

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第6195571号

(P6195571)

(45) 発行日 平成29年9月13日(2017.9.13)

(24) 登録日 平成29年8月25日(2017.8.25)

(51) Int.Cl. F I
G 0 6 F 11/34 (2006.01) G O 6 F 11/34 1 6 6
G 0 6 F 11/36 (2006.01) G O 6 F 11/36 1 3 6

請求項の数 12 (全 41 頁)

(21) 出願番号	特願2014-543380 (P2014-543380)	(73) 特許権者	390009531
(86) (22) 出願日	平成25年3月1日(2013.3.1)		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2015-514240 (P2015-514240A)		INTERNATIONAL BUSINESS MACHINES CORPORATION
(43) 公表日	平成27年5月18日(2015.5.18)		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/JP2013/001262		New Orchard Road, Armonk, New York 10504, United States of America
(87) 国際公開番号	W02013/136703		
(87) 国際公開日	平成25年9月19日(2013.9.19)	(74) 代理人	100108501
審査請求日	平成28年2月19日(2016.2.19)		弁理士 上野 剛史
(31) 優先権主張番号	13/422,598		
(32) 優先日	平成24年3月16日(2012.3.16)		
(33) 優先権主張国	米国 (US)		

最終頁に続く

(54) 【発明の名称】 低特権状態からのランタイム・インストルメンテーション制御の変更

(57) 【特許請求の範囲】

【請求項1】

スーパーバイザ状態又は低特権状態のいずれかで実行可能なランタイム・インストルメンテーション制御変更(MRIC)命令を実行するための方法をコンピュータに実行させるためのプログラムであって、前記MRIC命令は、ランタイム・インストルメンテーション制御のサブセットのみを設定するためのものであり、前記ランタイム・インストルメンテーション制御の全ては、ランタイム・インストルメンテーション制御ロード(LRIC)特権命令によってロード可能であり、前記方法は、前記コンピュータが、

前記MRIC命令をフェッチすることであって、前記MRIC命令はランタイム・インストルメンテーション制御ブロック(RICCB)のアドレスを含む、フェッチすることと、

プロセッサにより、前記RICCBをフェッチすることであって、前記RICCBは、前記プロセッサの前記ランタイム・インストルメンテーション制御のサブセットを変更するための複数の値を含み、前記ランタイム・インストルメンテーション制御のサブセットは、プログラム実行中、前記プロセッサによって認識されたイベントのランタイム・インストルメンテーション情報を保持するためのランタイム・インストルメンテーション・プログラム・バッファ(RIB)位置のランタイム・インストルメンテーション・プログラム・バッファ現在アドレス(RCA)を含む、フェッチすることと、

前記複数の値を前記ランタイム・インストルメンテーション制御ブロックにロードすることと、

10

20

前記ロードされた複数の値を用いて、ランタイム・インストルメンテーション・イベント情報を前記 R I B に与えることと、
を含む、コンピュータ・プログラム。

【請求項 2】

前記方法が、

前記 R I C C B の前記複数の値の少なくとも 1 つについてのモデル依存の制限された値を得ることと、

前記モデル依存の制限された値を更新値として 前記ランタイム・インストルメンテーション制御ブロック にロードすることと、

をさらに含む、請求項 1 に記載のコンピュータ・プログラム。

10

【請求項 3】

報告グループ・サイズにより前記モデル依存の制限された値がロードされ、

前記ランタイム・インストルメンテーション制御の K フィールドが第 1 の値を有することに基づいて、現在アドレスを前記 R I C C B の起点アドレスの値に設定することと、

前記ランタイム・インストルメンテーション制御の K フィールドが第 2 の値を有することに基づいて、前記現在アドレスを指定された R I C C B の現在アドレス値の値に設定することと、

をさらに含む、請求項 2 に記載のコンピュータ・プログラム。

【請求項 4】

前記 M R I C 命令内の前記アドレスがダブルワード境界と整列していないと判断すること、及び、

前記 M R I C 命令内に含まれる前記アドレスから前記 R I C C B をフェッチできないと判断すること、

のいずれか 1 つ又は複数の値に基づいて、例外が引き起こされることをさらに含む、請求項 1 ～ 3 のいずれか 1 項に記載のコンピュータ・プログラム。

20

【請求項 5】

前記 R I C C B 内の 1 つ又は複数の値が有効であると判断することをさらに含み、前記判断することは、

ランタイム・インストルメンテーション S ビットが、前記低特権状態からの前記 M R I C 命令の実行が許可されないことを示していると判断すること、

前記ランタイム・インストルメンテーション制御が有効でないと判断すること、及び
ランタイム・インストルメンテーションがイネーブルにされていないと判断すること、
のいずれか 1 つをさらに含む、請求項 1 ～ 4 のいずれか 1 項に記載のコンピュータ・プログラム。

30

【請求項 6】

前記 R I C C B の前記複数の値は、

ランタイム・インストルメンテーション・データ・サンプリングの詳細を管理するための制御と、

ランタイム・インストルメンテーション・データ収集の詳細を管理するための制御と、
プログラム・バッファへのランタイム・インストルメンテーション・データ報告の詳細
を管理するための制御と、

40

命令キャッシュミスの検出を管理するための制御と、

データ・キャッシュミスの検出を管理するための制御と、

報告グループのサイズを管理するための制御と、

次の報告グループが格納される出力プログラム・バッファ内の現在アドレスを管理するための制御と、

呼び出しタイプ、リターン・タイプ、及び転送タイプ分岐に関するデータのデータ収集を管理するための制御と、

正しく又は誤って予測され、また成立した又は成立しなかった分岐のデータ収集を管理するための制御と、

50

のうちの1つ又は複数をさらに含む、請求項1～5のいずれか1項に記載のコンピュータ・プログラム。

【請求項7】

前記MRIC命令は第1のコンピュータ・アーキテクチャに対して定められ、前記フェッチすること及び構文解析することは、代替的なコンピュータ・アーキテクチャの第2のプロセッサにより実行され、前記第2のプロセッサにより前記フェッチすること及び前記構文解析することは、

前記第1のコンピュータ・アーキテクチャにおける前記MRIC命令の実行をエミュレートするためのソフトウェア・ルーチンを識別することと、

前記ソフトウェア・ルーチンを用いて前記MRIC命令を実行することと、
を含む、請求項1～6のいずれか1項に記載のコンピュータ・プログラム。

10

【請求項8】

エラーに遭遇した場合、前記ランタイム・インストルメンテーション制御ブロックの値は更新されない、請求項1～7のいずれか1項に記載のコンピュータ・プログラム。

【請求項9】

スーパーバイザ状態又は低特権状態のいずれかからランタイム・インストルメンテーション制御変更(MRIC)命令を実行するためのコンピュータ実装方法であって、前記MRIC命令は、ランタイム・インストルメンテーション制御のサブセットのみを設定するためのものであり、前記ランタイム・インストルメンテーション制御の全ては、ランタイム・インストルメンテーション制御ロード(LRIC)特権命令によってロード可能であり、前記方法は、コンピュータが、

20

前記MRIC命令をフェッチすることであって、前記MRIC命令はランタイム・インストルメンテーション制御ブロック(RICCB)のアドレスを含む、フェッチすることと、

プロセッサにより、前記RICCBをフェッチすることであって、前記RICCBは、前記プロセッサの前記ランタイム・インストルメンテーション制御のサブセットを変更するための複数の値を含み、前記ランタイム・インストルメンテーション制御のサブセットは、プログラム実行中、前記プロセッサによって認識されたイベントのランタイム・インストルメンテーション情報を保持するためのランタイム・インストルメンテーション・プログラム・バッファ(RIB)位置のランタイム・インストルメンテーション・プログラム・バッファ現在アドレス(RCA)を含む、フェッチすることと、

30

前記複数の値を前記ランタイム・インストルメンテーション制御ブロックにロードすることと、

前記ロードされた複数の値を用いて、ランタイム・インストルメンテーション・イベント情報を前記RIBに与えることと、
を含む、方法。

【請求項10】

前記RICCBの前記複数の値の少なくとも1つについてのモデル依存の制限された値を得ることと、

前記モデル依存の制限された値を更新値として前記ランタイム・インストルメンテーション制御ブロックにロードすることと、
をさらに含む、請求項9に記載の方法。

40

【請求項11】

スーパーバイザ状態又は低特権状態のいずれかからランタイム・インストルメンテーション制御変更(MRIC)命令を実行するためのコンピュータ・システムであって、前記MRIC命令はランタイム・インストルメンテーション制御のサブセットのみを設定するためのものであり、前記ランタイム・インストルメンテーション制御の全ては、ランタイム・インストルメンテーション制御ロード(LRIC)特権命令によりロード可能である、前記コンピュータ・システムは、

方法を実施するように構成されたランタイム・インストルメンテーション・モジュール

50

を含むコンピュータ・プロセッサを含み、前記方法は、

前記MR I C命令をフェッチすることであって、前記MR I C命令はランタイム・インストールメンテーション制御ブロック（R I C C B）のアドレスを含む、フェッチすることと、

プロセッサにより、前記R I C C Bをフェッチすることであって、前記R I C C Bは、前記プロセッサの前記ランタイム・インストールメンテーション制御のサブセットを変更するための複数の値を含み、前記ランタイム・インストールメンテーション制御のサブセットは、プログラム実行中、前記プロセッサによって認識されたイベントのランタイム・インストールメンテーション情報を保持するためのランタイム・インストールメンテーション・プログラム・バッファ（R I B）位置のランタイム・インストールメンテーション・プログラム・バッファ現在アドレス（R C A）を含む、フェッチすることと、

10

前記複数の値を前記ランタイム・インストールメンテーション制御ブロックにロードすることと、

前記ロードされた複数の値を用いて、ランタイム・インストールメンテーション・イベント情報を前記R I Bに与えることと、

を含む、コンピュータ・システム。

【請求項12】

前記R I C C Bの前記複数の値の少なくとも1つについてのモデル依存の制限された値を得ることと、

前記モデル依存の制限された値を更新値として前記ランタイム・インストールメンテーション制御ブロックにロードすることと、

20

をさらに含む、請求項11に記載のシステム。

【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内の処理に関し、より具体的には、低特権状態（lesser-privileged state）からランタイム・インストールメンテーション制御（run-time-instrumentation controls）を変更することに関する。

【背景技術】

30

【0002】

コンピュータ・プロセッサは、ますます複雑化する分岐予測及び命令キャッシング論理を用いて、プログラム又は命令ストリームを実行する。これらのプロセスは、命令スループット、従って処理性能を高めるために導入されてきた。性能を向上させるための論理を導入すると、特定のソフトウェア・アプリケーションがコンピュータ・プロセッサ上でどのように実行されるかを確実に予測することが困難になる。ソフトウェア開発プロセスの際、多くの場合、機能性および性能の間にバランスがとられる。ソフトウェアは、そのソフトウェアを実行している基礎をなすハードウェアから1又は複数の抽象化レベルで実行される。ハードウェアが仮想化されるとき、付加的な抽象化層が導入される。性能向上論理及び種々の抽象化層が導入されると、プログラムが実行されているときにハードウェア・レベルで何が実際に行われているかを完全に理解することは困難である。この情報なしに、ソフトウェア開発者は、ソフトウェア・アプリケーションを最適化するために、実行継続時間、メモリ使用率、スレッドの数等といった、より抽象的な方法を使用する。

40

【先行技術文献】

【非特許文献】

【0003】

【非特許文献1】「IBM（登録商標） z/Architecture Principles of Operation」、出版番号SA22-7932-08、第9版、2010年8月

【発明の概要】

50

【発明が解決しようとする課題】

【0004】

ハードウェア特有の情報が利用可能であるとき、この情報は、典型的には、事後に開発者に与えられ、且つ、高レベルで、集約された形で与えられ、及び／又は他のプログラムの活動及びオペレーティング・システムに散在されており、そのため、ソフトウェア・アプリケーションの効率性及び正確性に影響を及ぼし得る問題の識別が困難になる。

【課題を解決するための手段】

【0005】

実施形態は、低特権状態からランタイム・インストルメンテーション制御変更 (M R I C, Modifying Run-time-Instrumentation Controls) のための方法、システム、及びコンピュータ・プログラム製品を含む。M R I C 命令がフェッチされる。M R I C 命令は、ランタイム・インストルメンテーション制御ブロック (R I C C B, Run-time-Instrumentation Control Block) のアドレスを含む。M R I C 命令内に含まれるアドレスに基づいて、R I C C B がフェッチされる。R I C C B は、プロセッサのランタイム・インストルメンテーション制御のサブセットを変更するための値を含む。ランタイム・インストルメンテーション制御のサブセットは、ランタイム・インストルメンテーション・プログラム・バッファ (R I B, Runtime Instrumentation program Buffer) 位置のランタイム・インストルメンテーション・プログラム・バッファ現在アドレス (R C A, Runtime instrumentation program buffer Current Address) を含む。R I B は、プログラム実行中、プロセッサにより認識されるイベントのランタイム・インストルメンテーション情報を保持する。R I C C B の値が、ランタイム・インストルメンテーション制御内にロードされる。ランタイム・インストルメンテーション制御内にロードされた値に基づいて、イベント情報が R I B に与えられる。

【0006】

本発明の技術を通じて、付加的な特徴及び利点の実現される。本発明の他の実施形態及び態様は、本明細書において詳細に説明され、特許請求される本発明の一部と見なされる。利点及び特徴に関して本発明をより良好に理解するために、説明及び図面を参照されたい。

【0007】

本発明と見なされる主題は、本明細書の最後にある特許請求の範囲において具体的に示され、明確に特許請求されている。本発明の上記及び他の特徴、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】

【0008】

【図1】 1つの実施形態における例示的なホスト・コンピュータ・システムを示す図である。

【図2】 1つの実施形態における例示的なエミュレーション・ホスト・コンピュータ・システムを示す図である。

【図3】 1つの実施形態における例示的なコンピュータ・システムを示す図である。

【図4】 1つの実施形態における例示的なコンピュータ・ネットワークを示す図である。

【図5】 1つの実施形態におけるコンピュータ・システムの要素を示す図である。

【図6】 1つの実施形態におけるコンピュータ・システムの詳細な要素を示す。

【図7】 1つの実施形態におけるコンピュータ・システムの詳細な要素を示す。

【図8】 1つの実施形態におけるコンピュータ・システムの詳細な要素を示す。

【図9】 1つの実施形態による、プロセッサのランタイム・インストルメンテーションのためのシステムの概略図を示す。

【図10】 1つの実施形態におけるランタイム・インストルメンテーション制御変更 (M R I C, Modify Run-time-Instrumentation controls) 命令を示す。

【図11】 1つの実施形態における、低特権状態のプログラムから M R I C 命令を開始するためのプロセス・フローを示す。

10

20

30

40

50

【図 1 2】付加的な実施形態における、低特権状態のプログラムから M R I C 命令を開始するためのプロセス・フローを示す。

【図 1 3】1つの実施形態における、特権状態 (privileged state) により設定可能な制御を含むランタイム・インストルメンテーション制御の制御ブロック (R I C C B, Run-time-Instrumentation Controls Control Block) の一部を示す。

【図 1 4】1つの実施形態における、準特権 (semi-privileged) ビット (K) が 1 に設定されるとき R I C C B 制御ブロックの一部を示す。

【図 1 5】1つの実施形態における、準特権ビット (K) が 0 に設定されるとき R I C C B 制御ブロックの一部を示す。

【図 1 6】1つの実施形態による報告グループ (reporting group) を示す。

【図 1 7】1つの実施形態におけるコンピュータ・プログラム製品を示す。

【発明を実施するための形態】

【0 0 0 9】

本発明の 1つの実施形態は、低特権状態からランタイム・インストルメンテーション制御のサブセットを更新するためのシステム、方法、及びコンピュータ・プログラム製品である。1つの実施形態において、現在のランタイム・インストルメンテーション制御の有効性、ランタイム・インストルメンテーション制御の現在の状態、及びフラグ (K) が低特権状態の構成が可能であることを示すことに基づいて、低特権状態プログラムは、ランタイム・インストルメンテーション制御の大部分又は全てを変更することができる。低特権状態プログラムは、ランタイム・インストルメンテーション制御変更 (M R I C, Modify Run-time-Instrumentation Controls) 命令を実行して、ランタイム・インストルメンテーション制御を更新する。M R I C 命令は、ランタイム・インストルメンテーション制御の許可の設定に基づいて、多数のランタイム・インストルメンテーション制御の更新を可能にするが、ランタイム・インストルメンテーション制御の大部分は、M R I C 命令を用いて低特権状態プログラムにより更新することができる。

【0 0 1 0】

図 1 は、1つの実施形態におけるホスト・コンピュータ・システム 5 0 の代表的なコンポーネントを示す。コンピュータ・システムにおいて、コンポーネントの他の構成を用いることもできる。代表的なホスト・コンピュータ・システム 5 0 は、主ストア (コンピュータ・メモリ) 2 と通信する 1つ又は複数のプロセッサ 1、並びにストレージ・デバイス 1 1 及び他のコンピュータ又は S A N 等と通信するためのネットワーク 1 0 への I/O インターフェースを含む。プロセッサ 1 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。プロセッサ 1 は、プログラム・アドレス (仮想アドレス) をメモリの実アドレスに変換するための動的アドレス変換 (dynamic address translation、D A T) 3 を有することができる。D A T 3 は、典型的には、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 (translation lookaside buffer、T L B) 7 を含むので、コンピュータ・メモリ 2 のブロックへの後のアクセスでは、アドレス変換の遅延を必要としない。典型的には、キャッシュ 9 が、コンピュータ・メモリ 2 とプロセッサ 1 との間に用いられる。キャッシュ 9 は、1つより多い C P U が利用可能な大型キャッシュと、大型キャッシュと各 C P U 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。幾つかの実施形態においては、下位レベル・キャッシュが分割されて、命令のフェッチ及びデータ・アクセスのための別個の下位レベル・キャッシュを与える。1つの実施形態において、命令は、キャッシュ 9 を介して、命令フェッチ・ユニット 4 により、コンピュータ・メモリ 2 からフェッチされる。命令は、命令デコード・ユニット 6 においてデコードされ、命令実行ユニット 8 に (幾つかの実施形態においては、他の命令とともに) ディスパッチされる。典型的には、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命令実行ユニット 8 が用いられる。命令は、命令実行ユニット 8 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コンピュータ・メモリ 2 からオペランドにアクセスする。コンピュータ・メモリ 2 からオペ

10

20

30

40

50

ランドにアクセスする（オペランドをロード又はストア（格納）する）場合、典型的には、実行される命令の制御下で、ロード・ストア・ユニット5が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。

【0011】

図2において、図1のホスト・コンピュータ・システム50などのホスト・アーキテクチャのホスト・コンピュータ・システム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21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21において、ホスト・プロセッサ（CPU）1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）29であり、ホスト・コンピュータ・システム50のプロセッサ1のもの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ネイティブ・プロセッサ27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21は、ネイティブ・プロセッサ27がアクセス可能なメモリ22を有する。1つの実施形態において、メモリ22は、コンピュータ・メモリ2の部分と、エミュレーション・ルーチン・メモリ23の部分とに区分化される。コンピュータ・メモリ2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・システム21のプログラムに利用可能である。ネイティブ・プロセッサ27は、エミュレーション・ルーチン・メモリ23から取得された、エミュレートされたプロセッサ29のもの以外のアーキテクチャのアーキテクチャ化命令セットのネイティブ命令を実行し、且つ、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、コンピュータ・メモリ2内のプログラムによる実行のためにホスト命令にアクセスすることができる。ホスト・コンピュータ・システム50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、（汎用レジスタ、及び仮想アドレスの動的変換などの）ネイティブ・プロセッサ27において利用可能な機能を利用することもできる。専用ハードウェア及びオフロード・エンジン

10

20

30

【0012】

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、プログラマ、今日では通常「C」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用いられる。ストレージ媒体内に格納されたこれらの命令は、z/Architecture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、pSeries（登録商標）サーバ及びxSeries（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）、Sun Microsystemsなどによって製造されたハードウェアを用いた種々のマシン上でLinux（登録商標）を実行しているマシンにおいて実行することができる。z/Architecture（登録商標）下でそのハードウェア上で実行することに加えて、Linux（登録商標）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、UMX、又はFundamental Software, Inc.（FSI）、又はPlatform Solutions, Inc.（PSI）によるエミュレーションを用いるマシンを使用することもできる。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

40

50

ッサのアーキテクチャをエミュレートする。

【0013】

エミュレートされたホスト・コンピュータ・システム21のコンポーネントの1つ又は複数は、その全体を引用により本明細書に組み入れる非特許文献1にさらに記載される。IBMは、米国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
ンズ・コーポレーションの登録商標である。本明細書で使用する他の名称は、インター
ナショナル・ビジネス・マシーンズ・コーポレーション又は他の会社の登録商標、商標、
又は製品名であり得る。

【0014】

ネイティブ・プロセッサ27は、典型的に、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・ルーチン・メモリ23内に格納
されたエミュレーション・ソフトウェアを実行して、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施する。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、1つ又は複数のエミュレートされたマシン命令を、ネイティブ
・プロセッサ27により実行するための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をキャッシュに入れて、より高速の変換を
達成することができる。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・アーキテクチャのアーキテクチャ規則を維持して、エミュレートされるプロセッサ
のために書かれたオペレーティング・システム及びアプリケーションが正確に動作するこ
とを保証する。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、エミュレートされるプロセッサ29上で実行する
ように設計されたオペレーティング・システム又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27上で実行できるよう
にする。

【0015】

エミュレートされる特定の命令がデコードされ、個々の命令の機能を実施するためのサ
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29の機能をエミュレートす
るエミュレーション・ソフトウェアの機能は、例えば、「C」サブルーチン若しくはドラ
イバにおいて、又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。

【0016】

1つの実施形態において、本発明は、ソフトウェア(ライセンス内部コード、ファーム
ウェア、マイクロコード、ミリコード、ピココードなどと呼ばれる場合もあるが、そのい
ずれも本発明の1つ又は複数の態様と整合性がある)により実施することができる。図1
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・コードには、ホスト・コ
ンピュータ・システム50のCPU(中央演算処理ユニット)1としても知られるプロセ
ッサにより、長期ストレージ媒体、CD-ROMドライブ、テープドライブ、又はハード
ドライブなどのストレージ・デバイス11から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などの、デ
ータ処理システムと共に使用す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よく、又はコンピュータ・メモリ
2からユーザに分散させてもよく、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10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

10

20

30

40

50

ータ・システムに分散させてもよい。

【0017】

代替的に、コンピュータ・メモリ2においてプログラム・コードを具体化し、プロセッサ・バス（図示せず）を用いて、プロセッサ1によりプログラム・コードにアクセスすることができる。こうしたプログラム・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・システムと、1つ又は複数のアプリケーション・プログラムとを含む。プログラム・コードは、通常、ストレージ・デバイス11などの高密度媒体からコンピュータ・メモリ2にページングされ、そこでプロセッサ1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、本明細書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、磁気テープ等を含む）上に格納されたとき、「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。

【0018】

図3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3のシステム100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101を含む。ベース・コンピュータ・システム101は、1つ又は複数のプロセッサ106と、周知の技術に従って1つ又は複数のプロセッサ106とベース・コンピュータ・システム101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（図示せず）とを含む。バスは、プロセッサ106を、メモリ105、及び、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長期ストレージ107に接続する。ベース・コンピュータ・システム101はまた、バスを介して、1つ又は複数のプロセッサ106を、キーボード104、マウス103、プリンタ／スキャナ110、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などの任意のユーザ・インターフェース機器であってもよい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、1つ又は複数のプロセッサを、LCDスクリーン又はモニタなどのディスプレイ装置102にも接続する。

【0019】

ベース・コンピュータ・システム101は、ネットワーク109と通信する108ことの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の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（登録商標）、又はモデムである。或いは、ベース・コンピュータ・システム101は、セルラー・デジタル・パケット・データ（CDPD）カードなどの無線インターフェースを用いて通信することができる。ベース・コンピュータ・システム101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）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。

【0020】

図4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200を示す。データ処理ネットワーク200は、各々が複数の個々のワークステーション201、202、203、204、及び／又は図3のベース・コンピュータ・システム101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。

【0021】

プログラミング・コード111をメモリ105において具体化し、プロセッサ・バスを用いてプロセッサ106により、このプログラミング・コード111にアクセスすることができる。こうしたプログラミング・コードは、種々のコンピュータ・コンポーネントの機能及び相互作用を制御するオペレーティング・システムと、1つ又は複数のアプリケーション・プログラム112とを含む。プログラム・コードは、通常、長期ストレージ107から高速メモリ105にページングされ、そこでプロセッサ106による処理のために利用可能になる。ソフトウェア・プログラミング・コードをメモリ内、物理媒体上で具体化し、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、本明細書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、磁気テープ等を含む）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。

10

【0022】

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、主ストア（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。

20

【0023】

引き続き図4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）206、又はデータ・リポジトリにアクセスすることができ、且つ、ワークステーション205から直接アクセスすることもできるアプリケーション・サーバ（遠隔サーバ）208などの、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206は、各々のネットワーク207へのエントリ・ポイントとして働く。ゲートウェイは、1つのネットワーキング・プロトコルを別のものに接続するときに必要なとされる。ゲートウェイ・コンピュータ206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207）に結合することが好ましい。ゲートウェイ・コンピュータ206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101、201、202、203及び204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・マシーンズ・コーポレーションから入手可能なIBM（登録商標）eServer（登録商標）zSeries（登録商標）z9（登録商標）サーバを用いて実装することができる。

30

【0024】

1つの実施形態において、本発明を具体化するソフトウェア・プログラミング・コードには、図3の長期ストレージ107などの長期ストレージ媒体から、ベース・コンピュータ・システム101のプロセッサ106によりアクセスされる。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、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、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210及び211に分散させてもよく、又はこうした他のシステムのユーザが用いるために、ネットワーク上で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。

40

【0025】

図5を参照すると、プロセッサ106に関する例示的なプロセッサの実施形態が示され

50

る。プロセッサ106の性能を向上させるために、バッファ・メモリ・ブロックに1つ又は複数のレベルのキャッシュ303が用いられる。キャッシュ303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バイト、128バイト、又は256バイトのメモリ・データである。1つの実施形態において、命令をキャッシュに入れるために、データをキャッシュに入れるのとは別個のキャッシュが用いられる。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・システムのメモリ105などの主ストレージは、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303を有するプロセッサ・システムにおいて、メモリ105は、典型的にはより高速であり、且つ、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。メモリ105は、オペレーティング・システムによってメモリ105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」する。

【0026】

プログラム・カウンタ（命令カウンタ）311は、実行される現在の命令（current instruction）のアドレスを常時監視している。z/Architecture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ビットに切り捨てされ得る。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、コンピュータのプログラム状況ワード（program status word、PSW）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（即ち、プログラム環境からオペレーティング・システム環境への現在のコンテキストが切り替わる。）。プログラムがアクティブでない間、プログラムのPSWはプログラム・カウンタ値を維持し、オペレーティング・システムが実行されている間、オペレーティング・システムの（PSW内の）プログラム・カウンタが用いられる。1つの実施形態において、プログラム・カウンタは、現在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。縮小命令セット・コンピューティング（RISC, Reduced Instruction Set Computing）命令は、典型的には固定長であり、複合命令セット・コンピューティング（CISC, Complex Instruction Set Computing）命令は、典型的には可変長である。IBM（登録商標） z/Architecture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CISC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）操作のいずれかにより、プログラム・カウンタ311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（条件コードなど）と共にPSW内に保存され、新しいプログラム・カウンタ値がロードされ、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。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311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。

【0027】

1つの実施形態において、プロセッサ106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305が用いられる。命令フェッチ・ユニット305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。1つの実施形態において、命令フェッチ・ユニット305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ためのプリフェッチ技術を用いる。例えば、命令フェッチ・ユニット305は、次の順次命令を含む16バイトの命令と、付加的なバイトの更なる順次命令とをフェッチすることができる。

【0028】

次に、フェッチされた命令が、プロセッサ106により実行される。1つの実施形態に

において、フェッチされた命令は、命令フェッチ・ユニット 305 のデコード／ディスパッチ・ユニット 306 に渡される。デコード／ディスパッチ・ユニット 306 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実行ユニット 307、308、及び／又は 310 に転送する。実行ユニット 307 は、命令フェッチ・ユニット 305 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、命令の操作コード（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、メモリ 105、アーキテクチャ化レジスタ 309、又は実行されている命令の即値フィールドのいずれかから、実行ユニット 307 に与えられる。実行の結果は、格納される場合には、メモリ 105、アーキテクチャ化レジスタ 309、又は他のマシン・ハードウェア（制御レジスタ、P SW レジスタ等のような）内に格納される。

10

【0029】

プロセッサ 106 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 307、308、及び 310 を有する。図 6 を参照すると、実行ユニット 307 は、インターフェース論理 407 を介して、アーキテクチャ化レジスタ 309、デコード／ディスパッチ・ユニット 306、ロード・ストア・ユニット 310、及びその他のプロセッサ・ユニット 401 と通信することができる。実行ユニット 307 は、幾つかのレジスタ回路 403、404、及び 405 を用いて、算術論理演算ユニット（ALU）402 が演算対象とする情報を保持することができる。ALU 402 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）などの論理関数を実行する。1 つの実施形態において、ALU 402 は、設計に依存する専用の演算をサポートする。他の回路は、例えば条件コード及び復元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 408 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 406 内に保持され、この出力レジスタ回路 406 が、結果を種々の他の処理機能に転送することができる。他の実施形態においては、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。

20

【0030】

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 307 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ユニット（図示せず）で実行される。実行ユニットは、オペランドに対して操作コードが定めた関数を実行することにより、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 つのアーキテクチャ化レジスタ 309 内に見出されるオペランドに対して、実行ユニット 307 により実行することができる。

30

【0031】

実行ユニット 307 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 のオペランドに格納し、ここで第 3 のオペランドは、第 3 のレジスタであっても又は 2 つのソース・レジスタの一方であってもよい。実行ユニット 307 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、除算のいずれかを含む、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）402 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 402 もあり、浮動小数点のために設計された ALU 402 もある。実施形態において、データは、アーキテクチャに応じて、ビッグエンディアン（最下位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（最下位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（登録商標）z/Architecture は、ビッグエンディアンである。符号付きフィールドは、符号及び大きさとすることができ、アーキテクチャに応じて、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 402 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

40

50

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）ブロックとして記述される。

【0032】

図7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308に送られ、この分岐ユニット308は、分岐履歴テーブル432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、現在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。1つの実施形態において、分岐ユニット308は、複数の入力レジスタ回路427、428及び429と、出力レジスタ回路430とを有するALU426を用いることができる。分岐ユニット308は、例えば、汎用レジスタ、デコード／ディスパッチ・ユニット306、又は他の回路425と通信することができる。

【0033】

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、又は（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。1つの実施形態において、コンテキスト・スイッチ動作により、現在実行中のプログラムについての状態情報が保存され、次いで、起動される別のプログラムについての状態情報がロードされる。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、条件コード、メモリ変換情報、及びアーキテクチャ化レジスタの内容を含む。コンテキスト・スイッチ活動は、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））により単独で又はその組み合わせで実施することができる。

【0034】

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、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、操作コード・フィールドによって、オペランドとして識別される暗黙指定のレジスタ（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、インデックス・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、互いに加算されてメモリ内のオペランドのアドレスをもたらす。ここでの位置（location）は、典型的には、特に断りのない限り、主メモリ（主ストレージ）内の記憶位置を意味する。

【0035】

図8を参照すると、プロセッサは、ロード／ストア・ユニット310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310は、キャッシュ・メモリ・インターフェースを通じてメモリ内のターゲット・オペランドのアドレスを取得し、オペランドをアーキテクチャ化レジスタ309若しくは別のメモリ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、又は、メモリ内のターゲット・オペランドのアドレスを取得し、アーキテクチャ化レジスタ309若しくは別のメモリ位置から取得したデータをメ

メモリ内のターゲット・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310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。ロード／ストア・ユニット310は、アーキテクチャ化レジスタ309、デコード／ディスパッチ・ユニット306、キャッシュ／メモリ・インターフェース若しくは他の要素455と通信することができ、ストレージ・アドレスを計算し、且つ、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、ALU458、及び制御論理463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、当技術分野において周知のように、アウト・オブ・オーダー式動作が、プログラムに対してイン・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。

10

【0036】

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」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、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、図5のDAT312のような種々の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て含み、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においては、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、仮想アドレスに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。DAT312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できる。TLBの内容は、最長時間未使用(Least Recently Used、LRU)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。

20

30

【0037】

プロセッサ106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。1つの実施形態において、キャッシュ・コヒーレンシを保持する際に、「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。

【0038】

図5のI/Oユニット304は、プロセッサ106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304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) z/Series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。

40

【0039】

インストルメンテーション・データは、プロセッサ106の動作に関連したデータである。1つの実施形態において、インストルメンテーション・データ及び他のシステム・レベル・メトリックへのアクセスが制限されていること又は利用できないことがある。コン

50

ピュータ・プロセッサは、特権状態（即ち、スーパーバイザ状態）、及び低特権状態（即ち、問題状態）の下で動作する。特権状態において、プログラムは、特権操作を介して、全てのシステム・リソースにアクセスする（例えば、全ての制御レジスタ及びスーパーバイザ・メモリ空間にアクセスする）ことができる。特権状態は、特権モード又はスーパーバイザ・モードとも呼ばれる。コンピュータ・プロセッサ上で実行されているオペレーティング・システムは、特権状態で動作し得る。低特権状態は、システム・リソースへのアクセスが制限されている非特権状態（non-privileged state）である。例えば、低特権状態で実行されているアプリケーション・プログラムは、制御レジスタへのアクセスが制限されるか又は制御レジスタにアクセスできないことがあり、且つ、オペレーティング・システムにより、アプリケーション・プログラムに割り当てられたユーザ・メモリ空間にしかアクセスすることができない。低特権状態は、典型的には、オペレーティング・システムの制御下で実行されるアプリケーション・プログラムに割り当てられ、特権操作を低特権状態で実施することはできない。低特権状態は、問題状態、問題モード、又はユーザ・モードとしても知られる。

【0040】

低特権状態で実行されているプログラムに書き込み・アクセス可能でない1つのこうした制限されたリソースが、プログラム状況ワード（PSW）である。PSWは、実行される次の命令のプログラム・カウンタと、分岐命令により使用可能な条件コード・フィールドと、インストルメンテーションがイネーブルにされているか又はディスエーブルにされているかを示すためのインストルメンテーション制御フィールドと、命令の順序を制御し、プログラムに割り当てられた特権状態を含むコンピュータ・プロセッサの状態を判断するために用いられる他の情報とを含むことができる。マルチスレッド処理環境においては、複数のプログラムが、利用可能なコンピュータ・プロセッサの容量を共有するか、又はタイムスライスする。プログラムの各々は、関連したPSW、プログラムに割り当てられた主ストレージにアクセスするためのアドレス変換テーブルの起点アドレス（origin address）、汎用レジスタの現在値のセット、制御レジスタ、浮動小数点レジスタ等を含む、コンテキスト情報を有する。現在アクティブな又は制御を行っているPSWは、現（current）PSWと呼ばれる。この現PSWは、現在実行されているプログラムを制御する。コンピュータ・プロセッサは、割り込み能力を有し、この割り込み能力により、例外条件及び外部からの刺激に応答して、コンピュータ・プロセッサが、別のプログラムにより迅速にコンテキストを切り替えることが可能になる。割り込みが発生すると、コンピュータ・プロセッサは、現PSWを、特定のクラスの割り込みのための、旧（old）PSW位置と呼ばれる割り当てられたストレージ位置に入れる。コンピュータ・プロセッサは、第2の割り当てられたストレージ位置から、新（new）PSWをフェッチする。この新しいコンテキストにより、実行される次のプログラムが判断される。1つの実施形態において、これらのストレージ位置は、コンピュータ・プロセッサがアクセス可能なメモリ位置に配置される。コンピュータ・プロセッサが割り込み処理を終了すると、割り込みを処理するプログラムは、旧PSWを含む旧コンテキストを再ロードし、これを再び現PSWとすることができ、これにより、割り込みを受けたプログラムが続行できるようになる。

【0041】

PSWのフィールドは、明示的に（例えば、命令の実行がPSWビットの一部を読み出す場合）、又は暗黙的に（例えば、命令フェッチ、オペランド・フェッチ、アドレス生成計算、アドレス生成ソース等の際に）参照することができる。明示的な参照は、一般に、実行時に実施されるのに対して、暗黙的な参照は、一般に、命令実行中のパイプラインの異なる段階（即ち、命令フェッチ、命令デコード、実行時、及び完了時）に実施される。PSW内の個々のフィールドは、互いに独立して参照又は更新することができる。

【0042】

1つの実施形態において、コンテキストを操作することにより、オペレーティング・システムは、コンピュータ・プロセッサによりランタイム・インストルメンテーションをイネーブルにすることを含む、コンピュータ処理リソースの制御を行う。ランタイム・イン

ストルメンテーションは、オペレーティング・システムの実行の際、並びに、オペレーティング・システムにより実行されるいずれかのソフトウェア・アプリケーションにより、イネーブル又はディスエーブルにすることができる。ランタイム・インストルメンテーションのイネーブル／ディスエーブル状態は、プログラムと関連したPSW内に、コンテキスト情報として保存される。

【0043】

ランタイム・インストルメンテーション(RI)ファシリティは、z/Architectureを実装するモデル上に組み込むことができる。RIファシリティがインストールされ、イネーブルにされると、プログラム実行中、データが、CPUの1つ又は複数の収集バッファ内に収集され、次いで、プログラム・バッファに報告される。格納された情報の各単位は、報告グループと呼ばれる。報告グループの内容は、その内容がプログラム実行中CPUにより認識されたイベントを表す複数の記録から成る。

10

【0044】

ランタイム・インストルメンテーション・ファシリティが構成内にインストールされている場合、PSWフィールド(RIビット)がランタイム・インストルメンテーションをイネーブルにする。ランタイム・インストルメンテーション制御の有効性が、RIビットをオンにする能力を決定するが、RIが1のとき、CPU制御は有効であり、ランタイム・インストルメンテーションがイネーブルにされる。ランタイム・インストルメンテーション・ファシリティは、以下の命令：ランタイム・インストルメンテーション制御ロード(load run-time-instrumentation controls)命令、ランタイム・インストルメンテーション制御変更(modify run-time-instrumentation controls)命令、ランタイム・インストルメンテーション発行(run-time-instrumentation emit)命令、ランタイム・インストルメンテーション・ネクスト(run-time-instrumentation next)命令、ランタイム・インストルメンテーション・オフ(run-time-instrumentation off)命令、ランタイム・インストルメンテーション・オン(run-time-instrumentation on)命令、ランタイム・インストルメンテーション制御ストア(store run-time-instrumentation controls)命令、及びランタイム・インストルメンテーション制御テスト(test run-time-instrumentation controls)命令を含むことができる。

20

【0045】

ランタイム・インストルメンテーション制御ロード(LRIC, Load Run-time-Instrumentation Controls)命令は、ランタイム・インストルメンテーションを制御するランタイム・インストルメンテーション制御を初期化する。ランタイム・インストルメンテーション制御変更(MRIC, Modify Run-time-Instrumentation Controls)命令は、初めにLRICによって確立されたランタイム・インストルメンテーション制御の全て又はサブセットを変更する。ランタイム・インストルメンテーション発行(RIEMIT, Run-time-Instrumentation EMIT)命令は、汎用レジスタの値を収集バッファ内に格納することにより、これを収集する。ランタイム・インストルメンテーション・ネクスト(RINEXT, Run-time-Instrumentation NEXT)命令は、RINEXTの後の次の順次命令(NSI, Next Sequential Instruction)の指向サンプリングを実施する。ランタイム・インストルメンテーション・オフ(RIOFF, Run-time-Instrumentation OFF)命令は、ランタイム・インストルメンテーションをディスエーブルにする。ランタイム・インストルメンテーション・オン(RION, Run-time-Instrumentation ON)命令は、ランタイム・インストルメンテーションをイネーブルにする。ランタイム・インストルメンテーション制御ストア(STRIC, STore Run-time-Instrumentation Controls)命令は、ランタイム・インストルメンテーション制御の現在の値を指定されたストレージ位置に入れる。ランタイム・インストルメンテーション制御テスト(TRIC, Test Run-time-Instrumentation Controls)命令は、ランタイム・インストルメンテーション制御を検査する。有効である場合、制御変更インジケータの状態が設定される。

30

40

【0046】

ランタイム・インストルメンテーション・ファシリティは、測定警報外部割り込み(me

50

asurement-alert external interruption) を保留中 (pending) にする能力を含む。ランタイム・インストルメンテーションによって収集され、プログラム・バッファに報告される情報の一部はモデル依存であり、従って、定義されない。ランタイム・インストルメンテーション・ファシリティにより与えられるサンプル及びデータは、性能特性の統計的推定を目的としており、概ね正確であり、繰り返し可能でないことがある。例えば、サンプリング・モードに関係なく、例外を発生させた又は特定のシステムの内部活動と関連付けられたサンプル命令が報告グループのストアをもたすかどうか、そして、ストアされる場合、ランタイム・インストルメンテーション・データ内に含まれるモデル依存データに影響を及ぼすかどうかは、予測不可能である。

【0047】

10

収集バッファは、その内容がプログラム実行中プロセッサにより認識されるイベントについて報告する記録のセットをキャプチャするために用いられる。例として、1又は複数の成立した分岐の実行、トランザクション実行アボート (abort) イベント、及び R I E M I T 命令のオペランドが挙げられる。R I E M I T 命令の実行は、汎用レジスタの値を収集バッファ内に格納することにより、これを収集する。付加的なデータを収集すること、及び／又は命令データ・バッファなどの他のバッファに格納することが可能である。

【0048】

報告は、報告制御に従う。サンプル命令が識別されると、各々の報告制御は、対応する条件のチェックをイネーブルにする。対応する条件が存在する場合、報告グループが形成され、格納される。報告制御がイネーブルにされないか、又はイネーブルにされた報告制御に関して対応する条件が存在しない場合、報告グループは格納されない。サンプル命令について報告されたデータは、命令データ・バッファ及び他のモデル依存のソースから取得され、次いで、このデータを用いて報告グループの1つ又は複数の記録の内容が作成され、1つのこうした記録が命令の記録となる。

20

【0049】

報告グループ・ストアにおいてキャプチャすることができる記録タイプには：フィラー (filler)、追加 (extra)、開始 (begin)、タイムスタンプ (timestamp)、命令 (instruction)、発行 (emit)、T X アボート (TX abort)、呼び出し (call)、リターン (return)、及び転送 (transfer) が含まれる。フィラー記録は、収集バッファ内の有効な記録の数が現在の報告グループ・サイズの報告グループを満たすのに十分でない場合に、報告グループにおいて使用される。追加記録は、報告グループの追加セクションにおいて使用することができる。開始記録は、第1の報告グループの最初の記録である。タイムスタンプ記録は、第1の報告グループ以外のあらゆる報告グループの記録0として格納される。命令記録は、サンプル命令に関して報告グループが格納されるときに、報告グループの最後の記録として作成される。発行記録は、R I E M I T の正常な実行により作成される。トランザクション実行 (T X, Transaction-Execution) モード・アボート記録は、暗黙的アボート、又は、トランザクション・アボート命令の実行によって作成される。呼び出し記録は、呼び出しタイプ分岐命令として分類される分岐命令の実行によって作成される。リターン記録は、リターン・タイプ命令として分類されるリターン・タイプ分岐命令の実行によって作成される。転送記録は、特定の条件コード基準を満たす分岐命令の実行によって作成される。

30

40

【0050】

図9は、1つの実施形態において実装することができるプロセッサのランタイム・インストルメンテーションのためのシステムの概略図を示す。1つの実施形態において、システム500は、図3のプロセッサ106などの中央演算処理ユニット (C P U) を含む。1つの実施形態において、プロセッサ106は、シングル・プロセッサである。代替的な実施形態において、プロセッサ106は、マルチコア・プロセッサのシングル処理コアである。1つの実施形態において、プロセッサ106は、様々な速度で動作し得る。

【0051】

1つの実施形態において、プロセッサ106は、レジスタ510をさらに含む。レジス

50

タ510は、プロセッサ106により使用するためのデータのワードを格納することができるハードウェア・レジスタである。レジスタ510は、プロセッサ106によりアクセス可能なデータ・ビットを格納するための1又は複数のラッチを含む。レジスタ510は、例えば、汎用レジスタ及び制御レジスタを含むことができる。プロセッサ106は、レジスタ510と通信するインストルメンテーション・モジュール506を付加的に含む。インストルメンテーション・モジュール506は、プロセッサ106のインストルメンテーションを制御する。インストルメンテーション・モジュール506は、1つ又は複数の成立した分岐の実行経路、トランザクション実行アボート・イベント、種々のランタイム・オペランド、タイムスタンプ情報等といったインストルメンテーション・データを、プロセッサ106から直接収集するように構成される。インストルメンテーション・モジュール506は、プロセッサ106からインストルメンテーション・データを収集し、そのインストルメンテーション・データを収集バッファ508に格納する。1つの実施形態において、収集バッファ508は、インストルメンテーション・モジュール506から受け取ったデータを収集する循環バッファ(circular buffer)であり、循環バッファが満たされると、該循環バッファは、最も古いデータを新しいデータで上書きする。

10

【0052】

プロセッサ106は、1つ又は複数のオペレーティング・システム516及び1つ又は複数のアプリケーション518を実行する。1つ又は複数のオペレーティング・システム516及び1つ又は複数のアプリケーション518は、ハードドライブ、CD-ROM、フラッシュ・メモリ等のようなストレージ520内に格納され、必要に応じてストレージ520からランタイム・メモリ504内にロードされる、ページと呼ばれる、現在実行されているオペレーティング・システム及び／又はアプリケーションの1つ又は複数のアクティブ部分を格納するために予約された主メモリ514のランタイム・メモリ504領域にロードされる。1つの実施形態において、オペレーティング・システムの各々は、ハイパーバイザ(図示せず)によって管理され、且つ、プロセッサ106により実行される仮想マシンとして実行される。

20

【0053】

1つの実施形態において、プロセッサ106は、主メモリ514から、現在実行されているオペレーティング・システム又はアプリケーションに関する主メモリ514内のPSWデータ512からのPSWをレジスタ510にロードし、例えばレジスタ510において、1つ又は複数のプロセッサ設定(processor settings)を設定する。1つの実施形態において、レジスタ510内のPSWは、インストルメンテーション・モジュール506をイネーブルにし、制御するための1又は複数のビットを含む。

30

【0054】

1つ又は複数のアプリケーション518は、特定のオペレーティング・システム上で実行されるようにコンパイルされたソフトウェア・アプリケーション、インタプリタ(例えば、Java(登録商標)上で実行される解釈済みコード、又はオペレーティング・システム・サポート・スレッド(例えば、プロセス管理、デーモン等)を含む。1つ又は複数のオペレーティング・システム516、及び／又は、1つ又は複数のアプリケーション518の各々は、インストルメンテーション・モジュール506をトリガして、インストルメンテーション・データの収集を開始又は停止するための命令を実行することができる。

40

【0055】

1つの実施形態において、1つ又は複数のアプリケーション518の1つは、サンプル命令であると判断された命令を実行し、それにより、サンプル命令の実行の完了時にサンプル・ポイントが作成され、その結果、インストルメンテーション・モジュール506が、収集バッファ508からのアプリケーションの収集データを、アプリケーションがアクセス可能な主メモリ514内のプログラム・バッファ522に移動させる。主メモリ514は、当技術分野において周知の任意のアドレス指定可能メモリとすることができる。1つの実施形態において、主メモリ514は、キャッシュと呼ばれることもある、高速アク

50

セス・バッファ・ストレージを含むことができる。各CPUは、関連したキャッシュを有することができる。付加的な実施形態において、主メモリ514は、ダイナミック・ランダム・アクセス・メモリ（DRAM）である。さらに別の実施形態において、主メモリは、アプリケーションによりアクセス可能な、コンピュータのハードドライブ又はフラッシュ・メモリなどのストレージ・デバイスである。

【0056】

ランタイム・インストルメンテーション機能は、実験室環境において又はオフライン分析のために使用できるだけでなく、実行時のプログラム内のライブ・ソフトウェア環境においても、プログラム制御下でも使用できる新しいファシリティである。特権状態は、プロセッサ106の制御を設定し、ランタイム・インストルメンテーションを管理することができる。低特権状態から制御の大部分を変更する能力をもたらすことにより、ランタイム・インストルメンテーション・ファシリティの柔軟性が強化される。低特権状態は、ランタイム・インストルメンテーションから利益を得る可能性が最も高い。低特権状態は、（1）特権状態が、これが実行している全ての低特権状態プログラムにわたる整合性および制御を維持すること、並びに（2）オペレーティング・システム自体がランタイム・インストルメンテーションを実施しているとき、低特権状態がランタイム・インストルメンテーションを実施する場合も実施しない場合もあることを保証するように、特定の設定の変更が禁止されている。どちらも、スーパーバイザがランタイム・インストルメンテーションを一時的にディスエーブルにし、そのランタイム・インストルメンテーション・コンテキストを保存し、低特権状態プログラムのランタイム・インストルメンテーション・コンテキストを復元し、次に、ランタイム・インストルメンテーションが再びイネーブルにされた状態で低特権状態プログラムを開始する場合、ランタイム・インストルメンテーションを実施することができる。コンテキストの変更が、ランタイム・インストルメンテーション制御の適切な保存／復元を含まない場合、特権状態及び低特権状態の両方において収集されたデータの衝突により、使用できないランタイム・インストルメンテーション・データの集まりになる可能性が高い。

【0057】

1つの実施形態において、低特権状態プログラムをより良好にサポートするために、ランタイム・インストルメンテーション制御変更（MRIC, Modify Run-time-Instruction）命令が定められる。MRIC命令は、レジスタ510において現在アクティブな、ランタイム・インストルメンテーションに関連した特定のプロセッサ106の制御を変更するために使用される準特権命令である。制御は、特権状態によるランタイム・インストルメンテーション制御ロード（LRIC, Load Run-time-Instruction Control）命令の正常な実行（successful execution）により、初めにロードされる。

【0058】

ランタイム・インストルメンテーション制御と相互作用するMRIC命令の能力は、以下で詳細に説明される。MRIC命令の実行が成功した場合、MRIC命令の実行が正常に完了したときにMRIC命令により指定される全ての制御が設定され、さもなければ、MRIC命令の実行が失敗した場合、制御に影響を及ぼさない。

【0059】

図10は、1つの実施形態におけるMRIC命令を示す。1つの実施形態において、MRIC命令600は、操作コード602及び604を含む（この特定の場合は「オペコード」又は「分割オペコード（split opcode）」とも呼ばれる）。オペコード602及び604は、図9のプロセッサ106のようなプロセッサに対してMRIC命令を識別する。MRIC命令600はまた、基底レジスタ・フィールド606（B₂）並びに変位フィールド608及び610のセットをまとめたものから決定されるオペランド・アドレスを含む。基底レジスタ+変位レジスタの内容の和は、MRIC命令によって更新されるランタイム・インストルメンテーション制御設定を含むランタイム・インストルメンテーション制御制御ブロック（RICC B, Run-time-Instrumentation Controls Control Block）の位置を指し示す。変位フィールド608及び610は、基底レジスタ・フィールド60

6により示される基底レジスタ内に含まれるアドレスからの変位を示し、その和はR I C C Bに一致する。図10は、明確にするための、M R I C命令の1つの実施形態を示す。当業者であれば、他の実施形態において、M R I C命令を異なるようにフォーマットすることができ、及び／又は、異なるオペランド及びオペコードを含み得ることを理解するであろう。

【0060】

M R I C命令600は、L R I C命令により更新可能な制御のうちのサブセットだけを更新するために使用される。M R I C命令600によって設定される制御は、L R I C命令によって正常に設定された制御のサブセットに制限される。ランタイム・インストルメンテーション制御は、種々のランタイム・インストルメンテーション制御を更新する能力を含む、M R I C命令の動作を制御する多数のビットを含む。1つの実施形態において、これらのビットは、L R I C命令によってのみ設定することができる全てのランタイム・インストルメンテーション制御のサブセット内に配置される。

10

【0061】

図11は、1つの実施形態における、低特権状態プログラムからM R I C命令を開始するためのプロセス・フローを示す。ブロック702において、プロセッサにより、M R I C命令がフェッチされる。ブロック704において、プロセッサにより、M R I C命令が実行される。ブロック706において、R I C C B内のランタイム・インストルメンテーション制御値が、ランタイム・インストルメンテーション制御内にロードされる。ブロック708において、ランタイム・インストルメンテーション制御は、ロードされた設定に基づいて、情報を図9のプログラム・バッファ522に与える。

20

【0062】

図12は、付加的な実施形態における、低特権状態プログラムからM R I C命令を開始するためのプロセス・フローを示す。1つの実施形態において、図12のプロセス・フローは、図9のインストルメンテーション・モジュール506によって実行される。ブロック802において、低特権状態プログラムより発行されたM R I C命令がフェッチされる。M R I C命令は、オペコード及びオペランドを含む。M R I C命令のオペランドは、ランタイム・インストルメンテーション制御を更新するのに使用されるランタイム・インストルメンテーション制御値を含むR I C C Bである。これらの値は、例えば、ランタイム・インストルメンテーション・データ・サンプリングの詳細を管理するための制御値、ランタイム・インストルメンテーション・データ収集の詳細を管理するための制御値、プログラム・バッファ内に報告するランタイム・インストルメンテーション・データの詳細を管理するための制御値、プログラム・バッファの位置内の現在アドレス等を含む。

30

【0063】

ブロック804において、M R I Cの実行が許可されるかどうか判断される。ランタイム・インストルメンテーションのSビット（L R I Cによってのみ制御される）は、低特権状態プログラムが、M R I C命令の実行を可能にするどうかを決定する。ランタイム・インストルメンテーションSビットが1に設定されていた場合、処理はブロック806に続く。

【0064】

ブロック806において、現在のランタイム・インストルメンテーション制御の有効性ビット（Vビットとも呼ばれる）が1に設定されているかどうか判断される。有効性ビットは、以前にL R I C命令によって設定されたとおりに、プロセッサにおけるランタイム・インストルメンテーション制御のセットの有効性を示す。現在のランタイム・インストルメンテーション制御が有効でない（即ち、以前のL R I C命令により無効とされた）場合、ブロック818において、M R I C命令の正常な実行を行うことができない。1つの実施形態において、M R I C命令の実行の失敗により従来の設定が変更されず、ランタイム制御は以前の値のままである。有効性ビットは、M R I C命令により更新することができない。ブロック808において、M R I C命令が指し示すR I C C Bがフェッチされる。

40

50

【0065】

妥当性ビットが1に設定される（即ち、ランタイム・インストールメンテーション制御が有効である）場合、処理はブロック810に続く。ブロック810において、Kビットがゼロである（即ち、低特権状態プログラムが、ランタイム・インストールメンテーション制御に関して準特権状態で実行されていない）場合、ブロック814において、起点アドレスも制限アドレスも更新されないが、MRICによる更新が許可される他の全ての制御が更新される。Kビットが1である（即ち、低特権状態プログラムが、ランタイム・インストールメンテーション制御に関して準特権状態で実行されている）場合、ブロック812において、起点アドレス及び制限アドレスを含めて、MRICによる更新が許可される全ての制御が更新される。

10

【0066】

図13は、1つの実施形態における、特権状態により設定可能な制御を含むランタイム・インストールメンテーション制御制御ブロック（RICCB）の一部を示す。制御ブロック部分900は、図13を参照して説明されるもの以外の付加的な値を含むことができる。LRIC命令により、制御ブロック部分900に対する変更を実施することができる。

【0067】

制御ブロック部分は、有効性ビット902（Vビット）を含む。有効性ビット902は、以前にLRIC命令により設定されたとおりに、プロセッサにおけるランタイム・インストールメンテーション制御のセットの有効性を示す。

【0068】

制御ブロックはまた、低特権状態プログラムがMRIC命令の実行を許可するかどうかを判断するのに使用されるSビット904も含む。Kビット906は、低特権状態プログラムが、ランタイム・インストールメンテーション制御の起点アドレス及び制限アドレスなどのランタイム・インストールメンテーション制御に関して、準特権状態での実行が許可されるかどうかを示す。Hビット908は、アドレス制御（即ち、起点アドレス、制限アドレス、及び現在のアドレス）が一次仮想アドレス空間を参照するか又はホーム仮想アドレス空間を参照するかを決定する。0ビット910は無視され、0として扱われる。

20

【0069】

低特権状態サンプル報告制御ビット912（Psビット）は、低特権状態プログラムと併せて使用される。低特権状態において、ランタイム・インストールメンテーション制御内のPsビット912がゼロである場合、ランタイム・インストールメンテーションがイネーブルにされているときには、ランタイム・インストールメンテーション制御の報告制御は無視され、従って、報告グループは格納されない。低特権状態にあり、ランタイム・インストールメンテーション制御のPsビット912が1である場合、報告制御がチェックされ、その定められた機能に従って使用される。

30

【0070】

スーパーバイザ状態サンプル報告制御ビット914（Qsビット）は、スーパーバイザ状態プログラムと併せて使用される。スーパーバイザ状態において、ランタイム・インストールメンテーション制御のQsビット914がゼロである場合、ランタイム・インストールメンテーションがイネーブルにされているとき、ランタイム・インストールメンテーション制御の報告制御は無視され、従って、報告グループは格納されない。スーパーバイザ状態において、ランタイム・インストールメンテーション制御のQsビット914が1である場合、報告制御がチェックされ、その定められた機能に従って使用される。

40

【0071】

低特権状態収集バッファ制御ビット916（Pcビット）は、図9の収集バッファ508に対する更新を制御する。低特権状態において、ランタイム・インストールメンテーション制御のPcビット916がゼロである場合、ランタイム・インストールメンテーションがイネーブルにされているときには、ランタイム・インストールメンテーション制御の収集バッファ制御は無視され、収集バッファ508の更新が防止される。低特権状態において、ランタイム・インストールメンテーション制御のPcビット916が1である場合、収集バ

50

ッファ制御がチェックされ、その定められた機能に従って使用される。

【0072】

スーパーバイザ状態収集バッファ制御ビット918（Qcビット）は、収集バッファ508に対する更新を制御する。スーパーバイザ状態において、ランタイム・インストルメンテーション制御内のQcビット918がゼロである場合、ランタイム・インストルメンテーションがイネーブルにされているときには、ランタイム・インストルメンテーション制御の収集バッファ制御は無視され、収集バッファ508の更新は防止される。スーパーバイザ状態において、ランタイム・インストルメンテーション制御内のQcビット918が1である場合、指し示された収集バッファ制御がチェックされ、その定められた機能に従って使用される。

10

【0073】

Gビット920は、停止割り込み（halted interruption）とも呼ばれるランタイム・インストルメンテーションの停止割り込みの保留制御である。Gビット920がゼロである場合、停止割り込みは保留中でない。Gビット920が1である場合、停止割り込みは保留中である。プログラム・バッファ522内の第1の報告グループが書き込まれると、Gビット920はゼロに設定される。つまり、プログラム・バッファの起点アドレスがプログラム・バッファの制限アドレスに等しい場合、Gビット920はゼロに設定される。プログラム・バッファ522内の第1の報告グループ以外のストアが試行された場合であって、ランタイム・インストルメンテーション停止条件が存在しないときには、Gビット920がゼロに設定され、報告グループが格納される。プログラム・バッファ522内の第1の報告グループ以外のストアが試行された場合であって、ランタイム・インストルメンテーション停止条件が存在するときには、Gビット920が1に設定され、報告グループは格納されない。

20

【0074】

Uビット922は、バッファ・フル割り込み（buffer-full interruption）及び停止割り込みに対するイネーブルメント制御である。Uビット922がゼロであるとき、割り込み要求の生成がディスエーブルにされ、保留中の場合は、保留中のままである。

【0075】

Lビット924は、バッファ・フル割り込みの保留制御である。Lビット924がゼロである場合、バッファ・フル割り込みは保留中でない。Lビット924が1である場合、バッファ・フル割り込みは保留中である。

30

【0076】

キー・フィールド926は、その値が報告グループのストアのためのストレージ保護キーとして使用される、4ビットの符号なし整数である。ストレージ・キーがストレージ・アクセス要求と関連したアクセス・キーと一致する場合にのみ、報告グループのストアが許可され、フェッチは、ストレージ・キーがアクセス・キーと一致する場合又はストレージ・キーのフェッチ保護ビットがゼロである場合のみ、フェッチが許可される。ストレージ・キーの4つのアクセス制御ビットがアクセス・キーと等しい場合、又はアクセス・キーがゼロである場合、キーが一致する。

【0077】

図14は、MRICが準特権モードでの実行が許可される（即ち、Kビットが1である）場合のRICCB制御ブロックの一部を示す。制御ブロック1000は、図14を参照して説明されるもの以外の付加的な値を含むことができる。1つの実施形態において、低特権状態プログラムによっては、MRIC命令オペランドのいずれのグレイアウトされたセクションにもアクセスすることができない。準特権モードが許可され場合、起点アドレス（ROA）1002及び制限アドレス1004は、低特権状態プログラムにより、MRIC命令を用いて設定される。

40

【0078】

1つの実施形態において、現在アドレス・フィールド（RCA）1006は、MRIC命令により更新することができる。現在アドレス・フィールド1006は、報告グループ

50

・サイズ (Reporting Group Size) フィールド 1 0 4 4 (R G S フィールド) を検査し、プログラム・バッファのアドレスを形成するのに使用される有効ビット位置の数に影響を与える。6 4 ビットのランタイム・インストルメンテーション・プログラム・バッファの現在アドレスは、ワード 0、ワード 1 のビット位置 0 乃至 2 6 - R G S、及び R G S + 右に付加された 5 つの 2 進数ゼロである。これが、プログラム・バッファ 5 2 2 内に格納されることになる後続の報告グループの、図 9 のプログラム・バッファ 5 2 2 における開始位置である。報告グループは、インストルメンテーション・モジュール 5 0 6 により作成され、後にプログラム・バッファ 5 2 2 内に格納される情報の単位である。1 つの実施形態において、現在アドレス・フィールド 1 0 0 6 により指定される R G S フィールド 1 0 4 4 が、ランタイム・インストルメンテーション制御の現在の報告グループ・サイズと等しくない（即ち、現在アドレス・フィールド 1 0 0 6 が R G S フィールド 1 0 4 4 を変更する）場合、現在アドレス・フィールド 1 0 0 6 は、起点アドレス 1 0 0 2 に設定される。

10

【0 0 7 9】

残りのサンプル間隔カウント (Remaining Sample Interval Count) フィールド 1 0 4 2 (R S I C フィールド) は、M R I C 命令を用いて低特権プログラムにより更新することができる。R S I C フィールド 1 0 4 2 は、残りのサンプル間隔カウントを示す 6 4 ビットの符号なし整数を含む。ランタイム・インストルメンテーション制御内の R S I C フィールド 1 0 4 2 の値がゼロであるか又はスケーリング・ファクタ・フィールド 1 0 4 0 (S F フィールド) 内の値に等しく、且つ、ランタイム・インストルメンテーションがイネーブルにされている場合、次のサンプル間隔は、サンプリング・モード・フィールド 1 0 0 8 (M フィールド) 及び S F フィールド 1 0 4 0 の値に基づく完全な間隔 (full interval) である。R S I C フィールド 1 0 4 2 が非ゼロであり、S F フィールド 1 0 4 0 を下回り、且つ、ランタイム・インストルメンテーションがイネーブルにされている場合、次のサンプル間隔は、部分的な間隔である。R S I C フィールド 1 0 4 2 が非ゼロであり、S F フィールド 1 0 4 0 の値を上回り、且つ、ランタイム・インストルメンテーションがイネーブルにされている場合、次のサンプル間隔は延長した間隔である。延長した間隔が期間満了すると、次の間隔は、S F フィールド 1 0 4 0 の値に基づく。R S I C フィールド 1 0 4 2 が非ゼロの値に設定されると、この R S I C フィールド 1 0 4 2 は、S F フィールド 1 0 4 0 も従う同じモデル依存の最大限度に従う。R S I C フィールド 1 0 4 2 のオリジナル値がゼロである場合、サンプリング・モードは、L R I C 命令及び M R I C 命令の実行中、R S I C フィールド 1 0 4 2 が S F フィールド 1 0 4 0 内の値に設定されるかどうか、又は、ランタイム・インストルメンテーションがイネーブルにされるまで、R S I C フィールド 1 0 4 2 が引き続きゼロとして表示されるかどうかを指示する。

20

30

【0 0 8 0】

S F フィールド 1 0 4 0 は、その値がスケーリング・ファクタの単位のカウントである 6 4 ビットの符号なし整数を含む。単位の大きさは、サンプリング・モード・フィールド 1 0 0 8 (M フィールド) から決定される。R S I C フィールド 1 0 4 2 内の値がゼロである場合、S F フィールド 1 0 4 0 は、現在の命令がサンプル命令として認識される、ゼロまでデクリメントされる R S I C フィールド 1 0 4 2 の初期値を提供し、間隔カウントが S F フィールド 1 0 4 0 の値からリフレッシュされる。S F フィールド 1 0 4 0 の有効値は、1 から $2^{64} - 1$ までの範囲内にある。ゼロが指定される場合、1 の値が推定される。しかしながら、各モデルは S F フィールド 1 0 4 0 の最小値及び最大値の両方を有し得る。最小値及び最大値はまた、サンプリング・モード・フィールド 1 0 0 8 に基づいて異なり得る。最小値を下回る値が指定される場合、モデル依存の最小値がロードされる。最大値を上回る値が指定される場合、モデル依存の最大値がロードされる。

40

【0 0 8 1】

D C 制御フィールド 1 0 3 6 は、その値がデータフェッチ又はストアのキャッシュミスと関連したキャッシュ待ち時間 (cash-latency) レベルを指し示す、4 ビットの符号なし整数である。つまり、サンプル命令がデータ・アクセス・キャッシュミスに遭遇した。別

50

のランタイム・インストルメンテーション制御により禁止されない限り、そのデータ・アクセスによりDC制御フィールド1036の値により指し示されるレベルを数値的に上回るか又はこれに等しいキャッシュ待ち時間レベルでミスが認識されたサンプル命令を表す報告グループのストアが試行される。データ・アクセスに関するキャッシュ構造及びキャッシュ待ち時間レベルは、モデル依存である。複数の又は長いオペランドを用いる命令については、もしあれば、どのオペランド・アクセスが報告制御に用いられるかは、モデル依存である。モデル依存の挙動は、DC制御フィールド1036の値を無視することがあり、従って、この値は、報告グループのストアのために使用しないことがある。

【0082】

ICフィールド1034は、その値が命令フェッチのキャッシュミスと関連したキャッシュ待ち時間レベルを指し示す、4ビットの符号なし整数である。つまり、サンプル命令のフェッチが命令フェッチ・キャッシュミスに遭遇した。ICフィールド1034及びDC制御フィールド1036の両方について、キャッシュ待ち時間レベルは、観察しているプロセッサから特定のキャッシュ・レベル・アクセスまでの距離を抽象化したものである。待ち時間レベルは、プロセッサと主ストレージとの間のネスト化されたキャッシュ・レベルの量と、こうしたキャッシュ・レベルが複数のプロセッサ間でどのように共有されるかの組み合わせに依存する。より大きい待ち時間レベルは、一般に、より時間のかかるアクセスに対応する。ICフィールド1034及びDC制御フィールド1036内の値は、キャッシュ待ち時間レベルのゼロを起点とした識別子と考えることができる。例えば、ゼロの値は、L1キャッシュ（即ち、プロセッサに最も近いキャッシュ）に対応する。従って、1の値は、L2キャッシュ、又は、幾つかのマシンにおいてはL1.5キャッシュとして知られ得る次のキャッシュ層である。2-15の値は、主メモリに到達するまでの付加的なキャッシュ待ち時間層の論理的進行を示すが、主メモリ自体は含まない。一般に、キャッシュ構造は、15の層の深さには至らない。従って、ICフィールド1034及びDC制御フィールド1036内の15という値は、特別な場合と解釈され、命令フェッチ又はデータ・アクセスにおけるキャッシュミスは、それぞれ、キャッシュ待ち時間レベルに関係なく、報告グループのストアを発生させる目的では認識されないことを意味する。別のランタイム・インストルメンテーション制御により禁止されない限り、そのフェッチにより、ICフィールド1034の値により指し示されるレベルを数値的に上回る又はこれに等しいキャッシュ待ち時間レベルでミスが認識されたサンプル命令を表す報告グループのストアが試行される。命令フェッチに関するキャッシュ構造及びキャッシュ待ち時間レベルは、モデル依存である。モデル依存の挙動は、ICフィールド1034の値を無視することがあり、従って、この値は、報告グループのストアのために使用しないことがある。

【0083】

キャッシュ待ち時間レベル・オーバーライド報告制御ビット1032（Fビット）は、非分岐命令及び分岐予測制御のためのものである。ランタイム・インストルメンテーション制御内のFビット1032がゼロである場合、ランタイム・インストルメンテーション制御のキャッシュ報告制御（ICフィールド1034及びDC制御フィールド1036）がチェックされ、その定められた機能に従って使用される。ランタイム・インストルメンテーション制御の分岐予測制御（BPxnビット1022、BPxtビット1024、BPtiビット1026、及びBPniビット1028）がチェックされ、その定められた機能に従って使用される。Fビット1032が1である場合、別の制御によって禁止されない限り、これらの同じ制御が無視され、報告グループが格納される。

【0084】

データ・キャッシュミス制御ビット1030（Dビット）は、報告グループを格納するかどうかを示す。Dビット1030が1である場合、追加タイプ記録が、サンプル命令についてのモデル依存データを含む報告グループの追加セクションに入れられ、又は入れられない場合がある。

【0085】

M R I C 命令は、分岐予測 (B P) 報告制御 (B P x n 1 0 2 2、B P x t 1 0 2 4、B P t i 1 0 2 6、及び B P n i 1 0 2 8) を含む。ランタイム・インストルメンテーション制御内の B P 報告制御ビットがゼロである場合、対応する条件はチェックされない。B P 報告制御ビットが 1 であり、対応する分岐予測条件が存在する場合、報告グループが格納される。

【0086】

B P x n ビット 1 0 2 2 は、1 であるとき、分岐予測情報のチェックをイネーブルにする。従って、サンプル分岐が成立すると誤って予測されたが成立しなかった場合、報告グループが格納される。

【0087】

B P x t ビット 1 0 2 4 は、1 であるとき、分岐予測情報のチェックをイネーブルにする。従って、サンプル分岐が成立しないと誤って予測されたが成立した場合、報告グループが格納される。

【0088】

B P t i ビット 1 0 2 6 は、1 であるとき、分岐予測情報のチェックをイネーブルにする。従って、サンプル分岐が成立すると正しく予測され、分岐が成立したが、分岐ターゲットが誤って予測された場合、報告グループが格納される。

【0089】

B P n i ビット 1 0 2 8 は、1 である場合、分岐予測情報のチェックをイネーブルにする。従って、サンプル分岐が成立しないと正しく予測され、分岐が成立せず、且つ、分岐ターゲットが誤って予測された場合、報告グループが格納される。

【0090】

トランザクション実行モード記録のイネーブルメント制御ビット 1 0 2 0 (X ビット) は、トランザクション実行モード・アボート記録の収集を制御する。ランタイム・インストルメンテーション制御内の X ビット 1 0 2 0 がゼロであるとき、トランザクション実行モード・アボート記録は収集されない。X ビット 1 0 2 0 が 1 であるとき、トランザクション実行モード・アボート記録が収集され、図 9 の収集バッファ 5 0 8 に入れられる。モデルにトランザクション実行ファシリティがインストールされていない場合、X ビット 1 0 2 0 は無視される。

【0091】

R I E M I T 命令制御ビット 1 0 1 8 (E ビット) は、R I E M I T 命令の実行を制御する。ランタイム・インストルメンテーションがイネーブルにされるとき、ランタイム・インストルメンテーション制御内の E ビット 1 0 1 8 がゼロであるか、又は無視されてゼロとして扱われる場合、R I E M I T は、操作を実行しない。E ビット 1 0 1 8 が 1 であり、他の形で無視されない場合、R I E M I T はイネーブルにされ、その定められた機能を実行する。

【0092】

J ビット 1 0 4 6 は、ゼロである場合、条件付き分岐 (B C, Branch on Condition) 命令が、マスク値に関係なく他のタイプの分岐カテゴリ内にあることを指定する。J ビット 1 0 4 6 が 1 である場合、1 5 のマスクを指定する B C 命令は、リターン・タイプ分岐カテゴリ内にある。B C 命令が 1 - 1 4 のマスクを指定する場合、B C 命令は J ビット 1 0 4 6 に影響を受けず、常にその他のタイプの分岐カテゴリ内にある。リターン・タイプ分岐カテゴリ内にある場合、R ビット 1 0 1 6 が、図 9 の収集バッファ 5 0 8 への包含を制御する。他のタイプの分岐カテゴリ内にある場合、B ビット 1 0 4 8 が収集バッファ 5 0 8 への包含を制御する。他のタイプの分岐カテゴリは、転送タイプ分岐カテゴリとして示すこともできる。

【0093】

命令アドレス・コード・ビット 1 0 1 4 (C ビット) は、呼び出しタイプ分岐のイネーブルメントを制御する。ランタイム・インストルメンテーション制御内の C ビット 1 0 1 4 が 1 であり、命令が呼び出しタイプ分岐である場合、収集バッファが更新される。呼び

10

20

30

40

50

出しタイプ分岐とリターン・タイプ分岐の両方のモデル依存検出が組み合わされる場合、Cビット1014は両方のタイプに対して動作し、Rビット1016は有効でない。

【0094】

Rビット1016は、リターン・タイプ分岐のイネーブルメント制御である。ランタイム・インストルメンテーション制御内のRビット1016が1であり、命令がリターン・タイプ分岐である場合、収集バッファ508が更新される。

【0095】

Bビット1048は、呼び出しタイプ分岐及びリターン・タイプ分岐以外の分岐のイネーブルメント制御である。ランタイム・インストルメンテーション制御内のBビット1048が1であり、命令が、ランタイム・インストルメンテーションにより認識される他のタイプの分岐である場合、収集バッファ508が更新される。

10

【0096】

最大アドレス超過ビット1012 (MAEビット) は、1に設定された場合、1に設定された命令アドレス・コード (Cフィールド) を有する1つ又は複数の報告グループが格納されたことを示す。ひとたびMAEビット1012が1に設定されると、続くランタイム・インストルメンテーションを実行することによっては、MAEビット1012が再びゼロに設定を戻されることはない。MAEビットをゼロとして指定するLRIC命令又はMRIC命令の実行により、MAEビットがゼロに設定される。

【0097】

ランタイム・インストルメンテーション・ネクスト (RINEXT) 制御ビット1010 (Nビット) は、サンプル命令の実行を制御する、ランタイム・インストルメンテーション・ネクスト命令のイネーブルメントを制御する。ランタイム・インストルメンテーション制御内のNビット1010がゼロであるか、又は無視されてゼロとして扱われる場合、RINEXTは、操作を実行しない。Nビット1010が1であり、他の形で無視されない場合、RINEXTがイネーブルにされ、その定められた機能を実行する。

20

【0098】

サンプリング・モード・フィールド1008 (Mフィールド) は、ランタイム・インストルメンテーション制御内のその値がランタイム・インストルメンテーション制御に関するサンプリング・モードを指定する、4ビットの符号なし整数である。

【0099】

報告グループ・サイズ・フィールド1044 (RGS) は、その値が報告グループの記録 (RRG, Records of a Reporting Group) の数を指定する、3ビットの符号なし整数である。報告グループ内の記録の数は、開始／タイムスタンプ記録及び命令の最後の記録を含む2つの記録から、最大256の記録まで変化し得る。1つの実施形態において、上限は、モデル依存であり得る。報告グループ内に入れられる16バイト記録の数は、 $2^{(RGS + 1)}$ である。

30

【0100】

一次CPU能力抑止制御ビット1038 (Yビット) 及び二次CPU能力抑止制御ビット1039 (Zビット) は、まとめて抑止制御 (suppression control) と呼ばれる。報告グループのストアの抑止とは、ストアする試みが実行されないことを意味する。構成内の全てのCPUのCPU能力が同じであるとき、抑止制御は有効でなく、抑止は行われな

い。構成において、あるCPUのCPU能力が別のCPUの能力と異なる場合、抑止制御は有効であり、少なくとも1つのCPUが、CPU能力又は一次CPU能力で動作していると言われ、一方、少なくとも1つの他のCPUが二次CPU能力で動作していると言われる。一次CPU能力と二次CPU能力は、異なる動作速度である。Yビット1038及びZビット1039の両方がゼロであるとき、抑止は行われな

い。Yビット1038がゼロであり、Zビット1039が1であるとき、例えばプロセッサ106などのCPUが二次CPU能力で動作している場合、抑止が行われる。Yビット1038が1であり、Zビット1039がゼロであるとき、例えばプロセッサ106などのCPUが一次CPU能力で動作している場合、抑止が行われる。Yビット1038及びZビット1039の両方と

40

50

も1である場合、抑止が行われる。

【0101】

図14の上記のフィールド及びビットは、フィールドの配置及び呼称の一例であり、明確にするために本明細書で与えられている。他の実施形態において、フィールドのサブセットだけを使用してもよいこと、及び、フィールドはいずれの順序又は位置としてもよく、及び／又は、異なる名称で表し得ることが理解されるであろう。

【0102】

前述のように、プログラム実行中にランタイム・インストルメンテーションがイネーブルにされると、ランタイム・インストルメンテーション・データが、プロセッサ106の収集バッファ508内に収集される。1つの実施形態において、収集バッファ508は、
10
収集された最新の記録の保存に使用されるプロセッサ106の内部バッファである。サンプル・トリガ・ポイントが検出されると、記録は、プログラム・バッファ522に書き込まれた報告グループの一部として、収集バッファ508からプログラム・バッファ522にコピーされる。1つの実施形態において、記録は、収集バッファ508から、非破壊的方法でコピーされる。

【0103】

収集バッファ508は、これがプロセッサ内に配置され、且つ、1つの実施形態においては、所定のイベントの命令アドレス及びイベント・メタデータを格納するためのレジスタ対のアレイとして実装されるので、「ハードウェア収集バッファ」と呼ぶことができる。
20
イベントの例は、分岐の成立であり、そこでレジスタ対は分岐の命令アドレスを保持することができ、メタデータは分岐のターゲット並びに分岐の挙動履歴 (histroic behavior) に関する情報を保持することができる。1つの実施形態においては、レジスタ対が順序付けられ、命令ストリーム内でイベントが生じると順次更新される。カウンタは、アレイ内の最後に更新されたエントリのインデックスを示すように維持される。1つの実施形態において、収集バッファ508は循環バッファであり、収集バッファ508が一杯になると、次のイベントがアレイ内の最初のエントリを上書きし、後続のイベントにおいて、アレイのレジスタ対の順次更新が再開する。従って、アレイをCB[0]乃至CB[N-1]で、及びカウンタiが最新の更新インデックスを示すとする、キャプチャされたイベントの記録は、系列CB[i]、CB[i-1]...CB[1]、CB[0]、CB[N-1]、CB[N-2]...CB[i+1]で表される。別の実施形態においては、
30
バッファ内の最も古いエントリを指し示すヘッド・ポインタ、及び、バッファ内の最も新しいエントリを指し示すテール／現在ポインタの2つのポインタが使用される。

【0104】

任意の所定の実行点でのプロセッサ106の状態を表すイベントが、収集バッファ508内に順次キャプチャされる。収集バッファ508は、その内容が、プログラム実行中にプロセッサ106により認識されるイベント（例えば、1つ又は複数の成立した分岐の実行、トランザクション実行アボート・イベント、RIEMIT命令のオペランド等）に関して報告する記録のセットをキャプチャするために使用される。1つの実施形態において、認識されるイベントは、図14に示されるRICCBの内容に依存する。示される収集バッファ508の実施形態におけるエントリは、イベント命令アドレスと、他の関連する
40
イベント・メタデータとを含む。イベント・メタデータの例として、これらに限定されるものではないが、成立した分岐の命令アドレス及びその分岐の挙動履歴についての何らかの情報を含むそのターゲット；RIEMIT命令の命令アドレス及びそれぞれのレジスタ値；並びに、トランザクション・アボート命令のアドレス及びそれぞれのトランザクション復元エントリ・ポイントが挙げられる。

【0105】

収集バッファ508の1つの実施形態は、最大32個までのエントリ（即ち、32のイベントについての情報）を格納し、各命令アドレスは64のビット（例えば、ビット0：63）により指定され、イベント・メタデータは64のビット（例えば、ビット64：127）により指定される。収集バッファのサイズ（RCB）は、記録の数を表すモデル依
50

存のカウンタである。1つの実施形態において、収集バッファ508のバイト・サイズは、16バイトの記録サイズの倍数である。収集バッファのサイズ(RCB)は、モデルの最大報告グループのカウンタ(RRG)と、収集バッファから取得されない報告グループの記録のカウンタ(RNC)との間の差を上回るか又はこれに等しい記録の数である。従って、1つの実施形態において、収集バッファのサイズは、

【数1】

$$RCB \geq (RRG - RNC)$$

10

のように表される。

【0106】

1つの実施形態において、収集バッファ508及び命令データ・バッファの内容は(ひとつが使用される場合)、(1)割り込み、(2)ランタイム・インストルメンテーション・ファシリティをオン又はオフにするPSWビット(例えば、ビット24)の1からゼロへの変更、及び(3)ランタイム・インストルメンテーション・ファシリティがトランザクション実行モードであるときにサンプル命令が識別された場合(この場合、収集バッファ508及び命令データ・バッファの更なる更新は停止し、トランザクション終了時に再開し、この時点では、報告グループのストアは保留中であり、収集バッファ508及び命令データ・バッファはパージされる)というイベントによって、パージされる又は他の形で影響を受ける。

20

【0107】

図2に示されるエミュレートされたホスト・コンピュータ・システムのような1つの実施形態において、収集バッファ508は、レジスタ及び／又はメモリを用いて実装される。この実施形態において、随意的な命令データ・バッファもまた、存在する場合には、レジスタ及び／又はメモリを用いて実装される。

【0108】

実施形態において、付加的な能力は、データ収集に影響を与えることがあり、且つ、前述した通常の命令カウンタ又はサイクル・カウンタのサンプリングを実質的に妨げずに、付加的なデータ収集ポイントをもたらすと考えることができる。これらには、収集バッファ508内に汎用レジスタの値を格納することにより、その値を収集するRIEMIT命令の実行が含まれる。さらに、前述したランタイム・インストルメンテーション制御内のデータ収集制御ビットを用いて、収集したデータのタイプをカスタマイズすることができる(例えば、E、C、R及びB制御ビット)。このように、収集するデータ・タイプは、プログラム可能である。

30

【0109】

1つの実施形態において、ランタイム・インストルメンテーション命令の記録を構築するために使用されるモデル依存のサンプル命令データを収集するために、命令データ・バッファが実装される。命令データ・バッファは、命令がサンプル命令として識別されたときに利用可能になると予期して、命令からデータを収集する。1つの実施形態において、命令データ・バッファは、サンプル・ポイントとしてトリガになる命令についての情報が保存される、プロセッサ内のハードウェア・バッファ／ストレージ位置であるので、ログアウト・プロセス中、情報を収集バッファ508からデータと共に書き出すことができる。収集バッファ508と同様に、命令データ・バッファは、命令アドレス、及びその命令に関連したメタデータを含む。命令データ・バッファ内のメタデータは、多くの場合、マシンに依存し、これらに限定されるものではないが、キャッシュミス関連情報、及び分岐予測関連情報を含むことができる。

40

【0110】

実施形態によれば、収集される他のデータは、収集バッファ508からのものでないことも、又は命令データ・バッファからのものでないこともある。例として、(1)報告グ

50

ループの最初の記録：タイムスタンプ又は開始記録、及び（２）あらゆる報告グループに関して作成することができ、従って、収集バッファ５０８内に格納されない付加的なタイプの記録の一部を形成するために使用されるデータが挙げられ、これらの記録は、存在する場合、報告グループの追加又はマシン依存セクションに入れることができる。これらの記録は、本明細書では「システム情報記録」と呼ばれる。

【０１１１】

図１５は、MRICが準特権モードでの実行を許可しない（即ち、Kビットがゼロである）場合のRICCB制御ブロックの一部を示す。制御ブロック１１００は、図１５を参照して説明されるもの以外の付加的な値を含むことができる。準特権モードが許可されない場合、起点アドレス・セクション１１０２及び制限アドレス・セクション１１０４は使用されず、低特権状態プログラムによって現在の値は変更されない。

10

【０１１２】

１つの実施形態において、ランタイム・インストルメンテーション制御設定の全てのうちのサブセットのみが、低特権状態プログラムによるMRIC命令の正常な実行によって更新される。

【０１１３】

RICCBがフェッチされると、多数のエラーに遭遇し得る。さらに、MRIC命令内のRICCBのアドレスが適切に整列（align）されていない場合、例外が引き起こされる。同様に、エラー状態又は無効アドレスのために、MRIC命令内のアドレスにアクセスできない場合、例外が引き起こされる。MRIC命令の実行中、RICCB値がいずれかの明確な形で競合する（即ち、内部的に矛盾する）と判断された場合には、同じく例外が引き起こされる。

20

【０１１４】

１つの実施形態においては、MRIC命令が発行され、ランタイム・インストルメンテーションがイネーブルにされる；現在のランタイム制御が有効でない；及び／又は、プロセッサが低特権状態にあり、現在のランタイム・インストルメンテーション制御内のランタイム・インストルメンテーション停止ビットがゼロである、のいずれかの理由により特別操作例外が認識される。

【０１１５】

１つの実施形態において、MRICのストレージ・オペランドがダブルワード（doubleword）境界に整列されていない；

30

プロセッサがスーパーバイザ状態にあるか、又は現在のランタイム・インストルメンテーション制御内のKビットが１であり、且つ、以下の条件、即ち、指定された制限アドレスが指定された起点アドレスより小さい、指定された現在アドレスが指定された起点アドレスより小さい、及び指定された現在アドレスが指定の生成制限アドレスに１を足した和より大きい、のいずれかが認識される；並びに、

プロセッサが低特権状態にあり、現在のR I制御の内のkビットがゼロであり、且つ、以下の条件、即ち、指定された現在アドレスが現在の起点アドレスより小さい、指定された現在アドレスが現在の制限アドレスに１を足した和より大きい、無効なモードが指定される、のいずれかが認識される；

40

のいずれかの理由により指定例外が認識される。

【０１１６】

１つの実施形態において、１つのアーキテクチャの特定のプロセッサに関して定められたMRIC命令は、異なるアーキテクチャの第２のプロセッサによって実行され得る。１つの実施形態において、第２のプロセッサは、ソフトウェア・ベースのエミュレーション・ルーチンを識別し、ソフトウェア・ベースのエミュレーション・ルーチンを用いて、MRIC命令を実行する。

【０１１７】

図１６は、サンプル・ポイントにおいてプログラム・バッファ５２２に格納された報告

50

グループ1200の高レベルの例を示す。記録内の報告グループのサイズは、 $2^{(R_{GS} + 1)}$ に等しい R_{RG} で表され、ここで、 R_{GS} は、指数としての報告グループ・サイズである。収集バッファ508以外の位置からコピーされた記録のモデル依存数(R_{NC})は、報告グループで使用されるとき、非破壊的にコピーされても又はされなくてもよい。図16の例において、 $R_{RG} = 8$ 、 $R_{GS} = 2$ 、及び $R_{NC} = 4$ である。図16に示される例示的な報告グループ1200は、ヘッダ・セクション1202、本体セクション1204、追加記録セクション1206、及びフッタ・セクション1208を含む。

【0118】

ヘッダ・セクション1202は、状態情報、追跡情報、及び／又はタイミング情報を保持するための開始記録又はタイムスタンプ記録を含むことができる。開始記録は、プログラム・バッファに格納された第1の報告グループに関するヘッダ・セクション1202内に格納される(即ち、 $RCA1006$ が $ROA1002$ に等しいとき)。1つの実施形態において、開始記録は、「02」の記録タイプ・フィールドと、現在幾つの報告グループがプログラム・バッファに格納されているかを示すための報告グループ数(NRG)フィールドと、報告グループのサイズを示すための RGS フィールドと、プログラム・バッファ522が一杯であるかどうかを示すための停止(S :stopped)フィールドと、ランタイム・インストルメンテーションが停止されたかどうかを示すための停止(H :halted)フィールドと、いつ開始記録が書き込まれたかを示すための時刻(TOD)クロック・フィールドとを含む。1つの実施形態において、開始記録内のフィールドの少なくとも1つのサブセットが、 RI 制御ブロック(例えば、 $RICCB$)から調達される。タイムスタンプ記録の1つの実施形態は、「03」の記録タイプを有し、その記録がいつ格納されたかを示すための TOD クロック・フィールドを含む。1つの実施形態において、タイムスタンプ記録は、第1の報告グループ以外の各報告グループについてヘッダ・セクション1202に格納される。

【0119】

報告グループの本体セクション1204は、収集バッファ508からサンプリングされたイベント及び情報に関する種々の記録を含むことができる。イベント及び情報は、例えば、発行命令、トランザクション実行アボート、呼び出し、リターン、分岐、及びフィルターによってキャプチャされた状態情報を表すことができる。

【0120】

1つの実施形態において、発行記録は、 $RIEMIT$ 命令の正常な実行時に作成され、収集バッファ508に格納される。発行記録の1つの実施形態は、「10」の記録タイプ・フィールドと、発行記録において、現 PSW の命令アドレス・ビット位置がどのように表されるかを示すための命令アドレス・コード・フィールドと、アドレッシング・モード(例えば、64ビット、31ビット、又は24ビット)に応じて変わり、且つ、 $RIEMIT$ 命令のアドレス、若しくは $RIEMIT$ が実行タイプ命令のターゲットであった場合には実行タイプ命令の命令アドレスを含む命令アドレス・フィールドと、 $RIEMIT$ 命令により指定される汎用レジスタからのデータを格納するための発行データ・フィールドとを含む。

【0121】

1つの実施形態において、トランザクション実行アボート記録が、暗黙的アボートにより、又はトランザクション・アボート命令の実行により作成され、収集バッファ508に格納される。アボート記録の1つの実施形態は、「11」の記録タイプ・フィールドと、トランザクション実行アボート記録において現 PSW の命令アドレス・ビット位置がどのように表されるかを示すための命令アドレス・コード・フィールドと、アドレッシング・モード(例えば、64ビット、31ビット、又は24ビット)に応じて変わり、且つ、アボートされる命令のアドレス、若しくはアボートされる命令が実行タイプ命令のターゲットであった場合には実行タイプ命令の命令アドレスを含む命令アドレス・フィールドと、アボートと関連したいずれかのモデル依存データに関するフィールドとを含む。

【0122】

1つの実施形態において、呼び出し記録は、R2フィールドが非ゼロであるときのBRANCH AND SAVE (BASR)、BRANCH AND SAVE (BAS)、BRANCH RELATIVE AND SAVE LONG、BRANCH RELATIVE AND SAVE、R2フィールドが非ゼロであるときのBRANCH AND LINK (BALR)、BRANCH AND LINK (BAL)、及びR2フィールドが非ゼロであるときのBRANCH AND SAVE AND SET MODEなどの、呼び出しタイプ分岐命令の実行によって作成される。呼び出し記録の1つの実施形態は、「12」の記録タイプ・フィールドと、呼び出し記録において現PSWの命令アドレス・ビット位置がどのように表されるかを示すための命令アドレス・コード・フィールドと、アドレッシング・モード（例えば、64ビット、31ビット、又は24ビット）に応じて変わり、且つ、分岐命令のアドレス、若しくは分岐命令が実行タイプ命令のターゲットであった場合には実行タイプ命令のアドレスを含む命令アドレス・フィールドと、分岐が正しく予測されたかどうかを示すための正常動作 (well-behaved) フィールドと、分岐ターゲット・アドレス（「呼び出された位置」とも呼ばれる）を含むターゲット・アドレス・フィールドとを含む。

10

【0123】

リターン記録及び転送記録は、呼び出し記録と同じフォーマットを有することができる。1つの実施形態において、リターン記録は、「13」の記録タイプ・フィールドを有し、R2フィールドが非ゼロであり、マスクが15であるときのBRANCH ON CONDITION (BCR) などのリターン・タイプ分岐命令の実行によって作成される。リターン記録に関して、命令アドレス・フィールドは、分岐命令のアドレス、又は分岐が実行タイプ命令のターゲットである場合には実行タイプ命令のアドレスを含み、ターゲット・アドレス・フィールドはリターン位置を含む。

20

【0124】

1つの実施形態において、転送記録は、「14」の記録タイプ・フィールドを有し、且つ、a. R2フィールドが非ゼロであり、マスクが1-14の範囲内にあるときのBRANCH ON CONDITION (BCR) ; b. Jビットがゼロであるか又はマスクが1-14の範囲内にあるときのBRANCH ON CONDITION (BC) ; c. BRANCH ON COUNT (BCT, BCTR, BCTG, BCTGR) ; d. BRANCH ON INDEX HIGH (BXH, BXHG) ; e. BRANCH ON INDEX LOW OR EQUAL (BXLE, BXLEG) ; f. BRANCH RELATIVE ON CONDITION (BRC) ; g. BRANCH RELATIVE ON CONDITION LONG (BRCL) ; h. BRANCH RELATIVE ON COUNT (BRCT, BRCTG) ; i. BRANCH RELATIVE ON COUNT HIGH (BRCTH) ; j. BRANCH RELATIVE ON INDEX HIGH (BRXH, BRXHG) ; k. BRANCH RELATIVE ON INDEX LOW OR EQUAL (BRXLE, BRXLG) ; l. COMPARE AND BRANCH (CRB, CGRB) ; m. COMPARE AND BRANCH RELATIVE (CRJ, CGRJ) ; n. COMPARE IMMEDIATE AND BRANCH (CIB, CGIB) ; o. COMPARE IMMEDIATE AND BRANCH RELATIVE (CIJ, CGIJ) ; p. COMPARE LOGICAL AND BRANCH (CLRB, CLGRB) ; q. COMPARE LOGICAL AND BRANCH RELATIVE (CLRJ, CLGRJ) ; r. COMPARE LOGICAL IMMEDIATE AND BRANCH (CLIB, CLGIB) ; 及びs. COMPARE LOGICAL IMMEDIATE AND BRANCH RELATIVE (CLIJ, CLGIJ) のような、リターン・タイプ分岐命令の実行により作成される。転送記録は、分岐が成立したときに作成される。転送記録に関して、命令アドレス・フィールドは、分岐命令のアドレス、又は分岐が実行タイプ命令のターゲットである場合には実行タイプ命令のアドレスを含み、ターゲット・アドレス・フィールドはリターン位置

30

40

50

を含む。

【0125】

フィルター記録は、収集バッファ508内の有効な記録の数が、現在のRGSの報告グループを満たすのに十分でない場合に、報告グループにおいて使用される。フィルター記録の1つの実施形態は、記録がフィルター記録であることを示すための「00」の記録タイプ・フィールドを含み、残りのバイトは定義されない。

【0126】

追加記録セクション1206は、存在する場合、モデル依存の記録を含むことができる。1つの実施形態において、追加記録のフォーマットは、記録が追加記録であることを示すように、記録タイプが「01」に設定され、追加記録の残りのバイトはモデル依存データを含むことができる点を除いて、フィルター記録に類似している。

10

【0127】

フッタ・セクション1208は、サンプル命令の実行についての情報を含む命令記録を含むことができる。命令記録は、サンプル命令に関して報告グループが格納されたときに作成される。命令記録の1つの実施形態は、「04」の記録タイプ・フィールドと、命令記録において、現PSWの命令アドレス・ビット位置がどのように表されるかを示すための命令アドレス・コード・フィールドと、アドレッシング・モード（例えば、64ビット、31ビット、又は24ビット）に応じて変わり、且つ、サンプル命令の命令アドレス、又はサンプル命令が実行タイプ命令のターゲットであった場合には実行タイプ命令の命令アドレスを含む命令アドレス・フィールドと、命令データ・バッファ（IDB, Instruction-data Buffer）から収集されたいずれかのモデル依存データを含むIDBフィールドとを含む。

20

【0128】

上述のように、実施形態は、コンピュータ実装プロセス及びそれらのプロセスを実施するための装置の形で具体化することができる。実施形態は、製造物品として有形媒体内に具体化された命令を含むコンピュータ・プログラム・コード論理1304を有するコンピュータ可読／使用可能媒体1302上の、図17に示されるようなコンピュータ・プログラム製品1300を含むことができる。コンピュータ可読／使用可能媒体1302に関する例示的な製造物品は、フロッピー（登録商標）ディスク、CD-ROM、ハードドライブ、ユニバーサル・シリアル・バス（USB）フラッシュ・ドライブ、又は他のいずれかのコンピュータ可読ストレージ媒体を含むことができ、ここで、コンピュータ・プログラム・コード論理1304がコンピュータにロードされ、コンピュータにより実行されると、コンピュータは、本発明を実施するための装置となる。実施形態は、例えば、ストレージ媒体内に格納される、コンピュータにロードされる及び／又はコンピュータによって実行される、又は電気配線若しくはケーブルにより光ファイバを通じて又は電磁放射を介してといった何らかの伝送媒体上で伝送される、コンピュータ・プログラム・コード論理1304を含み、ここで、コンピュータ・プログラム・コード論理1304がコンピュータにロードされ、コンピュータにより実行されると、コンピュータは、本発明を実施するための装置となる。汎用マイクロプロセッサ上に実装される場合、コンピュータ・プログラム・コード論理1304のセグメントは、特定の論理回路を作成するようにマイクロプロセッサを構成する。

30

40

【0129】

技術的効果及び利点として、低特権状態から実行時のランタイム・インストルメンテーション設定を変更するための、低特権状態プログラムにより実行可能なMRIC命令が挙げられる。

【0130】

本明細書で用いられる用語は、特定の実施形態を説明する目的のためのものにすぎず、本発明を限定することを意図したものではない。本明細書で用いられる場合、単数形「1つの(a)」、「1つの(an)」及び「その(the)」は、文脈が特に明示しない限り、複数形も同様に含むことを意図したものである。「含む(comprise)」及び／又は「含んで

50

いる (comprising)」という用語は、本明細書で用いられる場合、記述された特徴、整数、ステップ、動作、要素、及び／又はコンポーネントの存在を指定するが、1つ又は複数の他の特徴、整数、ステップ、動作、要素、コンポーネント、及び／又はそれらのグループの存在又は追加を排除するものではないこともさらに理解されるであろう。

【0131】

下記の特許請求の範囲におけるすべての機能付き手段 (ミーンズ・プラス・ファンクション) 又は機能付き工程 (ステップ・プラス・ファンクション) の対応する構造、材料、動作、及び均等物は、もしあれば、その機能を、明確に請求されている他の特許請求された要素と組み合わせて実行するためのあらゆる構造、材料、又は動作を含むことが意図されている。本発明の説明は、例証及び説明のために提示されたものであり、網羅的であること又は本発明を開示された形態に限定することを意図したものではない。当業者には、本発明の範囲および要旨から逸脱しない多くの修正及び変形が明らかとなるであろう。実施形態は、本発明の原理及び実際の適用を最も良く説明し、その他の当業者が企図される特定の使用に適した種々の修正を伴う種々の実施形態について本発明を理解できるように、選択され、説明された。

10

【0132】

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。

20

【0133】

1つ又は複数のコンピュータ可読媒体の任意の組み合わせを利用することができる。コンピュータ可読媒体は、コンピュータ可読信号媒体又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、電子的、磁氣的、光学的、電磁氣的、赤外線、若しくは半導体のシステム、装置、若しくはデバイス、又は上記のものの任意の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 (網羅的なリストではない) には、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・ディスクレット、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、又は上記のものの任意の適切な組み合わせが含まれる。本文書の文脈において、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、又はそれらとの関連で使用するためのプログラムを収容又は格納することができる任意の有形媒体とすることができる。

30

【0134】

コンピュータ可読信号媒体は、コンピュータ可読プログラム・コードが、例えばベースバンド内に又は搬送波の一部として内部に具体化された伝搬データ信号を含むことができる。このような伝搬信号は、これらに限定されるものではないが、電磁氣、光、又はそれらの任意の適切な組み合わせを含む種々の形態のいずれかを取ることができる。コンピュータ可読信号媒体は、コンピュータ可読ストレージ媒体ではなく、且つ、命令実行システム、装置若しくはデバイスによって、又はこれらとの関連で使用するためのプログラムを通信し、伝搬し、又は搬送することができる任意のコンピュータ可読媒体とすることができる。

40

【0135】

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、これらに限定されるも

50

のではないが、無線、有線、光ファイバ・ケーブル、RF等、又は上記のもののいずれかの適切な組み合わせを含む、いずれかの適切な媒体を用いて伝送することができる。

【0136】

本発明の態様の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（登録商標）、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記述することができる。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、一部がユーザのコンピュータ上で、独立したソフトウェア・パッケージとして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。後者のシナリオにおいて、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）を含む任意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・プロバイダを用いるインターネットを通じて）。

10

【0137】

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以下で説明される。フローチャート図及び／又はD-2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することが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り実行される命令が、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生成するようにすることができる。

20

【0138】

これらのコンピュータ・プログラム命令を、コンピュータ、他のプログラム可能データ処理装置、又は他のデバイスに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。

30

【0139】

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、そのコンピュータ、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、コンピュータ実装プロセスを生成するための一連の動作ステップを実施させて、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／動作を実装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。

40

【0140】

上述のように、実施形態は、コンピュータ実装プロセス及びそれらのプロセスを実施するための装置の形で具体化することができる。実施形態において、本発明は、1つ又は複数のネットワーク要素により実行されるコンピュータ・プログラム・コードにおいて具体化される。実施形態は、製造物品として有形媒体内に具体化された、命令を含むコンピュータ・プログラム・コード論理を有するコンピュータ使用可能媒体上のコンピュータ・プログラム製品を含む。コンピュータ使用可能媒体に関する例示的な製造物品は、フロッピー（登録商標）ディスク、CD-ROM、ハードドライブ、ユニバーサル・シリアル・バス（USB）フラッシュ・ドライブ、又は他のいずれかのコンピュータ可読ストレージ媒体を含むことができ、ここで、コンピュータ・プログラム・コード論理がコンピュー

50

タにロードされ、コンピュータにより実行されると、コンピュータは、本発明を実施するための装置となる。実施形態は、例えば、ストレージ媒体内に格納される、コンピュータにロードされる、及び／又はコンピュータによって実行される、又は電気配線若しくはケーブルにより、光ファイバを通じて、又は電磁放射を介してといった何らかの伝送媒体上で伝送される、コンピュータ・プログラム・コード論理を含み、ここで、コンピュータ・プログラム・コード論理がコンピュータにロードされ、コンピュータにより実行されると、コンピュータは、本発明を実施するための装置となる。汎用マイクロプロセッサ上に実装される場合、コンピュータ・プログラム・コード論理セグメントは、特定の論理回路を作成するようにマイクロプロセッサを構成する。

【0141】

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。

【符号の説明】

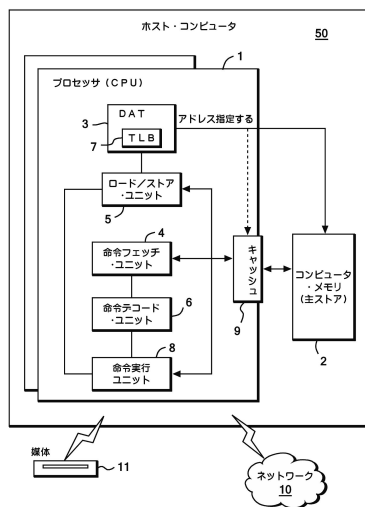
【0142】

- 1、106：プロセッサ
- 2、22、105：メモリ
- 11：ストレージ・デバイス
- 50：ホスト・コンピュータ・システム
- 100、500：システム
- 111：プログラミング・コード
- 112：アプリケーション・プログラム
- 200：データ処理ネットワーク
- 303：キャッシュ
- 304：I/Oユニット
- 305：命令フェッチ・ユニット
- 306：デコード／ディスパッチ・ユニット
- 307：実行ユニット
- 308：分岐ユニット
- 309：アーキテクチャ化レジスタ
- 310：ロード・ストア・ユニット
- 311：プログラム・カウンタ
- 504：ランタイム・メモリ
- 506：インストルメンテーション・モジュール
- 508：収集バッファ
- 510：レジスタ
- 512：PSWデータ
- 514：主メモリ
- 516：オペレーティング・システム
- 518：アプリケーション
- 520：ストレージ

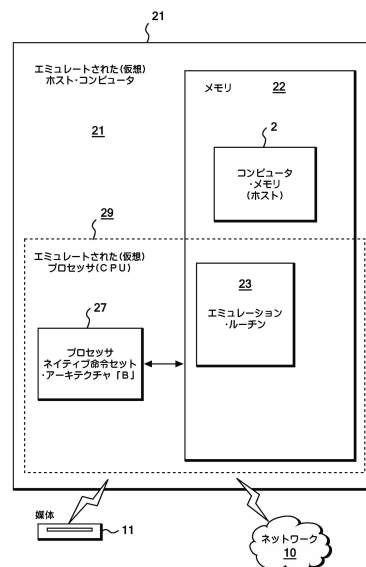
522：プログラム・バッファ
 600：MRIC命令
 602、604：オペコード
 606：基底レジスタ・フィールド
 608、610：変位フィールド
 1000、1100：制御ブロック
 1200：報告グループ
 1202：ヘッダ・セクション
 1204：本体セクション
 1206：追加記録セクション
 1208：フッタ・セクション
 1300：コンピュータ・プログラム製品
 1302：コンピュータ可読／使用可能媒体
 1304：コンピュータ・プログラム・コード論理

10

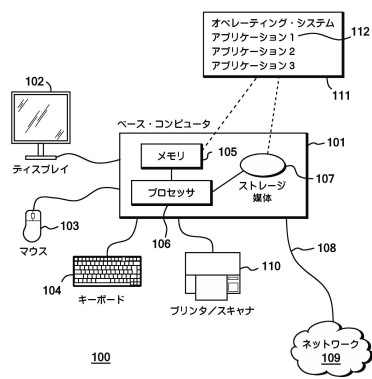
【図 1】



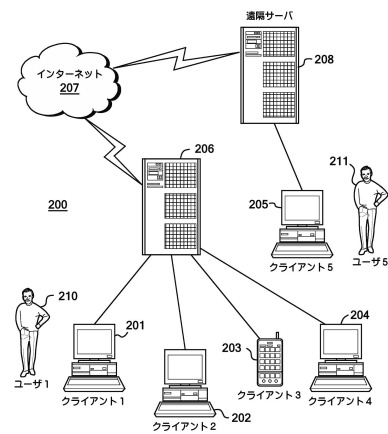
【図 2】



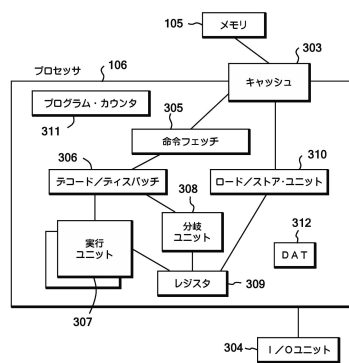
【図 3】



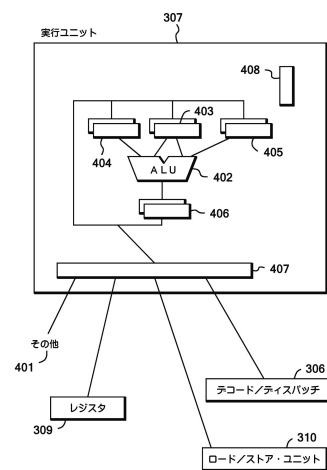
【図 4】



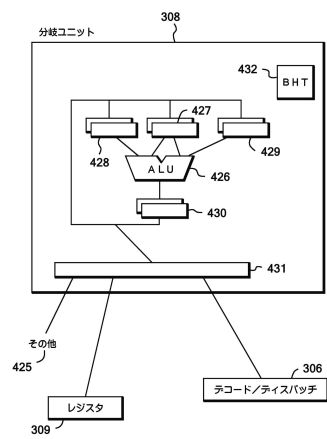
【図 5】



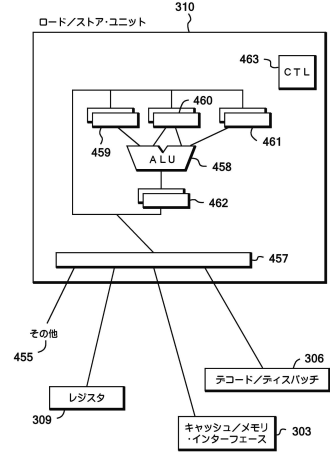
【図 6】



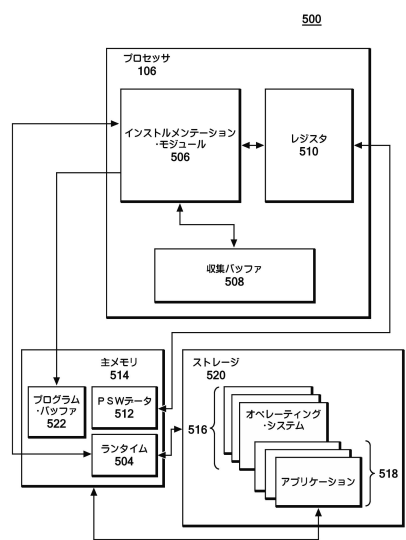
【図 7】



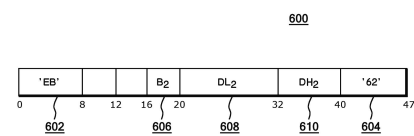
【図 8】



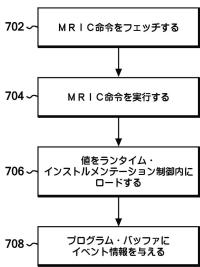
【図 9】



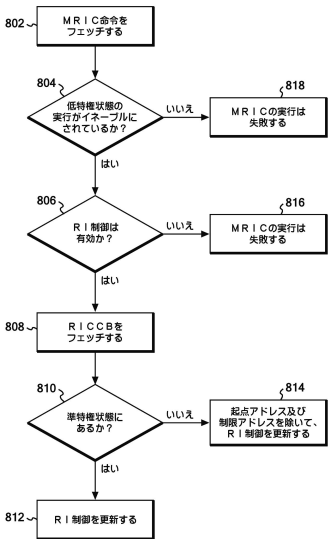
【図 10】



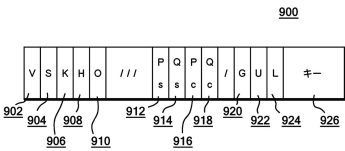
【図 1 1】



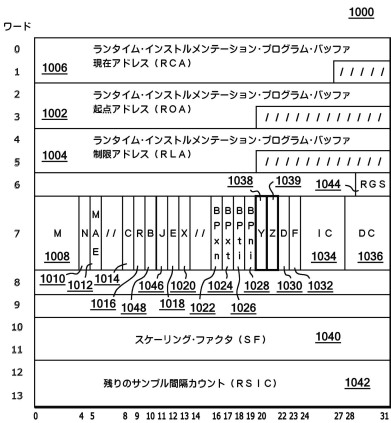
【図 1 2】



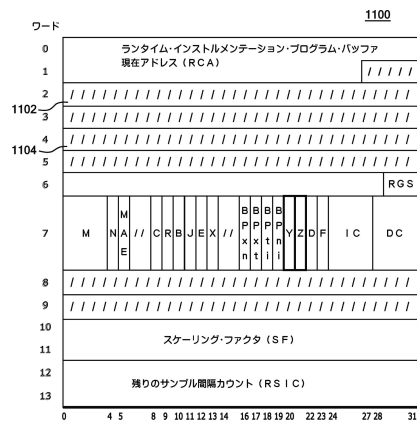
【図 1 3】



【図 1 4】



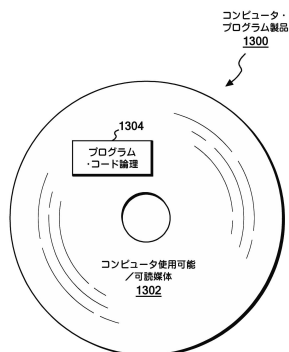
【図 15】



【図 16】

1200		
記録番号	記録タイプ	セクション
0	開始、タイムスタンプ	ヘッダ 1202
1	発行、TXポート、 呼び出し、リターン、分岐、フィラー	本体 1204
2		
3		
$R_{RG} \cdot R_{NC}(-4)$		
$R_{RG} \cdot R_{NC}+1(-5)$	追加、モデル依存	追加 1206
$R_{RG} \cdot 2(-6)$		
$R_{RG}-1$	命令	フッタ 1208

【図 17】



フロントページの続き

(74)代理人 100112690

弁理士 太佐 種一

(72)発明者 ファレル、マーク、エス

アメリカ合衆国 1 2 6 0 1 ニューヨーク州 ポキプシー サウス・ロード 2 4 5 5

(72)発明者 ゲイニー、ジュニア、チャールズ、ダブリュー

アメリカ合衆国 1 2 6 0 1 ニューヨーク州 ポキプシー サウス・ロード 2 4 5 5

(72)発明者 シャム、チャンルン、ケビン

アメリカ合衆国 1 2 6 0 1 ニューヨーク州 ポキプシー サウス・ロード 2 4 5 5

(72)発明者 スレゲル、ティモシー、ジェイ

アメリカ合衆国 1 2 6 0 1 ニューヨーク州 ポキプシー サウス・ロード 2 4 5 5

審査官 多胡 滋

(56)参考文献 特開2010-086520 (JP, A)

(58)調査した分野(Int.Cl., DB名)

G 0 6 F 1 1 / 3 4

G 0 6 F 1 1 / 3 6