA 要求、MSI 要求）を処理するために使用される情報、およびアダプタ機能に関連付けられた要求に関する情報（たとえば PCI 命令）を含む、そのアダプタ機能に関連付けられた情報を含む。各デバイス・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・スペースに関連付けられる。アダプタ機能は、アダプタ機能に割り当てられたシステム・メモリ内の 1 つまたは複数のアドレス・スペースを有することができる。

20

【 0 0 5 5 】

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。

永続的エラー状態インジケータ：このフィールドは、アダプタ機能が永続的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。

30

許可インジケータ：このフィールドは、アダプタ機能の制御を試行しているオペレーティング・システムが、それを実行する権限を有するかどうかを示す。

実行可能インジケータ：このフィールドは、アダプタ機能が実行可能であるかどうかを示す（たとえば、1 = 実行可能、0 = 実行不可）。

要求側識別子（R I D）：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、および機能番号を含む。

【 0 0 5 6 】

－一例では、このフィールドはアダプタ機能の構成スペースのアクセスに使用される。（アダプタのメモリは、たとえば構成スペース、I/Oスペース、または、1 つまたは複数のメモリ・スペース、あるいはそれらすべてを含む、アダプタ・スペースとして定義可能である。）一例では、構成スペースは、オペレーティング・システム（または他の構成）によってアダプタ機能に発行された命令内の構成スペースを指定することによってアクセス可能である。命令内に指定されるのは、構成スペース内へのオフセット、および、R I Dを含む適切な機能テーブル・エントリの位置を特定するために使用される機能ハンドルである。ファームウェアは命令を受信し、これが構成スペース用であるかどうかを決定する。したがって、I/Oハブへの要求を生成するために R I D を使用し、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。アダプタ機能の位置は R I D に基づき、オフセットはアダプタ機能の構成スペース内へのオフセットを指定する。

40

【 0 0 5 7 】

本明細書で使用される場合、ファームウェアは、たとえばプロセッサのマикроコード

50

、ミリコード、またはマクロコード、あるいはそれらすべてを含む。これはたとえば、ハードウェア・レベル命令、または、高位レベル機械コードの実装で使用されるデータ構造、あるいはその両方を含む。一実施形態では、これは、たとえば、通常、基礎となるハードウェア特有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、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送達されるプロプライエタリ・コードを含む。

【 0 0 5 8 】

ベース・アドレス・レジスタ (BAR) (1 から n) : このフィールドは、最初に指定されたアダプタ機能に関連付けられ、その値もアダプタ機能に関連付けられたベース・アドレス・レジスタ内に格納される、 $BAR_0 \sim BAR_n$ として指定された、複数の符号なし整数を含む。例として、各 BAR は、アダプタ機能内のメモリ・スペースまたは I/O スペースの開始アドレスを指定し、これが 64 または 32 ビットのメモリ・スペースであるか、あるいは 32 ビットの I/O スペースであるかの、アドレス・スペースのタイプも示す。

10

一例では、これは、アダプタ機能のメモリ・スペースまたは I/O スペースあるいはその両方にアクセスするために使用される。たとえば、アダプタ機能にアクセスするための命令内に提供されるオフセットは、アダプタ機能にアクセスするために使用されることになるアドレスを取得するために命令内に指定されたアドレス・スペースに関連付けられたベース・アドレス・レジスタ内の値に追加される。命令内に提供されるアドレス・スペース識別子は、アクセスされることになるアダプタ機能内のアドレス・スペースと、使用されることになる対応する BAR とを識別する。

20

サイズ 1 ... n : このフィールドは、 $SIZE_0 \sim SIZE_n$ として指定された、複数の符号なし整数を含む。サイズ・フィールドの値は、非ゼロの場合、以前に説明された BAR に対応する各エントリを備えた各アドレス・スペースのサイズを表す。

【 0 0 5 9 】

BAR およびサイズに関する詳細について、以下で説明する。

1 . BAR がアダプタ機能に関して実装されない場合、BAR フィールドおよびその対応するサイズ・フィールドはどちらのゼロとして格納される。

2 . BAR フィールドが I/O アドレス・スペースまたは 32 ビット・メモリ・アドレス・スペース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・スペースのサイズを表す。

30

3 . BAR フィールドが 64 ビット・メモリ・アドレス・スペースを表す場合、

a . BAR_n フィールドは最下位アドレス・ビットを表す。

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

c . 対応する $SIZE_n$ フィールドは、非ゼロであり、アドレス・スペースのサイズを表す。

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

【 0 0 6 0 】

内部ルーティング情報 : この情報は、アダプタへの特定のルーティングを実行するために使用される。例として、たとえばこれは、ノード、プロセッサ・チップ、およびハブ・アドレス指定情報を含む。

40

【 0 0 6 1 】

状況指示 : これは、ロード / 格納動作が阻止されているか、またはアダプタがエラー状態にあるかの指示、ならびに他の指示を提供する。

【 0 0 6 2 】

一例では、ビジー・インジケータ、永続的エラー状態インジケータ、および回復開始インジケータは、ファームウェアによって実行されるモニタリングに基づいて設定される。さらに、許可インジケータは、たとえばポリシーに基づいて設定され、BAR 情報は、プロセッサ (たと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く。他のフィールドは、構成、初期化、またはイベント、あるいはそれら

50

すべてに基づいて、設定可能である。他の実施形態では、機能テーブル・エントリは、より多い、より少ない、または異なる情報を、含むことができる。含まれる情報は、アダプタ機能によってサポートされるか、またはそれに対して実行可能化される、動作に依存することができる。

【 0 0 6 3 】

一例では、図 1 2 を参照すると、フィールド 2 は、関連付けられたアダプタ機能に関する情報を含む、P C I 機能情報ブロック (F I B) の論理アドレス 6 2 0 を指定する。機能情報ブロックは、アダプタ機能に関連付けられた、デバイス・テーブル・エントリまたは機能テーブル・エントリ (または他の位置)、あるいはその両方を更新するために使用される。情報は、アダプタの初期化時または構成時、あるいはその両方で、または特定のイベントにตอบสนองして、あるいはその両方で、F I B に格納される。

10

【 0 0 6 4 】

機能情報ブロック (F I B) に関する他の詳細について、図 1 3 を参照しながら説明する。一実施形態では、機能情報ブロック 6 5 0 は、以下のフィールドを含む。

フォーマット 6 5 1 : このフィールドは、F I B のフォーマットを指定する。

遮断制御 6 5 2 : このフィールドは、ページ可能 (pageable) モード・ゲストによる特有の命令のゲスト実行の結果、命令の遮断が生じるかどうかを示すために使用される。

エラー指示 6 5 4 : このフィールドは、ダイレクト・メモリ・アクセスおよびアダプタ割り込みに関するエラー状態指示を含む。ビットが設定された場合 (たとえば 1)、アダプタ機能に関するダイレクト・メモリ・アクセスまたはアダプタ割り込みの実行中に、1 つまたは複数のエラーが検出されている。

20

ロード / 格納阻止 (blocked) 6 5 6 : このフィールドは、ロード / 格納動作が阻止されるかどうかを示す。

P C I 機能有効 6 5 8 : このフィールドは、アダプタ機能に関する実行可能化制御を含む。ビットが設定された場合 (たとえば 1)、アダプタ機能は I / O 動作に対して実行可能化されるものとみなされる。

アドレス・スペース登録済み 6 6 0 : このフィールドは、アダプタ機能に関するダイレクト・メモリ・アクセス実行可能化制御を含む。フィールドが設定された場合 (たとえば 1)、ダイレクト・メモリ・アクセスは実行可能化される。

ページ・サイズ 6 6 1 : このフィールドは、D M A メモリ・アクセスによってアクセスされることになるページのサイズまたは他のメモリ単位を示す。

30

P C I ベース・アドレス (P B A) 6 6 2 : このフィールドは、アダプタ機能に割り当てられるシステム・メモリ内のアドレス・スペースに関するベース・アドレスである。これは、アダプタ機能が、指定された D M A アドレス・スペースへのダイレクト・メモリ・アクセスに使用可能である、最下位仮想アドレスを表す。

P C I アドレス限界 (P A L) 6 6 4 : このフィールドは、アダプタ機能が、指定された D M A アドレス・スペース内でアクセス可能である、最高位仮想アドレスを表す。

入力 / 出力アドレス変換ポインタ (I O A T) 6 6 6 : 入力 / 出力アドレス変換ポインタは、P C I 仮想アドレス変換によって使用される任意の変換テーブルのうちの第 1 を指定するか、または、変換の結果であるストレージのフレームの絶対アドレスを直接指定することができる。

40

割り込みサブクラス (I S C) 6 6 8 : このフィールドは、アダプタ機能に関するアダプタ割り込みを提示するために使用される、割り込みサブクラスを含む。

割り込みの数 (N O I) 6 7 0 : このフィールドは、アダプタ機能用に受け入れられる別個の割り込みコードの数を指定する。さらにこのフィールドは、アダプタ割り込みビット・ベクトル・アドレスおよびアダプタ割り込みビット・ベクトル・オフセット・フィールドによって指定された、アダプタ割り込みビット・ベクトルのサイズを、ビット単位でも定義する。

アダプタ割り込みビット・ベクトル・アドレス (A I B V) 6 7 2 : このフィールドは、アダプタ機能に関するアダプタ割り込みビット・ベクトルのアドレスを指定する。こ

50

のベクトルは割り込み処理で使用される。

アダプタ割り込みビット・ベクトル・オフセット 674：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・ビットのオフセットを指定する。

アダプタ割り込み要約ビット・アドレス (AISB) 676：このフィールドは、オプションで、割り込み処理で使用される、アダプタ割り込み要約ビットを指定するアドレスを提供する。

アダプタ割り込み要約ビット・オフセット 678：このフィールドは、アダプタ割り込み要約ビット・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) 680：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ために使用される、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 682：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。

要約ビット通知制御 684：このフィールドは、使用されている要約ビット・ベクトルが存在するかどうかを示す。

命令認証トークン 686：このフィールドは、ページ可能ストレージ・モード・ゲストがPCI命令の実行を認証されているかどうかを決定するために使用される。

【0065】

一例では、`z / Architecture (R)`において、ページ可能ゲストは、レベル2の解釈で、解釈実行開始 (SIE) 命令を介して解釈的に (interpretively) 実行される。たとえば、論理区画 (LPAR) ハイパーバイザは、物理的な固定メモリにおける論理区画を開始するために、SIE命令を実行する。`z / VM (R)`がその論理区画におけるオペレーティング・システムである場合、その $V = V$ (仮想) ストレージにおいてそのゲスト (仮想) 機会を実行するために、SIE命令を発行する。したがって、LPAR ハイパーバイザはレベル1のSIEを使用し、`z / VM (R)` ハイパーバイザはレベル2のSIEを使用し、さらに

アドレス変換フォーマット 687：このフィールドは、変換で使用されることになる最高レベル変換テーブルのアドレス変換のために選択されたフォーマット (たとえば、最高レベル・テーブル (たとえば、セグメント・テーブル、領域第3など) の指示) を示す。

【0066】

PCI機能制御修正命令内に指定された機能情報ブロックは、選択されたデバイス・テーブル・エントリ、機能テーブル・エントリ、または、命令内に指定されたアダプタ機能に関連付けられた他のファームウェア制御、あるいはそれらすべてを、修正するために使用される。デバイス・テーブル・エントリ、機能テーブル・エントリ、または他のファームウェア制御、あるいはそれらすべてを修正することによって、あるサービスがアダプタに提供される。これらのサービスは、たとえばアダプタ割り込み、アドレス変換、エラー状態リセット、ロード/格納阻止リセット、機能測定パラメータ設定、および遮断制御設定を含む。

【0067】

PCI機能制御修正命令に関連付けられた論理の一実施形態について、図14を参照しながら説明する。一例では、この命令は、オペレーティング・システム (または他の構成) によって発行され、オペレーティング・システムを実行しているプロセッサ (たとえばファームウェア) によって実行される。本明細書の例では、命令およびアダプタ機能はPCIベースである。しかしながら、他の例では、異なるアダプタ・アーキテクチャおよび対応する命令が使用可能である。

【0068】

一例では、オペレーティング・システムは、PCI機能ハンドル、DMAアドレス・スペース識別子、動作制御、および機能情報ブロックのアドレスの、オペランドを、(たとえば、命令によって指定された1つまたは複数のレジスタにおいて) 命令に提供する。

【0069】

10

20

30

40

50

図14を参照すると、最初に、照会700で、PCI機能制御修正命令を可能にする機構がインストールされるかどうかについて決定される。この決定は、たとえば制御ブロックに格納されたインジケータをチェックすることによって実行される。機構がインストールされない場合、ステップ702で例外条件が提供される。インストールされる場合、照会704で、ページ可能ストレージ・モード・ゲスト（または他のゲスト）によって命令が発行されたかどうか決定される。発行された場合、ステップ706でホスト・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ことになる。

【0070】

あるいは、照会708で、オペランドのうちの一つまたは複数位置合わせされるかどうかについて決定される。たとえば、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決定される。一例では、これはオプションである。オペランドが位置合わせされない場合、ステップ710で例外条件が提供される。位置合わせされる場合、照会712で、機能情報ブロックにアクセス可能であるかどうかについて決定される。アクセス可能でない場合、ステップ714で例外条件が提供される。アクセス可能である場合、照会716で、PCI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実行可能であるかどうかについて決定される。一例では、この決定は、ハンドル内の実行可能インジケータをチェックすることによって実行される。ハンドルが実行可能でない場合、ステップ718で例外条件が提供される。

【0071】

ハンドルが実行可能である場合、ステップ720で、機能テーブル・エントリの位置を特定するためにハンドルが使用される。すなわち、動作パラメータが確立されることになるアダプタ機能に対応する機能テーブル・エントリの位置を特定するために、ハンドルの少なくとも一部が機能テーブルへのインデックスとして使用される。

【0072】

照会722で、機能テーブル・エントリが見つかったかどうかについて決定される。見つからなかった場合、ステップ724で例外条件が提供される。見つかった場合、照会726で、命令を発行している構成がゲストであれば、ステップ728で例外条件（たとえばホストに対する遮断）が提供される。この照会は、指定されていれば、構成がゲストでない場合、または他の認証がチェックできる場合、無視することができる。

【0073】

照会730で、機能が実行可能であるかどうかについて決定される。一例では、この決定は、機能テーブル・エントリ内の実行可能インジケータをチェックすることによって実行される。実行可能でない場合、ステップ732で例外条件が提供される。

【0074】

機能が実行可能である場合、照会734で、回復がアクティブであるかどうかについて決定される。機能テーブル・エントリ内の回復インジケータによって決定された回復がアクティブである場合、ステップ736で例外条件が提供される。しかしながら回復がアクティブでない場合、照会738で、機能がビジーであるかどうかについてさらに決定される。この決定は、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って実行される。機能がビジーである場合、ステップ740でビジー条件が提供される。このビジー条件で、ドロップする代わりに命令を再試行することができる。

【0075】

機能がビジーでない場合、照会742で、機能情報ブロック・フォーマットが有効であるかどうかについて、さらに決定される。たとえば、このフォーマットがシステムによってサポートされているかどうかを決定するために、FIBのフォーマット・フィールドがチェックされる。無効である場合、ステップ744で例外条件が提供される。機能情報ブロック・フォーマットが有効である場合、照会746で、命令のオペランド内に指定された動作制御が有効であるかどうかについて、さらに決定される。すなわち、動作制御が、この命令に関して指定された動作制御のうちの一つであるかどうか、である。無効である場合、ステップ748で例外条件が提供される。しかしながら動作制御が有効である場合

10

20

30

40

50

、指定されている特定の動作制御で処理が続行する。

【0076】

指定可能な1つの動作制御は、アダプタに関するアドレス変換を制御する際に使用される、I/Oアドレス変換パラメータ登録動作である。この動作では、I/Oアドレス変換に関連するPCI機能パラメータが、DTE、FTE、または、FIBの適切なパラメータからの他の位置、あるいはそれらすべてで設定され、これは命令に対するオペランドである。これらのパラメータは、たとえば、PCIベース・アドレス、PCIアドレス限界（別名、PCI限界または限界）、アドレス変換フォーマット、ページ・サイズ、および、I/Oアドレス変換ポインタを含み、これらはこの動作に対するオペランドである。DMA開始アドレス(SDMA)およびDMA終了アドレス(EDMA)を含む暗黙(implied)オペランドも存在し、これらは、命令を実行しているプロセッサがアクセス可能な位置に格納される。

10

【0077】

I/Oアドレス変換に関する動作パラメータを確立するための論理の一実施形態について、図15を参照しながら説明する。最初に、照会800で、FIB内のPCIベース・アドレスがFIB内のPCI限界よりも大きいかどうかについて決定される。ベース・アドレスと限界とのこの比較で、ベース・アドレスの方が限界よりも大きいものとして示された場合、ステップ802で例外条件が認識される。しかしながら、ベース・アドレスの方が限界よりも小さいかまたは等しい場合、照会804で、アドレス変換フォーマットおよびページ・サイズが有効であるかどうかについてさらに決定される。これらが無効である場合、ステップ806で例外条件が提供される。しかしながら、これらが有効である場合、照会808で、(ベース・アドレスおよび限界に基づいて)アドレス・スペースのサイズが変換容量を超えているかどうかについて決定される。一例では、上位レベル・テーブルのフォーマットに基づいて、アドレス・スペースのサイズと、可能な最大のアドレス変換容量とが比較される。たとえば、上位レベル・テーブルがDAT適合セグメント・テーブルである場合、最大変換容量は2ギガバイトである。

20

【0078】

アドレス・スペースのサイズが変換容量を超える場合、ステップ810で例外条件が提供される。超えない場合、照会812で、ベース・アドレスがDMA開始アドレスよりも小さいかどうかについてさらに決定される。小さい場合、ステップ814で例外条件が提供される。小さくない場合、照会816で、アドレス限界がDMA終了アドレスよりも大きいかどうかについてさらに決定される。大きい場合、ステップ818で例外条件が提供される。一例では、DMA開始アドレスおよびDMA終了アドレスは、システム規模のポリシーに基づくものである。

30

【0079】

その後、照会820で、必要であれば、I/Oアドレス変換を実行するために十分なリソースが使用可能であるかどうかについて決定される。使用可能でない場合、ステップ822で例外条件が提供される。使用可能である場合、照会824で、I/Oアドレス変換パラメータがFTEおよびDTE内にすでに登録されているかどうかについて決定される。これは、FTE/DTE内のパラメータの値をチェックすることによって決定される。たとえば、FTE/DTE内の値がゼロかまたは他の定義済みの値である場合、登録は実行されていない。FTEの位置を特定するためには命令内に提供されたハンドルが使用され、DTEの位置を特定するためにはFTE内のデバイス・インデックスが使用される。

40

【0080】

アドレス変換に関してアダプタ機能がすでに登録されている場合、ステップ826で例外条件が提供される。登録されていない場合、照会828で、指定されたDMAアドレス・スペースが有効であるかどうか(すなわち、DTEが実行可能なアドレス・スペースであるかどうか)について決定される。有効でない場合、ステップ830で例外条件が提供される。すべてのチェックが首尾よく行われた場合、ステップ832で、変換パラメータがデバイス・テーブル・エントリ内に、およびオプションで、対応する機能テーブル・エ

50

ントリ（または他の指定された位置）内に、配置される。たとえば、I/Oアドレス変換に関するPCI機能パラメータが機能情報ブロックからコピーされ、DTE/FTE内に配置される。これらのパラメータは、たとえばPCIベース・アドレス、PCIアドレス限界、変換フォーマット、ページ・サイズ、およびI/Oアドレス変換ポイントを含む。この動作により、指定されたDMAアドレス・スペースへのDMAアクセスが実行可能となる。これにより、アダプタ機能に関するI/Oアドレス変換が実行可能となる。

【0081】

PCI機能制御修正命令によって指定可能な他の動作制御は、I/Oアドレス変換パラメータ登録解除動作であり、その例について図16を参照しながら説明する。この動作では、I/Oアドレス変換に関連する機能パラメータがゼロにリセットされる。この動作で、指定されたDMAアドレス・スペースへのDMAアクセスが実行不可となり、そのDMAアドレス・スペースに対するI/O変換索引（lookaside）バッファ・エントリのページ（purge）が実行される。これにより、アドレス変換が実行不可となる。

10

【0082】

図16を参照すると、一実施形態では、照会900で、I/Oアドレス変換パラメータが登録されていないかどうかについて決定される。一例では、この決定は、FTEまたはDTE内の適切なパラメータの値をチェックすることによって実行される。それらのフィールドがゼロまたは何らかの指定された値である場合、それらは登録されていない。したがって、ステップ902で例外条件が提供される。それらが登録されている場合、照会904で、DMAアドレス・スペースが有効であるかどうかについて決定される。これが無効である場合、ステップ906で例外条件が提供される。DMAアドレス・スペースが有効である場合、ステップ908で、デバイス・テーブル・エントリ内、およびオプションで、対応する機能テーブル・エントリ内の、変換パラメータがクリアされる。

20

【0083】

上記で、アダプタによってシステム・メモリ・アドレスに提供された入力/出力アドレスを変換するための効率的なメカニズムについて、詳細に説明した。一例では、メモリにアクセスする際に、フルPCIアドレス（たとえば64ビット・アドレス全体）が使用されるが、変換のためにアドレスの一部のみを使用することによって、アドレス変換ルックアップが最小限に抑えられる。フル・アドレスの使用によって保護は追加されるが、たとえば変換のために低位ビットのみを使用することで、低レベルの変換ルックアップによってより効率的な変換が可能となる。これにより、オペレーティング・システム仮想アドレスと共存可能な、または同じであることが可能なアドレスを使用する際に、オペレーティング・システムの柔軟性が可能となる。さらに、アダプタがCPUアドレス変換テーブルを共有できるようになり、I/Oはスキャンが必要なテーブルの数を減らすことができる。さらに、異なるアダプタまたはオペレーティング・システムあるいはその両方が、アドレス・スペース領域の切り離しを使用できるようにすることで、保護が追加される。

30

【0084】

本明細書で説明される実施形態では、アダプタはPCIアダプタである。本明細書で使用されるPCIは、PCIまたはPCIeを含むがこれに限定されない、Peripheral Component Interconnect Special Interest Group (PCI-SIG)によって定義されるPCIベース規格に従って実装される、任意のアダプタを言い表す。特定の一例では、Peripheral Component Interconnect Express (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に関する双方向通信プロトコルを定義する、構成要素レベルの相互接続標準である。PCI通信は、PCIeバス上の伝送に関するPCIe標準に従ってパケット内にカプセル化される。I/Oアダプタから発せられ、ホスト・システムで終端するトランザクションは、上り方向（upbound）トランザクションと呼ばれる。ホスト・システムから発せられ、I/Oアダプタで終端するトランザクションは下り方向（downbound）トランザクションと呼ばれる。PCIeトポロジは、PCIeバスを形成するためにペア（たとえば、一方は上り方向リンク、一方は下り方向リ

40

50

ンク)を成す2地点間一方向リンクに基づく。P C I e 標準はP C I - S I Gによって維持および公開される。

【0085】

当業者であれば理解されるように、本発明の態様は、システム、方法、またはコンピュータ・プログラム製品として具体化することができる。したがって、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、マイクロコードなど)、または、本明細書ではすべてが全体として「回路」、「モジュール」、または「システム」と呼ばれることのあるソフトウェアおよびハードウェアの態様を組み合わせた実施形態の、形を取ることができる。さらに、本発明の態様は、具体化されたコンピュータ読み取り可能プログラム・コードを有する1つまたは複数のコンピュータ読み取り可能メディア内に具体化された、コンピュータ・プログラム製品の形を取ることができる。

10

【0086】

1つまたは複数のコンピュータ読み取り可能メディアの任意の組み合わせを使用することができる。コンピュータ読み取り可能メディアは、コンピュータ読み取り可能ストレージ・メディアとすることができる。コンピュータ読み取り可能ストレージ・メディアは、たとえば、電子、磁気、光、電磁、赤外線、または半導体の、システム、装置、またはデバイス、あるいはそれらの任意の好適な組み合わせとすることができるが、これらに限定されるものではない。コンピュータ読み取り可能ストレージ・メディアのさらに特定の例(非網羅的リスト)は、1本または複数本のワイヤを有する電気接続、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)、読み取り専用メモリ(ROM)、消去可能プログラマブル読み取り専用メモリ(EEPROMまたはフラッシュ・メモリ)、光ファイバ、ポータブル・コンパクト・ディスク読み取り専用メモリ(CD-ROM)、光ストレージ・デバイス、磁気ストレージ・デバイス、またはそれらの任意の好適な組み合わせを含む。本書との関連で、コンピュータ読み取り可能ストレージ・メディアは、命令実行のシステム、装置、またはデバイスによって、またはそれらに関連して使用するためのプログラムを、含むかまたは格納可能な、任意の有形メディアとすることができる。

20

【0087】

次に図17を参照すると、一例では、コンピュータ・プログラム製品1000は、たとえば、本発明の1つまたは複数の態様を提供および容易にするためにコンピュータ読み取り可能プログラム・コード手段または論理1004をその上に格納するための、1つまたは複数のコンピュータ読み取り可能ストレージ・メディア1002を含む。

30

【0088】

コンピュータ読み取り可能メディア上に具体化されたプログラム・コードは、無線、有線、光ファイバ・ケーブル、RFなど、またはそれらの任意の好適な組み合わせを含むがこれらに限定されない、適切なメディアを使用して伝送可能である。

【0089】

本発明の態様に関する動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++などのオブジェクト指向プログラミング言語と、「C」プログラミング言語、アセンブラ、または同様のプログラミング言語などの、従来の手続き型プログラミング言語とを含む、1つまたは複数のプログラミング言語の任意の組み合わせで作成可能である。プログラム・コードは、完全にユーザのコンピュータ上で、部分的にユーザのコンピュータ上で、スタンドアロン型ソフトウェア・パッケージとして、部分的にユーザのコンピュータ上および部分的にリモート・コンピュータ上で、あるいは、完全にリモートのコンピュータまたはサーバ上で、実行可能である。後者のシナリオでは、リモート・コンピュータを、ローカル・エリア・ネットワーク(LAN)またはワイド・エリア・ネットワーク(WAN)を含む任意のタイプのネットワークを介してユーザのコンピュータに接続するか、あるいは、(たとえば、インターネット・サービス・プロバイダを使用し、インターネットを介して)外部コンピュータに接続することができる

40

50

【0090】

本明細書では、本発明の実施形態に従った方法、装置（システム）、およびコンピュータ・プログラム製品の、流れ図またはブロック図あるいはその両方を参照しながら、本発明の態様について説明する。流れ図またはブロック図あるいはその両方の各ブロック、および、流れ図またはブロック図あるいはその両方におけるブロックの組み合わせが、コンピュータ・プログラム命令によって実装可能であることを理解されよう。これらのコンピュータ・プログラム命令は、機械を生成するために、汎用コンピュータ、特定用途向けコンピュータ、または他のプログラマブル・データ処理装置の、プロセッサに提供可能であり、結果としてコンピュータまたは他のプログラマブル・データ処理装置のプロセッサを介して実行する命令が、流れ図またはブロック図あるいはその両方のブロック内に指定された機能／動作を実装するための手段を作成する。

10

【0091】

コンピュータ、他のプログラマブル・データ処理装置、または他のデバイスに、特定の様式で機能するように指示することが可能な、これらのコンピュータ・プログラム命令を、コンピュータ読み取り可能メディアに格納することも可能であり、結果として、コンピュータ読み取り可能メディア内に格納された命令が、流れ図またはブロック図あるいはその両方のブロック内に指定された機能／動作を実装する命令を含む製品を生成する。

【0092】

コンピュータ実装プロセスを生成するために、コンピュータ、他のプログラマブル装置、または他のデバイス上で一連の動作ステップを実行させるように、コンピュータ・プログラム命令を、コンピュータ、他のプログラマブル・データ処理装置、または他のデバイス上にロードすることも可能であり、結果として、コンピュータまたは他のプログラマブル装置上で実行する命令は、流れ図またはブロック図あるいはその両方のブロック内に指定された機能／動作を実装するための処理を提供する。

20

【0093】

図面内の流れ図およびブロック図は、本発明の様々な実施形態に従った、システム、方法、およびコンピュータ・プログラム製品の可能な実装のアーキテクチャ、機能、および動作を示す。この点で、流れ図またはブロック図内の各ブロックは、指定された論理機能を実装するための1つまたは複数の実行可能命令を含む、モジュール、セグメント、またはコードの一部を表すことが可能である。いくつかの代替の実装では、ブロック内に示された機能が、図に示された順序外で実行可能であることにも留意されたい。たとえば、関連する機能に応じて、連続して示された2つのブロックを実際にはほぼ同時に実行することが可能であるか、またはこのブロックを逆の順序で実行することが可能である。ブロック図または流れ図あるいはその両方の各ブロック、および、ブロック図または流れ図あるいはその両方内のブロックの組み合わせが、指定された機能または動作を実行する特定用途向けハードウェアベース・システム、あるいは、特定用途向け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、実装可能であることにも留意されよう。

30

【0094】

上記に加えて、本発明の1つまたは複数の態様は、顧客環境の管理を提示するサービス・プロバイダによる、提供、提示、展開、管理、サービス提供などが可能である。たとえばサービス・プロバイダは、1人または複数の顧客に対して本発明の1つまたは複数の態様を実行する、コンピュータ・コードまたはコンピュータ・インフラストラクチャあるいはその両方を、作成、維持、サポートなどを実行することができる。次に、サービス・プロバイダは、例として、加入または料金契約あるいはその両方の下で、顧客からの支払いを受け取ることができる。加えて、または別の方法として、サービス・プロバイダは、1人または複数の第三者への広告コンテンツの販売から支払いを受け取ることができる。

40

【0095】

本発明の一態様では、本発明の1つまたは複数の態様を実行するためのアプリケーションが展開可能である。一例として、アプリケーションの展開は、本発明の1つまたは複数

50

の態様を実行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。

【0096】

本発明の他の態様として、コンピュータ読み取り可能コードをコンピューティング・システムに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が展開可能である。コードとコンピューティング・システムとを組み合わせると、本発明の1つまたは複数の態様を実行することができる。

【0097】

本発明のさらに他の態様として、コンピュータ読み取り可能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可能である。コンピュータ・システムはコンピュータ読み取り可能メディアを備え、コンピュータ・メディアは本発明の1つまたは複数の態様を含む。コードとコンピュータ・システムとを組み合わせると、本発明の1つまたは複数の態様を実行することができる。

【0098】

上記で様々な実施形態について説明したが、こ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は、本発明の1つまたは複数の態様を組み込んで使用することができる。例として、インターナショナル・ビジネス・マシーンス・コーポレーションによって提示されたPower Systemsサーバまたは他のサーバ、あるいは他の企業のサーバなどの、System z (R)サーバ以外のサーバは、本発明の1つまたは複数の態様を含むこと、使用すること、またはそれらから恩恵を得ること、あるいはそれらすべてが可能である。さらに、本明細書の例では、アダプタおよびPCIハブはサーバの一部とみなされるが、他の実施形態では、それらは必ずしもサーバの一部とみなされる必要はなく、単に、コンピューティング環境のシステム・メモリまたは他の構成要素あるいはその両方に結合されているものとみなすことができる。コンピューティング環境はサーバである必要はない。さらに、変換テーブルについて説明しているが、任意のデータ構造を使用することが可能であり、テーブルという用語はすべてのこうしたデータ構造を含むものである。さらに、アダプタはPCIベースであるが、本発明の1つまたは複数の態様は、他のアダプタまたは他のI/O構成要素と共に使用可能である。アダプタおよびPCIアダプタは、単なる例である。さらに、本発明から逸脱することなく、他のサイズのアドレス・スペース、アドレス・テーブル、またはページ、あるいはそれらすべてを使用することができる。さらにDTEは、より多い、より少ない、または異なる情報を含むことができる。またさらに、本発明の1つまたは複数の態様を使用して、他のタイプのアドレスを変換することができる。多くの他の変形が可能である。

【0099】

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つまたは複数の態様からの恩恵を得ることができる。例として、システム・バスを介して直接または間接的にメモリ要素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・コードの格納または実行あるいはその両方を行うために好適なデータ処理システムが使用可能である。メモリ要素は、たとえば、プログラム・コードの実際の実行中に採用されるローカル・メモリと、大容量ストレージと、実行中に大容量ストレージからコードを取り出さなければならない回数を低減させるために少なくともいくつかのプログラム・コードの一時ストレージを提供するキャッシュ・メモリとを含む。

【0100】

入力/出力または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、および他のメモリ・メディアなど)を、直接、または介在するI/Oコントローラを介して、システムに結合することができる。データ処理システムを、介在する専用または公衆のネットワークを介して、他のデータ処理システムあるいはリモートのプリンタまたはストレージ・デバイスに結合できるようにするために、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

10

20

30

40

50

デム、ケーブル・モデム、およびイーサネット・カードは、使用可能なネットワーク・アダプタのほんのいくつかのタイプである。

【0101】

図18を参照すると、本発明の1つまた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の代表的な構成要素が示されている。代表的なホスト・コンピュータ5000は、コンピュータ・メモリ(すなわち中央ストレージ)5002と通信する1つまたは複数のCPU5001、ならびに、他のコンピュータまたはSANなどと通信するためのストレージ・メディア・デバイス5011およびネットワーク5010へのI/Oインターフェースを備える。CPU5001は、アーキテクチャ命令セットおよびアーキテクチャ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、ダイナミック・アドレス変換(DAT)5003を有することができる。DATは、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、通常、変換をキャッシュに入れるための変換索引バッファ(TLB)5007を含む。通常、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が採用される。キャッシュ5009は、複数のCPUが使用可能な大型キャッシュと、大型キャッシュと各CPUとの間の小型、高速(低レベル)キャッシュとを有する、階層型とすることができる。いくつかの実装では、低レベル・キャッシュは、命令フェッチおよびデータ・アクセスのための別の低レベル・キャッシュを提供するために、分割される。一実施形態では、命令は、キャッシュ5009を介して命令フェッチ・ユニット5004によってメモリ5002からフェッチされる。命令は、命令復号ユニット5006内で復号され、命令実行ユニット5008へと(いくつかの実施形態では他の命令と共に)ディスパッチされる。通常、たとえば演算実行ユニット、浮動小数点実行ユニット、および分岐命令実行ユニットなどの、いくつかの実行ユニット5008が採用される。この命令は、必要に応じて、命令指定レジスタまたはメモリからオペランドにアクセスする、実行ユニットによって実行される。オペランドがメモリ5002からアクセス(ロードまたは格納)されるものである場合、ロード/格納ユニット5005は、通常、実行される命令の制御の下に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、あるいはそれらの組み合わせによって、実行可能である。

【0102】

上記のように、コンピュータ・システムは、ローカル(またはメイン)ストレージ内に、情報、ならびにアドレス指定、保護、および参照、ならびに変更記録を含む。アドレス指定のいくつかの態様は、アドレスのフォーマット、アドレス・スペースの概念、様々なタイプのアドレス、および、1つのタイプのアドレスが他のタイプのアドレスに変換される様式を含む。いくつかのメイン・ストレージは、永続的に割り当てられたストレージ位置を含む。メイン・ストレージは、直接アドレス指定可能なデータの高速度アクセス・ストレージを、システムに提供する。データおよびプログラムはどちらも、処理される前に(入力デバイスから)メイン・ストレージ内にロードされることになる。

【0103】

メイン・ストレージは、時にはキャッシュと呼ばれる、1つまたは複数の小型高速アクセス・バッファ・ストレージを含むことができる。キャッシュは、通常、CPUまたはI/Oプロセッサに物理的に関連付けられる。性能に関する以外の、別個のストレージ・メディアの物理構造および使用の効果は、通常、プログラムによって観察することはできない。

【0104】

別々のキャッシュを、命令およびデータ・オペランドに対して維持することができる。キャッシュ内の情報は、キャッシュ・ブロックまたはキャッシュ・ライン(または単にライン)と呼ばれる整数境界(integral boundary)上の連続バイト内で維持される。モデルは、バイト単位のキャッシュ・ラインのサイズを戻す、キャッシュ属性抽出命令を提供

10

20

30

40

50

する。モデルは、データまたは命令キャッシュへのストレージのプリフェッチまたはキャッシュからのデータの解放を実行する、データのプリフェッチ命令および比較的長いデータのプリフェッチ命令を提供することもできる。

【 0 1 0 5 】

ストレージは、長い水平のビットの文字列とみなされる。たいていの動作では、ストレージへのアクセスは左から右の順に進行する。ビットの文字列は、8ビット単位に細分化される。8ビット単位がバイトと呼ばれ、これがすべての情報フォーマットの基本構築ブロックである。ストレージ内の各バイト位置は、そのバイト位置のアドレスであるか、または単にバイト・アドレスである、固有の非負整数によって識別される。隣接するバイト位置は、左の0から始まり、左から右へと順に進んでいく、連続アドレスを有する。アドレスは符号なし2進整数であり、24、31、または64ビットである。

10

【 0 1 0 6 】

情報は、ストレージとCPUまたはチャンネル・サブシステムとの間で、1回に1バイトまたはバイト・グループずつ送信される。特に指定のない限り、たとえばz / Architecture (R)では、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ってアドレス指定される。グループ内のバイト数は、実行される動作によって暗黙的または明示的に指定される。CPU動作で使用される場合、バイトのグループはフィールドと呼ばれる。バイトの各グループ内で、たとえばz / Architecture (R)では、ビットは左から右の順に番号付けされる。z / Architecture (R)では、左端のビットは時には「高位」ビットと呼ばれ、右端のビットは「低位」ビットと呼ばれる。しかしながら、ビット番号はストレージ・アドレスではない。バイトのみがアドレス指定可能である。ストレージ内のバイトの個別のビット上で動作するために、バイト全体がアクセスされる。バイト内のビットには、左から右へと、0から7の番号が付けられる(たとえばz / Architecture (R)の場合)。アドレス内のビットには、24ビット・アドレスの場合は8~31または40~63、あるいは31ビット・アドレスの場合は1~31または33~63の、番号を付けることが可能であり、64ビット・アドレスの場合は0~63が番号付けされる。任意の他の複数バイトの固定長フォーマットでは、フォーマットを構成するビットには、0から始まる連続番号が付けられる。エラー検出のため、および好ましくは訂正のために、1つまたは複数のチェック・ビットを、各バイトまたはバイト・グループと共に伝送することができる。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、プログラムによって直接制御することはできない。ストレージ容量はバイト数で表される。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・コードによって暗示される場合、フィールドは、1、2、4、8、または16バイトとすることが可能な、固定長を有するものと言われる。いくつかの命令では、さらに大きいフィールドが暗示される場合がある。ストレージ・オペランド・フィールドの長さが暗示されず、明示的に示される場合、フィールドは可変長を有するものと言われる。可変長オペランドは、1バイトずつの増分(またはいくつかの命令では、2バイトの倍数または他の倍数)での長さの変化が可能である。情報がストレージ内に配置される場合、たとえストレージへの物理経路の幅が格納されているフィールドの長さよりも長い場合であっても、指定されたフィールドに含まれるそれらのバイト位置のコンテンツのみが置き換えられる。

20

30

40

【 0 1 0 7 】

情報の一定の単位は、ストレージ内の整数境界とされる。境界は、そのストレージ・アドレスがバイト単位の長さの倍数である場合、情報の単位に関する整数と呼ばれる。整数境界上の2、4、8、および16バイトのフィールドには、特別の名前が与えられる。ハーフワードは、2バイト境界上の2つの連続するバイトのグループであり、命令の基本構築ブロックである。ワードは、4バイト境界上の4つの連続するバイトのグループである。ダブルワードは、8バイト境界上の8つの連続するバイトのグループである。クワドワード(quadword)は、16バイト境界上の16の連続するバイトのグループである。ストレージ・アドレスがハーフワード、ワード、ダブルワード、およびクワドワードを指定す

50

る場合、アドレスの2進表現は、それぞれ右端に1つ、2つ、3つ、または4つのゼロビットを含む。命令は、2バイト整数境界上にあるものとされる。ほとんどの命令のストレージ・オペランドは、境界整列要件を有さない。

【0108】

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装するデバイスでは、後で命令がフェッチされる元であるキャッシュ・ラインに、プログラムが格納する場合、格納が後でフェッチされる命令を変更するかどうかにかかわらず、大幅な遅延を経験する可能性がある。

【0109】

一実施形態では、本発明は、ソフトウェア（時にはライセンス付き内部コード、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、そのいずれかは本発明と一致することになる）によって実施可能である。図18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、通常、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、長期ストレージ・メディア・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、ディスク、ハード・ドライブ、またはCD-ROMなどの、データ処理システムと共に使用するための様々な知られたメディアのいずれかで、具体化することができる。コードは、こうしたメディア上で配布可能であるか、またはコンピュータ・メモリ5002からユーザへ、あるいは、ネットワーク5010を介して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・システムへ、こうした他のシステムのユーザが使用するために、配布可能である。

【0110】

ソフトウェア・プログラム・コードは、様々なコンピュータ構成要素および1つまたは複数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、ストレージ・メディア・デバイス5011から、プロセッサ5001による処理に使用可能な比較的高速のコンピュータ・ストレージ5002へと、ページングされる。物理メディア上のメモリ内のソフトウェア・プログラム・コードを具体化するため、またはネットワークを介してソフトウェア・コードを配布するため、あるいはその両方のための技法および方法は、良く知られており、本明細書ではこれ以上考察しない。プログラム・コードは、有形メディア（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これらに限定されない）上で作成および格納される場合、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・プログラム製品メディアは、通常、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。

【0111】

図19は、内部で本発明が実施可能な、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19のシステム5020は、オプションの周辺デバイスを含む、パーソナル・コンピュータ、ワークステーション、またはサーバなどの、代表的な基本コンピュータ・システム5021を備える。基本コンピュータ・システム5021は、1つまたは複数のプロセッサ5026、ならびに、既知の技法に従ってプロセッサ5026とシステム5021の他の構成要素との間の通信を接続および実行可能にするために採用されたバスを含む。バスは、プロセッサ5026を、メモリ5025と、たとえばハード・ドライブ（たとえば磁気メディア、CD、DVD、およびフラッシュ・メモリのうちのいずれかを含む）またはテープ・ドライブを含むことが可能な長期ストレージ5027と、に接続する。システム5021は、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、または、タッチ・スクリーン、デジタル入力パッドなどの任意のユーザ・インターフェース・デバイスとすることが可能な他のインターフェース・デバイス、あるいはそれらすべてなどの、1つまたは複数のインターフェース・デバイスに接続する、ユーザ・インターフェース・アダ

10

20

30

40

50

プタも含む場合がある。バスは、ディスプレイ・アダプタを介して、LCDスクリーンまたはモニタなどのディスプレイ・デバイス5022を、マイクロプロセッサ5026にも接続する。

【0112】

システム5021は、ネットワーク5029との通信5028が可能なネットワーク・アダプタを用いて、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可能である。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。別の方法として、システム5021は、CDPD(セルラ式デジタル・パケット・データ)カードなどの無線インターフェースを使用して、通信可能である。システム5021は、ローカル・エリア・ネットワーク(LAN)またはワイド・エリア・ネットワーク(WAN)におけるこうした他のコンピュータと関連付けることが可能であるか、または、システム5021は、他のコンピュータとのクライアント/サーバ配置構成におけるクライアントなどとする事ができる。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のすべてが、当分野で知られている。

10

【0113】

図20は、内部で本発明が実施可能なデータ処理ネットワーク5040を示す。データ処理システム5040は、それぞれが複数の個別のワークステーション5041、5042、5043、5044を含むことが可能な、無線ネットワークおよび有線ネットワークなどの複数の個別のネットワークを含むことができる。加えて、当業者であれば、1つまたは複数のLANを含めることが可能であることを理解し、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備えることができる。

20

【0114】

さらに図20を参照すると、ネットワークは、ゲートウェイ・コンピュータ(クライアント・サーバ5046)、またはアプリケーション・サーバ(データ・リポジトリにアクセス可能であり、ワークステーション5045から直接アクセスすることも可能な、リモート・サーバ5048)などの、メインフレーム・コンピュータまたはサーバを含むこともできる。ゲートウェイ・コンピュータ5046は、各個別のネットワークへの入力ポイントとして働く。ゲートウェイは、一方のネットワーク・プロトコルを他方に接続する場合に必要である。ゲートウェイ5046は、ゲートウェイ5046は、好ましくは、通信リンクによって他のネットワーク(たとえばインターネット5047)に結合することが可能である。ゲートウェイ5046は、通信リンクを使用して1つまた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可能である。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・マシーンズ・コーポレーションから入手可能な、IBM eServer(TM) System z(R)サーバを使用して実装可能である。

30

【0115】

図19および図20を同時に参照すると、CD-ROMドライブまたはハード・ドライブなどの、長期ストレージ・メディア5027から、システム5020のプロセッサ5026によって、本発明を具体化することが可能なソフトウェア・プログラミング・コードにアクセスすることができる。ソフトウェア・プログラミング・コードは、ディスク、ハード・ドライブ、またはCD-ROMなどの、データ処理システムで使用するために、様々な既知のメディアのいずれかで具体化することができる。コードは、こうしたメディア上で配布可能であるか、または、1つのコンピュータ・システムのメモリまたはストレージから、ネットワークを介して他のコンピュータ・システムへと、こうした他のシステムのユーザが使用するために、ユーザ5050、5051に配布可能である。

40

【0116】

別の方法として、プログラミング・コードは、メモリ5025内で具体化し、プロセッサ・バスを使用してプロセッサ5026によってアクセス可能である。こうしたプログラム・コードは、様々なコンピュータ構成要素および1つまたは複数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。

50

プログラム・コードは、通常、ストレージ・メディア5027から、プロセッサ5026による処理に使用可能な高速メモリ5025へとページングされる。物理メディア上のメモリ内のソフトウェア・プログラム・コードを具体化するため、またはネットワークを介してソフトウェア・コードを配布するため、あるいはその両方のための技法および方法は、良く知られており、本明細書ではこれ以上考察しない。プログラム・コードは、有形メディア（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これらに限定されない）上で作成および格納される場合、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・プログラム製品メディアは、通常、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。

10

【0117】

プロセッサが最も容易に使用できるキャッシュ（通常はプロセッサの他のキャッシュより高速かつ小型）は、最低（L1またはレベル1）キャッシュであり、メイン・ストア（メイン・メモリ）は、最高レベル・キャッシュ（3レベルある場合はL3）である。最低レベル・キャッシュはしばしば、実行されることになる機械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。

【0118】

図21を参照すると、例示的プロセッサの実施形態がプロセッサ5026に関して示されている。通常、プロセッサの性能を向上させるために、キャッシュ5053の1つまたは複数のレベルがメモリ・ブロックをバッファに入れるために採用される。キャッシュ5053は、使用さ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、128、または256バイトのメモリ・データである。別々のキャッシュが、データのキャッシュよりも命令のキャッシュにしばしば採用される。キャッシュ・コヒーレンス（メモリおよびキャッシュ内のラインのコピーの同期化）は、しばしば、当分野で周知の様々な「スヌープ（snoop）」アルゴリズムによって提供される。プロセッサ・システムのメイン・メモリ・ストレージ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・システムでは、メイン・ストレージ5025は、通常はより高速であり、コンピュータ・システムが使用可能な不揮発性ストレージ（DASD、テープなど）の一部のみを保持するため、時にはレベル5（L5）と呼ばれる。メイン・ストレージ5025は、オペレーティング・システムによってメイン・ストレージ5025内およびメイン・ストレージ5025外にページングされたデータのページを「キャッシュ」する。

20

30

【0119】

プログラム・カウンタ（命令カウンタ）5061は、実行されることになる現行命令のアドレスを追跡する。z/Architecture（R）プロセッサ内のプログラム・カウンタは64ビットであり、前のアドレス指定限界をサポートするために31または24ビットまで切り詰めることができる。プログラム・カウンタは通常、コンテキスト切り換え時に持続するように、コンピュータのPSW（プログラム状況ワード）で具体化される。したがって、プログラム・カウンタ値を有する進行中のプログラムに、たとえばオペレーティング・システムが割り込むことができる（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切り換え）。プログラムのPSWは、プログラムがアクティブでない間、プログラム・カウンタ値を維持し、オペレーティング・システムが実行している間、オペレーティング・システムの（PSWの）プログラム・カウンタが使用される。通常、プログラム・カウンタは、現行命令のバイト数に等しい数ずつ増分される。RISC（縮小命令セット・コンピューティング）命令は、通常、固定長であり、CISC（複数命令セット・コンピューティング）命令は、通常、可変長である。IBM z/Architecture（R）の命令は、2、4、または6バイトを有するCISC命令である。プログラム・カウンタ5061は、コンテキスト切り換え動作、または、たとえば分岐命令の分岐実行動作のいずれかによって修正される。コンテキスト切り換え動作

40

50

では、実行されているプログラムに関する（条件コードなどの）他の状態情報と共に、現行プログラムのカウンタ値がプログラム状況ワードに保存され、実行されることになる新しいプログラム・モジュールの命令をポイントする、新しいプログラムのカウンタ値がロードされる。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、プログラムがプログラム内で意思決定またはループできるようにするために、分岐実行動作が実行される。

【0120】

通常、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採用される。フェッチ・ユニットは、「次の順次命令」、分岐実行命令のターゲット命令、またはコンテキスト切り換えに続くプログラムの第1の命令の、いずれかをフェッチする。現在の命令フェッチ・ユニットは、しばしば、プリフェッチされた命令が使用され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めに、プリフェッチ技法を採用する。たとえば、フェッチ・ユニットは、次の順次命令および他の順次命令の追加バイトを含む、16バイトの命令をフェッチすることができる。

10

【0121】

フェッチされた命令は、その後、プロセッサ5026によって実行される。ある実施形態では、フェッチされた命令がフェッチ・ユニットのディスパッチ・ユニット5056に渡される。ディスパッチ・ユニットは命令を復号し、復号された命令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、通常、復号された算術命令に関する情報を、命令フェッチ・ユニット5055から受信し、命令のオペレーション・コードに従って、オペランド上で算術演算を実行することになる。オペランドは、好ましくはメモリ5025、アーキテクチャ・レジスタ5029、または実行されている命令の即時フィールドの、いずれかから、実行ユニット5057に提供される。実行の結果は、格納される場合、メモリ5025、レジスタ5029、または（制御レジスタ、PSWレジスタなどの）他の機械ハードウェアの、いずれかに格納される。

20

【0122】

プロセッサ5026は、通常、命令の機能を実行するための1つまたは複数のユニット5057、5058、5060を有する。図22を参照すると、実行ユニット5057は、アーキテクチャ汎用レジスタ5059、復号/ディスパッチ・ユニット5056、ロード/格納ユニット5060、およびその他5065のプロセッサ・ユニットと、インターフェース論理5071によって通信することができる。実行ユニット5057は、算術論理ユニット（ALU）5066が動作することになる情報を保持するために、いくつかのレジスタ回路5067、5068、5069を採用することができる。ALUは、加算、減算、乗算、および除算などの算術演算、ならびに、論理積、論理和、および排他的論理和（XOR）、回転およびシフトなどの、論理機能を実行する。好ましくは、ALUは、設計に依存した特別な演算をサポートする。他の回路は、たとえば条件コードおよび回復サポート論理を含む、他のアーキテクチャ機構5072を提供することができる。通常、ALU演算の結果は、結果を様々な他の処理機能に転送できる出力レジスタ回路5070内に保持される。プロセッサ・ユニットには多くの配置構成があるが、この説明では、一実施形態の代表的な理解を与えることのみが意図されている。

30

40

【0123】

たとえばADD命令は、算術および論理機能を有する実行ユニット5057内で実行されるが、たとえば浮動小数点命令は、特別な浮動小数点機能を有する浮動小数点実行内で実行される。好ましくは、実行ユニットは、オペランド上でオペレーション・コード定義機能を実行することによって、命令によって識別されたオペランド上で動作する。たとえば、加算命令は、命令のレジスタ・フィールドによって識別された2つのレジスタ5059内で見つかったオペランド上で、実行ユニット5057によって実行されることが可能である。

【0124】

実行ユニット5057は、2つのオペランド上で算術加算を実行し、その結果を第3の

50

オペランドに格納するが、この第3のオペランドは、第3のレジスタ、または2つのソース・レジスタのうちの1つとすることができる。実行ユニットは、好ましくは、シフト、回転、論理積、論理和、および排他的論理和などの、様々な論理機能、ならびに、加算、減算、乗算、除算のいずれかを含む、様々な算術機能を実行することが可能な、演算論理ユニット（ALU）5066を使用する。ALU5066のいくつかはスカラー演算用に、いくつかは浮動小数点用に設計されている。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）（最高バイト・アドレスに最下位バイトがある）またはリトル・エンディアン（最低バイト・アドレスに最下位バイトがある）とすることができる。IBM z/Architecture（R）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、符号付き絶対値、1の補数、または2の補数とすることが可能である。2の補数内の負の値または正の値のいずれかがALU内で加算のみを必要とすることから、ALUは減算機能を設計する必要がないという点において、2の補数有利である。数は通常、省略表現で記述され、たとえば12ビット・フィールドは4096バイト・ブロックのアドレスを画定し、一般に4Kバイト（キロバイト）ブロックと記述される。

【0125】

図23を参照すると、分岐命令を実行するための分岐命令情報は、通常、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ために、分岐履歴テーブル5082などの分岐予測アルゴリズムをしばしば採用する、分岐ユニット5058へと送信される。現行の分岐命令のターゲットはフェッチされ、条件付き演算が完了する前に投機的に実行される。条件付き演算が完了した場合、投機的に実行された分岐命令は、条件付き演算の条件および投機された結果に基づいて、完了または廃棄される。典型的な分岐命令は、条件コードをテストし、条件コードが分岐命令の分岐要件に合致する場合、ターゲット・アドレスに分岐することが可能であり、ターゲット・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時フィールド内で見つかった中のいくつかの数に基づいて算出可能である。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、ALU5074を採用することができる。分岐ユニット5058は、汎用レジスタ5059、復号ディスパッチ・ユニット5056、または、たとえば他の回路5073と通信することができる。

【0126】

命令グループの実行は、たとえば、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切り換え、コンテキスト切り換えを発生させるプログラム例外またはエラー、複数のプログラム（マルチスレッド環境の場合）のコンテキスト切り換えまたはマルチスレッド・アクティビティを発生させるI/O割り込み信号を含む、多様な理由で、割り込まれる可能性がある。好ましくは、コンテキスト切り換え動作は、現在実行中のプログラムに関する状態情報を保存し、その後、呼び出された他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内に保存することができる。状態情報は、好ましくは、実行されることになる次の命令をポイントするプログラム・カウンタ値、条件コード、メモリ変換情報、およびアーキテクチャ・レジスタ・コンテンツを含む。コンテキスト切り換えアクティビティは、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（マイクロコード、ピココード、またはライセンス付き内部コード（LIC））単独で、あるいはそれらの組み合わせによって、行使することが可能である。

【0127】

プロセッサは、命令定義の方法に従ってオペランドにアクセスする。命令は、命令の一部の値を使用して即時オペランドを提供すること、汎用レジスタまたは特定用途向けレジスタ（たとえば浮動小数点レジスタ）のいずれかを明示的にポイントする1つまたは複数のレジスタ・フィールドを提供することが、可能である。命令は、オペレーション・コード・フィールドによってオペランドとして識別された、暗黙レジスタを使用することができる。命令は、オペランドに関するメモリ位置を使用することができる。命令が、たとえ

10

20

30

40

50

ばメモリ内のオペランドのアドレスを提供するためにまとめて追加される、基本レジスタ、インデックス・レジスタ、および即時フィールド（変位フィールド）を定義する、 $z/Architecture(R)$ の長変位機構（long displacement facility）によって例示されるように、オペランドのメモリ位置は、レジスタ、即時フィールド、またはレジスタと即時フィールドとの組み合わせによって提供可能である。本明細書では、特に指定がない限り、位置とはメイン・メモリ（メイン・ストレージ）内の位置を示唆する。

【0128】

図24を参照すると、プロセッサはロード/格納ユニット5060を使用してストレージにアクセスする。ロード/格納ユニット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すること、およびこのオペランドをレジスタ5059または他のメモリ5053位置内にロードすることによって、ロード動作を実行することが可能であるか、あるいは、メモリ5053内のターゲット・オペランドのアドレスを取得すること、および、レジスタ5059または他のメモリ5053位置から取得したデータをメモリ5053内のターゲット・オペランド位置に格納することによって、格納動作を実行することが可能である。ロード/格納ユニット5060は投機的とすることが可能であり、命令シーケンスとは順序外れのシーケンスでメモリにアクセスすることが可能であるが、ロード/格納ユニット5060は、その命令が順序通りに実行されたようにプログラムに対して表示することを維持するものである。ロード/格納ユニット5060は、汎用レジスタ5059、復号/ディスパッチ・ユニット5056、キャッシュ/メモリ・インターフェース5053、または他の要素5083との通信が可能であり、ストレージ・アドレスを計算するため、および、動作を順番に維持するためのパイプライン・シーケンシングを提供するための、様々なレジスタ回路、ALU5085、および制御論理5090を備える。いくつかの動作は順序外れとすることが可能であるが、ロード/格納ユニットは、当分野で周知のように、プログラムに対して順序外れ動作を順序通りに実行されたように表すための機能を提供する。

【0129】

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、時には「論理アドレス」および「有効アドレス」と呼ばれる。これらの仮想アドレスは、単に仮想アドレスにオフセット値をプレフィックス付けすること、仮想アドレスを1つまたは複数の変換テーブルを介して変換することを含むが、これらに限定されない、様々な動的アドレス変換（DAT）技術のうちの1つによって物理メモリ位置にリダイレクトされるという点で、仮想であり、変換テーブルは、好ましくは少なくともセグメント・テーブルおよびページ・テーブルを単独またはそれらの組み合わせで含み、好ましくは、セグメント・テーブルはページ・テーブルをポイントするエントリを有する。 $z/Architecture(R)$ では、領域第1テーブル、領域第2テーブル、領域第3テーブル、セグメント・テーブル、およびオプション・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付けられた物理メモリ位置にマッピングするエントリを備える変換索引バッファ（TLB）を使用することによって、しばしば向上される。エントリは、DATが変換テーブルを使用して仮想アドレスを変換した場合に作成される。その後の仮想アドレスの使用は、低速逐次変換テーブル・アクセスではなく高速TLBのエントリを使用することができる。TLBコンテンツは、LRU（最長時間未使用法）を含む様々な置換アルゴリズムによって管理可能である。

【0130】

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、コヒーレンシに関してインターロックされた、I/O、キャッシュ、TLB、およびメモリなどの、共有リソースを維持する責務を有する。通常、キャッシュ・コヒーレンシを維持する際には、「スヌープ」技術が使用される。スヌープ環境では、各キャッシュ・ラインは、共有を容易にするために、共有状態、排他的状態、変更状態、無効状態などのうちのいずれかとしてマーク付けすることが可能である。

10

20

30

40

50

【0131】

I/Oユニット5054(図21)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・プログラムに提示される。IBM(R)からのSystem z(R)などのメインフレームでは、チャンネル・アダプタおよびオープン・システム・アダプタが、オペレーティング・システムと周辺デバイスとの間に通信を提供するメインフレームのI/Oユニットである。

【0132】

さらに、他のタイプのコンピューティング環境も、本発明の1つまたは複数の態様から恩恵を受けることができる。例として、環境は、エミュレータ(たとえばソフトウェアまたは他のエミュレーション・メカニズム)を含むことが可能であり、ここでは、特定のアーキテクチャ(たとえば命令実行、アドレス変換などのアーキテクチャ機能、およびアーキテクチャ・レジスタを含む)またはそのサブセットが(たとえば、プロセッサおよびメモリを有するネイティブ(native)・コンピュータ・システム上で)エミュレートされる。こうした環境では、たとえ、エミュレータを実行するコンピュータがエミュレートされる機能以外の異なるアーキテクチャを有する可能性がある場合であっても、エミュレータの1つまたは複数のエミュレーション機能は、本発明の1つまたは複数の態様を実装することが可能である。一例として、エミュレーション・モードでは、エミュレートされている特定の命令または動作が復号され、個別の命令または動作を実装するために適切なエミュレーション機能が構築される。

【0133】

エミュレーション環境では、ホスト・コンピュータは、たとえば命令およびデータを格納するためのメモリと、メモリから命令をフェッチするため、およびオプションでフェッチされた命令に対してローカル・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るため、およびフェッチされた命令のタイプを決定するための、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、データをメモリからレジスタにロードすること、データをレジスタからメモリに再度格納すること、または、復号ユニットによって決定されるように、いくつかのタイプの算術または論理演算を実行することを、含むことができる。一例では、各ユニットはソフトウェア内に実装される。たとえば、ユニットによって実行されている動作は、1つまたは複数のサブルーチンとしてエミュレータ・ソフトウェア内に実装される。

【0134】

より具体的には、現在では通常「C」プログラムであるプログラムによって、しばしばコンパイラ・アプリケーションを用いて、メインフレーム内でアーキテクチャ機械命令が使用される。ストレージ・メディアに格納されたこれらの命令は、z/Architecture(R) IBM(R)では、あるいは他のアーキテクチャを実行する機械では、ネイティブに実行することができる。それらは、既存の、および今後のIBM(R)メインフレーム・サーバ内で、およびIBM(R)の他の機械(たとえばPower SystemサーバおよびSystem x(R)サーバ)で、エミュレートすることができる。それらは、IBM(R)、Intel(R)、AMD(TM)、その他によって製造されるハードウェアを使用して、多様な機械上でLinuxを実行している機械内で実行可能である。z/Architecture(R)の下でのそのハードウェア上での実行に加えて、Linux、ならびに、TurboHercules(www.turbohercules.com/)、Hercules(www.hercules-390.org/)、またはFSI(Fundamental Software, Inc)(www.funsoft.com/)によるエミュレーションを使用する、一般に実行がエミュレーション・モードである機械が使用可能である。エミュレーション・モードでは、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ために、エミュレーション・ソフトウェアがネイティブ・プロセッサによって実行される。

【 0 1 3 5 】

ネイティブ・プロセッサは、通常、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・アーキテクチャのフェッチおよび実行命令に対する責務を負う。エミュレーション・ソフトウェアは、一度に1つまたは複数のエミュレートされた機械命令をフェッチし、ネイティブ・プロセッサによる実行のために、この1つまたは複数のエミュレートされた機械命令を対応するネイティブ機械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、高速変換が達成可能なようにキャッシュすることができる。それにもかかわらず、エミュレーション・ソフトウェアは、エミュレートされたプロセッサ用に作成されたオペレーティング・システムおよびアプリケーションが正しく動作するのを保証するように、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するものである。さらに、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上で実行するように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・プロセッサ上で実行可能なように、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・テーブルおよびページ・テーブルを含む動的アドレス変換機能、割り込みメカニズム、コンテキスト切り換えメカニズム、時刻(TOD)時計、および、I/Oサブシステムへのアーキテクチャ・インターフェースを含むが、これらに限定されない、エミュレートされたプロセッサ・アーキテクチャによって識別されたリソースを提供するものである。

10

20

【 0 1 3 6 】

エミュレートされている特定の命令が復号され、個別の命令の機能を実行するために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、好ましい実施形態の説明を理解した当業者であればわかるように、たとえば、「C」サブルーチンまたはドライバ内に、あるいは、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内に、実装される。Beausoleil等による「Multiprocessor for Hardware Emulation」という名称の米国特許証第5551013号、および、Scalzi等による「Preprocessing of Stored Target Routines for Emulating Incompatible Instructions on a Target Processor」という名称の米国特許証第6009261号、および、Davidian等による「Decoding Guest Instruction to Directly Access Emulation Routines that Emulate the Guest Instructions」という名称の米国特許証第5574873号、および、Grorishiek等による「Symmetrical Multiprocessing Bus and Chipset Used for Coprocessor Support Allowing Non-Native Code to Run in a System」という名称の米国特許証第6308255号、および、Lethin等による「Dynamic Optimizing Object Code Translator for Architecture Emulation and Dynamic Optimizing Object Code Translation Method」という名称の米国特許証第6463582号、および、Eric Trautによる「Method for Emulating Guest Instructions on a Host Computer Through Dynamic Recompile of Host Instructions」という名称の米国特許証第5790825号、および多くのその他を含むが、それらに限定されない、様々なソフトウェアおよびハードウェアのエミュレーション特許は、当業者が使用可能なターゲット機械に対して、異なる機械のために構成された命令フォーマットのエミュレーションを

30

40

50

達成する、様々な知られた方法を示している。

【0137】

図25では、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の例が提供されている。エミュレートされたホスト・コンピュータ・システム5092では、ホスト・プロセッサ(CPU)5091はエミュレートされたホスト・プロセッサ(または仮想ホスト・プロセッサ)であり、ホスト・コンピュータ5000'のプロセッサ5091のそれ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を備える。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。この実施形態例では、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部分とに区分される。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って、エミュレートされたホスト・コンピュータ5092のプログラムによって使用可能である。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以外のアーキテクチャのアーキテクチャ命令セットのネイティブ命令を実行し、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、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決定するために、アクセスされたホスト命令を復号することが可能な、シーケンスおよびアクセス/復号ルーチンで取得された1つまたは複数の命令を採用することによって、ホスト・コンピュータ・メモリ5096内のプログラムから実行のためにホスト命令にアクセスすることが可能である。ホスト・コンピュータ・システム5000'アーキテクチャ用に定義された他の機構は、たとえば、汎用レジスタ、制御レジスタ、ダイナミック・アドレス変換、ならびにI/Oサブシステム・サポートおよびプロセッサ・キャッシュなどの機構を含む、アーキテクチャ機構ルーチンによってエミュレート可能である。エミュレーション・ルーチンは、エミュレーション・ルーチンの性能を向上させるために、エミュレーション・プロセッサ5093内で使用可能な(汎用レジスタおよび仮想アドレスの動的変換などの)機能を利用することも可能である。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を支援するために特別なハードウェアおよびオフロード・エンジンを提供することも可能である。

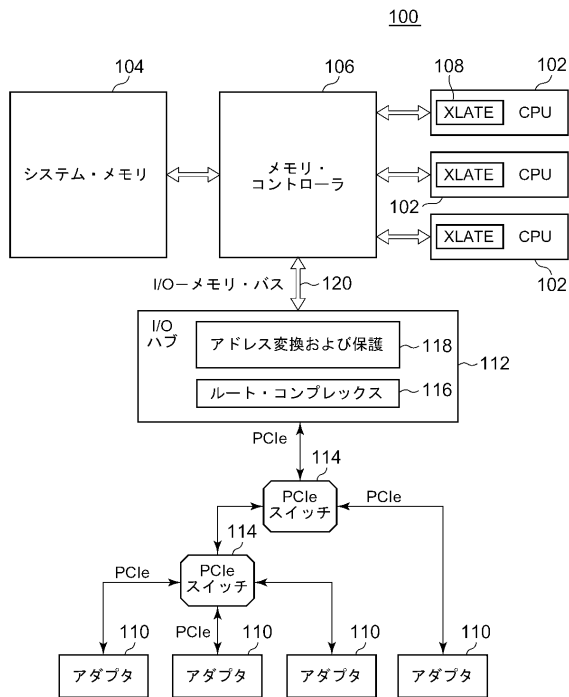
【0138】

本明細書で使用される用語は、特定の実施形態を説明するためのみのものであり、本発明を限定することは意図されていない。本明細書で使用される場合、単数形「a」、「an」、および「the」は、コンテキストが特に明確に示していない限り、複数形も含むことが意図されている。「備える(comprises)」または「備えている(comprising)」あるいはその両方の用語は、本明細書で使用される場合、示された機能、整数、ステップ、動作、要素、または構成要素、あるいはそれらすべての存在を指定するが、1つまたは複数の他の機能、整数、ステップ、動作、要素、構成要素、またはそれらのグループ、あるいはそれらすべての存在または追加を排除するものではないことを、さらに理解されよう。

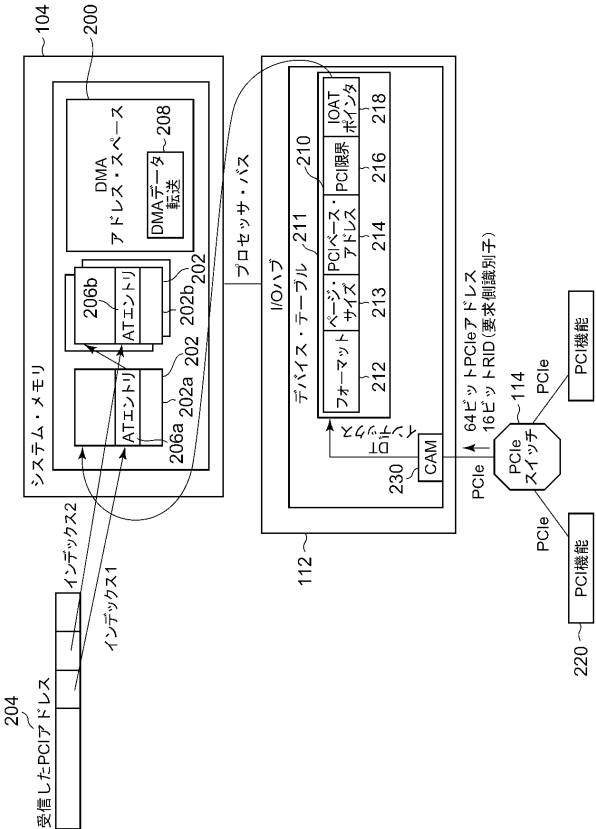
【0139】

以下の特許請求の範囲内のすべての手段またはステップならびに機能要素の、対応する構造、材料、動作、および等価物は、もしもあれば、具体的に請求された他の請求要素と組み合わせてこの機能を実行するための任意の構造、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で提示されたものであるが、開示された形の本発明を網羅するかまたはこれに限定されることは意図されていない。当業者であれば、本発明の範囲を逸脱することのない多くの修正および変形が明らかとなろう。実施形態は、本発明の原理および実際の適用例を最も適切に説明するために、ならびに、企図された特定の用途に好適な様々な修正を伴う様々な実施形態について、他の当業者が本発明を理解できるように、選択および説明されたものである。

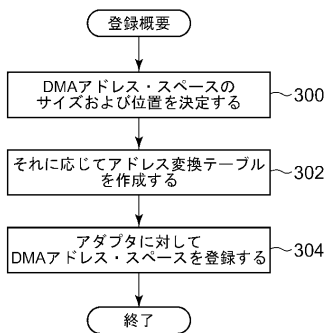
【図1】



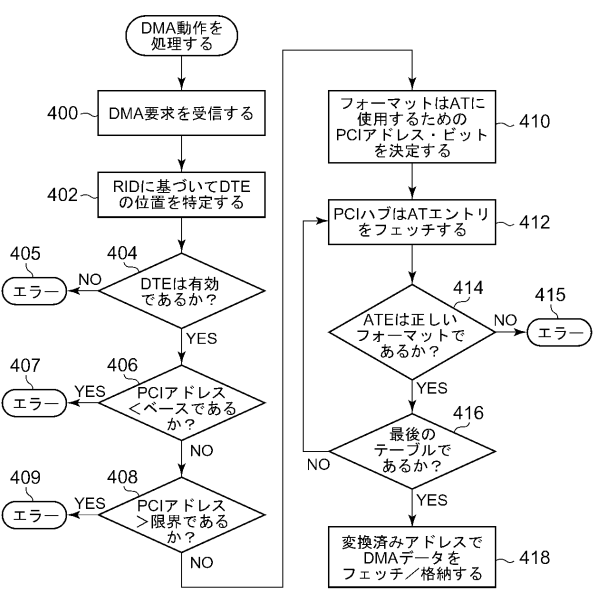
【図2】



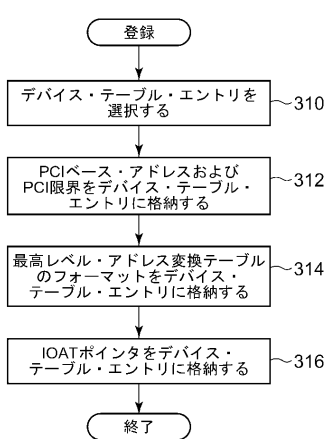
【図3】



【図5】

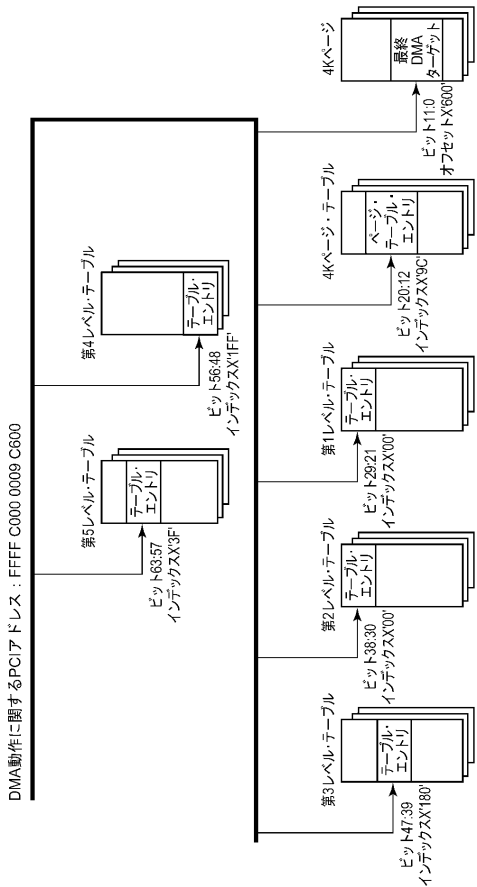


【図4】



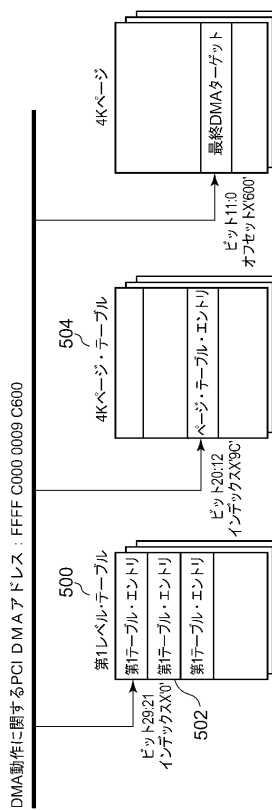
【 図 6 】

DMAAS 6Mサイズ
ページ・テーブルを含む6レベルのテーブルを必要とする (各4Kページは51208バイト・エントリを含む)
DMA動作に関するPCIアドレス : FFFF C000 0009 C600



【 図 7 】

DMAAS 6Mサイズ
1つの4K第1レベル・テーブルおよび3つの4Kページ・テーブルを必要とする (各4Kページは51208バイト・エントリを含む)
ベースPCIアドレス : FFFF C000 0000 0000
PCI履歴アドレス : FFFF C000 005F FFFF
DMA動作に関するPCIアドレス : FFFF C000 0009 C600
PCIベース/履歴チェックにより、PCIアドレス・ビット63:30を無視することができる



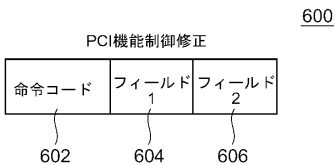
【 図 8 】

フォーマット		CPU DAT適合フォーマット				554	
550	4Kページ DAT適合	11ビット 領域第1 DMAAS 16E	11ビット 領域第2 DMAAS 8P	11ビット 領域第3 DMAAS 4T	8ビット ページ・ テーブル DMAAS 1M	12ビット バイト・ オフセット 4K	556
		11ビット	11ビット	11ビット	20ビット		
552	1Mページ DAT適合	11ビット 領域第1 DMAAS 16E	11ビット 領域第2 DMAAS 8P	11ビット 領域第3 DMAAS 4T	20ビット バイト・オフセット 1M		

【 図 9 】

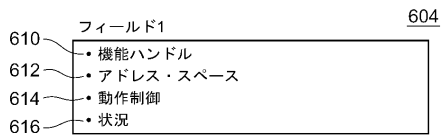
フォーマット		IO拡張アドレス変換フォーマット				576	
570	4Kページ 4K ATテーブル	7ビット 第5レベル DMAAS 16E	9ビット 第3レベル DMAAS 128P	9ビット 第2レベル DMAAS 512G	9ビット 第1レベル DMAAS 1G	9ビット IOPT DMAAS 2M	12ビット バイト・オフセット 4K
		17ビット	17ビット	17ビット	17ビット	12ビット	
572	4Kページ 1M ATテーブル	10ビット 第2レベル DMAAS 16E	17ビット 第1レベル DMAAS 8E	17ビット 第1レベル DMAAS 64T	17ビット IOPT DMAAS 512M	20ビット バイト・オフセット 4K	
574	1Mページ 1M ATテーブル	10ビット 第2レベル DMAAS 16E	17ビット 第1レベル DMAAS 16P	17ビット IOPT DMAAS 128G	20ビット バイト・オフセット 1M		

【図10】



600

【図11】



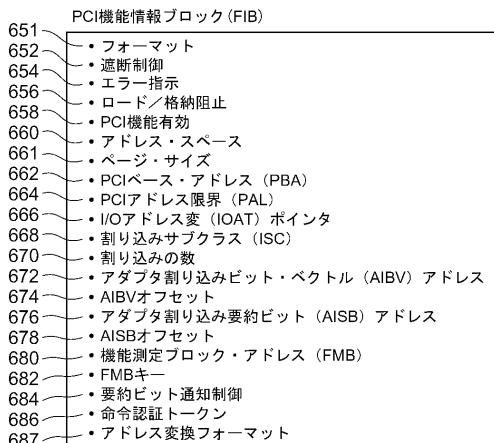
604

【図12】



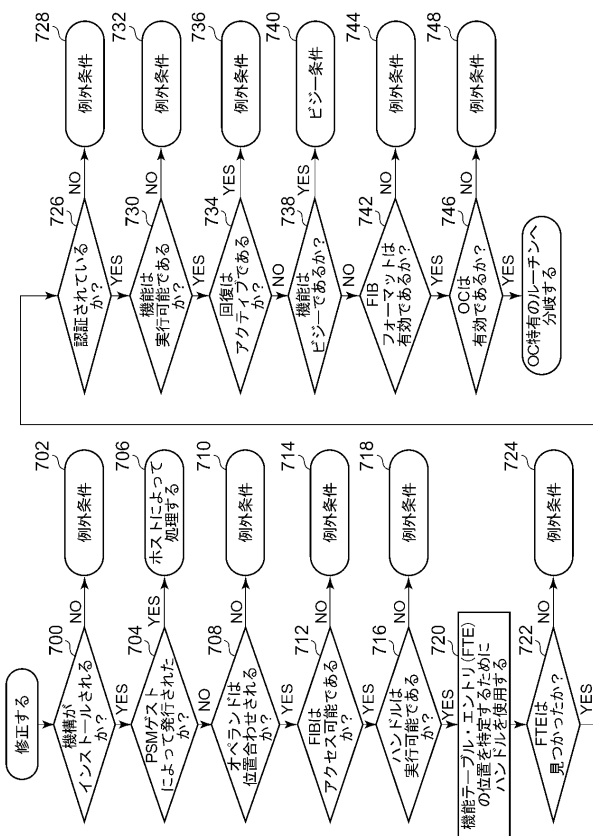
606

【図13】

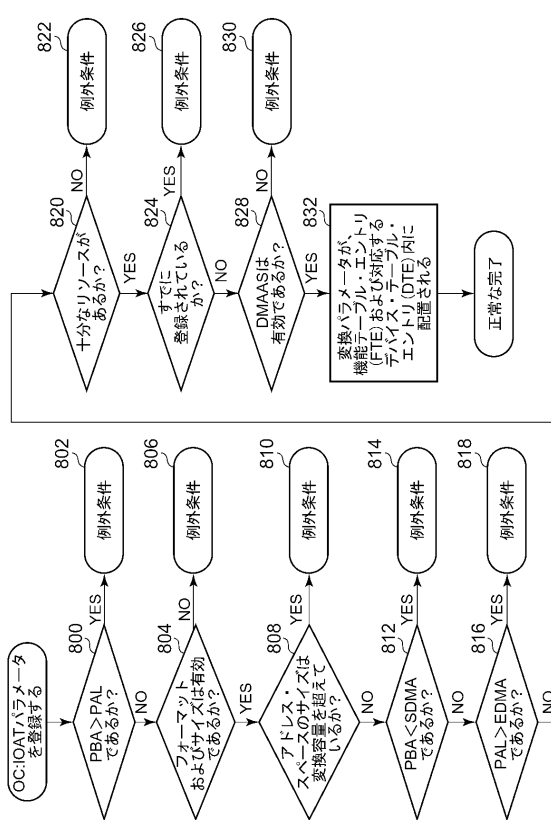


650

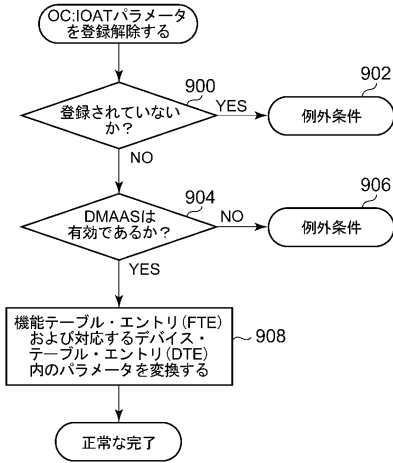
【図14】



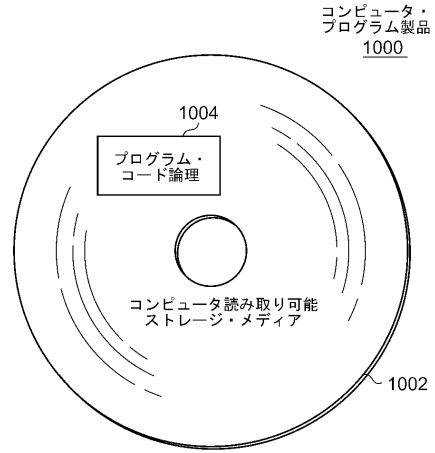
【図15】



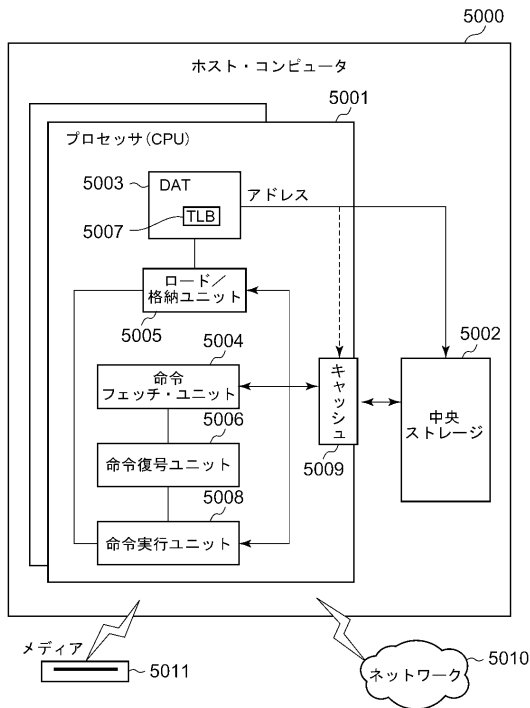
【図16】



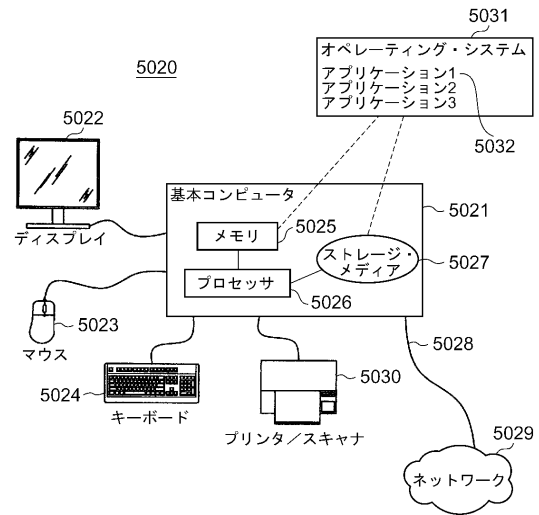
【図17】



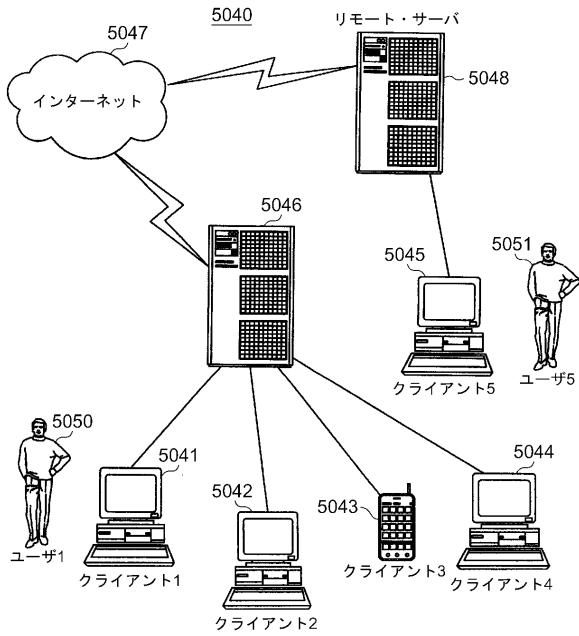
【図18】



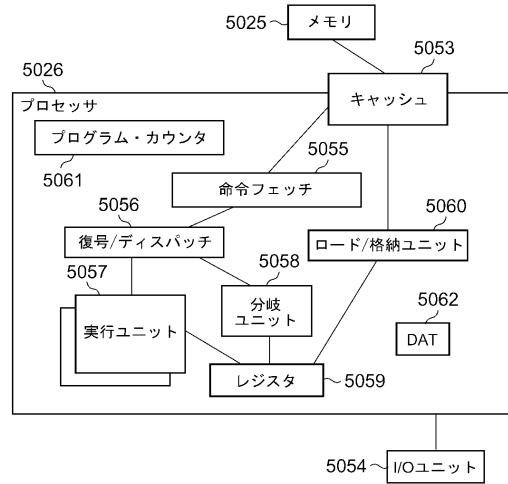
【図19】



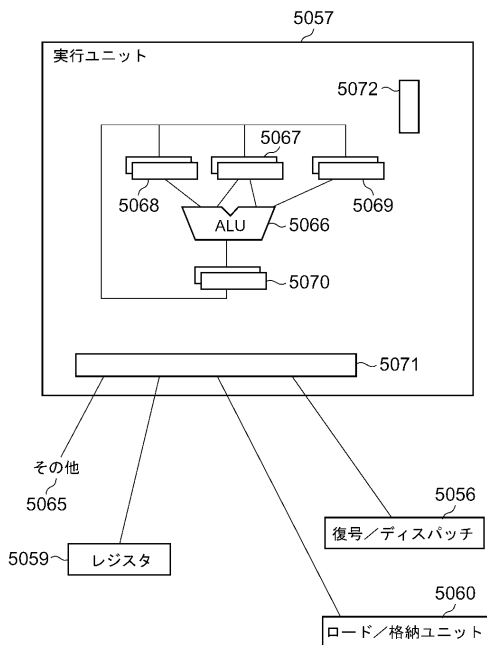
【図20】



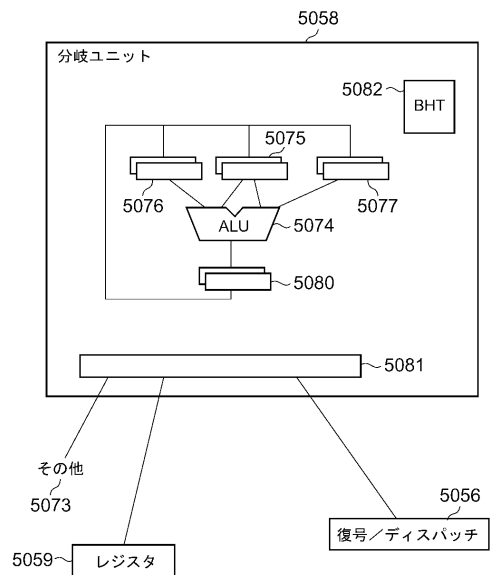
【図21】



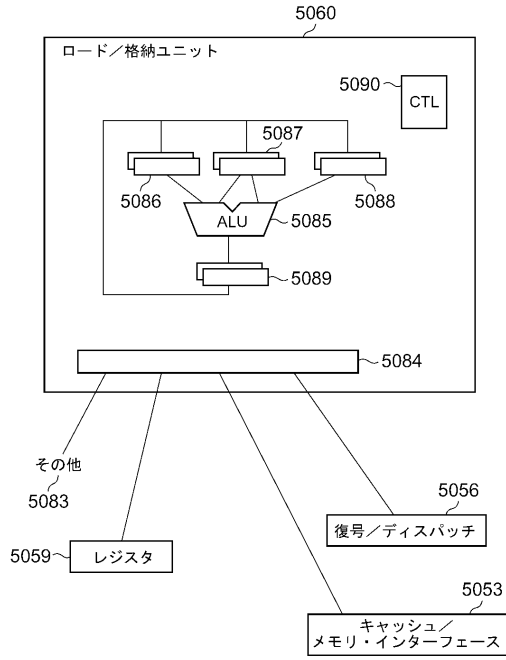
【図22】



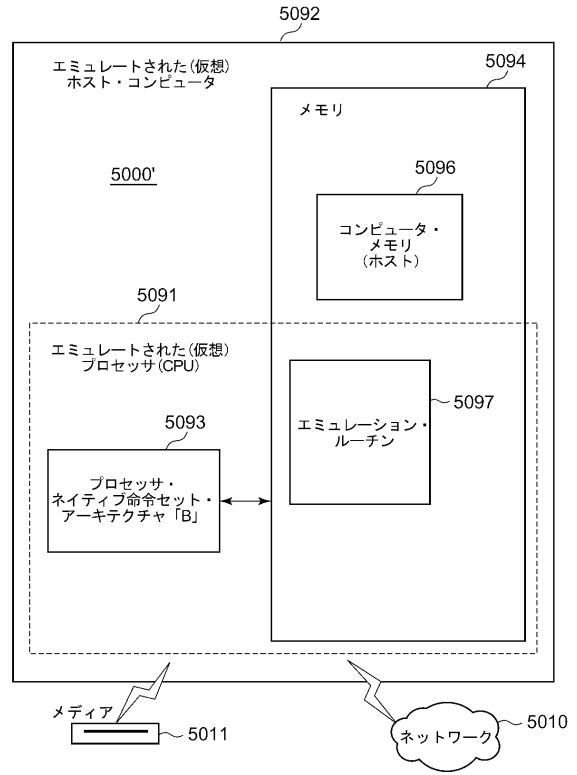
【図23】



【図24】



【図25】



フロントページの続き

- (72)発明者 クラドック、デイヴィッド
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2455
- (72)発明者 グレッグ、トーマス
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2455
- (72)発明者 グレイナー、ダン
アメリカ合衆国95141-1003 カリフォルニア州サンノゼ ベイリー・アヴェニュー555
- (72)発明者 ライス、エリック、ノーマン
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2455

審査官 桜井 茂行

- (56)参考文献 特開昭62-073347(JP,A)
国際公開第2009/087133(WO,A1)
特表2011-509470(JP,A)
特開平02-050744(JP,A)
特開平06-019798(JP,A)

- (58)調査した分野(Int.Cl., DB名)
G06F 12/08-12/12
G06F 13/14
G06F 13/28