

(19) 日本国特許庁(JP)

(12) 公表特許公報(A)

(11) 特許出願公表番号

特表2015-523653  
(P2015-523653A)

(43) 公表日 平成27年8月13日(2015.8.13)

(51) Int.Cl.	F I			テーマコード (参考)		
<b>G06F 9/30 (2006.01)</b>	G06F	9/30	350B	5B033		
<b>G06F 9/34 (2006.01)</b>	G06F	9/34	350A			
<b>G06F 9/46 (2006.01)</b>	G06F	9/46	430			

審査請求 未請求 予備審査請求 未請求 (全 64 頁)

(21) 出願番号 特願2015-516692 (P2015-516692)  
 (86) (22) 出願日 平成24年11月22日 (2012.11.22)  
 (85) 翻訳文提出日 平成26年9月29日 (2014.9.29)  
 (86) 国際出願番号 PCT/IB2012/056625  
 (87) 国際公開番号 W02013/186601  
 (87) 国際公開日 平成25年12月19日 (2013.12.19)  
 (31) 優先権主張番号 13/524,887  
 (32) 優先日 平成24年6月15日 (2012.6.15)  
 (33) 優先権主張国 米国 (US)

(71) 出願人 390009531  
 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  
 INTERNATIONAL BUSINESS MACHINES CORPORATION  
 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  
 (74) 代理人 100108501  
 弁理士 上野 剛史  
 (74) 代理人 100112690  
 弁理士 太佐 種一

最終頁に続く

(54) 【発明の名称】 NONTRANSACTIONALSTORE 命令

(57) 【要約】

【課題】 コンピューティング環境内で命令を実行するためのコンピュータ・プログラム製品、コンピュータ・システム、及び方法を提供する。

【解決手段】 トランザクション実行モードで実行される、NONTRANSACTIONAL STORE 命令が、命令と関連したトランザクションがアポートした場合にも保持されるストアを実行する。ストアは、アポートされたトランザクションのデバッグを容易にすることができるユーザ指定の情報を含む。

【選択図】 図 13

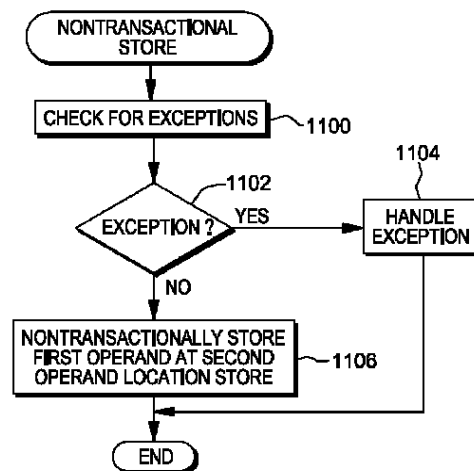


FIG. 11A

**【特許請求の範囲】****【請求項 1】**

コンピューティング環境内で命令を実行するためのコンピュータ・プログラム製品であって、前記コンピュータ・プログラム製品は、

処理回路により読み出し可能であり、且つ、方法を実施するための、前記処理回路により実行される命令を格納するコンピュータ可読ストレージ媒体を含み、前記方法は、

プロセッサにより、実行のためのマシン命令を取得することであって、前記マシン命令は、コンピュータ・アーキテクチャによるコンピュータ実行のために定められたものであり、且つ、

非トランザクション・ストア操作を指定するためのオペコードと、

第 1 のオペランドと、

前記第 1 のオペランドの位置を指し示すための第 2 のオペランドと、

を含む、取得することと、

前記プロセッサにより、前記マシン命令を実行することと、

を含み、前記実行することは、

前記第 1 のオペランドを、前記第 2 のオペランドにより指定された位置に非トランザクション的に配置することを含み、前記第 2 のオペランドに格納された情報は、前記マシン命令と関連したトランザクションのアポートにもかかわらず保持され、前記非トランザクション的に配置することは前記プロセッサのトランザクション実行モードの終了まで遅延される、コンピュータ・プログラム製品。

10

20

**【請求項 2】**

前記トランザクション実行モードの終了は、前記マシン命令と関連した最外トランザクションの終了又はアポート条件によって生じる、請求項 1 に記載のコンピュータ・プログラム製品。

**【請求項 3】**

複数の非トランザクション・ストレージは、他のプロセッサに対して同時実行ストアのように見える、請求項 1 に記載のコンピュータ・プログラム製品。

**【請求項 4】**

前記方法は、

前記プロセッサがトランザクション実行モードにあるかどうかを判断することと、

前記プロセッサが前記トランザクション実行モードにあることに基づいて、前記トランザクションが制約付きトランザクションであるか又は制約なしトランザクションであるかを判断することと、

前記トランザクションが制約なしトランザクションであることに基づいて、前記マシン命令の実行を続行することと、

をさらに含む、請求項 1 に記載のコンピュータ・プログラム製品。

30

**【請求項 5】**

前記トランザクションが制約付きトランザクションであることに基づいて、プログラム例外をもたらし、前記マシン命令の実行を終了させる、請求項 4 に記載のコンピュータ・プログラム製品。

40

**【請求項 6】**

前記トランザクションがトランザクション実行モードにないことに基づいて、前記マシン命令を STORE 命令として実行する、請求項 4 に記載のコンピュータ・プログラム製品。

**【請求項 7】**

前記第 2 のオペランドは、前記マシン命令のインデックス・フィールドにより指定されるレジスタの内容、前記マシン命令のベース・フィールドにより指定されるレジスタの内容、及び少なくとも 1 つの変位フィールドの内容の結合から形成される、請求項 1 に記載のコンピュータ・プログラム製品。

**【請求項 8】**

50

前記マシン命令はNONTRANSACTIONAL STORE命令を含み、前記方法は、

TRANSACTION BEGIN命令の実行に基づいて前記トランザクションを開始することであって、前記トランザクションは、選択されたトランザクションの完了まで主メモリへのトランザクション・ストアのコミットを効果的に遅延させる、開始することと、

前記選択されたトランザクションの実行を完了するTRANSACTION END命令の実行に基づいて、主メモリへの前記トランザクション・ストアをコミットすることと、

前記選択されたトランザクションの実行を終了するアボートに基づいて、前記トランザクション・ストアを廃棄することと、  
をさらに含み、

前記非トランザクション的に配置することは、前記コミットすること又は前記廃棄することに関係なく実行される、請求項1に記載のコンピュータ・プログラム製品。

【請求項9】

前記第1のオペランドは、前記マシン命令により指定されるレジスタの内容を含み、前記内容はユーザ指定のものである、請求項1に記載のコンピュータ・プログラム製品。

【請求項10】

前記トランザクションは、制約なしトランザクションのネスト化における制約なしトランザクションである、請求項1に記載のコンピュータ・プログラム製品。

【請求項11】

コンピューティング環境内で命令を実行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、

を含み、前記コンピュータ・システムは方法を実行するように構成され、前記方法は、

プロセッサにより、実行のためのマシン命令を取得することであって、前記マシン命令は、コンピュータ・アーキテクチャによるコンピュータ実行のために定められたものであり、且つ、

非トランザクション・ストア操作を指定するためのオペコードと、

第1のオペランドと、

前記第1のオペランドの位置を指し示すための第2のオペランドと、

を含む、取得することと、

前記プロセッサにより、前記マシン命令を実行することと、

を含み、前記実行することは、

前記第1のオペランドを、前記第2のオペランドにより指定された位置に非トランザクション的に配置することを含み、前記第2のオペランドに格納された情報は、前記マシン命令と関連したトランザクションのアボートにもかかわらず保持され、前記非トランザクション的に配置することは前記プロセッサのトランザクション実行モードの終了まで遅延される、コンピュータ・システム。

【請求項12】

複数の非トランザクション・ストレージは、他のプロセッサに対する同時実行ストアのように見える、請求項11に記載のコンピュータ・システム。

【請求項13】

前記方法は、

前記プロセッサがトランザクション実行モードにあるかどうかを判断することと、

前記プロセッサが前記トランザクション実行モードにあることに基づいて、前記トランザクションが制約付きトランザクションであるか又は制約なしトランザクションであるかを判断することと、

前記トランザクションが制約なしトランザクションであることに基づいて、前記マシン命令の実行を続行することと、

10

20

30

40

50

をさらに含む、請求項 11 に記載のコンピュータ・システム。

【請求項 14】

前記トランザクションが制約付きトランザクションであることに基づいて、プログラム例外をもたらす、前記マシン命令の実行を終了させる、請求項 13 に記載のコンピュータ・システム。

【請求項 15】

前記トランザクションがトランザクション実行モードにないことに基づいて、前記マシン命令を STORE 命令として実行する、請求項 13 に記載のコンピュータ・システム。

【請求項 16】

前記第 1 のオペランドは、前記マシン命令により指定されるレジスタの内容を含み、前記内容はユーザ指定のものであり、前記第 2 のオペランドは、前記マシン命令のインデックス・フィールドにより指定されるレジスタの内容、前記マシン命令のベース・フィールドにより指定されるレジスタの内容、及び少なくとも 1 つの変位フィールドの内容の結合から形成される、請求項 11 に記載のコンピュータ・システム。

10

【請求項 17】

前記トランザクションは、制約なしトランザクションのネスト化における制約なしトランザクションである、請求項 11 に記載のコンピュータ・システム。

【請求項 18】

コンピューティング環境内で命令を実行するための方法であって、前記方法は、プロセッサにより、実行のためのマシン命令を取得することであって、前記マシン命令は、コンピュータ・アーキテクチャによるコンピュータ実行のために定められたものであり、且つ、

20

非トランザクション・ストア操作を指定するためのオペコードと、

第 1 のオペランドと、

前記第 1 のオペランドの位置を指し示すための第 2 のオペランドと、

を含む、取得することと、

前記プロセッサにより、前記マシン命令を実行することと、

を含み、前記実行することは、

前記第 1 のオペランドを、前記第 2 のオペランドにより指定された位置に非トランザクショナルに配置することを含み、前記第 2 のオペランドに格納された情報は、前記マシン命令と関連したトランザクションのアポートにもかかわらず保持され、前記非トランザクショナルに配置することは前記プロセッサのトランザクション実行モードの終了まで遅延される、方法。

30

【請求項 19】

複数の非トランザクション・ストアは、他のプロセッサに対して同時実行ストアのように見える、請求項 18 に記載の方法。

【請求項 20】

前記プロセッサがトランザクション実行モードにあるかどうかを判断することと、

前記プロセッサが前記トランザクション実行モードにあることに基づいて、前記トランザクションが制約付きトランザクションであるか又は制約なしトランザクションであるかを判断することと、

40

前記トランザクションが制約なしトランザクションであることに基づいて、前記マシン命令の実行を続行することと、

をさらに含む、請求項 18 に記載の方法。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、一般に、マルチプロセッシング・コンピューティング環境に関し、具体的には、こうしたコンピューティング環境内のトランザクション処理に関する。

【背景技術】

50

## 【 0 0 0 2 】

複数の中央演算処理装置（CPU）による同一のストレージ位置の更新は、マルチプロセッサ・プログラミングの長年の課題である。ANDなどの単純な論理演算さえも含めて、ストレージ位置を更新する多くの命令が、その位置への多重アクセスにより更新を行う。例えば、最初にストレージ位置をフェッチし、次いで、更新結果を戻して格納（ストア）する。

## 【 0 0 0 3 】

複数のCPUが安全に同一のストレージ位置を更新するために、その位置へのアクセスがシリアル化される。1つの命令、即ちインターナショナル・ビジネス・マシーンス・コーポレーションにより以前に提供されたS/360アーキテクチャで導入されたTEST AND SET命令が、ストレージ位置のインターロック更新を提供した。インターロック更新とは、他のCPU及び入力/出力（I/O）サブシステム（例えば、チャンネル・サブシステム）が観察するとき、命令のストレージ・アクセス全体がアトミックに行われるように見えることを意味する。後にインターナショナル・ビジネス・マシーンス・コーポレーションにより提供されるS/370アーキテクチャは、インターロック更新を実行するより高度な手段を提供し、且つ、ロックワード（又は、セマフォ）として一般的に知られるものの実装を可能にするCOMPARE AND SWAP及びCOMPARE DOUBLE AND SWAP命令を導入した。COMPARE AND SWAP AND PURGE及びCOMPARE AND SWAP AND STOREを含む、近年付加された命令は、付加的なインターロック更新能力を提供している。しかしながら、これら命令は全て、単一のストレージ位置のみに対するインターロックを提供するものである。

## 【 0 0 0 4 】

要素を二重連結リストに加える際などの、より複雑なプログラム技術では、複数のストレージ位置のインターロック更新が必要とされ得る。他のCPU及びI/Oサブシステムが観察するとき、こうした操作においては、正方向ポインタ及び逆方向ポインタの両方が同時に更新されるように見える。こうした複数の位置更新を行うために、プログラムは、ロックワード等の別個の単一のシリアル化ポイントを用いるように強制される。しかしながら、ロックワードは、保証されるよりかなり粗いレベルのシリアル化を提供することがあり、例えば、2つの要素だけが更新されるとしても、ロックワードは何百万の要素のキュー全体をシリアル化することがある。プログラムは、より細かい粒度のシリアル化（例えば、ロック・ポイントの階層構造）を使用するように、データを構築できるが、そのことは、階層構造に違反した場合の潜在的なデッドロック状況、及び1つ又は複数のロックを保持する間にプログラムがエラーに遭遇した場合又はロックを取得できない場合の復旧の問題といった、付加的な問題をもたらす。

## 【 0 0 0 5 】

上記に加えて、プログラムは、例外条件をもたらすことも又はもたらさないこともある一連の命令を実行することができる多数のシナリオがある。例外条件が発生しない場合、プログラムは続行するが、例外条件が認識された場合、プログラムは、修正アクションをとり、その例外条件を取り除くことができる。一例として、Java（登録商標）は、例えば、投機的実行、関数の部分的インライン化、及び/又はポインタ・ヌル・チェックの再順序付けなどにおいて、こうした実行を活用することができる。

## 【 0 0 0 6 】

インターナショナル・ビジネス・マシーンス・コーポレーションにより提供されるz/OS（登録商標）及びその前身のような、典型的なオペレーティング・システム環境において、プログラムは、回復環境を確立して、プログラムが遭遇し得るあらゆるプログラム例外条件をインターセプトする。プログラムが例外条件をインターセプトしない場合、典型的には、オペレーティング・システムは、対処する準備ができていない例外のために、プログラムを異常終了する。こうした環境の確立及び活用は、費用がかかり複雑である。

## 【 先行技術文献 】

10

20

30

40

50

## 【特許文献】

## 【0007】

【特許文献1】米国特許第5,551,013号明細書

【特許文献2】米国特許第6,009,261号明細書

【特許文献3】米国特許第5,574,873号明細書

【特許文献4】米国特許第6,308,255号明細書

【特許文献5】米国特許第6,463,582号明細書

【特許文献6】米国特許第5,790,825号明細書

## 【非特許文献】

## 【0008】

【非特許文献1】「z/Architecture Principles of Operation」、IBM(登録商標)出版番号SA22-7832-08、第9版、2010年8月

## 【発明の概要】

## 【発明が解決しようとする課題】

## 【0009】

コンピューティング環境内で命令を実行するためのコンピュータ・プログラム製品、コンピュータ・システム、及び方法を提供する。

## 【課題を解決するための手段】

## 【0010】

コンピューティング環境内で命令を実行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り、従来技術の欠点に対処し、利点をもたらされる。コンピュータ・プログラム製品は、処理回路により読み出し可能であり、且つ、方法を実施するための、処理回路により実行される命令を格納するコンピュータ可読ストレージ媒体を含む。この方法は、プロセッサにより、実行のためのマシン命令を取得することであって、このマシン命令は、コンピュータ・アーキテクチャによるコンピュータ実行のために定められたものであり、且つ、非トランザクション・ストア操作を指定するためのオペコードと、第1のオペランドと、第1のオペランドの位置を指し示すための第2のオペランドとを含む、取得することと、プロセッサにより、マシン命令を実行することとを含み、この実行することは、第1のオペランドを、第2のオペランドにより指定された位置に非トランザクション的に配置することを含み、第2のオペランドに格納された情報は、マシン命令と関連したトランザクションのアポートにもかかわらず保持され、非トランザクション的に配置することは、プロセッサのトランザクション実行モードの終了まで遅延される。

## 【0011】

1つ又は複数の実施形態に関連する方法及システムも、本明細書で説明され、特許請求される。

付加的な特徴及び利点の実現される。他の実施形態及び態様は、本明細書で詳細に説明され、特許請求される本発明の一部と見なされる。

ここで、添付図面を参照して、単なる例として本発明の実施形態を説明する。

## 【図面の簡単な説明】

## 【0012】

【図1】コンピューティング環境の一実施形態を示す。

【図2】TRANSACTION BEGIN(TBEGIN)命令の一例を示す。

【図3】図2のTBEGIN命令のフィールドのさらなる詳細の一実施形態を示す。

【図4】TRANSACTION BEGIN constrained(TBEGIN C)命令の一例を示す。

【図5】図4のTBEGIN C命令のフィールドのさらなる詳細の一実施形態を示す。

【図6】TRANSACTION END(TEND)命令の一例を示す。

【図7】TRANSACTION ABORT(TABORT)命令の一例を示す。

【図8】ネスト化されたトランザクションの一例を示す。

10

20

30

40

50

【図9】NONTRANSACTIONAL STORE (NTSTG) 命令の一例を示す。

【図10】EXTRACT TRANSACTION NESTING DEPTH (ETND) 命令の一例を示す。

【図11】トランザクション診断ブロックの一例を示す。

【図12】関連したアポート・コード及び条件コードと共に、アポートの例示的な理由を示す。

【図13】NONTRANSACTIONAL STORE 命令と関連した論理の一例を示す。

【図14】NONTRANSACTIONAL STORE 命令と関連した付加的な論理の一例を示す。

【図15】コンピュータ・プログラム製品の一実施形態を示す。

【図16】ホスト・コンピュータ・システムの一実施形態を示す。

【図17】コンピュータ・システムの更に別の例を示す。

【図18】コンピュータ・ネットワークを含むコンピュータ・システムの別の例を示す。

【図19】コンピュータ・システムの種々の要素の一実施形態を示す。

【図20】図19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21】図19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22】図19のコンピュータ・システムのロード/ストア・ユニットの一実施形態を示す。

【図23】エミュレートされたホスト・コンピュータ・システムの一実施形態を示す。

【発明を実施するための形態】

【0013】

一実施形態によると、トランザクション実行 (TX) ファシリティが提供される。このファシリティは、命令についてのトランザクション処理を提供し、1つ又は複数の実施形態においては、以下に説明されるような異なる実行モード、並びにネスト化されたレベルのトランザクション処理を提供する。

【0014】

トランザクション実行ファシリティは、トランザクション実行 (TX) モードと呼ばれる CPU 状態を導入する。CPU のリセット後、CPU は、TX モードにはない。CPU は、TRANSACTION BEGIN 命令により TX モードに入る。CPU は、(a) 最も外側の (最外) TRANSACTION END 命令 (「内」及び「外」の詳細については後述) により、又は (b) トランザクションがアポートされることにより、TX モードを終了する。TX モードにある間、他の CPU 及び I/O サブシステムが観察するとき、CPU によるストレージ・アクセスは、ブロック・コンカレントであるように見える。ストレージ・アクセスは、(a) 最外トランザクションがアポートすることなく終了したときに、ストレージにコミットされる (即ち、例えば、キャッシュ又は CPU にローカルなバッファにおいて行われた更新が伝搬され、実メモリに格納され、他の CPU に可視となる) か、又は (b) トランザクションがアポートされた場合に廃棄される。

【0015】

トランザクションをネスト化することができる。つまり、CPU は TX モードにある間、CPU は、別の TRANSACTION BEGIN 命令を実行することができる。CPU が TX モードに入るようにする命令は、最外 TRANSACTION BEGIN と呼ばれ、同様に、プログラムは最外トランザクション内にあると言われる。TRANSACTION BEGIN の後続の実行は、内部命令と呼ばれ、プログラムは、内部トランザクションを実行している。モデルは、最小ネスト化深さ及びモデル依存の最大ネスト化深さを提供する。EXTRACT TRANSACTION NESTING DEPTH 命令は、現在のネスト化深さ値を返し、更に別の実施形態においては、最大ネスト化深さ値を返すことができる。この技術は、「平坦化されたネスト化」と呼ばれるモデルを使用し、そこでは、任意のネスト化深さにおけるアポート条件により、全てのレベルのト

10

20

30

40

50

ランザクションがアボートされ、制御は、最外 TRANSACTION BEGIN に続く命令に返される。

【0016】

ランザクションの処理中、両方のアクセスが同じキャッシュ・ライン内のいずれかの記憶場所に対するものであり、且つ、アクセスの一方又は両方がストアである場合、1つのCPUによりなされるランザクション・アクセスは、(a)別のCPUによりなされるランザクション・アクセス又は非ランザクション・アクセス、又は(b)I/Oサブシステムによりなされる非ランザクション・アクセスのいずれかと競合すると言われる。換言すれば、ランザクション実行が生産的であるために、CPUは、これがコミットするまで、ランザクション・アクセスを行っているように見られない。このプログラム・モデルは、例えば百万個の要素の二重連結リスト内の2つのポイントを更新するといった特定の環境においては、非常に有効であり得る。しかしながら、このプログラム・モデルは、ランザクション・アクセスが行われるストレージ位置に対する競合が多数ある場合、あまり有効ではない。

10

【0017】

ランザクション実行の1つのモデル(本明細書では、制約なしランザクションと呼ぶ)においては、ランザクションがアボートされると、プログラムは、アボート条件がもはや存在しないことを期待してランザクションの再駆動を試みることで、又は、プログラムが同等の非ランザクション経路に「フォールバックすること」ができる。ランザクション実行の別のモデル(本明細書では、制約付きランザクションと呼ぶ)においては、アボートされたランザクションは、CPUにより自動的に再駆動され、制約違反がない限り、制約付きランザクションが最終的に完了することが保証される。

20

【0018】

ランザクションを開始する際、プログラムは、(a)ランザクションがアボートされた場合、どの汎用レジスタをオリジナルの内容に復元するか、(b)ランザクションが、例えば浮動小数点レジスタ及び浮動小数点制御レジスタを含む、浮動小数点レジスタのコンテキストの変更を許可されているかどうか、(c)ランザクションは、アクセス・レジスタ(AR)の変更を許可されているかどうか、及び(d)特定のプログラム例外条件が、割り込みを発生させないようにブロックされているかどうかといった、種々の制御を指定することができる。制約なしランザクションがアボートされた場合、種々の診断情報を提供することができる。例えば、制約なしランザクションを開始する最外 TRANSACTION BEGIN 命令は、プログラムが指定したランザクション診断ブロック(TDB)を指定することができる。さらに、ランザクションが、プログラム割り込み又は解釈実行を終了させる条件に起因してアボートされた場合、それぞれ、CPUのプリフィックス領域内のTDB又はホストの状態記述により指定されるTDBを用いることもできる。

30

【0019】

種々のタイプのレジスタが上述されている。これらは、本明細書でさらに詳細に説明される。汎用レジスタは、一般算術演算及び論理演算においてアキュムレータとして使用される。一実施形態において、各々のレジスタは、64個のビット位置を含み、16個の汎用レジスタがある。汎用レジスタは、番号0-15により識別され、命令内の4ビットのRフィールドによって指定される。ある命令は、数個のRフィールドを有することによって、複数の汎用レジスタをアドレス指定することを規定する。命令によっては、特定の汎用レジスタの使用は、命令のRフィールドによって明示的に指定されるのではなく、暗黙指定される。

40

【0020】

一般算術演算及び論理演算においてアキュムレータとして使用されるのに加えて、16個の汎用レジスタのうち15個は、アドレス生成におけるベース・アドレス及び指標レジスタとしても用いられる。これらの場合、レジスタは、命令内の4ビットのBフィールド又はXフィールドによって指定される。Bフィールド又はXフィールドにおける0の値は、適用されるべきベース又は指標が存在しないことを指定し、従って、汎用レジスタ0

50



は、ベース・アドレス又は指標を含むものとして指定されない。

【0021】

浮動小数点命令は、1組の浮動小数点レジスタを使用する。一実施形態において、CPUは、16個の浮動小数点レジスタを有する。浮動小数点レジスタは、番号0 - 15により識別され、浮動小数点命令内の4ビットのRフィールドによって指定される。各々の浮動小数点レジスタは、64ビットの長さであり、且つ、短い(32ビット)又は長い(64ビット)浮動小数点オペランドを含むことができる。

【0022】

浮動小数点制御(FPC)レジスタは、マスク・ビット、フラグ・ビット、データ例外コード及び丸めモード・ビットを含む32ビット・レジスタであり、浮動小数点演算の処理の際に使用される。

10

【0023】

さらに、一実施形態において、CPUは、各々が64ビット位置を有する16個の制御レジスタを有する。レジスタにおけるビット位置は、プログラム・イベント記録(PER)(以下に説明される)のような、システム内の特定のファシリティに割り当てられ、且つ、演算が行われることを規定するため又はファシリティが必要とする特別な情報を与えるために用いられる。一実施形態においては、以下に説明されるように、トランザクション・ファシリティについては、CR0(ビット8及び9)及びCR2(ビット61 - 63)が用いられる。

【0024】

20

例えば、CPUは、0 - 15の番号が付された16個のアクセス・レジスタを有する。アクセス・レジスタは、アドレス空間制御要素(ASCE)の間接的指定を含む32個のビット位置から成る。アドレス空間制御要素は、対応するアドレス空間への参照を変換するための動的アドレス変換(DAT)機構により用いられるパラメータである。CPUがアクセス・レジスタ・モード(プログラム状況ワード(PSW)内のビットにより制御される)と呼ばれるモードにあるとき、ストレージ・オペランド参照のための論理アドレスを指定するために用いられる命令のBフィールドがアクセス・レジスタを指定し、アクセス・レジスタにより指定されるアドレス空間制御要素が、行われている参照用にDATにより使用される。一部の命令では、Bフィールドの代わりにRフィールドが使用される。アクセス・レジスタの内容をロードし、格納するための命令、及び1つのアクセス・レジスタの内容を別のアクセス・レジスタに移動するための命令が与えられる。

30

【0025】

アクセス・レジスタ1 - 15の各々は、任意のアドレス空間を指定することができる。アクセス・レジスタ0は、一次命令空間を指定する。アクセス・レジスタ1 - 15の1つを用いてアドレス空間を指定する場合、CPUは、アクセス・レジスタの内容を変換することにより、どのアドレス空間が指定されているのかを判断する。アクセス・レジスタ0を使用してアドレス空間を指定する場合は、CPUは、そのアクセス・レジスタを、一次命令空間を指定しているものとして扱い、アクセス・レジスタの実際の内容を検査しない。従って、16個のアクセス・レジスタは、任意の一時点において、一次命令空間と、最大15個の他の空間を指定することができる。

40

【0026】

一実施形態において、複数のタイプのアドレス空間がある。アドレス空間は、各々の番号をストレージ内のバイト位置と関連付けることを可能にする特別な変換パラメータとともに、連続した整数番号のシーケンス(仮想アドレス)である。シーケンスは0から始まり、左から右へ進む。

【0027】

例えば、z/Architectureにおいて、主ストレージ(主メモリともいう)にアクセスするためにCPUにより仮想アドレスが使用されると、その仮想アドレスは、最初に、動的アドレス変換(DAT)により実アドレスに変換され、次に、プリフィックス付加により絶対アドレスに変換される。DATは、1乃至5レベルのテーブル(ページ

50

・テーブル、セグメント・テーブル、領域第3テーブル、領域第2テーブル、及び領域第1テーブル)を変換パラメータとして使用することができる。特定のアドレス空間のための最高レベルのテーブルの指定(起点及び長さ)は、アドレス空間制御要素と呼ばれ、これは、DATが使用するために制御レジスタにおいて見出されるか、又はアクセス・レジスタにより指定される。代替的に、アドレス空間のためのアドレス空間制御要素は、実空間指定とすることができ、この実空間指定は、DATが、いずれのテーブルも使用することなく、単に仮想アドレスを実アドレスとして扱うことにより、仮想アドレスを変換することを示す。

【0028】

DATは、その時々において、異なる制御レジスタ内のアドレス空間制御要素又はアクセス・レジスタにより指定されるアドレス空間制御要素を使用する。その選択は、現PSWにおいて指定される変換モードによって決定される。一次空間モード、二次空間モード、アクセス・レジスタ・モード及びホーム空間モードの4つの変換モードが利用可能である。変換モードに応じて、異なるアドレス空間がアドレス指定可能である。

10

【0029】

CPUが一次空間モード又は二次空間モードにある何れの時点でも、CPUは、2つのアドレス空間即ち一次アドレス空間及び二次アドレス空間に属する仮想アドレスを変換することができる。CPUがアクセス・レジスタ・モードにある何れの時点でも、CPUは、最大16個のアドレス空間(一次アドレス及び最大15個のAR指定アドレス空間)に変換することができる。CPUがホーム空間モードにある何れの時点でも、CPUは、ホーム・アドレス空間の仮想アドレスを変換することができる。

20

【0030】

一次アドレス空間は、一次仮想アドレスからなるため、そのように識別され、一次仮想アドレスは、一次アドレス空間制御要素(ASCE)により変換される。同様に、二次アドレス空間は、二次ASCEによって変換された二次仮想アドレスからなり、AR指定アドレス空間は、AR指定ASCEによって変換されたAR指定仮想アドレスからなり、ホーム・アドレス空間は、ホームASCEによって変換されたホーム仮想アドレスからなる。一次ASCE及び二次ASCEは、それぞれ、制御レジスタ1及び7内にある。AR指定ASCEは、制御レジスタ2、5及び8を用いてアクセス・レジスタ変換(ART)と呼ばれるプロセスを通して配置されるASN第2テーブル・エントリ内にある。ホームASCEは、制御レジスタ13内にある。

30

【0031】

図1を参照して、本明細書で説明されるトランザクション・ファシリティの1つ又は複数の態様を組み込み、使用するためのコンピューティング環境の一実施形態を説明する。

【0032】

図1を参照すると、一例において、コンピューティング環境100は、ニューヨーク州アーモンク所在のインターナショナル・ビジネス・マシーンス・コーポレーションにより提供されるz/Architectureに基づいている。z/Architectureは、全体が引用により本明細書に組み入れられる非特許文献1に記載されている。

【0033】

IBM、並びにZ/OS及びZ/VM(以下に参照される)は、ニューヨーク州アーモンク所在のインターナショナル・ビジネス・マシーンス・コーポレーションの登録商標である。本明細書で用いられる他の名称は、インターナショナル・ビジネス・マシーンス・コーポレーション又は他の企業の登録商標、商標、又は製品名であり得る。

40

【0034】

一例として、コンピューティング環境100が、1つ又は複数の制御ユニット108を介して1つ又は複数の入力/出力(I/O)デバイス106に結合された中央プロセッサ・コンプレックス(CPC)102を含む。中央プロセッサ・コンプレックス(CPC)102は、例えば、1つ又は複数の中央プロセッサ110、1つ又は複数のパーティション112(例えば論理パーティション(LP))、論理パーティション・ハイパーバイザ

50

114、及び入力/出力サブシステム115を含み、これらの各々を以下に説明する。

【0035】

1つ又は複数の中央プロセッサ110は、論理パーティションに割り当てられた物理プロセッサ・リソースである。特に、それぞれの論理パーティション112は、各々がパーティションに割り当てられた物理プロセッサ110の全て又は割り当て分を表す、1つ又は複数の論理プロセッサを有する。特定のパーティション112の論理プロセッサは、そのパーティションに対して基礎をなすプロセッサ・リソース110が予約されるように、そのパーティション専用であっても、又は、基礎をなすプロセッサ・リソースが別のパーティションに対して潜在的に利用可能であるように、別のパーティションと共有してもよい。

10

【0036】

論理パーティションは、別個のシステムとして機能し、且つ、1つ又は複数のアプリケーション、及び随意的に、各々の論理パーティションについて異なり得る常駐オペレーティング・システムを有する。一実施形態において、オペレーティング・システムは、ニューヨーク州アーモンク所在のインターナショナル・ビジネス・マシーンズ・コーポレーションにより提供されるz/OSオペレーティング・システム、z/VMオペレーティング・システム、z/Linuxオペレーティング・システム、又はTPFオペレーティング・システムである。論理パーティション112は、プロセッサ110上で実行されているファームウェアが実装する論理パーティション・ハイパーバイザ114により管理される。本明細書で用いられるファームウェアは、例えば、プロセッサのマイクロコード及び/又はミリコードを含む。ファームウェアは、例えば、より上位レベルのマシン・コードの実装に用いられる、ハードウェア・レベルの命令及び/又はデータ構造体を含む。一実施形態において、ファームウェアは、例えば、典型的には、信頼できるソフトウェアを含むマイクロコード又は基礎をなすハードウェアに特有のマイクロコードとして配信される独自のコードを含み、システム・ハードウェアへのオペレーティング・システムのアクセスを制御する。

20

【0037】

論理パーティション及び論理パーティション・ハイパーバイザの各々は、中央プロセッサと関連した中央ストレージのそれぞれのパーティション内に常駐する1つ又は複数のプログラムを含む。論理パーティション・ハイパーバイザ114の一例は、ニューヨーク州アーモンク所在のインターナショナル・ビジネス・マシーンズ・コーポレーションが提供するプロセッサ・リソース/システム・マネージャ(PR/SM)である。

30

【0038】

入力/出力サブシステム115は、入力/出力デバイス106と主ストレージ(主メモリともいう)との間の情報の流れを方向付ける。この入力/出力サブシステム115は、中央処理コンプレックスに結合され、中央処理コンプレックスの一部とすることも又はそれとは別個のものとすることもできる。I/Oサブシステムは、中央プロセッサを、I/Oデバイスと直接通信するタスクから解放し、データ処理が、入力/出力処理と同時に進行することを可能にする。通信を提供するために、I/Oサブシステムは、I/O通信アダプタを用いる。例えば、チャンネル、I/Oアダプタ、PCIカード、イーサネット・カード、Small Computer System Interface(SCSI)カード等を含む、種々のタイプの通信アダプタがある。本明細書で説明される特定の例において、I/O通信アダプタはチャンネルであり、従って、I/Oサブシステムは、本明細書ではチャンネル・サブシステムと呼ばれる。しかしながら、これは一例にすぎない。他のタイプのI/Oサブシステムを用いることもできる。

40

【0039】

I/Oサブシステムは、入力/出力デバイス106との間の情報の流れを管理する際に、1つ又は複数の入力/出力経路を通信リンクとして使用する。この特定の例において、通信アダプタはチャンネルであるので、これらの経路は、チャンネル経路と呼ばれる。

【0040】

50

上述のコンピューティング環境は、用いることができるコンピューティング環境の一例にすぎない。これらに限定されるものではないが、区分化されていない環境、他の区分化された環境、及び/又はエミュレートされた環境を含む他の環境を用いることができ、実施形態は、いずれか1つの環境に限定されるものではない。

【0041】

1つ又は複数の態様によると、トランザクション実行ファシリティは、CPUの機能強化であり、これは、CPUが、(トランザクションとして知られる)一連の命令を実行することができる手段を提供し、それらの記憶場所の更新を含む、複数のストレージ位置にアクセスすることができる。他のCPU及びI/Oサブシステムにより観察されるとき、トランザクションは、(a)単一のアトミック操作として全体的に完了されるか、又は(b)潜在的に、トランザクションがこれまでに実行されたという証拠を残さずにアボートされる(本明細書で説明される特定の条件を除いて)。従って、成功裏に完了したトランザクションは、典型的なマルチプロセッシング・モデルにおいて必要とされるいずれのロック機構もなしに、多数のストレージ位置を更新することができる。

10

【0042】

トランザクション実行ファシリティは、例えば、1つ又は複数の制御、1つ又は複数の命令、制約付き実行及び制約なし実行を含むトランザクション処理、アボート処理を含み、その各々を以下にさらに説明する。

【0043】

一実施形態において、トランザクション・アボート・プログラム状況ワード(PSW)、トランザクション診断ブロック(TDB)アドレス、及びトランザクション・ネスト化深さを含む3つの特殊用途の制御、5つの制御レジスタのビット、及びTRANSACTION BEGIN(制約付き及び制約なし)、TRANSACTION END、EXTRACT TRANSACTION NESTING DEPTH、TRANSACTION ABORT、及びNONTRANSACTIONAL STOREを含む6つの一般命令を用いて、トランザクション実行ファシリティを制御する。ファシリティがインストールされる際、ファシリティは、例えば、構成内の全てのCPU内にインストールされる。ファシリティ表示は、1つの実施においてはビット73であり、1の場合、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たことを示す。

20

【0044】

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ると、構成は、制約なしトランザクション実行ファシリティと、随意的に、制約付きトランザクション実行ファシリティとを提供し、それらの各々を以下に説明する。例証として、ファシリティ表示50及び73が共に1であるとき、制約付き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る。両方のファシリティ表示は、メモリの特定の位置に格納される。

30

【0045】

本明細書で用いられる場合、命令名TRANSACTION BEGINは、簡略記号TBEGIN(制約なしトランザクションに対するTRANSACTION BEGIN)及び簡略記号TBEGINC(制約付きトランザクションに対するTRANSACTION BEGIN)を有する命令を指す。特定の命令に関する説明は、命令名と、それに続く丸括弧又は角括弧内の簡略記号、又は単に簡略記号によって示される。

40

【0046】

TRANSACTION BEGIN(TBEGIN)命令の形式の一実施形態が、図2-図3に示される。一例として、TBEGIN命令200は、制約なしトランザクション開始操作を指定するオペコードを含むオペコード・フィールド202と、ベース・フィールド( $B_1$ )204と、変位フィールド( $D_1$ )206と、即値フィールド( $I_2$ )208とを含む。 $B_1$ フィールドが0でないとき、 $B_1$ 204が指定する汎用レジスタの内容を $D_1$ 206に付加して、第1のオペランド・アドレスを取得する。

【0047】

$B_1$ フィールドが0でないとき、以下が適用される。:

50

\* トランザクション・ネスト化深さが最初に0のとき、第1のオペランド・アドレスは、トランザクションがアボートされた場合に種々の診断情報を格納することができる、T B E G I N 指定 T D B (以下にさらに説明される)と呼ばれる、256バイトのトランザクション診断ブロックの記憶位置を指定する。C P U が一次空間モード又はアクセス・レジスタ・モードにあるとき、第1のオペランド・アドレスは、一次アドレス空間内の記憶位置を指定する。C P U が二次空間モード又はホーム空間モードにあるとき、第1のオペランド・アドレスは、それぞれ、二次アドレス空間又はホーム・アドレス空間内の記憶位置を指定する。D A T がオフのとき、トランザクション診断ブロック ( T D B ) アドレス ( T D B A ) は、実ストレージ内の記憶位置を指定する。

【0048】

第1のオペランドに対するストア・アクセス可能性が判断される。アクセス可能である場合、オペランドの論理アドレスは、トランザクション診断ブロック・アドレス ( T D B A ) 内に入れられ、T D B A は有効である。

【0049】

\* C P U が既に制約なシトランザクション実行モードにあるとき、T D B A は変更されず、アクセス可能性について第1のオペランドが試験されるかどうかは予測不能である。

B<sub>1</sub> フィールドが0のとき、第1のオペランドについてアクセス例外は検出されず、最外 T B E G I N 命令について、T D B A は無効である。

【0050】

一例において、I<sub>2</sub> フィールドのビットは、次のように定められる。:

汎用レジスタ保存マスク ( G R S M ) 210 ( 図3 ) : I<sub>2</sub> フィールドのビット0 - 7 は、汎用レジスタ保存マスク ( G R S M ) を含む。G R S M の各ビットは、汎用レジスタの偶数 - 奇数対を表し、ここでビット0はレジスタ0及び1を表し、ビット1はレジスタ2及び3を表し、以下同様である。最外 T B E G I N 命令の G R S M におけるビットが0のとき、対応するレジスタ対は保存されない。最外 T B E G I N 命令の G R S M におけるビットが1のとき、対応するレジスタ対は、プログラムが直接アクセスできないモデル依存位置に保存される。

【0051】

トランザクションがアボートした場合、最外 T B E G I N 命令が実行されたときに、保存されたレジスタ対は、その内容に復元される。トランザクションがアボートするとき、全ての他の ( 保存されていない ) 汎用レジスタの内容が復元されるとは限らない。

汎用レジスタ保存マスクは、最外のものを除いて、全ての T B E G I N 命令において無視される。

【0052】

A R 変更許可 ( A ) 212 : A 制御、即ち I<sub>2</sub> フィールドのビット12は、トランザクションがアクセス・レジスタを変更するのを許可されるかどうかを制御する。有効な A R 変更許可制御は、現在のネスト化レベル及び全ての外部レベルに対する T B E G I N 命令における A 制御の論理 A N D である。

【0053】

有効な A 制御が0のとき、いずれかのアクセス・レジスタを変更しようとする試みがなされた場合、トランザクションは、アボート・コード11 ( 限定された命令 ) でアボートされる。有効な A 制御が1のとき、アクセス・レジスタが変更される場合には、トランザクションはアボートされない ( いずれの他のアボート条件もない場合 ) 。

【0054】

浮動小数点演算許可 ( F ) 214 : F 制御、即ち I<sub>2</sub> フィールドのビット13は、トランザクションが特定の浮動小数点命令を実行するのを許可されているかどうかを制御する。有効な浮動小数点演算許可制御は、現在のネスト化レベル及び全ての外部レベルに対する T B E G I N 命令における F 制御の論理 A N D である。

【0055】

有効な F 制御が0である場合、( a ) 浮動小数点命令を実行しようとする試みがなされ

10

20

30

40

50

た場合、トランザクションは、アボート・コード 11 (限定された命令) でアボートされ、且つ、(b) 浮動小数点制御レジスタ (FPCR) のバイト 2 におけるデータ例外コード (DXC) は、いずれのデータ例外プログラム例外条件によっても設定されない。有効な F 制御が 1 である場合、(a) 浮動小数点命令を実行しようとする試みがなされた場合、トランザクションはアボートされず (いずれの他のアボート条件もない場合)、且つ、(b) データ例外プログラム例外条件により FPCR における DXC を設定することができる。

【0056】

プログラム割り込みフィルタ処理制御 (PIFC) 216 : I<sub>2</sub> フィールドのビット 14 - 15 は、プログラム割り込みフィルタ処理制御 (PIFC) である。PIFC は、CPU がトランザクション実行モードにある間に発生するプログラム例外条件 (例えば、アドレス指定例外、データ例外、操作例外、保護例外等) の特定のクラスが割り込みをもたらすかどうかを制御する。

10

【0057】

有効な PIFC は、現在のネスト化レベル及び全ての外部レベルに対する TBEGIN 命令における PIFC の最高値である。有効な PIFC が 0 のとき、全てのプログラム例外条件は、割り込みをもたらす。有効な PIFC が 1 のとき、1 及び 2 のトランザクション実行クラスを有するプログラム例外条件は、割り込みをもたらす。(例外の重大度に応じて、各々のプログラム例外条件が少なくとも 1 つのトランザクション実行クラスに割り当てられる。重大度は、トランザクションの実行を反復する際の復元の尤度、及び、オペレーティング・システムが割り込みを見る必要があるかどうかに基づく。) 有効な PIFC が 2 のとき、トランザクション実行クラスが 1 であるプログラム例外条件は、割り込みをもたらす。3 の有効な PIFC は予約される。

20

【0058】

I<sub>2</sub> フィールドのビット 8 - 11 (命令のビット 40 - 43) は予約され、0 を含む必要があり、他の場合には、プログラムは、将来互換性をもって動作することができなくなる可能性がある。

【0059】

図 4 - 図 5 を参照して、TRANSACTION BEGIN constrained (TBEGINC) 命令の形式の一実施形態を説明する。一例において、TBEGINC 300 は、制約付きトランザクション開始操作を指定するオペコードを含むオペコード・フィールド 302 と、ベース・フィールド (B<sub>1</sub>) 304 と、変位フィールド (D<sub>1</sub>) 306 と、即値フィールド (I<sub>2</sub>) 308 とを含む。B<sub>1</sub> 304 が指定する汎用レジスタの内容を D<sub>1</sub> 306 に付加して、第 1 のオペランド・アドレスを取得する。しかしながら、TRANSACTION BEGIN constrained 命令においては、ストレージにアクセスするために、第 1 のオペランド・アドレスは使用されない。代わりに、命令の B<sub>1</sub> フィールドは 0 を含み、他の場合には、指定例外が認識される。

30

【0060】

一実施形態において、I<sub>2</sub> フィールドは種々の制御を含み、その一例が図 5 に示される。

40

【0061】

一例において、I<sub>2</sub> フィールドのビットは、次のように定められる。:

【0062】

汎用レジスタ保存マスク (GRSM) 310 : I<sub>2</sub> フィールドのビット 0 - 7 は、汎用レジスタ保存マスク (GRSM) を含む。GRSM の各ビットは、汎用レジスタの偶数 - 奇数対を表し、ここで、ビット 0 はレジスタ 0 及び 1 を表し、ビット 1 はレジスタ 2 及び 3 を表し、以下同様である。GRSM におけるビットが 0 のとき、対応するレジスタ対は保存されない。GRSM におけるビットが 1 のとき、対応するレジスタ対は、プログラムが直接アクセスできないモデル依存位置に保存される。

【0063】

50

トランザクションがアポートした場合、最外 T B E G I N 命令が実行されたときに、保存されたレジスタ対は、その内容に復元される。トランザクションがアポートするとき、全ての他の（保存されていない）汎用レジスタの内容が復元されるとは限らない。

【 0 0 6 4 】

T B E G I N C を使用して制約なしトランザクション実行モードで実行を継続するとき、汎用レジスタ保存マスクは無視される。

【 0 0 6 5 】

A R 変更許可 ( A ) 3 1 2 : A 制御、即ち  $I_2$  フィールドのビット 1 2 は、トランザクションがアクセス・レジスタを変更するのを許可されるかどうかを制御する。有効な A R 変更許可制御は、現在のネスト化レベル及びあらゆる外部 T B E G I N 又は T B E G I N C 命令に対する T B E G I N C 命令における A 制御の論理 A N D である。

10

【 0 0 6 6 】

有効な A 制御が 0 のとき、いずれかのアクセス・レジスタを変更しようとする試みがなされた場合、トランザクションは、アポート・コード 1 1 ( 限定された命令 ) でアポートされる。有効な A 制御が 1 のとき、アクセス・レジスタが変更される場合には、トランザクションはアポートされない ( いずれの他のアポート条件もない場合 ) 。

【 0 0 6 7 】

$I_2$  フィールドのビット 8 - 1 1 及びビット 1 3 - 1 5 ( 命令のビット 4 0 - 4 3 及びビット 4 5 - 4 7 ) は予約され、0 を含む必要がある。

【 0 0 6 8 】

T R A N S A C T I O N B E G I N 命令の終わりは、その形式を図 6 に示す T R A N S A C T I O N E N D ( T E N D ) 命令により指定される。一例として、T E N D 命令 4 0 0 は、トランザクション終了操作を指定するオペコードを含むオペコード・フィールド 4 0 2 を含む。

20

【 0 0 6 9 】

トランザクション実行ファシリティに関して多数の用語が用いられ、従って、便宜のためだけに、用語のリストを以下にアルファベット順で提供する。一実施形態において、これらの用語は、次のように定められる。

【 0 0 7 0 】

アポート : T R A N S A C T I O N E N D 命令の前にトランザクションが終了し、0 のトランザクション・ネスト化深さをもたらすとき、トランザクションがアポートする。一実施形態において、トランザクションがアポートすると、以下のことが行われる。

30

\* 全てのレベルのトランザクションにより行われたトランザクション・ストア・アクセスが廃棄される ( 即ち、コミットされない ) 。

\* 全てのレベルのトランザクションにより行われた非トランザクション・ストア・アクセスがコミットされる。

\* 最外 T R A N S A C T I O N B E G I N 命令の汎用レジスタ保存マスク ( G R S M ) により指定されたレジスタが、トランザクション実行より前のレジスタの内容に復元される ( 即ち、最外 T R A N S A C T I O N B E G I N 命令の実行時のレジスタの内容に ) 。最外 T R A N S A C T I O N B E G I N 命令の汎用レジスタ保存マスクにより指定されていない汎用レジスタは、復元されない。

40

\* アクセス・レジスタ、浮動小数点レジスタ、及び浮動小数点制御レジスタは、復元されない。トランザクションがアポートすると、トランザクション実行中にこれらのレジスタに対してなされた全ての変更が保持される。

【 0 0 7 1 】

トランザクションは、限定された命令の実行を試みることに、限定されたりソースの変更を試みることに、トランザクションの競合、種々の C P U リソースの超過、いずれかの解釈実行インターセプト条件、いずれかの割り込み、T R A N S A C T I O N A B O R T 命令及び他の理由を含む、様々な理由に起因してアポートすることがある。トランザクション・アポート・コードは、トランザクションをアポートできる具体的な理由を提供する。

50

## 【 0 0 7 2 】

図7を参照して、TRANSACTION ABORT (TABORT) 命令の形式の一例を説明する。一例として、TABORT 命令500は、トランザクション・アボート操作を指定するオペコードを含むオペコード・フィールド502と、ベース・フィールド( $B_2$ )504と、変位フィールド( $D_2$ )506とを含む。 $B_2$ フィールドが0でないとき、 $B_2$ 504により指定される汎用レジスタの内容を $D_2$ 506に付加して、第2のオペランド・アドレスを取得し、他の場合には、第2のオペランド・アドレスは、 $D_2$ フィールドのみから形成され、 $B_2$ フィールドは無視される。第2のオペランド・アドレスは、データをアドレス指定するためには用いられず、代わりに、アドレスは、アボート処理中にトランザクション診断ブロックに配置されるトランザクション・アボート・コードを形成する。第2のオペランド・アドレスについてのアドレス計算は、アドレス算術演算の規則に従う。24ビット・アドレッシング・モードにおいては、ビット0 - 29が0に設定され、31ビット・アドレッシング・モードにおいては、ビット0 - 32が0に設定される。

10

## 【 0 0 7 3 】

コミット：最外TRANSACTION END命令の完了時に、CPUは、トランザクション(即ち、最外トランザクション及びあらゆるネスト化レベル)によって行われたストア・アクセスをコミットし、それらが他のCPU及びI/Oサブシステムに見えるようにする。他のCPU及びI/Oサブシステムにより観察されるとき、トランザクションの全てのネスト化レベルによって行われた全てのフェッチ及びストア・アクセスは、コミットが行われたときに単一の同時操作として行われるように見える。

20

## 【 0 0 7 4 】

汎用レジスタ、アクセス・レジスタ、浮動小数点レジスタ及び浮動小数点制御レジスタの内容は、コミット・プロセスによって変更されない。トランザクションのストアがコミットされる際、トランザクション実行中にこれらのレジスタに対して行われるいずれの変更も保持される。

## 【 0 0 7 5 】

競合：1つのCPUによって行われたトランザクション・アクセスは、両方のアクセスが同じキャッシュ・ライン内のいずれかの記憶場所に対するものであり、且つ、アクセスの1つ又は複数がストアである場合、(a)別のCPUによって行われたトランザクション・アクセス若しくは非トランザクション・アクセス、又は(b)I/Oサブシステムによって行われた非トランザクション・アクセスと競合する。

30

## 【 0 0 7 6 】

競合が概念的シーケンスにおいて検出できない場合でも、CPUの命令の投機的実行により、競合を検出することができる。

## 【 0 0 7 7 】

制約付きトランザクション：制約付きトランザクションは、制約付きトランザクション実行モードで実行されるトランザクションであり、以下の制限を受ける。

- \* 一般命令のサブセットが利用可能である。
- \* 制限された数の命令を実行することができる。
- \* 制限された数のストレージ・オペランド位置にアクセスすることができる。
- \* トランザクションは、単一のネスト化レベルに制限される。

40

## 【 0 0 7 8 】

割り込みの反復又は他のCPU又はI/Oサブシステムとの競合がない場合、制約付きトランザクションは最終的に完了し、従って、アボート・ハンドラ・ルーチンは必要ない。制約付きトランザクションを以下に詳しく説明する。

## 【 0 0 7 9 】

CPUが既に制約なしトランザクション実行モードにある間にTRANSACTION BEGIN constrained (TBEGINC) 命令が実行されると、実行は、ネスト化された制約なしトランザクションとして継続する。

50



## 【 0 0 8 0 】

制約付きトランザクション実行モード：トランザクション・ネスト化深さが0であり、トランザクションがT B E G I N C命令により開始されたとき、C P Uは、制約付きトランザクション実行モードに入る。C P Uが制約付きトランザクション実行モードにある間、トランザクション・ネスト化深さは1である。

## 【 0 0 8 1 】

ネスト化されたトランザクション：C P Uが制約なしトランザクション実行モードにある間にT R A N S A C T I O N B E G I N命令が出されると、トランザクションがネスト化される。

## 【 0 0 8 2 】

トランザクション実行ファシリティは、平坦化されたネスト化と呼ばれるモデルを使用する。平坦化されたネスト化モードにおいて、内部トランザクションにより行われたストアは、最外トランザクションがそのストアをコミットするまで、他のC P U及びI / Oサブシステムにより観察することができない。同様に、トランザクションがアポートした場合、全てのネスト化されたトランザクションがアポートし、全てのネスト化されたトランザクションの全てのトランザクション・ストアが廃棄される。

## 【 0 0 8 3 】

ネスト化されたトランザクションの一例を図8に示す。図示されるように、第1のT B E G I N 6 0 0が最外トランザクション6 0 1を開始し、T B E G I N 6 0 2は第1のネスト化されたトランザクションを開始し、T B E G I N 6 0 4は、第2のネスト化されたトランザクションを開始する。この例において、T B E G I N 6 0 4及びT E N D 6 0 6は、最内(innermost)トランザクション6 0 8を定める。T E N D 6 1 0が実行されると、トランザクション・ストアが、最外トランザクション及び全ての内部トランザクションに対してコミットされる6 1 2。

## 【 0 0 8 4 】

制約なしトランザクション：制約なしトランザクションとは、制約なしトランザクション実行モードで実行されるトランザクションである。制約なしトランザクションは、制約付きトランザクションのような方法では制限されないが、依然として様々な理由によりアポートされ得る。

## 【 0 0 8 5 】

制約なしトランザクション実行モード：トランザクションがT B E G I N命令により開始されると、C P Uは制約なしトランザクション実行モードに入る。C P Uが制約なしトランザクション実行モードにある間、トランザクション・ネスト化深さは、1から最大トランザクション・ネスト化深さまで変化し得る。

## 【 0 0 8 6 】

非トランザクション・アクセス：非トランザクション・アクセスは、C P Uがトランザクション実行モードにないとき、C P Uにより行われるストレージ・オペランド・アクセスである(即ち、トランザクション以外の典型的なストレージ・アクセス)。さらに、I / Oサブシステムにより行われるアクセスは、非トランザクション・アクセスである。さらに、N O N T R A N S A C T I O N A L S T O R E命令を用いて、C P Uが制約なしトランザクション実行モードにある間に非トランザクション・ストア・アクセスをもたらすことができる。

## 【 0 0 8 7 】

図9を参照して、N O N T R A N S A C T I O N A L S T O R E命令の形式の一実施形態を説明する。一例として、N O N T R A N S A C T I O N A L S T O R E命令7 0 0は、非トランザクション・ストア操作を指示するオペコードを指定する複数のオペコード・フィールド7 0 2 a、7 0 2 bと、その内容が第1のオペランドと呼ばれるレジスタを指定するレジスタ・フィールド(R<sub>1</sub>)7 0 4と、インデックス・フィールド(X<sub>2</sub>)7 0 6と、ベース・フィールド(B<sub>2</sub>)7 0 8と、第1の変位フィールド(DL<sub>2</sub>)7 1 0と、第2の変位フィールド(DH<sub>2</sub>)7 1 2とを含む。X<sub>2</sub>フィールド及びB<sub>2</sub>フィー

10

20

30

40

50

ルドにより指定される汎用レジスタの内容を、 $DH_2$  フィールド及び  $DL_2$  フィールドの内容の連結の内容に付加して、第2のオペランド・アドレスを形成する。 $X_2$  フィールド若しくは  $B_2$  フィールドのいずれか又は両方が0のとき、対応するレジスタは、加算に関与しない。

**【0088】**

64ビットの第1のオペランドは、第2のオペランド位置において、変更されずに非ランザクシヨンの配置される。

$DH_2$  フィールド及び  $DL_2$  フィールドにより形成された変位は、20ビットの符号付き2進整数として扱われる。

第2のオペランドは、ダブルワード境界上にアラインされ、他の場合には、指定例外が認識され、操作は抑止される。

10

**【0089】**

外部/最外ランザクシヨン：外部ランザクシヨンとは、より低い番号のランザクシヨン・ネスト化深さを有するランザクシヨンである。ランザクシヨン・ネスト化深さが1のランザクシヨンが、最外ランザクシヨンである。

**【0090】**

最外 `TRANSACTION BEGIN` 命令は、ランザクシヨン・ネスト化深さが最初に0であるときに実行される。最外 `TRANSACTION END` 命令は、ランザクシヨン・ネスト化深さを1から0に移行させるものである。本実施形態において、制約付きランザクシヨンは、最外ランザクシヨンである。

20

**【0091】**

プログラム割り込みフィルタ処理：特定のプログラム例外条件が原因でランザクシヨンがアポートされると、プログラムは、随意的に割り込みが発生するのを防止することができる。この技術は、プログラム割り込みフィルタ処理と呼ばれる。プログラム割り込みフィルタ処理は、割り込みのランザクシヨン・クラス、`TRANSACTION BEGIN` 命令からの有効なプログラム割り込みフィルタ処理制御、及び制御レジスタ0におけるランザクシヨン実行プログラム割り込みフィルタ処理オーバーライドを受ける。

**【0092】**

ランザクシヨン：ランザクシヨンは、CPUがランザクシヨン実行モードにある間に行われたストレージ・オペランド・アドレス、及び変更された選択された汎用レジスタを含む。制約なしランザクシヨンの場合、ストレージ・オペランド・アクセスは、ランザクシヨン・アクセス及び非ランザクシヨン・アクセスの両方を含むことができる。制約付きランザクシヨンの場合、ストレージ・オペランド・アクセスは、ランザクシヨン・アクセスに制限される。他のCPU及びI/Oサブシステムにより観察されるとき、ランザクシヨン実行モードにある間にCPUにより行われた全ストレージ・オペランド・アクセスは、単一の同時操作として行われるように見える。ランザクシヨンがアポートされると、ランザクシヨン・ストア・アクセスは廃棄され、最外 `TRANSACTION BEGIN` 命令の汎用レジスタ保存マスクにより指定されるいずれのレジスタも、ランザクシヨン実行より前のレジスタの内容に復元される。

30

**【0093】**

ランザクシヨン・アクセス：ランザクシヨン・アクセスとは、`NONTRANSACTIONAL STORE` 命令により行われたアクセスを除いた、CPUがランザクシヨン実行モードにある間に行われたストレージ・オペランド・アクセスである。

40

**【0094】**

ランザクシヨン実行モード：ランザクシヨン実行モードという用語は、制約なしランザクシヨン実行モード及び制約付きランザクシヨン実行モードの両方の共通操作を説明する。従って、操作を説明する際、制約なし及び制約付きという用語は、ランザクシヨン実行モードを限定するために用いられる。

**【0095】**

ランザクシヨン・ネスト化深さが0のとき、CPUはランザクシヨン実行モードに

50

ない（非トランザクション実行モードとも呼ばれる）。

【0096】

CPUにより観察されるとき、トランザクション実行モードで行われたフェッチ及びストアは、トランザクション実行モードにない間に行われたものと変わらない。

【0097】

z/Architectureの一実施形態において、トランザクション実行ファシリティは、制御レジスタ0のビット8 - 9、制御レジスタ2のビット61 - 63、トランザクション・ネスト化深さ、トランザクション診断ブロック・アドレス及びトランザクション・アポート・プログラム状況ワード（PSW）の制御下にある。

【0098】

最初のCPUリセットに続いて、制御レジスタ0のビット位置8 - 9、制御レジスタ2のビット位置62 - 63及びトランザクション・ネスト化深さが0に設定される。トランザクション実行制御、即ち制御レジスタ0のビット8が0のとき、CPUをトランザクション実行モードに入れることはできない。

【0099】

種々の制御に関するさらなる詳細を以下に説明する。

示されるように、トランザクション実行ファシリティは、制御レジスタ0の2個のビット及び制御レジスタ2の3個のビットによって制御される。例えば：

【0100】

制御レジスタ0のビット：一実施形態において、ビット割り当ては、以下の通りである。

【0101】

トランザクション実行制御（TXC）：制御レジスタ0のビット8は、トランザクション実行制御である。このビットは、制御プログラム（例えば、オペレーティング・システム）が、トランザクション実行ファシリティがプログラムによって使用可能かどうかを示すことができる機構を提供する。ビット8は、トランザクション実行モードに成功裏に入るために、1であるべきである。

制御レジスタ0のビット8が0のとき、EXTRACT TRANSACTION NESTING DEPTH、TRANSACTION BEGIN及びTRANSACTION END命令を実行しようとする試みは、特別な操作実行をもたらす。

図10を参照して、EXTRACT TRANSACTION NESTING DEPTH命令の形式の一実施形態を説明する。一例として、EXTRACT TRANSACTION NESTING DEPTH 800は、トランザクション・ネスト化深さ抽出操作を示すオペコードを指定するオペコード・フィールド802と、汎用レジスタを指定するレジスタ・フィールドR<sub>1</sub> 804とを含む。

現在のトランザクション・ネスト化深さは、汎用レジスタR<sub>1</sub>のビット48 - 63に配置される。レジスタのビット0 - 31は変更されないままであり、レジスタのビット32 - 47は0に設定される。

更に別の実施形態において、最大トランザクション・ネスト化深さもまた、ビット16 - 31内といった、汎用レジスタR<sub>1</sub>内に配置される。

【0102】

トランザクション実行プログラム割り込みフィルタ処理オーバーライド（PIFO）：制御レジスタ0のビット9は、トランザクション実行プログラム割り込みフィルタ処理オーバーライドである。このビットは、制御プログラムが、TRANSACTION BEGIN命令により指定又は暗黙指定された有効なプログラム割り込みフィルタ処理制御に関係なく、CPUがトランザクション実行モードにある間に発生するあらゆるプログラム例外条件が割り込みをもたらすのを保証することができる機構を提供する。

【0103】

制御レジスタ2のビット：一実施形態において、割り当ては以下の通りである。：

【0104】

10

20

30

40

50

トランザクション診断範囲 ( T D S ) : 次のように、制御レジスタ 2 のビット 6 1 は、レジスタのビット 6 2 - 6 3 におけるトランザクション診断制御 ( T D C ) の適用性を制御する。 :

T D S

【表 1】

値	意味	
0	CPUが問題状態にあるか又はスーパーバイザ状態にあるかに関わらず、TDCが適用される。	
1	CPUが問題状態にあるときにのみTDCが適用される。CPUがスーパーバイザ状態にあるとき、処理はあたかもTDCが0を含んでいるかのようである。	10

【 0 1 0 5 】

トランザクション診断制御 ( T D C ) : 制御レジスタのビット 6 2 - 6 3 は、2 ビットの符号なし整数であり、これを用いて、診断目的のために、トランザクションをランダムにアポートすることができる。一実施形態において、TDCの符号化は、以下の通りである。 :

T D C

【表 2】

値	意味	
0	正常動作 ; TDCの結果として、トランザクションはアポートされない。	
1	ランダム命令において、しかし最外 TRANSACTION END命令の実行前に、あらゆるトランザクションをアポートする。	
2	ランダム命令において、ランダム・トランザクションをアポートする。	
3	予約される	30

【 0 1 0 6 】

0 でない T D C に起因してトランザクションがアポートしたとき、以下のいずれかを行うことができる。

\* アポート・コードは、コード 7 - 1 1、1 3 - 1 6 又は 2 5 5 のいずれかに設定され、コードの値は CPU によりランダムに選択され、条件コードはアポート・コードに対応して設定される。アポート・コードを以下にさらに説明する。

\* 制約なしトランザクションの場合、条件コードは 1 に設定される。この場合、アポート・コードは適用可能でない。

【 0 1 0 7 】

T D C 値 1 が実装されるかどうかはモデル依存である。実装されない場合、1 の値は、あたかも 2 が指定されたかのように機能する。

制約付きトランザクションの場合、1 の T D C 値は、あたかも 2 の T D C 値が指定されたかのように扱われる。

3 の T D C 値が指定された場合、結果は予測不能である。

【 0 1 0 8 】

トランザクション診断ブロック・アドレス ( T D B A )

命令の B<sub>1</sub> フィールドが 0 でないとき、最外 TRANSACTION BEGIN ( T B E G I N ) 命令の第 1 のオペランド・アドレスから、有効なトランザクション診断ブロック・アドレス ( T D B A ) が設定される。CPU が一次空間又はアクセス・レジスタ・

モードにあるとき、TDBAは、一次アドレス空間内の記憶位置を指定する。CPUが二次空間又はホーム空間モードにあるとき、TDBAは、それぞれ、二次アドレス空間又はホーム・アドレス空間内の記憶位置を指定する。DAT（動的アドレス変換）がオフのとき、TDBAは、実ストレージ内の記憶位置を指定する。

【0109】

トランザクションが後でアボートされる場合、TBEGIN指定TDBと呼ばれるトランザクション診断ブロックを見つけるために、CPUによりTDBAが用いられる。TDBAの右端の3つのビットは0であり、TBEGIN指定TDBがダブルワード境界上にあることを意味する。

【0110】

最外TRANSACTION BEGIN (TBEGIN) 命令のB<sub>1</sub>フィールドが0のとき、トランザクション診断ブロック・アドレスは無効であり、トランザクションが後でアボートされる場合、TBEGIN指定TDBは格納されない。

【0111】

トランザクション・アボートPSW (TAPSW)

TRANSACTION BEGIN (TBEGIN) 命令の実行中、ネスト化深さが最初に0であるとき、トランザクション・アボートPSWは現PSWの内容に設定され、トランザクション・アボートPSWの命令アドレスは、次の順次命令を指定する（即ち、最外TBEGINに続く命令）。TRANSACTION BEGIN constrained (TBEGINC) 命令の実行中、ネスト化深さが最初に0であるとき、トランザクション・アボートPSWは、トランザクション・アボートPSWの命令アドレスが（TBEGINCに続く次の順次命令ではなく）TBEGINCを指定することを除いて、現PSWの内容に設定される。

【0112】

トランザクションがアボートされると、トランザクション・アボートPSWにおける条件コードが、アボート条件の重大度を示すコードに置き換えられる。その後、割り込みをもたらさない原因に起因してトランザクションがアボートされた場合、PSWは、トランザクション・アボートPSWからロードされ、割り込みをもたらす理由に起因してトランザクションがアボートされた場合、トランザクション・アボートPSWは、割り込み旧PSWとして格納される。

トランザクション・アボートPSWは、いずれの内部TRANSACTION BEGIN命令の実行中にも変更されない。

【0113】

トランザクション・ネスト化深さ (TND)

トランザクション・ネスト化深さは、例えば、TRANSACTION BEGIN命令が条件コード0で完了するたびにインクリメントし、且つ、TRANSACTION END命令が完了するたびにデクリメントする、16ビットの符号なし値である。トランザクション・ネスト化深さは、トランザクションがアボートされたとき、又はCPUのリセットにより、0に設定される。

【0114】

一実施形態において、15の最大TNDが実装される。

1つの実装において、CPUが制約付きトランザクション実行モードにあるとき、トランザクション・ネスト化深さは1である。さらに、最大TNDは4ビットの値として表すことができるが、TNDは、トランザクション診断ブロックにおける検査を容易にするために、16ビットの値となるように定められる。

【0115】

トランザクション診断ブロック (TDB)

トランザクションがアボートされた場合、種々の状態情報を、以下のようなトランザクション診断ブロック (TDB) 内に保存することができる。

1. TBEGIN指定TDB: 制約なしトランザクションについて、最外TBEGIN

10

20

30

40

50

N命令のB<sub>1</sub>フィールドが0でないとき、命令の第1のオペランド・アドレスは、TBEGIN指定TDBを指定する。これは、アプリケーションのアポート・ハンドラにより検査することができるアプリケーション・プログラムにより指定される記憶位置である。

2. プログラム割り込み(PI)TDB: 制約なシランザクションが、非フィルタ処理(unfiltered)プログラム例外条件に起因してアポートされた場合、又は制約付きシランザクションがいずれかのプログラム例外条件に起因してアポートされた場合(即ち、プログラム割り込みをもたらすいずれかの条件が認識される)、PI-TDBは、プリフィックス領域内の記憶位置に格納される。これは、オペレーティング・システムが提供できるあらゆる診断報告において、オペレーティング・システムが検査又はログアウトするのに利用可能である。

3. インターセプトTDB: シランザクションが、インターセプトをもたらす(即ち、条件により解釈実行が終了し、制御がホスト・プログラムに戻る)いずれかのプログラム例外条件に起因してアポートされた場合、TDBは、ゲスト・オペレーティング・システムについての状態記述ブロックにおいて指定された記憶位置に格納される。

#### 【0116】

一実施形態において、TBEGIN指定TDBは、TDBアドレスが有効であるとき(即ち、最外TBEGIN命令のB<sub>1</sub>フィールドが0でないとき)にのみ格納される。

非フィルタ処理プログラム例外条件に起因するアポートの場合、PI-TDB又はインターセプトTDBのいずれか1つだけが格納される。従って、1つのアポートに対して、0、1又は2つのTDBが格納され得る。

#### 【0117】

TDBの各々の一例に関するさらなる詳細を以下に説明する。:

TBEGIN指定TDB: 有効なシランザクション診断ブロック・アドレスにより指定された256バイトの記憶位置である。シランザクション診断ブロック・アドレスが有効であるとき、TBEGIN指定TDBは、シランザクション・アポート上に格納される。TBEGIN指定TDBは、最外TRANSACTION BEG IN命令の実行時に有効である全てのストレージ保護機構の影響下にある。TBEGIN指定TDBの任意の部分についてのPER(プログラム・イベント記録)ストレージ変更イベントは、シランザクション・アポート処理中ではなく、最外TBEGINの実行中に検出される。

#### 【0118】

PERの1つの目的は、プログラムのデバッグを支援することである。これにより、プログラムに、例示するような以下のタイプのイベントをアラートすることが可能になる。

\* 分岐命令の実行の成功: 分岐ターゲット位置が指定したストレージ領域(記憶領域)内にあるときにのみイベントが生じるようにするための選択肢が提供される。

\* 指定したストレージ領域からの命令のフェッチ。

\* 指定したストレージ領域の内容の変更。ストレージ領域が指定したアドレス空間内にあるときにのみイベントを生じるようにするための選択肢が提供される。

\* STORE USING REAL ADDRESS命令の実行。

\* TRANSACTION END命令の実行。

#### 【0119】

プログラムは、上記のイベント・タイプの1つ又は複数が認識されることを選択的に指定することができるが、STORE USING REAL ADDRESSに関するイベントは、ストレージ変更イベントと一緒にしか指定することができない。PERイベントに関する情報は、プログラム割り込みを用いてプログラムに与えられ、割り込みの原因は、割り込みコードで識別される。

#### 【0120】

シランザクション診断ブロック・アドレスが有効でないとき、TBEGIN指定TDBは格納されない。

#### 【0121】

プログラム割り込みTDB: 実記憶位置6, 144-6, 399(1800-18FF

10

20

30

40

50

hex)である。プログラム割り込みに起因してトランザクションがアボートされるとき、プログラム割り込みTDBが格納される。他の原因のためにトランザクションがアボートされるとき、プログラム割り込みTDBの内容は予測不能である。

【0122】

プログラム割り込みTDBは、いずれの保護機構の影響下にもない。プログラム割り込み中にプログラム割り込みTDBが格納されたとき、プログラム割り込みTDBについて、PERストレージ変更イベントは検出されない。

【0123】

インターセプトTDB：状態記述の記憶位置488 - 495により指定される256バイトのホストの実記憶位置である。インターセプトTDBは、アボートされたトランザクションが、ゲスト・プログラム割り込みインターセプト（つまり、インターセプト・コード8）をもたらすときに格納される。他の原因のためにトランザクションがアボートされるとき、インターセプトTDBの内容は予測不能である。インターセプトTDBは、いずれの保護機構の影響下にもない。

【0124】

図11に示すように、一実施形態において、トランザクション診断ブロック900のフィールドは、次のとおりである。

【0125】

形式902：バイト0は、次のように、妥当性及び形式の表示を含む。：

【表3】

値	意味
0	TDBの残りのフィールドは予測不能である。
1	形式-1 TDB、その残りのフィールドは以下に説明される。
2-255	予約される

【0126】

形式フィールドが0であるTDBは、ヌルTDBと呼ばれる。

【0127】

フラグ904：バイト1は、次のような種々の表示を含む。：

競合トークン妥当性（CTV）：トランザクションが、フェッチ又はストア競合のためにアボートされるとき（つまり、それぞれ、アボート・コード9又は10）、バイト1のビット0は、競合トークン妥当性表示である。CTV表示が1のとき、TDBのバイト16 - 23における競合トークン910は、競合が検出された論理アドレスを含む。CTV表示が0であるとき、TDBのバイト16 - 23は予測不能である。

トランザクションが、フェッチ又はストア競合以外のいずれかの理由のためにアボートされたとき、バイト1のビット0は、0として格納される。

制約付きトランザクション表示（CTI）：CPUが制約付きトランザクション実行モードにあるとき、バイト1のビット1は1に設定される。CPUが制約なしトランザクション実行モードにあるとき、バイト1のビット1は0に設定される。

予約：バイト1のビット2 - 7は予約され、0として格納される。

【0128】

トランザクション・ネスト化深さ（TND）906：トランザクションがアボートされたとき、バイト6 - 7はトランザクション・ネスト化深さを含む。

【0129】

トランザクション・アボート・コード（TAC）908：バイト8 - 15は、64ビットの符号なしトランザクション・アボート・コードを含む。各々のコード・ポイントは、トランザクションがアボートされる理由を示す。

【0130】

トランザクションがプログラム割り込み以外の条件のためにアボートされたとき、トラ

10

20

30

40

50

ンザクション・アボート・コードがプログラム割り込み T D B 内に格納されるかどうかは、モデル依存である。

【 0 1 3 1 】

競合トークン 9 1 0 : フェッチ又はストア競合 (つまり、それぞれ、アボート・コード 9 及び 1 0 ) に起因してアボートされたトランザクションの場合、バイト 1 6 - 2 3 は、競合が検出されたストレージ位置の論理アドレスを含む。競合トークンは、C T V ビット、即ちバイト 1 のビット 0 が 1 であるときに意味をもつ。

C T V ビットが 0 のとき、バイト 1 6 - 2 3 は予測不能である。

【 0 1 3 2 】

C P U による投機的実行のため、競合トークンは、トランザクションの概念的実行シーケンスより必ずしもアクセスされないストレージ位置を指定することができる。

10

【 0 1 3 3 】

アボートされたトランザクション命令アドレス ( A T I A ) 9 1 2 : バイト 2 4 - 3 1 は、アボートが検出されたときに実行していた命令を識別する命令アドレスを含む。トランザクションがアボート・コード 2、5、6、11、13 若しくは 2 5 6、又はこれより大きいアボート・コードに起因してアボートされたとき、或いはトランザクションがアボート・コード 4 又は 1 3 に起因してアボートされ、且つ、プログラム例外条件が無効化しているとき、A T I A は、実行されていた命令を直接指し示す。アボート・コード 4 又は 1 2 に起因してトランザクションがアボートされ、且つ、プログラム例外条件が無効化していないとき、A T I A は、実行されていた命令を超えて指し示す。

20

【 0 1 3 4 】

トランザクションがアボート・コード 7 - 1 0、14 - 1 6 又は 2 5 5 に起因してアボートされたとき、A T I A は必ずしもアボートを引き起こすまさにその命令を示さないが、トランザクション内のそれより前又は後の命令を指し示し得る。

【 0 1 3 5 】

トランザクションが、e x e c u t e 型命令のターゲットである命令に起因してアボートされた場合、A T I A は、e x e c u t e 型命令を識別し、上述のようなアボート・コードに応じて、命令を指し示すか、又は命令を超えて指し示す。A T I A は、e x e c u t e 型命令のターゲットを示さない。

【 0 1 3 6 】

トランザクションがアボートされると、A T I A は、アドレッシング・モードの影響下にある。2 4 ビット・アドレッシング・モードにおいて、フィールドのビット 0 - 4 0 は 0 を含む。3 1 ビット・アドレッシング・モードにおいて、フィールドのビット 0 - 3 2 は 0 を含む。

30

【 0 1 3 7 】

プログラム割り込み以外の条件に起因してトランザクションがアボートされたとき、アボートされたトランザクション命令アドレスがプログラム割り込み T D B に格納されるかどうかは、モデル依存である。

【 0 1 3 8 】

アボート・コード 4 又は 1 2 に起因してトランザクションがアボートされ、且つ、プログラム例外条件が無効化していないとき、A T I A は、アボートを引き起こす命令を指し示さない。A T I A から、割り込み長さコード ( I L C ) により示されるハーフワードの数を減算することにより、抑止又は終了している条件において、又は完了している非 P E R イベントに対して、アボートを引き起こす命令を識別することができる。P E R イベントに起因してトランザクションがアボートされ、且つ、他のプログラム例外条件が存在しないとき、A T I A は予測不能である。

40

【 0 1 3 9 】

トランザクション診断ブロック・アドレスが有効であるとき、T B E G I N 指定 T D B のバイト 3 6 - 3 9 におけるプログラム例外識別 ( P I I D ) において、I L C を検査することができる。フィルタ処理が適用されないときは、実記憶装置において P I I D の位

50



置 1 4 0 - 1 4 3 において I L C を検査することができる。

【 0 1 4 0 】

例外アクセス識別 ( E A I D ) 9 1 4 : 特定のフィルタ処理済みプログラム例外条件に起因してアボートされたトランザクションについては、 T B E G I N 指定 T D B のバイト 3 2 は、例外アクセス識別を含む。上記に引用により組み入れられた非特許文献 1 に述べられるように、 z / A r c h i t e c t u r e の一例においては、 E A I D の形式及びこれが格納される事例は、例外条件が割り込みをもたらすときの实記憶位置 1 6 0 に記述されるものと同じである。

【 0 1 4 1 】

プログラム割り込みをもたらすあらゆる例外条件を含む、他の理由のためにアボートされたトランザクションの場合、バイト 3 2 は予測不能である。プログラム割り込み T D B において、バイト 3 2 は予測不能である。

10

【 0 1 4 2 】

このフィールドは、トランザクション診断ブロック・アドレスにより指定される T D B 内にのみ格納され、他の場合には、フィールドは予約される。 E A I D は、制御されたアクセス・リスト又は D A T 保護、 A S C E タイプ、ページ変換、領域第 1 変換、領域第 2 変換、領域第 3 変換及びセグメント変換プログラムの例外条件に対してのみ格納される。

【 0 1 4 3 】

データ例外コード ( D X C ) 9 1 6 : フィルタ処理済みデータ例外プログラム例外条件に起因してアボートされたトランザクションについては、 T B E G I N 指定 T D B のバイト 3 3 は、データ例外コードを含む。上記に引用により組み入れられる非特許文献 1 に述べられるように、 z / A r c h i t e c t u r e の一例において、 D X C の形式及びこれが格納される事例は、例外条件をもたらすときの实記憶位置 1 4 7 に記述されるものと同じである。一例において、記憶位置 1 4 7 は D X C を含む。

20

【 0 1 4 4 】

プログラム割り込みをもたらすあらゆる例外条件を含む他の理由のためにアボートされたトランザクションに対して、バイト 3 3 は予測不能である。プログラム割り込み T D B において、バイト 3 3 は予測不能である。

【 0 1 4 5 】

このフィールドは、トランザクション診断ブロック・アドレスにより指定される T D B 内にのみ格納され、他の場合には、フィールドは予約される。 D X C は、データ・プログラム例外条件に対してのみ格納される。

30

【 0 1 4 6 】

プログラム割り込み識別 ( P I I D ) 9 1 8 : フィルタ処理済みプログラム例外条件に起因してアボートされたトランザクションに対して、 T B E G I N 指定 T D B のバイト 3 6 - 3 9 は、プログラム割り込み識別を含む。 z / A r c h i t e c t u r e の一例において、 P I I D の形式は、 P I I D のビット 1 3 - 1 4 における命令長コードが、例外条件が検出された命令に対して個別のものであることを除いて、(上記に引用により組み入れられる非特許文献 1 に述べられるように)条件が割り込みをもたらすときの实記憶位置 1 4 0 - 1 4 3 に記述されるものと同じである。

40

【 0 1 4 7 】

プログラム割り込みをもたらす例外条件を含む他の理由によりアボートされたトランザクションについては、バイト 3 6 - 3 9 は予測不能である。プログラム割り込み T D B において、バイト 3 6 - 3 9 は予測不能である。

【 0 1 4 8 】

このフィールドは、トランザクション診断ブロック・アドレスにより指定された T D B 内にのみ格納され、他の場合には、フィールドは予約される。プログラム割り込み識別は、プログラム例外条件に対してのみ格納される。

【 0 1 4 9 】

変換例外識別 ( T E I D ) 9 2 0 : 以下のフィルタ処理済みプログラム例外条件のいず

50

れかに起因してアボートされたトランザクションに対して、T B E G I N 指定 T D B のバイト 4 0 - 4 7 は、変換例外識別を含む。

- \* 制御されたアクセス・リスト又は D A T 保護
- \* A S C E タイプ
- \* ページ変換
- \* 領域第 1 変換
- \* 領域第 2 変換
- \* 領域第 3 変換
- \* セグメント変換例外

#### 【 0 1 5 0 】

上記に引用により組み入れられる非特許文献 1 に述べられるように、z / A r c h i t e c t u r e の一例において、条件が割り込みをもたらすとき、T E I D の形式は、実記憶位置 1 6 8 - 1 7 5 に記述されるものと同じである。

10

#### 【 0 1 5 1 】

プログラム割り込みをもたらす例外条件を含む、他の理由によりアボートされたトランザクションに対して、バイト 4 0 - 4 7 は予測不能である。プログラム割り込み T D B において、バイト 4 0 - 4 7 は予測不能である。

#### 【 0 1 5 2 】

このフィールドは、トランザクション診断ブロック・アドレスにより指定された T D B 内のみ格納され、他の場合には、フィールドは予約される。

20

#### 【 0 1 5 3 】

ブレーキング・イベント・アドレス 9 2 2 : フィルタ処理済みプログラム例外条件に起因してアボートされたトランザクションに対して、T B E G I N 指定 T D B のバイト 4 8 - 5 5 は、ブレーキング・イベント・アドレスを含む。上記に引用により組み入れられる非特許文献 1 に述べられるように、z / A r c h i t e c t u r e の一例において、ブレーキング・イベント・アドレスの形式は、例外条件が割り込みをもたらすときの実記憶位置 2 7 2 - 2 7 9 に記述されるものと同じである。

#### 【 0 1 5 4 】

プログラム割り込みをもたらす例外条件を含む、他の理由によりアボートされたトランザクションに対して、バイト 4 8 - 5 5 は予測不能である。プログラム割り込み T D B において、バイト 4 8 - 5 5 は予測不能である。

30

#### 【 0 1 5 5 】

このフィールドは、トランザクション診断ブロック・アドレスにより指定された T D B 内のみ格納され、他の場合には、フィールドは予約される。

#### 【 0 1 5 6 】

以下にブレーキング・イベントに関連するさらなる詳細を説明する。

z / A r c h i t e c t u r e の一実施形態において、P E R - 3 ファシリティがインストールされると、この P E R - 3 ファシリティは、プログラムに、C P U の順次実行においてブレークを生じさせる最後の命令のアドレスを与える。ブレーキング・イベント・アドレスの記録を、ワイルド分岐検出のためのデバッグ支援として用いることができる。このファシリティは、例えば、ブレーキング・イベント・アドレス・レジスタと呼ばれる、C P U における 6 4 ビット・レジスタを与える。T R A N S A C T I O N A B O R T 以外の命令が、順次命令の実行においてブレークを生じさせる（つまり、P S W 内の命令アドレスが、命令長によりインクリメントされるのではなく、置き換えられる）たびに、その命令のアドレスが、ブレーキング・イベント・アドレス・レジスタ内に配置される。P E R が表示されるにせよ又はされないにせよ、プログラム割り込みが発生するたびに、ブレーキング・イベント・アドレス・レジスタの現在の内容が実ストレージ位置 2 7 2 - 2 7 9 に配置される。

40

#### 【 0 1 5 7 】

ブレーキング・イベントを引き起こす命令が、e x e c u t e 型命令 ( E X E C U T E

50

又はEXECUTE RELATIVE LONG)のターゲットである場合、そのexecute型命令をフェッチするために用いられた命令アドレスが、ブレーキング・イベント・アドレス・レジスタ内に配置される。

【0158】

z/Architectureの一実施形態において、次の命令：即ち、BRANCH AND LINK (BAL、BALR)；BRANCH AND SAVE (BAS、BASR)；BRANCH AND SAVE AND SET MODE (BASSM)；BRANCH AND SET MODE (BSM)；BRANCH AND STACK (BAKR)；BRANCH ON CONDITION (BC、BCR)；BRANCH ON COUNT (BCT、BCTR、BCTG、BCTGR)；BRANCH ON INDEX HIGH (BXH、BXHG)；BRANCH ON INDEX LOW OR EQUAL (BXLE、BXLEG)；BRANCH RELATIVE ON CONDITION (BRC)；BRANCH RELATIVE ON CONDITION LONG (BRCL)；BRANCH RELATIVE ON COUNT (BRCT、BRCTG)；BRANCH RELATIVE ON INDEX HIGH (BRXH、BRXHG)；BRANCH RELATIVE ON INDEX LOW OR EQUAL (BRXLE、BRXLG)；COMPARE AND BRANCH (CRB、CGRB)；COMPARE AND BRANCH RELATIVE (CRJ、CGRJ)；COMPARE IMMEDIATE AND BRANCH (CIB、CGIB)；COMPARE IMMEDIATE AND BRANCH RELATIVE (CIJ、CGIJ)；COMPARE LOGICAL AND BRANCH (CLRB、CLGRB)；COMPARE LOGICAL AND BRANCH RELATIVE (CLRJ、CLGRJ)；COMPARE LOGICAL IMMEDIATE AND BRANCH (CLIB、CLGIB)；及びCOMPARE LOGICAL IMMEDIATE AND BRANCH RELATIVE (CLIJ、CLGIJ)の1つが分岐をもたらすたびに、ブレーキング・イベントが発生すると考えられる。

10

20

【0159】

次の命令：即ち、BRANCH AND SET AUTHORITY (BSA)；BRANCH IN SUBSPACE GROUP (BSG)；BRANCH RELATIVE AND SAVE (BRAS)；BRANCH RELATIVE AND SAVE LONG (BRASL)；LOAD PSW (LPSW)；LOAD PSW EXTENDED (LPSWE)；PROGRAM CALL (PC)；PROGRAM RETURN (PR)；PROGRAM TRANSFER (PT)；PROGRAM TRANSFER WITH INSTANCE (PTI)；RESUME PROGRAM (RP)；及びTRAP (TRAP2、TRAP4)の1つが完了するたびに、ブレーキング・イベントが発生すると考えられる。

30

【0160】

ブレーキング・イベントは、(暗黙的に、又はTRANSACTION ABORT命令の結果として)トランザクションがアボートされた結果として生じるのではないと考えられる。

40

【0161】

モデル依存診断情報924：バイト112 - 127は、モデル依存診断情報を含む。

【0162】

12(フィルタ処理済みプログラム割り込み)を除く全てのアボート・コードに対して、モデル依存診断情報が、格納されるTDBの各々に保存される。

【0163】

一実施形態において、モデル依存診断情報は次のものを含む。

\*バイト112 - 119は、トランザクション実行分岐表示(TXBI)と呼ばれる64ビットのベクトルを含む。ベクトルの最初の63ビットの各々は、次のように、CP

50

U がトランザクション実行モードにある間の分岐命令の実行の結果を示す。

【表 4】

値	意味
0	命令は分岐なしに完了する
1	命令は分岐して完了する

【0164】

ビット 0 は、最初のこうした分岐命令の結果を表し、ビット 1 は、第 2 のこうした命令の結果を表し、以下同様である。

10

【0165】

CPU がトランザクション実行モードにある間、63 より少ない分岐命令が実行された場合、分岐命令に対応しない最右端のビットが 0 に設定される (ビット 63 を含む)。63 より多い分岐命令が実行されると、TXBI のビット 63 は 1 に設定される。

【0166】

TXBI 内のビットは、以下のものを除いて、上記に列挙した、ブレーキング・イベントを生じさせることができる命令によって設定される。：即ち、

- いずれの限定された命令も、TXBI 内にビットを設定させない

- 例えば `z / Architecture` の命令において、`BRANCH ON CONDITION`、`BRANCH RELATIVE ON CONDITION`、又は `BRANCH RELATIVE ON CONDITION LONG` 命令の  $M_1$  フィールドが 0 であるとき、又は、以下の命令の  $R_2$  フィールドが 0 のとき、命令の実行により、TXBI 内にビットが設定されるかどうかは、モデル依存である。

20

\* `BRANCH AND LINK (BALR)`; `BRANCH AND SAVE (BASR)`; `BRANCH AND SAVE AND SET MODE (BASSM)`; `BRANCH AND SET MODE (BSM)`; `BRANCH ON CONDITION (BCR)`; 及び `BRANCH ON COUNT (BCTR, BCTR)`

【0167】

\* ホスト・アクセス例外により生じたアボート条件について、バイト 127 のビット位置 0 が 1 に設定される。他の全てのアボート条件について、バイト 127 のビット位置 0 が 0 に設定される。

30

【0168】

\* ロード/ストア・ユニット (LSU) により検出されたアボート条件に対して、バイト 127 の右端の 5 ビットは、原因の表示を含む。LSU により検出されなかったアボート条件に対して、バイト 127 が予約される。

【0169】

汎用レジスタ 930 : バイト 128 - 255 は、トランザクションがアボートされたときの汎用レジスタ 0 - 15 の内容を含む。レジスタは、昇順に格納され、バイト 128 - 135 の汎用レジスタ 0 から始まり、次にバイト 136 - 143 の汎用レジスタ 1、以下同様である。

40

【0170】

予約 : 他の全てのフィールドは予約される。特に断りのない限り、予約されるフィールドの内容は予測不能である。

【0171】

他の CPU 及び I/O サブシステムにより観察されるとき、トランザクション・アボート中の TDB のストアは、あらゆる非トランザクション・ストア後に行われる多重アクセス参照である。

【0172】

トランザクションが実行する即値構成の範囲外の原因のために、トランザクションがア

50

ポートされることがある。例えば、ハイパーバイザ（LPAR又はz/VMなどの）により認識された一時的イベントにより、トランザクションがアポートされることがある。

【0173】

トランザクション診断ブロック内に与えられる情報は、診断目的であることを意図しており、且つ、実質的に正しい。しかしながら、即値構成の範囲外のイベントによりアポートが発生することがあるため、アポート・コード又はプログラム割り込み識別のような情報が構成内の条件を正確に反映しないことがあるので、プログラム・アクションを決定するのに、この情報を使用すべきではない。

【0174】

TDB内に保存された診断情報に加えて、いずれかのデータ例外プログラム例外条件に起因してトランザクションがアポートされ、且つ、AFPレジスタ制御、即ち制御レジスタ0のビット45、並びに有効な浮動小数点演算許可（F）制御の両方が1であるとき、フィルタ処理がプログラム例外条件に適用されるかどうかに関係なく、データ例外コード（DXC）が、浮動小数点制御レジスタ（FPCR）のバイト2に配置される。トランザクションがアポートされ、且つ、AFPレジスタ制御又は有効な浮動小数点演算許可制御のいずれか又は両方が0であるとき、DXCは、FPCRに配置されない。

10

【0175】

一実施形態においては、本明細書に示されるように、トランザクション実行ファシリテイがインストールされたとき、次の一般命令が与えられる。

```
*EXTRACT TRANSACTION NESTING DEPTH
*NONTRANSACTIONAL STORE
*TRANSACTION ABORT
*TRANSACTION BEGIN
*TRANSACTION END
```

20

【0176】

CPUがトランザクション実行モードにあるとき、特定の命令を実行しようとする試みが制限され、トランザクションがアポートされる。

【0177】

制約付きトランザクション実行モードで発行されたとき、特定の命令を実行しようとする試みが、トランザクション制約プログラム割り込みをもたらすこともあり、又はあたかもトランザクションが制約されていないかのように実行を進行させ得る。

30

【0178】

z/Architectureの一例において、制限された命令は、例として、次の非特権命令：即ち、COMPARE AND SWAP AND STORE；MODIFY RUNTIME INSTRUMENTATION CONTROLS；PERFORM LOCKED OPERATION；M<sub>1</sub>フィールドのコードが6又は7であるときのPREFETCH DATA（RELATIVE LONG）；M<sub>3</sub>フィールドが0であり、R<sub>1</sub>フィールドのコードが6又は7であるときのSTORE CHARACTERS UNDER MASK HIGH；STORE FACILITY LIST EXTENDED；STORE RUNTIME INSTRUMENTATION CONTROLS；SUPERVISOR CALL；及びTEST RUNTIME INSTRUMENTATION CONTROLSを含む。

40

【0179】

上記のリストにおいて、COMPARE AND SWAP AND STORE及びPERFORM LOCKED OPERATIONは、TXモードにおいて基本命令を用いてより効果的に実施できる複合命令である。PREFETCH DATA及びPREFETCH DATA RELATIVE LONGについての事例は、6及び7のコードがキャッシュ・ラインを解放するときに限定され、トランザクションの完了前に潜在的にデータのコミットを必要とする。SUPERVISOR CALLは、これが割り込みを発生させる（トランザクションをアポートさせる）ときに制限される。

50

## 【0180】

以下に列挙される条件下で、次の命令が制限される。：

\* 命令の R<sub>2</sub> フィールドが 0 でなく、且つ、分岐トレースがイネーブルにされたときの BRANCH AND LINK (BALR)、BRANCH AND SAVE (BASR) 及び BRANCH AND SAVE AND SET MODE

\* 命令の R<sub>2</sub> フィールドが 0 でなく、且つ、モード・トレースがイネーブルにされたときの BRANCH AND SAVE AND SET MODE 及び BRANCH AND SET MODE、モード・トレースがイネーブルにされたときの SET ADDRESSING MODE

\* 監視イベント条件が認識されたときの MONITOR CALL

10

## 【0181】

上記のリストは、トレース・エントリを形成できる命令を含む。これらの命令がランザクシヨンの実行することを許可され、且つ、トレース・エントリを形成し、その後ランザクシヨンがアポートされた場合、制御レジスタ 12 のトレース・テーブル・ポインタが進められるが、トレース・テーブルへのストアは廃棄される。このことは、トレース・テーブル内に一貫性のないギャップを残す。従って、トレース・エントリを形成する場合に、命令は制限される。

## 【0182】

CPU がランザクシヨン実行モードにあるとき、次の命令：即ち、CIPHER MESSAGE ; CIPHER MESSAGE WITH CFB ; CIPHER MESSAGE WITH CHAINING ; CIPHER MESSAGE WITH COUNTER ; CIPHER MESSAGE WITH OFB ; COMPRESSION CALL ; COMPUTE INTERMEDIATE MESSAGE DIGEST ; COMPUTE LAST MESSAGE DIGEST ; COMPUTE MESSAGE AUTHENTICATION CODE ; CONVERT UNICODE - 16 TO UNICODE - 32 ; CONVERT UNICODE - 16 TO UNICODE - 8 ; CONVERT UNICODE - 32 TO UNICODE - 16 ; CONVERT UNICODE - 32 TO UNICODE - 8 ; CONVERT UNICODE - 8 TO UNICODE - 16 ; CONVERT UNICODE - 8 TO UNICODE - 32 ; PERFORM CRYPTOGRAPHIC COMPUTATION ; RUNTIME INSTRUMENTATION OFF ; 及び RUNTIME INSTRUMENTATION ON が制限されるかどうかは、モデル依存である。

20

30

## 【0183】

上記命令の各々は、ハードウェア・コプロセッサにより現在実施されているか、又は過去のマシンで実施されたかのいずれかであり、従って制限されると考えられる。

## 【0184】

有効な AR 変更許可 (A) 制御が 0 のとき、次の命令：即ち、COPY ACCESS ; LOAD ACCESS MULTIPLE ; LOAD ADDRESS EXTENDED ; 及び SET ACCESS が制限される。

40

## 【0185】

上記の命令の各々により、アクセス・レジスタの内容が変更される。TRANSACTION BEGIN 命令の A 制御が 0 である場合、プログラムは、アクセス・レジスタの変更が許可されないことを明示的に表示する。

## 【0186】

有効な浮動小数点演算許可 (F) 制御が 0 であるとき、浮動小数点命令が制限される。

## 【0187】

特定の環境下で、次の命令：即ち、EXTRACT CPU TIME ; EXTRACT PSW ; STORE CLOCK ; STORE CLOCK EXTENDED ; 及び STORE CLOCK FAST を制限することができる。

50

## 【0188】

上記の命令の各々は、解釈実行状態記述におけるインターセプト制御を受ける。ハイパーバイザーが、これら命令についてのインターセプト制御を設定した場合、これら命令の実行は、ハイパーバイザの実装のために延長され、従って、インターセプトが発生する場合に制限されると考えられる。

## 【0189】

制限された命令を実行しようとする試みにより、制約なしトランザクションがアボートされると、トランザクション診断ブロック内のトランザクション・アボート・コードは11（制限された命令）に設定され、条件コードは3に設定されるが、次の場合を除く。即ち、制約なしトランザクションが、他の場合には特権操作例外をもたらす命令を実行しようとする試みに起因してアボートされたとき、アボート・コードが11（制限された命令）に設定されるか、又は4（特権操作プログラム割り込みの認識の結果もたらされ非フィルタ処理プログラム割り込み）に設定されるかは、予測不能である。制約なしトランザクションが、M<sub>1</sub>フィールド内のコードが6又は7のときのPREFETCH DATA（RELATIVE LONG）又はM<sub>3</sub>フィールドが0であり、且つ、R<sub>1</sub>フィールド内のコードが6又は7のときのSTORE CHARACTERS UNDER MASK HIGHを実行しようとする試みに起因してアボートされたとき、アボート・コードが11（制限された命令）に設定されるか又は16（キャッシュ・アザー（その他をキャッシュに入れる））に設定されるかは、予測不能である。制約なしトランザクションが、MONITOR CALLを実行しようとする試みに起因してアボートされ、且つ、監視イベント条件及び指定例外条件の両方が存在するとき、アボート・コードが11若しくは4に、又はプログラム割り込みがフィルタ処理されている場合に12に設定されるかどうかは予測不能である。

10

20

## 【0190】

制約付きトランザクションにおいて、付加的な命令を制限することができる。これらの命令は、制約なしトランザクションにおいて制限されるように現在定義されていないが、将来のプロセッサ上での制約なしトランザクションにおける特定の環境下で制限され得る。

## 【0191】

特定の制限された命令は、将来のプロセッサ上ではトランザクション実行モードにおいて許容され得る。従って、プログラムは、制限された命令を実行しようとする試みに起因してアボートされるトランザクションに依存すべきではない。TRANSACTION ABORT命令は、安定的にトランザクションをアボートさせるために使用すべきである。

30

## 【0192】

制約なしトランザクションにおいて、プログラムは、制限された命令に起因してアボートするトランザクションを収容するための、代替的な非トランザクション・コード経路を提供しなければならない。

## 【0193】

動作において、トランザクション・ネスト化深さが0であるとき、条件コード0をもたらすTRANSACTION BEGIN（TBEGIN）命令の実行により、CPUが制約なしトランザクション実行モードに入る。トランザクション・ネスト化深さが0であるとき、条件コード0をもたらすTRANSACTION BEGIN constrained（TBEGINC）命令の実行により、CPUが制約付きトランザクション実行モードに入る。

40

## 【0194】

特に明示的に断りがある場合を除いて、非トランザクション実行に適用される全ての原則は、トランザクション実行にも適用される。以下はCPUがトランザクション実行モードにある間の処理の付加的な特徴である。

## 【0195】

50

CPUが制約なしトランザクション実行モードにあるとき、条件コード0をもたらすTRANSACTION BEGIN命令の実行により、CPUは制約なしトランザクション実行モードのままである。

【0196】

CPUにより観察される時、トランザクション実行モードで行われたフェッチ及びストアは、トランザクション実行モードでない時に行われたフェッチ及びストアと変わらない。他のCPU及びI/Oサブシステムにより観察される時、CPUがトランザクション実行モードにある間に行われた全てのストレージ・オペランド・アクセスは、単一のブロック同時アクセスであるように見える。つまり、他のCPU及びI/O（例えば、チャンネル）プログラムにより観察される時、ハーフワード、ワード、ダブルワード又はクワッドワード内の全てのバイトへのアクセスは、ブロック・コンカレントに見えるように指定される。ハーフワード、ワード、ダブルワード又はクワッドワードは、このセクションではブロックと呼ばれる。フェッチ・タイプ参照が、ブロック内でコンカレントに見えるように指定されると、ブロック内に含まれるバイトがフェッチされている間、別のCPU又はI/Oプログラムによるそのブロックへのストア・アクセスは許可されない。ストア・タイプ参照が、ブロック内でコンカレントに見えるように指定されると、ブロック内に含まれるバイトがストアされる間、別のCPU又はI/Oプログラムによるブロックへのアクセス、即ちフェッチ又はストアは許可されない。

10

【0197】

命令及びDATのためのストレージ・アクセス、並びにART（アクセス・レジスタ・テーブル）テーブル・フェッチは、非トランザクション原則に従う。

20

【0198】

CPUは、通常、トランザクション・ネスト化深さを0に移行させる、この場合はトランザクションが完了するTRANSACTION END命令により、トランザクション実行モードを終了する。

【0199】

CPUが、TRANSACTION END命令の完了によりトランザクション実行モードを終了すると、トランザクション実行モードにある間に行われた全てのストアがコミットされる。つまり、他のCPU又はI/Oサブシステムにより観察される時、ストアは、単一のブロック・コンカレント操作として行われるように見える。

30

【0200】

制約付きトランザクション及び制約なしトランザクションと関連した処理に関する更なる詳細が、それぞれTBEGINC及びTBEGINによるトランザクションの開始と関連した処理を含めて、以下に与えられる。

【0201】

一例において、例えばTBEGINCによる制約付きトランザクションの開始と関連した処理は、以下の通りである。：

\* 既に制約付きTXモードにある場合、制約付きトランザクション例外が認識される。

\* 現在のTND（トランザクション・ネスト化深さ）> 0の場合、実行は、あたかも制約なしトランザクションであるかのように進行する。

40

）有効なF制御は0に設定される

）有効なPIFCは変更されない

）外部制約なしTXが、制約付きTXを使用することも又は使用しないこともあるサービス機能呼び出すことを可能にする

\* 現在のTND = 0のとき：

）トランザクション診断ブロック・アドレスは無効である

- アポート時に命令指定TDBは格納されない

）トランザクション・アポートPSWはTBEGINCのアドレスに設定される

- 次の順次命令ではない

）モデル依存の記憶位置に保存されたGRSMにより指定された汎用レジスタ対は、

50



プログラムによりアクセス可能でない

) トランザクション・トークンが随意的に形成される ( $D_2$  オペランドから)。トランザクション・トークンは、トランザクションの識別子である。トランザクション・トークンは、ストレージ・オペランド・アドレス又は他の値と等しい場合がある。

\* 有効  $A = T B E G I N C \quad A$  & 任意の外部  $A$

\*  $T N D$  がインクリメントされる

)  $T N D$  が 0 から 1 に移行する場合、 $C P U$  は制約付き  $T X$  モードに入る

) 他の場合には、 $C P U$  は制約なし  $T X$  モードにあるままである

\* 命令は  $C C 0$  により完了する

\* 例外：

)  $B_1$  フィールドが 0 でない場合、指定例外 ( $P I C$  (プログラム割り込みコード)  $0 0 0 6$ )

) トランザクション実行制御 ( $C R 0 . 8$ ) が 0 の場合、特殊操作例外 ( $P I C \quad 0 0 1 3 \quad h e x$ )

) 制約付き  $T X$  モードで発行された場合、トランザクション制約例外 ( $P I C \quad 0 0 1 8 \quad h e x$ )

) 制約付き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、操作例外 ( $P I C \quad 0 0 0 1$ )

) 命令が  $e x e c u t e$  型命令のターゲットである場合、実行例外 ( $P I C \quad 0 0 0 3$ )

) ネスト化深さ超過の場合、アボート・コード 1 3

\* 制約付きトランザクションにおけるアボート条件：

) アボート  $P S W$  は  $T B E G I N C$  命令を指し示す

- これに続く命令ではない

- アボート条件により、 $T X$  全体が再駆動される

\* フェイル経路なし

)  $C P U$  は、再駆動時に成功裡の完了を確実にするための特別な手段をとる

) 持続的な競合、割り込み、又は制約違反がないと仮定すると、トランザクションは最終的に完了することが保証される

\* 制約違反：

)  $P I C \quad 0 0 1 8 \quad h e x$  - は、トランザクション制約の違反を示す

) 又は、トランザクションは、あたかも制約なしであるかのように実行される

#### 【0202】

さらに、一実施形態において、例えば  $T B E G I N$  による制約なしトランザクションの開始と関連した処理は、以下の通りである。：

\*  $T N D = 0$  の場合：

)  $B_1 = 0$  の場合、トランザクション診断ブロック・アドレスは第 1 のオペランド・アドレスから設定される。

) トランザクション・アボート  $P S W$  は、次の順次命令アドレスに設定される。

)  $I_2$  フィールドにより指定される汎用レジスタ対は、モデル依存記憶位置に保存される。

- プログラムにより直接アクセス可能でない

\* 有効な  $P I F C$ 、 $A$ 、&  $F$  制御が計算される

) 有効な  $A = T B E G I N \quad A$  & 任意の外部  $A$

) 有効な  $F = T B E G I N \quad F$  & 任意の外部  $F$

) 有効な  $P I F C =$  最大値 ( $T B E G I N \quad P I F C$ 、任意の外部  $P I F C$ )

\* トランザクション・ネスト化深さ ( $T N D$ ) がインクリメントされる

\*  $T N D$  が 0 から 1 に移行する場合、 $C P U$  はトランザクション実行モードに入る

\* 条件コードが 0 に設定される。

10

20

30

40

50

- ) T B E G I Nに続く命令が制御を受け取ると：
  - T B E G I Nの成功が C C 0により示される
  - アボートされたトランザクションが、0でない C Cにより示される

\* 例外：

- ) ネスト化深さ超過の場合、アボート・コード 1 3
- ) B<sub>1</sub>フィールドが0でなく、且つ、ストア操作に関してストレージ・オペランドにアクセスできない場合、アクセス例外（種々の P I Cの1つ）
- ) T B E G I N命令が e x e c u t e型命令のターゲットである場合、実行例外（ P I C 0 0 0 3）
- ) 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、操作例外（ P I C 0 0 0 1）
- ) 次のいずれかの場合、 P I C 0 0 0 6
  - P I F Cが無効である（3の値）
  - 第2のオペランド・アドレスがダブルワードにアラインされていない
- ) トランザクション実行制御（ C R 0 . 8）が0である場合、 P I C 0 0 1 3 h e x
- ) 制約付き T Xモードで発行された場合、 P I C 0 0 1 8 h e x

#### 【 0 2 0 3】

トランザクションは、種々の原因のために暗黙的にアボートされ得る、又は T R A N S A C T I O N A B O R T命令により明示的にアボートされ得る。トランザクション・アボートの例示的な可能な原因、対応するアボート・コード、トランザクション・アボート P S Wに配置される条件コードを、以下に説明する。

#### 【 0 2 0 4】

外部割り込み：トランザクション・アボート・コードは2に設定され、且つ、トランザクション・アボート P S Wの条件コードは2に設定される。トランザクション・アボート P S Wは、外部割り込み処理の一部として外部旧 P S Wとして格納される。

#### 【 0 2 0 5】

プログラム割り込み（非フィルタ処理）：割り込みをもたらすプログラム例外条件（つまり、非フィルタ処理条件）により、トランザクションがコード4でアボートされる。トランザクション・アボート P S Wの条件コードは、そのプログラム割り込みコード特有に設定される。トランザクション・アボート P S Wは、プログラム割り込み処理の一部としてプログラム旧 P S Wとして格納される。

#### 【 0 2 0 6】

他の場合には操作例外に起因してアボートされるトランザクションをもたらす命令が、代替的な結果をもたらすことがある。制約なしトランザクションにおいては、トランザクションは、代わりにアボート・コード11（制限された命令）でアボートすることができ、制約付きトランザクションの場合、操作例外の代わりに、トランザクション制約プログラム割り込みが認識され得る。

#### 【 0 2 0 7】

いずれかの他の非フィルタ処理プログラム例外条件と共に P E R（プログラム・イベント記録）イベントが認識されると、条件コードは3に設定される。

#### 【 0 2 0 8】

マシン・チェック割り込み：トランザクション・アボート・コードは5に設定され、トランザクション・アボート P S Wの条件コードは2に設定される。トランザクション・アボート P S Wは、マシン・チェック割り込み処理の一部として、マシン・チェック旧 P S Wとして格納される。

#### 【 0 2 0 9】

I / O割り込み：トランザクション・アボート・コードは6に設定され、トランザクション・アボート P S Wの条件コードは2に設定される。トランザクション・アボート P S Wは、I / O割り込み処理の一部としてI / O旧 P S Wとして格納される。

## 【0210】

フェッチ・オーバーフロー：トランザクションが、CPUがサポートするより多い記憶位置からフェッチしようとしたとき、フェッチ・オーバーフロー条件が検出される。トランザクション・アポート・コードは7に設定され、条件コードは2又は3のいずれかに設定される。

## 【0211】

ストア・オーバーフロー：トランザクションが、CPUがサポートするより多い記憶位置にストアしようとしたとき、ストア・オーバーフロー条件が検出される。トランザクション・アポート・コードは8に設定され、条件コードは2又は3に設定される。

## 【0212】

フェッチ又はストア・オーバーフローに応じて条件コードを2又は3とするのを可能にすることにより、CPUが潜在的に再試行可能な状況を示すことが可能になる（例えば、条件コード2は、トランザクションの再実行が生産的であり得ることを示し、条件コード3は、再実行を推奨しない）。

## 【0213】

フェッチ競合：フェッチ競合条件は、別のCPU又はI/Oサブシステムが、このCPUによってトランザクション的にフェッチされた記憶位置に格納しようとするときに検出される。トランザクション・アポート・コードは9に設定され、条件コードは2に設定される。

## 【0214】

ストア競合：ストア競合条件は、別のCPU又はI/Oサブシステムが、このCPUによりトランザクション実行中に格納された記憶位置にアクセスしようとするときに検出される。トランザクション・アポート・コードは10に設定され、条件コードは2に設定される。

## 【0215】

制限された命令：CPUがトランザクション実行モードにあるとき、制限された命令の実行を試みるにより、トランザクションがアポートされる。トランザクション・アポート・コードは11に設定され、条件コードは3に設定される。

## 【0216】

CPUが制約付きトランザクション実行モードにあるとき、制限された命令を実行しようとする試みは、トランザクション制約プログラム割り込みをもたらすか、又は制限された命令が原因のアポートをもたらすかは予測不能である。トランザクションは、依然としてアポートされるが、アポート・コードはいずれの原因も表示し得る。

## 【0217】

プログラム例外条件（フィルタ処理済み）：割り込みをもたらさないプログラム例外条件（即ち、フィルタ処理済み条件）は、トランザクション・アポート・コード12でトランザクションをアポートさせる。条件コードは、3に設定される。

## 【0218】

ネスト化深さ超過：ネスト化深さ超過条件は、トランザクション・ネスト化深さが、構成についての最大許容可能値であり、且つ、TRANSACTION BEGIN命令が実行されたときに検出される。トランザクションは、トランザクション・アポート・コード13でアポートされ、条件コードは3に設定される。

## 【0219】

キャッシュ・フェッチ関連条件：トランザクションによりフェッチされたストレージ位置に関連する条件が、CPUのキャッシュ回路により検出される。トランザクションは、トランザクション・アポート・コード14でアポートされ、条件コードは2又は3のいずれかに設定される。

## 【0220】

キャッシュ・ストア関連条件：トランザクションにより格納されたストレージ位置に関連する条件は、CPUのキャッシュ回路により検出される。トランザクションは、トラン

10

20

30

40

50

ザクション・アボート・コード 15 でアボートされ、条件コードは 2 又は 3 のいずれかに設定される。

【 0 2 2 1 】

キャッシュ・アザー条件：キャッシュ・アザー条件は、CPU のキャッシュ回路により検出される。トランザクションは、トランザクション・アボート・コード 16 でアボートされ、条件コードは 2 又は 3 のいずれかに設定される。

【 0 2 2 2 】

トランザクションの実行中、CPU が同じ絶対アドレスにマッピングされた異なる論理アドレスを用いて命令又はストレージ・オペランドにアクセスする場合、トランザクションがアボートされるかどうかは、モデル依存である。トランザクションが、同じ絶対アドレスにマッピングされた異なる論理アドレスを用いるアクセスに起因してアボートされた場合、条件に応じて、アボート・コード 14、15 又は 16 が設定される。

10

【 0 2 2 3 】

その他の条件：その他の条件とは、トランザクションをアボートさせる CPU により認識される任意の他の条件である。トランザクション・アボート・コードは 255 に設定され、且つ、条件コードは 2 又は 3 のいずれかに設定される。

【 0 2 2 4 】

複数の構成が同じマシン（例えば、論理パーティション又は仮想マシン）内で実行されているとき、トランザクションは、外部マシン・チェック又は異なる構成で行われた I/O 割り込みに起因してアボートされ得る。

20

【 0 2 2 5 】

上記に例が与えられたが、対応するアボート・コード及び条件コードでのトランザクション・アボートの他の原因を提供することができる。例えば、原因が再起動割り込みである場合があり、この場合、トランザクション・アボート・コードは 1 に設定され、トランザクション・アボート P S W の条件コードは 2 に設定される。トランザクション・アボート P S W は、再起動処理の一部として再起動 - 旧 P S W として格納される。更に別の例として、原因は、スーパーバイザ呼び出し条件とすることができ、この場合、アボート・コードは 3 に設定され、トランザクション・アボート P S W の条件コードは 3 に設定される。他の又は異なる例も可能である。

【 0 2 2 6 】

30

注記：

1 . その他の条件は、以下のいずれかからもたらされ得る。

\* z / Architecture における、COMPARE AND REPLACE DAT TABLE ENTRY、COMPARE AND SWAP AND PURGE、INVALIDATE DAT TABLE ENTRY、INVALIDATE PAGE TABLE ENTRY、NQ 制御が 0 であり、且つ、SK 制御が 1 である PERFORM FRAME MANAGEMENT FUNCTION、NQ 制御が 0 であり、構成内の別の CPU により実行され、条件コードが 2 に設定される SET STORAGE KEY EXTENDED のような命令。

\* リセット、再起動、又は停止、又は同等の SIGNAL PROCESSOR 命令のようなオペレータ機能が CPU 上で実行される。

40

\* 上記に列挙されないいずれかの他の条件；条件コードは 2 又は 3 に設定される。

2 . フェッチ及びストア競合が検出された記憶位置は、同一キャッシュ・ライン内のどの場所であってもよい。

3 . 特定の条件下で、CPU は、類似のアボート条件を区別できないことがある。例えば、フェッチ又はストア・オーバーフローは、それぞれのフェッチ又はストア競合と区別できないことがある。

4 . CPU による複数の命令経路の投機的実行により、こうした条件が概念的シーケンスで行われない場合でも、競合又はオーバーフロー条件に起因して、トランザクションがアボートされる。制約付きトランザクション・モードにある間、CPU は、投機的実行を

50

一時的に禁止し、こうした競合又はオーバーフローを投機的に検出することなく、トランザクションを完了しようと試みることを可能にする。

【0227】

複数のアポード条件が適用される場合、CPUにより、どのアポード・コードが報告されるかは予測不能である。

【0228】

図12は、トランザクション診断ブロックに格納された例示的なアポード・コード及び対応する条件コード(CC)を要約する。図12の説明は、1つの特定の実施を示す。他の実施及び値の符号化も可能である。一実施形態において、アポード・コード1、3及び17-254は、起こり得る拡張に備えて予約される。こうした拡張は、条件コード2又は3のいずれのかの設定をもたらし得る。さらに、この実施形態においては、アポード・コード0は予約され、意味のあるアポード指示には割り当てられない。モデルによっては、CPUが、特定のアポード原因を区別できないことがある。例えば、全ての状況において、CPUにより、フェッチ/ストア・オーバーフローとフェッチ/ストア競合とを区別することはできない。

10

【0229】

CPUによりトランザクション的にアクセスされる記憶位置への他のCPU又はI/Oサブシステムによるアクセスの競合が、トランザクションのアポードを引き起こすことがある。トランザクションがアポードされると、トランザクションにより実施されたトランザクション・ストアは廃棄される。このことは、トランザクション・コードのデバッグを極めて困難にし得る。従って、一実施形態において、トランザクションがアポードされた場合でも、トランザクション・モードで行われたストアを保持することを可能にするNONTRANSACTIOINAL STORE (NTSTG)命令が提供される。

20

【0230】

図9を参照して上述したように、一例において、NONTRANSACTIOINAL STORE命令700は、非トランザクション・ストア操作を指示するオペコードを指定する複数のオペコード・フィールド702a、702bと、その内容が第1のオペランドと呼ばれるレジスタを指定するレジスタ・フィールド( $R_1$ )704と、インデックス・フィールド( $X_2$ )706と、ベース・フィールド( $B_2$ )708と、第1の変位フィールド( $DL_2$ )710と、第2の変位フィールド( $DH_2$ )712とを含む。 $X_2$ フィールド及び $B_2$ フィールドにより指定される汎用レジスタの内容を、 $DH_2$ フィールド及び $DL_2$ フィールドの内容の連結の内容により定められる値に付加して、第2のオペランド・アドレスを形成する。 $X_2$ フィールド若しくは $B_2$ フィールドのいずれか又は両方が0のとき、対応するレジスタは、加算に関与しない。

30

【0231】

64ビットの第1のオペランドは、第2のオペランド位置に変更されずに非トランザクション的に配置される。一実施形態において、第1のオペランドはユーザ指定のものである。

$DH_2$ フィールド及び $DL_2$ フィールドの内容の連結により形成される変位は、20ビットの符号付き2進整数として扱われる。

40

【0232】

第2のオペランドは、ダブルワード境界上にアラインされ、他の場合には、指定例外(プログラム割り込みコード(PIC)0006)が認識され、操作は抑止される。

【0233】

条件コード：コードは、変更されないままである。

プログラム例外：

\*アクセス例外(ストア、第2のオペランド)

\*操作例外(トランザクション実行ファシリティはインストールされていない)

\*指定例外

\*トランザクション制約例外(限定された命令に起因する)；PIC 0018 he

50

x)

【0234】

さらに、一実施形態において：

NONTRANSACTIONAL STOREは、トランザクションがアボートした場合でも、制約なしトランザクション実行モードで実行されているプログラムが、保持されるストアを実行することができる手段を提供する。このことは、トランザクションがアボートされた場合のトランザクションのデバッグを容易にし得る。

【0235】

CPUがトランザクション実行モードにないとき、NONTRANSACTIONAL STOREの操作は、z/ArchitectureのSTORE (STG)のようなSTORE命令と同じである。ただし、一実施形態においては、NONTRANSACTIONAL STOREの第2のオペランドがダブルワード境界上にあり(トランザクション実行モードにも関わらず)、一方、STGの第2のオペランドがアライメント要件を有さない。従って、この状況において、プロセッサがトランザクション実行モードにない場合に例外は与えられず、代わりに、NONTRANSACTIONAL STOREが、あたかもそれがSTOREのように実行される。

10

【0236】

STORE命令は、プロセッサにより実行されるとき、第1のオペランド(例えば、命令のレジスタ・フィールドにより指定されるレジスタの内容)を第2のオペランド位置(X<sub>2</sub>フィールド及びB<sub>2</sub>フィールドにより指定される汎用レジスタの内容を、DH<sub>2</sub>フィールド及びDL<sub>2</sub>フィールドの内容の連結により定められた値に付加することにより形成される)に入れる。

20

【0237】

プロセッサ・モデルによっては、NONTRANSACTIONAL STOREの性能は、STORE (STG)の性能より遅いことがある。

【0238】

CPUが1つのトランザクション内で同じ記憶位置に対してトランザクション・ストア及び非トランザクション・ストアを行い、次に、トランザクションがアボートした場合、トランザクション・ストア又は非トランザクション・ストアのいずれかによって変更された全ての記憶位置の内容は予測不能である。

30

【0239】

NONTRANSACTIONAL STOREと関連した論理の一実施形態が、図13と関連して説明される。一実施形態において、プロセッサ(例えば、CPU)は、この論理を実施する。

【0240】

図13を参照すると、NONTRANSACTIONAL STORE命令の実行に基づいて、特定の例外に関してチェックを行い(ステップ1100)、例外があった場合には(問い合わせ1102)、これを処理する(ステップ1104)。例えば、第2のオペランドがダブルワード境界上にアラインされていない場合、指定例外が認識され、命令の実行が抑止される。さらに、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、操作例外が認識され、命令の実行が抑止され；トランザクションが制約付き命令である場合、トランザクション制約例外が認識され、実行が抑止され；第2のオペランド位置に関する問題(例えば、保護例外、無効アドレス等)がある場合、アクセス例外が認識され、命令の実行は抑止される。他の場合には、例外がない場合、命令の第1のオペランドが、第2のオペランドにより指定される位置に、変更されずに非トランザクション的に配置される(即ち、ストアされる)(ステップ1106)。

40

【0241】

一実施形態において、NONTRANSACTIONAL STORE命令は、トランザクション実行モードで実行され、非トランザクション的配置は、TENDによる最外トランザクションの完了、又はトランザクション実行モードを終了するアボートの完了まで

50

といった、トランザクション実行モードの終わりまで遅延される。(更に別の実施形態においては、NONTRANSACTIONAL STORE命令は、命令が実行される特定のトランザクションの終わりまで遅延されるだけである)。さらに、複数の非トランザクション・ストアがある場合があり、これらは、他のプロセッサには、同時実行ストアのように見える。

#### 【0242】

更に別の実施形態において、図14を参照して説明されるように、NONTRANSACTIONAL STOREの処理は、プロセッサの実行モード、及び/又はNONTRANSACTIONAL STOREがその一部であるトランザクションのタイプによって決まる。

10

#### 【0243】

図14を参照すると、一実施形態において、プロセッサによるNONTRANSACTIONAL STORE命令の実行に基づいて、プロセッサがトランザクション実行モードにあるか(即ち、トランザクション・ネスト化深さがゼロより大きい)かどうかについての判断が行われる(問い合わせ1150)。プロセッサがトランザクション実行モードにある場合、NONTRANSACTIONAL STORE命令が制約なしトランザクション(TBEGINにより開始されるトランザクション)の一部であるかどうかについての判断が行われる(問い合わせ1152)。本明細書で説明されるように、命令が制約なしトランザクションの一部である場合、その命令は実行される(ステップ1154)。しかしながら、命令が制約付きトランザクション(例えば、TBEGINCにより開始され、制約なしトランザクションとして扱われない)の一部である場合は、プログラム例外がもたらされ、命令の更なる実行は抑止される(ステップ1156)。

20

#### 【0244】

問い合わせ1150に戻ると、プロセッサがトランザクション実行モードにない場合、NONTRANSACTIONAL STORE命令は、STORE命令として扱われ、処理は、STORE命令のように進む(ステップ1158)。

#### 【0245】

上述されたのは、命令と関連したトランザクションのアポート(即ち、命令を直接含むトランザクションのアポート、又は命令を含むトランザクションが内部にネスト化された別のトランザクションのアポート)の後でさえも持続する、トランザクション実行モードにある間に情報を明示的に格納するための能力である。情報はユーザ指定のものである。

30

#### 【0246】

さらに、上記に与えられたのは、マルチプロセッサの性能を著しく向上させる可能性をもたらす、ロックのような典型的な(粗い粒度の)シリアル化を用いずに、メモリにおける複数の不連続オブジェクトを更新する有効な手段である。つまり、ロック及びセマフォのような典型的な技術により与えられるより粗い粒度のストレージ・アクセス順序付けを実施することなく、複数の不連続オブジェクトが更新される。単純な小さいフットプリント更新のために、制約付きトランザクションが与えられる。

#### 【0247】

トランザクション実行は、これらに限定されるものではないが、部分的インライン化、投機的処理、及びロック省略(lock elision)を含む、様々なシナリオにおいて用いることができる。部分的インライン化において、実行される経路内に含まれる部分的領域は、TBEGIN/TENDにおいてワードラップが行われる。TABORTを内部に含ませ、側部出口(side-exit)上の状態をロールバックすることができる。Javaにおけるような投機については、トランザクションを用いて、逆参照ポインタ上でのヌルチェックを遅延させて、エッジをループさせることができる。ポインタがヌルである場合、トランザクションは、TBEGIN/TEND内に含まれるTABORTを用いて安全にアポートすることができる。

40

#### 【0248】

本明細書で用いられる場合、ストレージ、中央ストレージ、主ストレージ、メモリ及び

50

主メモリは、慣例によって暗黙的に又は明示的に、特に断りのない限り、交換可能に用いられる。さらに、一実施形態においては、トランザクションの効果的な遅延は、選択されたトランザクションの完了まで、主メモリへのトランザクション・ストアのコミットを遅延させることを含み、別の実施形態においては、トランザクションの効果的な遅延は、メモリに対するトランザクション更新を可能にし、且つ、古い値を保持し、アポート時にメモリを古い値に復元することを含む。

【0249】

当業者であれば理解するように、1つ又は複数の態様は、システム、方法又はコンピュータ・プログラムとして具体化することができる。従って、1つ又は複数の態様は、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態をとることができ、本明細書においては、これらは全て、一般的に「回路」、「モジュール」又は「システム」と呼ぶことがある。さらに、1つ又は複数の態様は、媒体内に具体化されたコンピュータ可読プログラム・コードを有する、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・プログラムの形態をとることができる。

10

【0250】

1つ又は複数のコンピュータ可読媒体のいずれかの組み合わせを用いることもできる。コンピュータ可読媒体は、コンピュータ可読ストレージ媒体とすることができる。コンピュータ可読ストレージ媒体は、例えば、これらに限定されるものではないが、電子、磁気、光学、電磁気、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、又はこれらのいずれかの適切な組み合わせとすることができる。コンピュータ可読ストレージ媒体のより具体的な例（非網羅的なリスト）として、以下のもの、即ち、1つ又は複数の配線を有する電氣的接続、ポータブル・コンピュータ・ディスク、ハード・ディスク、ランダム・アクセス・メモリ（RAM）、読み出し専用メモリ（ROM）、消去可能なプログラム可能読み出し専用メモリ（EPROM又はフラッシュ・メモリ）、光ファイバ、ポータブル・コンパクト・ディスク型読み出し専用メモリ（CD-ROM）、光記憶装置、磁気記憶装置、又は上記のいずれかの適切な組み合わせが挙げられる。本明細書の文脈においては、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関連して用いるためのプログラムを収容又は格納することが可能な、任意の有形媒体とすることができる。

20

30

【0251】

図15を参照すると、一例において、コンピュータ・プログラム1200は、例えば、1つ又は複数の実施形態を提供し、容易にするように、その上にコンピュータ可読プログラム・コード手段又は論理1204を格納するための、1つ又は複数の一時的でないコンピュータ可読ストレージ媒体1202を含む。

【0252】

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のではないが、無線、有線、光ファイバ・ケーブル、RF等、又はこれらの任意の適切な組み合わせを含む、適切な媒体を用いて伝送することができる。

40

【0253】

1つ又は複数の実施形態のため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、Smalltalk（登録商標）、C++など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などのような従来の手続き型プログラミング言語、アセンブラ、又は類似の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任意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、全体をユーザのコンピュータ上で実行することができ、独立型ソフトウェア・パッケージとして部分的にユーザのコンピュータ上で実行することができ、一部をユーザのコンピュータ上で実行し、一部を遠隔コンピュータ上で実行することができ、又は全体を遠隔コンピュータ若しくはサーバ上で実行することができる。後者のシナリオにおいては、

50



遠隔コンピュータは、ローカル・エリア・ネットワーク（LAN）若しくは広域ネットワーク（WAN）を含むいずれかのタイプのネットワークを通じてユーザのコンピュータに接続されるか、又は、若しくはコンバインド・ネットワークを含むがこれらに限定されるものではない、いずれかのタイプのネットワーク若しくは通信システムを通じてユーザのコンピュータに接続することができ、又は（例えば、インターネット・サービス・プロバイダを用いたインターネットを通じて）外部コンピュータへの接続をなすことができる。

【0254】

1つ又は複数の実施形態は、本明細書において、方法、装置（システム）及びコンピュータ・プログラムのフローチャート図及び／又はブロック図を参照して説明される。フローチャート図及び／又はブロック図の各ブロック、並びにフローチャート図及び／又はブロック図におけるブロックの組み合わせは、コンピュータ・プログラム命令によって実装できることが理解されるであろう。これらのコンピュータ・プログラム命令を、汎用コンピュータ、専用コンピュータ、又は他のプログラム可能データ処理装置のプロセッサに与えて、マシンを製造し、その結果、コンピュータ又は他のプログラム可能データ処理装置のプロセッサによって実行される命令が、フローチャート及び／又はブロック図の1つ又は複数のブロックにおいて指定された機能／動作を実装するための手段を生成することができる。

10

【0255】

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、又は他のデバイスを特定の方式で機能させるように指示することができるコンピュータ可読媒体内に格納し、その結果、そのコンピュータ可読媒体内に格納された命令が、フローチャート及び／又はブロック図の1つ又は複数のブロックにおいて指定された機能／動作を実装する命令を含む製品を製造するようにすることもできる。

20

【0256】

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装置、又は他のデバイス上で一連の動作ステップを行わせてコンピュータ実装プロセスを生成し、それにより、その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／動作を実施するためのプロセスを提供するようにすることもできる。

30

【0257】

図面内のフローチャート及びブロック図は、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行するための1つ又は複数の実行可能な命令を含む、モジュール、セグメント又はコードの一部を表すことができる。幾つかの代替的な実施において、ブロック内に記された機能は、図面内に記された順序とは異なる順序で行われることがあることにも留意すべきである。例えば、連続して示された2つのブロックは、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、又はこれらのブロックは、ときには逆の順序で実行されることもある。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、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することができることにも留意されたい。

40

【0258】

上記に加えて、1つ又は複数の態様は、顧客環境の管理を提供するサービス・プロバイダによって供与、提供、配置、管理、サービス等を行うことができる。例えば、サービス・プロバイダは、1又は複数の顧客のために1つ又は複数の態様を実施するコンピュータ・コード及び／又はコンピュータ・インフラストラクチャの作成、保守、サポート等を行うことができる。見返りに、サービス・プロバイダは、例として、予約申し込み及び／又は報酬契約の下で顧客から支払いを受けることができる。付加的に又は代替的に、サービ

50

ス・プロバイダは、1又は複数の第三者に対する広告内容の販売から支払いを受けることができる。

【0259】

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

【0260】

更に別の態様として、コンピュータ可読コードをコンピュータ・システムに統合することを含む、コンピュータ・インフラストラクチャを配置することができ、そこでは、コードは、コンピューティング・システムと協働して、1つ又は複数の実施形態を実施することができる。

10

【0261】

更に別の態様として、コンピュータ可読コードをコンピュータ・システムに統合することを含む、プロセスを提供することができる。コンピュータ・システムは、コンピュータ可読媒体を含み、ここで、コンピュータ媒体は、1つ又は複数の実施形態を含む。コードは、コンピュータ・システムと協働して、1つ又は複数の実施形態を実施することができる。

【0262】

種々の実施形態が上述されたが、これらは例にすぎない。例えば、他のアーキテクチャのコンピューティング環境を使用することができる。さらに、異なる命令、命令形式、命令フィールド、及び/又は命令の値を使用することができる。加えて、異なる他の及び/又は付加的な制限/制約を提供/使用することもできる。多くの変形が可能である。

20

【0263】

さらに、他のタイプのコンピューティング環境を使用することができる。一例として、プログラム・コードを格納及び/又は実行するのに適しており、システム・バスを介してメモリ要素に直接又は間接的に結合された少なくとも2つのプロセッサを含む、データ処理システムを使用することができる。メモリ要素は、例えば、プログラム・コードの実行中に用いられるローカル・メモリ、大容量記憶装置、及び実行中に大容量記憶装置からコードを取り出さなければならない回数を減らすために少なくとも幾つかのプログラム・コードの一時的なストレージを提供するキャッシュ・メモリを含む。

30

【0264】

入力/出力即ちI/Oデバイス(これらに限定されるものではないが、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、サムドライブ及び他のメモリ媒体等)は、直接システムに結合することもでき、又は介在する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。ネットワーク・アダプタをシステムに結合させて、データ処理システムが、介在する私的ネットワーク又は公衆ネットワークを通じて他のデータ処理システム又は遠隔プリンタ若しくはストレージ・デバイスに結合できるようにすることもできる。モデム、ケーブル・モデム及びイーサネット・カードは、ネットワーク・アダプタの利用可能なタイプのうちのほんの数例である。

40

【0265】

図16を参照すると、1つ又は複数の実施形態を実装するためのホスト・コンピュータ・システム5000の代表的なコンポーネントが描かれる。代表的なホスト・コンピュータ5000は、コンピュータ・メモリ(即ち、中央ストレージ)5002と通信する1つ又は複数のCPU5001と、他のコンピュータ若しくはSANなどとの通信のためのストレージ媒体デバイス5011及びネットワーク5010に対するI/Oインターフェースとを含む。CPU5001は、アーキテクチャ化命令セット及びアーキテクチャ化機能を有するアーキテクチャに準拠している。CPU5001は、アクセス・レジスタ変換(ART)5012を有し、これは、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(DAT)5003により用いられるアドレ

50

ス空間を選択するためのARTルックアサイド・バッファ(ALB)5013を含む。DAT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(TLB)5007を含み、後でコンピュータ・メモリ5002のブロックにアクセスしたときにアドレス変換による遅延を必要とせず済むように変換をキャッシュに入れるための、変換ルックアサイド・バッファ(TLB)5007を含む。典型的には、キャッシュ5009は、コンピュータ・メモリ5002とプロセッサ5001との間で用いられる。キャッシュ5009は、1つより多くのCPUが利用できる大型キャッシュと、大型キャッシュと各CPUとの間のより小型で高速な(下位レベルの)キャッシュとを有する階層構造とすることができる。いくつかの実施において、下位レベルのキャッシュは、命令フェッチ及びデータ・アクセスのための個別の下位レベル・キャッシュを提供するように分割される。一実施形態において、TXファシリティについては、トランザクション診断ブロック(TDB)5100及び1つ又は複数のバッファ5101を、キャッシュ5009及びメモリ5002の1つ又は複数に格納することができる。一例においては、TXモードにおいて、データは最初にTXバッファに格納され、TXモードが終了すると(例えば、最外TEND)、バッファ内のデータは、メモリに格納(コミット)されるか、又は、アポートがある場合、バッファ内のデータが廃棄される。

10

20

30

40

50

**【0266】**

一実施形態において、命令は、命令フェッチ・ユニット5004によりメモリ5002からキャッシュ5009を介してフェッチされる。命令は、命令デコード・ユニット5006内でデコードされ(幾つかの実施形態においては他の命令と共に)命令実行ユニット5008にディスパッチされる。典型的には、幾つかの実行ユニット5008、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が用いられる。さらに、TXファシリティの一実施形態において、種々のTX制御5110を用いることができる。命令は、実行ユニットにより、必要に応じて命令が指定するレジスタ又はメモリからのオペランドにアクセスすることにより実行される。オペランドがメモリ5002からアクセスされる(ロードされる又はストアされる)場合には、典型的には、ロード/ストア・ユニット5005が、実行されている命令の制御下でアクセスを取り扱う。命令は、ハードウェア回路若しくは内部マイクロコード(ファームウェア)、又はこの両方の組み合わせにより実行することができる。

**【0267】**

TXファシリティの態様によると、プロセッサ5001は、PSW5102(例えばTX及び/又はアポートPSW)、ネスト化深さ5104、TDBA5106及び1つ又は複数の制御レジスタ5108も含む。

**【0268】**

既述のように、コンピュータ・システムは、ローカル(又は、主)ストレージ内の情報、並びにアドレス指定、保護、並びに参照及び変更記録を含む。アドレス指定の幾つかの態様は、アドレスの形式、アドレス空間の概念、アドレスの種々のタイプ及び1つのタイプのアドレスが別のタイプのアドレスに変換される方法を含む。主ストレージの一部は、恒久的に割り当てられたストレージ位置を含む。主ストレージは、システムに、直接アドレス可能なデータの高速アクセス・ストレージを提供する。データ及びプログラムの両方とも、これらが処理される前に(入力デバイスから)主ストレージにロードされる。

**【0269】**

主ストレージは、キャッシュと呼ばれることがある、1つ又は複数のより小型の高速アクセス・バッファ・ストレージを含むことができる。キャッシュは、典型的には、CPU又はI/Oプロセッサと物理的に関連付けられる。物理的構造の、性能を除いた効果及び別個のストレージ媒体の使用は、一般に、プログラムにより観察することができない。

**【0270】**

命令及びデータ・オペランドに対して、別個のキャッシュを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・ブロック又はキャッシュ・ライン(又は、簡単に言えばライン)と呼ばれる整数境界上の連続バイトで維持される。モデルは、キャッシュ・ラ

インのサイズをバイト単位で戻すEXTRACT CACHE ATTRIBUTE命令を提供することができる。モデルはまた、データ又は命令キャッシュへのストレージのプリフェッチ又はキャッシュからのデータの解放を行うPREFETCH DATA及びPREFETCH DATA RELATIVE LONG命令も提供することができる。  
【0271】

ストレージは、ビットの水平の長い文字列として見ることができる。殆どの操作では、ストレージへのアクセスは、左から右への順で進行する。ビットの文字列は、8ビット単位で細分される。この8ビットの単位はバイトと呼ばれ、これは全ての情報形式の基本構成単位である。ストレージ内の各々のバイト位置は、負でない固有の整数により識別され、この整数がバイト位置のアドレス、即ち、簡単にバイト・アドレスである。隣接するバイト位置は連続するアドレスを有し、左端の0から始まって左から右へ順に進行する。アドレスは、符号なしの2進整数であり、24ビット、31ビット又は64ビットである。

10

【0272】

情報は、ストレージとCPU又はチャンネル・サブシステムとの間で、一度に1バイトずつ、又は1グループ分のバイトで伝送される。特に断りのない限り、例えばz/Architectureにおいて、ストレージ内のバイト・グループは、グループの左端のバイトによりアドレス指定される。グループ内のバイト数は、実行される操作により暗黙的に決定される場合、又は明示的に決定される場合がある。CPU操作に使用される場合、バイト・グループはフィールドと呼ばれる。各々のバイト・グループ内において、例えばz/Architectureにおいて、ビットは、左から右の順に番号付けされる。z/Architectureにおいて、左端のビットを「最上位」ビットと呼び、右端のビットを「最下位」ビットと呼ぶことがある。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。アドレス指定できるのはバイトだけである。ストレージ内の1つのバイトの個々のビットに対して操作を行うためには、そのバイト全体にアクセスされる。1バイトの中のビットには、(z/Architectureにおいて)左から右に0から7までの番号が付けられる。1つのアドレスの中のビットには、24ビット・アドレスの場合は、8-31又は40-63の番号が付けられ、又は31ビット・アドレスの場合は、1-31又は33-63の番号が付けられ、又は64ビット・アドレスの場合は、0-63の番号が付けられる。一例においては、ビット8-31及び1-31は、32ビット長の記憶位置(例えばレジスタ)にあるアドレスに適用され、一方、ビット40-63及び33-63は、64ビット長の記憶位置にあるアドレスに適用される。複数バイトの任意の他の固定長形式において、形式を構成するビットは、0から始まって連続的に番号が付けられる。エラー検出のため、また好ましくは訂正のため、各々のバイト又はバイト・グループと共に1又は複数の検査ビットを伝送することができる。こうした検査ビットは、マシンにより自動的に生成され、プログラムにより直接制御することはできない。ストレージ容量は、バイト数で表現される。ストレージ・オペランド・フィールドの長さが命令のオペコードで暗黙指定される場合、そのフィールドは固定長を有するといわれ、この長さは1バイト、2バイト、4バイト、8バイト又は16バイトとすることができる。幾つかの命令に対しては、より大きいフィールドが暗黙指定される。ストレージ・オペランドの長さが暗黙指定されず、明示的に指定される場合は、そのフィールドは可変長を有するといわれる。可変長オペランドは、1バイトのインクリメント(又は幾つかの命令では、2バイトの倍数又は他の倍数で)で長さが変化し得る。情報がストレージ内に配置されると、ストレージへの物理的パスの幅が格納されるフィールドの長さより大きい場合であっても、指定したフィールドに含まれているバイト位置の内容のみが置き換えられる。

20

30

40

【0273】

情報の特定の単位は、ストレージ内の整数境界上にあるべきである。境界は、そのストレージ・アドレスがバイトでの単位の長さの倍数である場合に、情報の単位に対して整数であると呼ばれる。整数境界上にある2バイト、4バイト、8バイト、16バイト及び32バイトのフィールドには、特別な名称が与えられる。ハーフワードは、2バイト境界上にある2個の連続したバイトのグループであり、これは命令の基本構成単位である。ワー

50

ドは、4バイト境界上にある4個の連続したバイトのグループである。ダブルワードは、8バイト境界上にある8個の連続したバイトのグループである。オクトワードは、32バイト境界上にある32個の連続したバイトのグループである。クワッドワードは、16バイト境界上にある16個の連続したバイトのグループである。ストレージ・アドレスが、ハーフワード、ワード、ダブルワード、クワッドワード及びオクトワードを指定するとき、そのアドレスの2進表現では、それぞれ1個、2個、3個、4個又は5個の右端の0ビットを含む。命令は、2バイト整数境界上にあるべきである。殆どの命令のストレージ・オペランドは、境界位置合わせ要件を有さない。

#### 【0274】

命令及びデータ・オペランドに対して別個のキャッシュを実装するデバイスにおいては、ストアが後にフェッチされる命令を変更するかどうかに関係なく、プログラムが、後にフェッチされるキャッシュ・ラインに格納される場合、著しい遅延が生じ得る。

10

#### 【0275】

一例において、実施形態は、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、ピココードなどとも呼ばれる場合があるが、そのいずれも1つ又は複数の実施形態と整合性がある）により実施することができる。図16を参照すると、1つ又は複数の態様を具体化するソフトウェア・プログラム・コードは、CD-ROMドライブ、テープドライブ、又はハードドライブといった長期ストレージ媒体デバイス5011から、ホスト・システム5000のプロセッサ5001により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のような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、又はコンピュータ・メモリ5002からユーザに分散させても、又はこうした他のシステムのユーザが使用するために、ネットワーク5010上の1つのコンピュータ・システムのストレージから他のコンピュータ・システムに分散させてもよい。

20

#### 【0276】

ソフトウェア・プログラム・コードは、種々のコンピュータ・コンポーネント及び1つ又は複数のアプリケーション・プログラムの機能及び相互作用を制御するオペレーティング・システムを含む。プログラム・コードは、通常、ストレージ媒体デバイス5011から相対的により高速のコンピュータ・ストレージ5002にページングされ、そこでプロセッサ5001による処理のために利用可能になる。ソフトウェア・プログラム・コードをメモリ内、物理的媒体上で具体化し、及び/又は、ネットワークを介してソフトウェア・コードを配布する技術及び方法は周知であり、ここではこれ以上論じない。プログラム・コードは、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）上に作成され格納されたとき、「コンピュータ・プログラム」と呼ばれることが多い。コンピュータ・プログラム媒体は、典型的には、処理回路による実行のために、好ましくはコンピュータ・システム内の処理回路によって読み取り可能である。

30

#### 【0277】

図17は、1つ又は複数の実施形態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17のシステム5020は、随意的な周辺機器を含む、パーソナル・コンピュータ、ワークステーション、又はサーバなどの代表的なベース・コンピュータ・システム5021を含む。ベース・コンピュータ・システム5021は、1つ又は複数のプロセッサ5026と、周知の技術に従ってプロセッサ5026とシステム5021の他のコンポーネントを接続し、これらの間の通信を可能にするために用いられるバスとを含む。バスは、プロセッサ5026を、ハードドライブ（例えば、磁気媒体、CD、DVD及びフラッシュ・メモリのいずれかを含む）又はテープドライブを含むことができる、メモリ5025及び長期ストレージ5027に接続する。システム5021はまた、バスを介して、マイクロプロセッサ5026を、キーボード5024、マウス5023、プリンタ/スキャナ5030、及び/又はタッチ・センシティブ

40

50

・スクリーン、デジタル化された入力パッド等のいずれかのユーザ・インターフェース機器とすることができる他のインターフェース機器といった、1つ又は複数のインターフェース機器に接続する、ユーザ・インターフェース・アダプタを含むこともできる。バスはまた、ディスプレイ・アダプタを介して、LCDスクリーン又はモニタなどのディスプレイ装置5022をマイクロプロセッサ5026にも接続する。

#### 【0278】

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

#### 【0279】

図18は、1つ又は複数の実施形態を実施することができるデータ処理ネットワーク5040を示す。データ処理ネットワーク5040は、各々が複数の個々のワークステーション5041、5042、5043、5044を含むことができる、無線ネットワーク及び有線ネットワークのような複数の個々のネットワークを含むことができる。さらに、当業者であれば理解するように、1つ又は複数のLANを含ませることができ、そこで、LANは、ホスト・プロセッサに結合された複数のインテリジェント・ワークステーションを含むことができる。

#### 【0280】

さらに図18を参照すると、ネットワークはまた、ゲートウェイ・コンピュータ（クライアント・サーバ5046）、又はアプリケーション・サーバ（データ・リポジトリにアクセスすることができ、且つ、ワークステーション5045から直接アクセスすることもできる遠隔サーバ5048）のような、メインフレーム・コンピュータ又はサーバを含むこともできる。ゲートウェイ・コンピュータ5046は、各々の個々のネットワークへの入力点のとして働く。ゲートウェイは、1つのネットワーク・プロトコルを別のものに接続するとき必要とされる。ゲートウェイ5046は、通信リンクによって別のネットワーク（例えば、インターネット5047）に結合することが好ましい。ゲートウェイ5046はまた、通信リンクを用いて、1つ又は複数のワークステーション5041、5042、5043、5044に直接結合することもできる。ゲートウェイ・コンピュータは、インターナショナル・ビジネス・マシーニズ・コーポレーションから入手可能なIBM eServer System zサーバを用いて実装することができる。

#### 【0281】

図17及び図18を同時に参照すると、1つ又は複数の実施形態を具体化することができるソフトウェア・プログラム・コード5031には、一般的に、CD-ROMドライブ又はハードドライブといった長期ストレージ媒体5027から、システム5020のプロセッサ5026によってアクセスすることができる。ソフトウェア・プログラム・コードは、ディスク、ハードドライブ、又はCD-ROMといった、データ処理システムと共に用いるための種々の周知の媒体のいずれかの上で具体化することができる。コードは、そのような媒体上で分散させても、又はメモリからユーザ5050、5051に分散させても、又は、こうした他のシステムのユーザが用いるために、ネットワーク上の1つのコンピュータ・システムのメモリ若しくは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。

#### 【0282】

代替的に、プログラム・コードをメモリ5025内で具体化し、プロセッサ・バスを用

10

20

30

40

50

いてプロセッサ 5 0 2 6 によってプログラム・コードにアクセスすることができる。このようなプログラム・コードは、種々のコンピュータ・コンポーネント及び1つ又は複数のアプリケーション・プログラム 5 0 3 2 の機能及び相互作用を制御するオペレーティング・システムを含む。プログラム・コードは、通常、ストレージ媒体 5 0 2 7 から高速メモリ 5 0 2 5 にページングされ、そこでプロセッサ 5 0 2 6 による処理のために利用可能になる。ソフトウェア・プログラム・コードをメモリ内、物理的媒体上で具体化し、及び/又は、ネットワークを介してソフトウェア・コードを配布する技術及び方法は周知であり、ここではこれ以上論じない。プログラム・コードは、有形の媒体（これらに限定されるものではないが、電子メモリ・モジュール（RAM）、フラッシュ・メモリ、コンパクト・ディスク（CD）、DVD、磁気テープなどを含む）に格納されたとき、「コンピュータ・プログラム」と呼ばれることが多い。コンピュータ・プログラム媒体は、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

#### 【0283】

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

#### 【0284】

図19を参照すると、プロセッサ 5 0 2 6 についての例示的なプロセッサの実施形態が示される。典型的には、メモリ・ブロックをバッファに入れてプロセッサ性能を向上させるために、1つ又は複数のレベルのキャッシュ 5 0 5 3 が用いられる。キャッシュ 5 0 5 3 は、用いられる可能性が高いメモリ・データのキャッシュ・ラインを保持する高速バッファである。典型的なキャッシュ・ラインは、64バイト、128バイト、又は256バイトのメモリ・データである。データをキャッシュに入れるのではなく、命令をキャッシュに入れるために、別個のキャッシュが用いられることが多い。キャッシュ・コヒーレンス（メモリ及びキャッシュ内のラインのコピーの同期）は、多くの場合、当技術分野において周知の種々の「スヌープ」アルゴリズムによって与えられる。プロセッサ・システムの主メモリ・ストレージ 5 0 2 5 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 5 0 5 3 を有するプロセッサ・システムにおいて、主ストレージ 5 0 2 5 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。主ストレージ 5 0 2 5 は、オペレーティング・システムによって主ストレージ 5 0 2 5 との間でページングされるデータのページを「キャッシュに入れる」。

30

#### 【0285】

プログラム・カウンタ（命令カウンタ） 5 0 6 1 は、実行される現行の命令のアドレスを常時監視している。z / A r c h i t e c t u r e プロセッサのプログラム・カウンタは64ビットであり、従来のアドレッシング制限をサポートするために、31ビット又は24ビットに切り捨てることができる。プログラム・カウンタは、典型的には、コンテキスト・スイッチの際に持続するように、コンピュータのPSW（プログラム状況ワード）内で具体化される。従って、例えば、オペレーティング・システムにより、プログラム・カウンタ値を有する進行中のプログラムに割り込みをかけることが可能である（プログラム環境からオペレーティング・システム環境へのコンテキスト・スイッチ）。プログラムのPSWは、プログラムがアクティブでない間、プログラム・カウンタ値を保持し、オペレーティング・システムが実行されている間、オペレーティング・システムの（PSW内の）プログラム・カウンタが用いられる。典型的には、プログラム・カウンタは、現行の命令のバイト数に等しい量だけインクリメントされる。RISC命令は、典型的には固定

40

50

長であり、CISC命令は、典型的には可変長である。IBM 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操作により、プログラム・カウンタ5061が変更される。コンテキスト・スイッチ操作において、現行のプログラム・カウンタ値は、実行されるプログラムについての他の状態情報（条件コードのような）と共にプログラム状況ワード内に保存され、実行される新しいプログラム・モジュールの命令を指し示す新しいプログラム・カウンタ値がロードされる。分岐成立操作を行い、分岐命令の結果をプログラム・カウンタ5061にロードすることにより、プログラムが判断を下すこと又はプログラム内でループすることを可能にする。

#### 【0286】

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

#### 【0287】

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

#### 【0288】

仮想アドレスは、動的アドレス変換5062、随機的にアクセス・レジスタ・トランザクション5063を用いて、実アドレスに変換される。

#### 【0289】

プロセッサ5026は、典型的には、命令の機能を実行するための1つ又は複数の実行ユニット5057、5058、5060を有する。図20を参照すると、実行ユニット5057は、インターフェース論理5071を介して、アーキテクチャ化された汎用レジスタ5059、デコード/ディスパッチ・ユニット5056、ロード・ストア・ユニット5060、及び他のプロセッサ・ユニット5065と通信する5071ことができる。実行ユニット5057は、幾つかのレジスタ回路5067、5068、5069を用いて、算術論理演算ユニット（ALU）5066が動作する情報を保持することができる。ALUは、加算、減算、乗算、及び除算などの算術演算、並びに、論理積、論理和、及び排他的論理和、ローテート及びシフトのような論理関数を実行する。ALUは、設計に依存する専用の演算をサポートすることが好ましい。他の回路は、例えば条件コード及び回復サポート論理を含む、他のアーキテクチャ化ファシリティ5072を提供することができる。典型的には、ALU演算の結果は、出力レジスタ回路5070に保持され、この出力レジスタ回路5070が、結果を種々の他の処理機能に転送することができる。多数のプロセッサ・ユニットの構成が存在し、本説明は、一実施形態の代表的な理解を与えることのみを意図している。

#### 【0290】

例えばADD命令は、算術及び論理機能を有する実行ユニット5057で実行され、一方、例えば浮動小数点命令は、特化された浮動小数点能力を有する浮動小数点実行部で実

10

20

30

40

50



行される。実行ユニットは、オペランドに対してオペコードが定めた関数を行うことにより、命令が特定したオペランドに対して動作することが好ましい。例えば、ADD命令は、命令のレジスタ・フィールドによって特定された2つのレジスタ5059内に見出されるオペランドに対して、実行ユニット5057により実行することができる。

#### 【0291】

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

10

20

#### 【0292】

図21を参照すると、分岐命令を実行するための分岐命令情報が、典型的には、分岐ユニット5058に送られ、この分岐ユニット5058は、多くの場合、分岐履歴テーブル5082のような分岐予測アルゴリズムを用いて、他の条件付き演算が完了する前に分岐の結果を予測する。条件付き演算が完了する前に、現行の分岐命令のターゲットがフェッチされ、投機的に実行される。条件付き演算が完了すると、投機的に実行された分岐命令は、条件付き演算の条件及び投機された結果に基づいて、完了されるか又は破棄される。典型的な分岐命令は、条件コードを試験し、条件コードが分岐命令の分岐要件を満たす場合、ターゲット・アドレスに分岐することができ、ターゲット・アドレスは、例えば、命令のレジスタ・フィールド又は即値フィールド内に見出されるものを含む幾つかの数に基づいて計算することができる。分岐ユニット5058は、複数の入力レジスタ回路5075、5076、5077と、出力レジスタ回路5080とを有するALU5074を用いることができる。分岐ユニット5058は、例えば、汎用レジスタ5059、デコード・ディスパッチ・ユニット5056、又は他の回路5073と通信する5081ことができる。

30

#### 【0293】

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。コンテキスト・スイッチ動作は、現在実行中のプログラムについての状態情報を保存し、次いで、起動される別のプログラムについての状態情報をロードすることが好ましい。状態情報は、例えば、ハードウェア・レジスタ又はメモリ内に保存することができる。状態情報は、実行される次の命令を指し示すプログラム・カウンタ値と、条件コードと、メモリ変換情報と、アーキテクチャ化されたレジスタの内容とを含むことが好ましい。コンテキスト・スイッチの活動は、ハードウェア回路、アプリケーション・プログラム、オペレーティング・システム・プログラム、又はファームウェア・コード(マイクロコード、ピココード、又はライセンス内部コード(LIC))単独で又はその組み合わせで実施することができる。

40

50

## 【0294】

プロセッサは、命令により定義された方法に従ってオペランドにアクセスする。命令は、命令の一部の値を用いて即値オペランドを与えることができ、汎用レジスタ又は専用レジスタ（例えば、浮動小数点レジスタ）のいずれかを明示的に指し示す1つ又は複数のレジスタ・フィールドを与えることができる。命令は、オペコード・フィールドによって、オペランドとして識別されるインプライド・レジスタ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、ベース・レジスタ、索引レジスタ、及び即値フィールド（変位フィールド）を定め、これらが、例えば互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。ここでの位置は、典型的には、特に断りのない限り、主メモリ（主ストレージ）内の記憶位置を意味する。

10

## 【0295】

図22を参照すると、プロセッサは、ロード/ストア・ユニット5060を用いて、ストレージにアクセスする。ロード/ストア・ユニット5060は、メモリ5053内のターゲット・オペランドのアドレスを取得し、オペランドをレジスタ5059又は別のメモリ5053の記憶位置にロードすることによってロード操作を行うことができ、或いは、メモリ5053内のターゲット・オペランドのアドレスを取得し、レジスタ5059又は別のメモリ5053の記憶位置から取得したデータをメモリ5053内のターゲット・オペランドの記憶位置に格納することによって、ストア操作を行うことができる。ロード/ストア・ユニット5060は、投機的なものであってもよく、命令シーケンスに対してアウト・オブ・オーダー式の順序でメモリにアクセスすることができるが、プログラムに対して、命令がインオーダー式に実行されたという外観を維持することになる。ロード/ストア・ユニット5060は、汎用レジスタ5059、デコード/ディスパッチ・ユニット5056、キャッシュ/メモリ・インターフェース5053、又は他の要素5083と通信することができ、ストレージ・アドレスを計算し、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、ALU5085、及び制御論理5090を含む。一部の動作は、アウト・オブ・オーダー式とすることができるが、ロード/ストア・ユニットは、アウト・オブ・オーダー式動作が、プログラムに対して、当技術分野において周知のようなインオーダー式に実行されたように見えるようにする機能を提供する。

20

30

## 【0296】

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、単に仮想アドレスをオフセット値にプリフィックス付加すること、1つ又は複数の変換テーブルを介して仮想アドレスを変換することを含む、種々の動的アドレス変換（DAT）技術の1つによって、物理的メモリ位置にリダイレクトされるという点で仮想のものであり、変換テーブルは、少なくともセグメント・テーブル及びページ・テーブルを単独で又は組み合わせて含むことが好ましく、セグメント・テーブルは、ページ・テーブルを指し示すエントリを有することが好ましい。z / Architectureでは、領域第1テーブル、領域第2テーブル、領域第3テーブル、セグメント・テーブル、及び随意的なページ・テーブルを含む、変換の階層構成が提供される。アドレス変換の性能は、仮想アドレスを関連した物理的メモリ位置にマッピングするエントリを含む変換ルックアサイド・バッファ（TLB）を用いることにより改善されることが多い。DATが変換テーブルを用いて仮想アドレスを変換したときに、エントリが作成される。次いで、後に仮想アドレスを用いることで、低速の順次変換テーブル・アクセスではなく、高速のTLBのエントリを用いることが可能になる。TLBの内容は、LRU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。

40

50

## 【0297】

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

## 【0298】

I/Oユニット5054(図19)は、プロセッサに、例えば、テープ、ディスク、プリンタ、ディスプレイ、及びネットワークを含む周辺機器に取り付けるための手段を与える。I/Oユニットは、ソフトウェア・ドライバによってコンピュータ・プログラムに提示されることが多い。IBMによるSystem z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。

10

## 【0299】

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

20

## 【0300】

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

30

## 【0301】

より具体的には、メインフレームにおいて、アーキテクチャ化されたマシン命令は、通常、プログラマによって、多くの場合コンパイラ・アプリケーションを介して、今日では「C」プログラマによって用いられる。ストレージ媒体内に格納されたこれらの命令は、z/ArchitectureのIBM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メインフレーム・サーバにおいて、及び、IBMの他のマシン(例えば、Power Systemsサーバ及びSystem x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社、Intel社、AMD社などによって製造されたハードウェアを用いて種々のマシン上でLinux(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下でそのハードウェア上で実行することに加えて、Linuxを用いること、並びに、一般に実行がエミュレーション・モードにあるHercules、UMX、又はFSI(Fundame

40

50

ntal Software, Inc)によるエミュレーションを用いるマシンを用いることもできる。エミュレーション・モードにおいては、ネイティブ・プロセッサによって、エミュレーション・ソフトウェアが実行され、エミュレートされたプロセッサのアーキテクチャをエミュレートする。

【0302】

ネイティブ・プロセッサは、一般的に、エミュレートされたプロセッサのエミュレーションを実行するためにファームウェア又はネイティブ・オペレーティング・システムのいずれかを含むエミュレーション・ソフトウェアを実行する。エミュレーション・ソフトウェアは、エミュレートされたプロセッサ・アーキテクチャの命令のフェッチと実行を担当する。エミュレーション・ソフトウェアは、エミュレートされたプログラム・カウンタを維持し、命令境界を常時監視している。エミュレーション・ソフトウェアは、一度に1つ又は複数のエミュレートされたマシン命令をフェッチし、ネイティブ・プロセッサにより実行するために、その1つ又は複数のエミュレートされたマシン命令を、対応するネイティブマシン命令のグループに変換することができる。これらの変換された命令は、より速い変換を達成できるようにキャッシュに入れることができる。それにも関わらず、エミュレーション・ソフトウェアは、エミュレートされたプロセッサ・アーキテクチャのアーキテクチャ規則を維持して、オペレーティング・システム及びエミュレートされたプロセッサのために書かれたアプリケーションが正確に動作することを保証しなければならない。さらに、エミュレーション・ソフトウェアは、これらに限定さ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(TOD)クロック、及びI/Oサブシステムへのアーキテクチャ化インターフェースを含む、エミュレートされたプロセッサのアーキテクチャによって識別されるリソースを提供し、オペレーティング・システム又はエミュレートされたプロセッサ上で実行するように設計されたアプリケーション・プログラムが、エミュレーション・ソフトウェアを有するネイティブ・プロセッサ上で実行することができる。

10

20

【0303】

エミュレートされた特定の命令がデコードされ、個々の命令の機能を実行するためのサブルーチンが呼び出される。エミュレートされたプロセッサの機能をエミュレートするエミュレーション・ソフトウェア機能は、例えば、「C」サブルーチン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。その各々の全体が引用により本明細書に組み入れられる、Beausoleil他による「Multiprocessor for Hardware Emulation」という名称の特許文献1、Scalzi他による「Preprocessing of Stored Target Routines for Emulating Incompatible Instructions on a Target Processor」という名称の特許文献2、Davidian他による「Decoding Guest Instruction to Directly Access Emulation Routines that Emulate the Guest Instructions」という名称の特許文献3、Gorishkek他による「Symmetrical Multiprocessing Bus and Chipset Used for Coprocessor Support Allowing Non-Native Code to Run in a System」という名称の特許文献4、Lethin他による「Dynamic Optimizing Object Code Translator for Architecture Emulation and Dynamic Optimizing Object Code Translation Method」という名称の特許文献5、Eric Trautによる「Method for Emulating Guest Instructions on a Host Computer Through Dynamic Reco

30

40

50

mpilation of Host Instructions」という名称の特許文献6及び他の多くを含むがこれらに限定されない、種々のソフトウェア及びハードウェア・エミュレーションの特許は、当業者が利用可能なターゲット・マシンのための異なるマシン用に設計された命令形式のエミュレーションを達成する様々な既知の方法を示す。

【0304】

図23において、ホスト・アーキテクチャのホスト・コンピュータ・システム5000'をエミュレートする、エミュレートされたホスト・コンピュータ・システム5092の一例が提供される。エミュレートされたホスト・コンピュータ・システム5092では、ホスト・プロセッサ(CPU)5091は、エミュレートされたホスト・プロセッサ(又は、仮想ホスト・プロセッサ)であり、且つ、ホスト・コンピュータ5000'のプロセッサ5091のものとは異なるネイティブな命令セット・アーキテクチャを有するエミュレーション・プロセッサ5093を含む。エミュレートされたホスト・コンピュータ・システム5092は、エミュレーション・プロセッサ5093がアクセス可能なメモリ5094を有する。例示的な実施形態において、メモリ5094は、ホスト・コンピュータ・メモリ5096の部分と、エミュレーション・ルーチン5097の部分とに区分化される。ホスト・コンピュータ・メモリ5096は、ホスト・コンピュータ・アーキテクチャに従い、エミュレートされたホスト・コンピュータ・システム5092のプログラムに利用可能である。エミュレーション・プロセッサ5093は、エミュレートされたプロセッサ5091のもの以外のアーキテクチャのアーキテクチャ化された命令セットのネイティブ命令を実行し、このネイティブ命令はエミュレーション・ルーチン・メモリ5097から取得されたものであり、且つ、エミュレーション・プロセッサ5093は、シーケンス及びアクセス/デコード・ルーチンにおいて取得される1つ又は複数の命令を用いることにより、ホスト・コンピュータ・メモリ5096の中のプログラム由来の実行のためのホスト命令にアクセスすることができ、このシーケンス及びアクセス/デコード・ルーチンは、アクセスされたホスト命令をデコードして、アクセスされたホスト命令の機能をエミュレートするためのネイティブ命令実行ルーチンを判断することができる。ホスト・コンピュータ・システム5000'のアーキテクチャのために定められた、例えば、汎用レジスタ、制御レジスタ、動的アドレス変換、及びI/Oサブシステムのサポート、並びにプロセッサ・キャッシュといったファシリティを含む他のファシリティを、アーキテクチャ化ファシリティ・ルーチンによってエミュレートすることができる。エミュレーション・ルーチンは、エミュレーション・ルーチンの性能を高めるために、エミュレーション・プロセッサ5093において利用可能な(汎用レジスタ、及び仮想アドレスの動的変換といった)機能を利用することもできる。ホスト・コンピュータ5000'の機能をエミュレートする際にプロセッサ5093を補助するために、専用のハードウェア及びオフ・ロード・エンジンを設けることもできる。

【0305】

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示するが、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/又はそれらの群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。

【0306】

以下の特許請求の範囲に存在する場合、「手段又はステップと機能との組合せ(ミーンズ又はステップ・プラス・ファンクション)」要素の対応する構造、材料、動作及び均等物は、明確に特許請求された他の請求要素と共に機能を実行するための任意の構造体、材料、又は行為を含むことを意図したものである。本発明の説明は、例証及び説明のためだけに提示されたものであり、網羅的であること又は本発明を開示した形態に限定すること

10

20

30

40

50

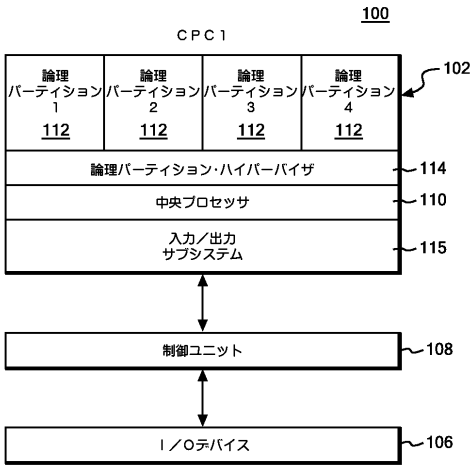
を意図したものではない。当業者には、本発明の範囲から逸脱しない多くの修正物及び変形物が明らかとなるであろう。実施形態は、本発明の原理及び実際の用途を最もよく説明するため、及び、当業者が、企図した特定の用途に適するように種々の修正を有する種々の実施形態に関して本発明を理解することができるように、選択され記述された。

【符号の説明】

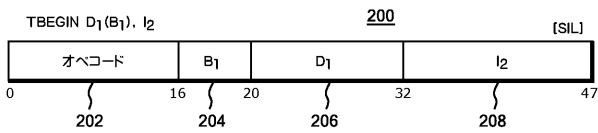
【0307】

100	: コンピューティング環境	
102	: 中央プロセッサ・コンプレックス (CPC)	
106	: I/Oデバイス	
108	: 制御ユニット	10
110	: 中央プロセッサ	
112	: パーティション	
114	: 論理パーティション・ハイパーバイザ	
115	: 入力/出力サブシステム	
200	: TRANSACTION BEGIN (TBEGIN) 命令	
202、302、402、502、702a、702b、802	: オペコード・フィールド	
204、304、504、708	: ベース・フィールド	
206、306、506、710	: 変位フィールド	
208、308	: 即値フィールド	20
210、310	: 汎用レジスタ保存マスク (GRSM)	
212、312	: AR変更許可 (A)	
214	: 浮動小数点演算許可 (F)	
216	: プログラム割り込みフィルタ処理制御 (PIFC)	
300	: TRANSACTION BEGIN constrained (TBEGIN C) 命令	
400	: TRANSACTION END (TEND) 命令	
500	: TRANSACTION ABORT (TABORT) 命令	
600	: ネスト化されたトランザクション	
700	: NONTRANSACTIONAL STORE (NTSTG) 命令	30
704、804	: レジスタ・フィールド	
706	: インデックス・フィールド	
800	: EXTRACT TRANSACTION NESTING DEPTH (ETND) 命令	
900	: トランザクション診断ブロック (TDB)	
1200	: コンピュータ・プログラム製品	

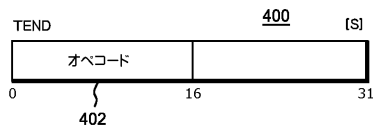
【 図 1 】



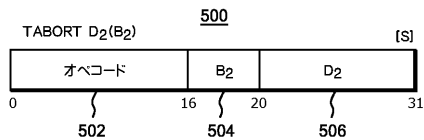
【 図 2 】



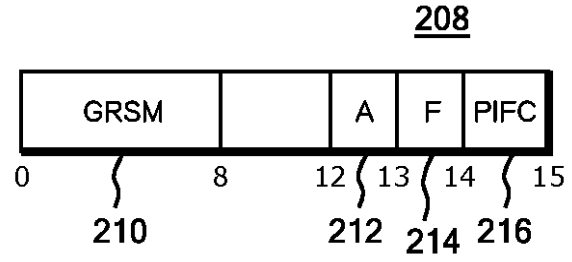
【 図 6 】



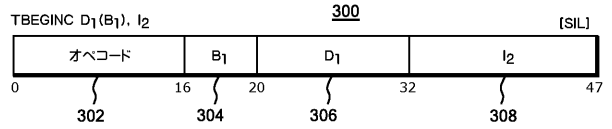
【 図 7 】



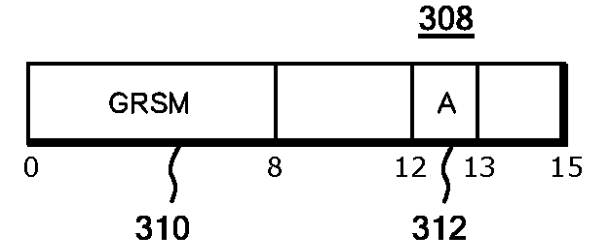
【 図 3 】



【 図 4 】



【 図 5 】



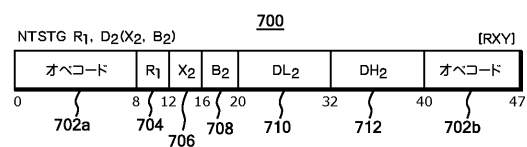
【 図 8 】

ネスト化されたトランザクション

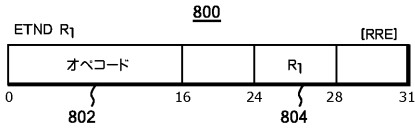
ネスト化 深さ	A CSECT		
0	...		
0	TBEGIN...~600		
1	...		
1	BAS 14, B		
1		B CSECT	608
1	...		
1	TBEGIN...~602		
2	...		
2	BAS 7, C		
2		C CSECT	
2	...		
2	604 TBEGIN...		
3	...		
3	TEND...		
2	...		606 ...
2	...		BR 7
2	...		
2	TEND...		
1	...		612
1	610	BR 14	
1	...		
1	TEND...		
0	...		

Callouts in the diagram:  
 - 601: 最外トランザクション (Outermost transaction)  
 - 608: 最内トランザクション (Innermost transaction)  
 - 612: トランザクション・スタアがコミットされる (Transaction is committed)

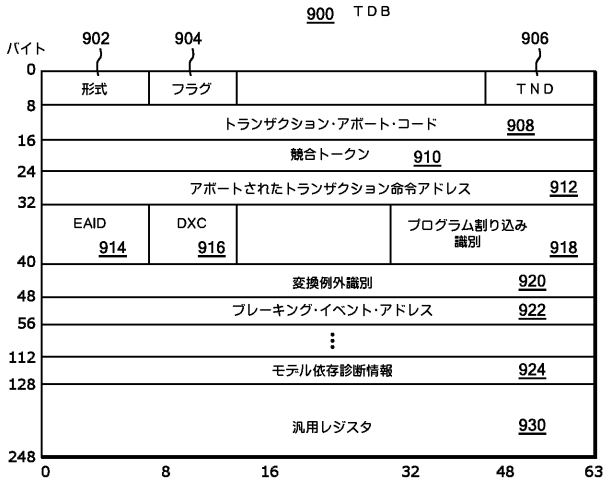
【 図 9 】



【 図 1 0 】



【 図 1 1 】

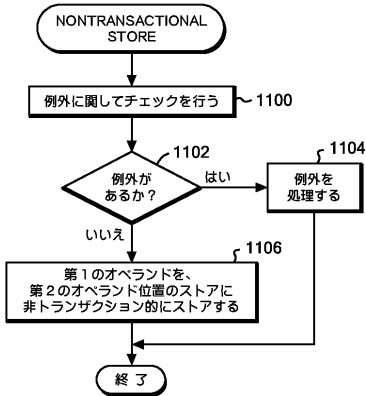


【 図 1 2 】

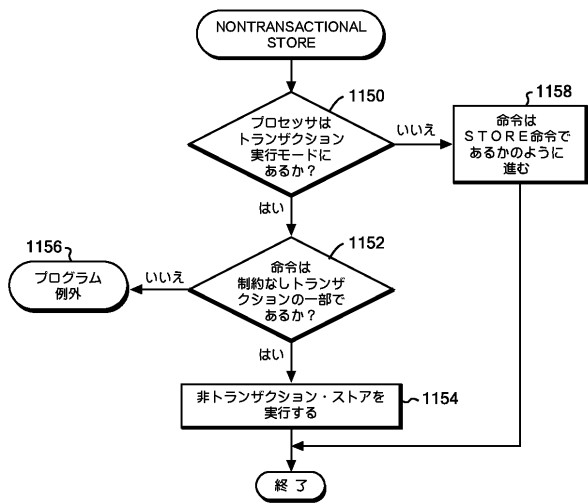
コード	アボートの理由	設定される条件コード
2	外部割り込み	2
4	プログラム割り込み (非フィルタ処理)	2又は3+
5	マシン・チェック割り込み	2
6	I/O割り込み	2
7	フェッチ・オーバーフロー	2又は3
8	ストア・オーバーフロー	2又は3
9	フェッチ競合	2
10	ストア競合	2
11	制限された命令	3
12	プログラム例外条件 (フィルタ処理済み)	3
13	ネスト化深さ超過	3
14	キャッシュ・フェッチ関連	2又は3
15	キャッシュ・ストア関連	2又は3
16	キャッシュ・アザー	2又は3
255	その他の条件	2又は3
>255	T A B O R T命令	2又は3
‡	判断不能; TDBは格納されない	1

説明:  
 + 条件コードは割り込みコードに基づく  
 † この状況は、トランザクションがアボートしたが、最外T B E G I N命令の成功後の実行後にTDBがアクセス不能になった時に発生する。T B E G I N指定TDBは格納されず、且つ、条件コードは1に設定される。

【 図 1 3 】

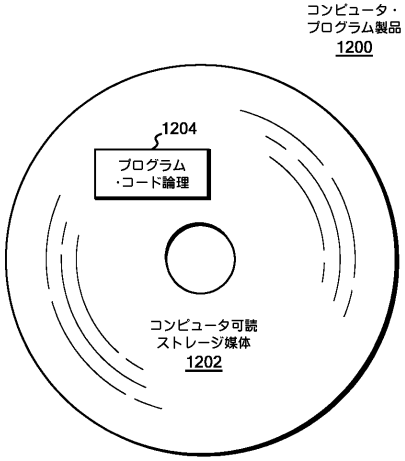


【 図 1 4 】

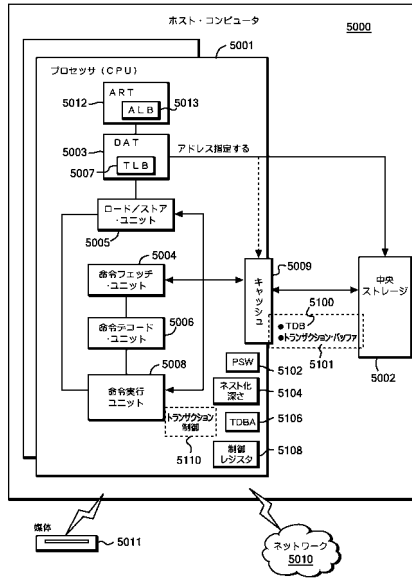




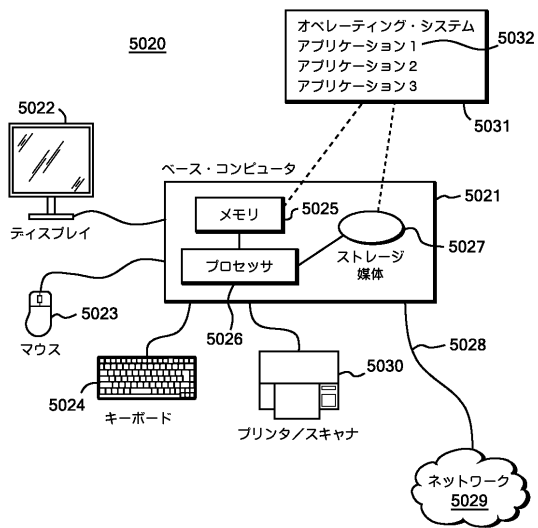
【 図 1 5 】



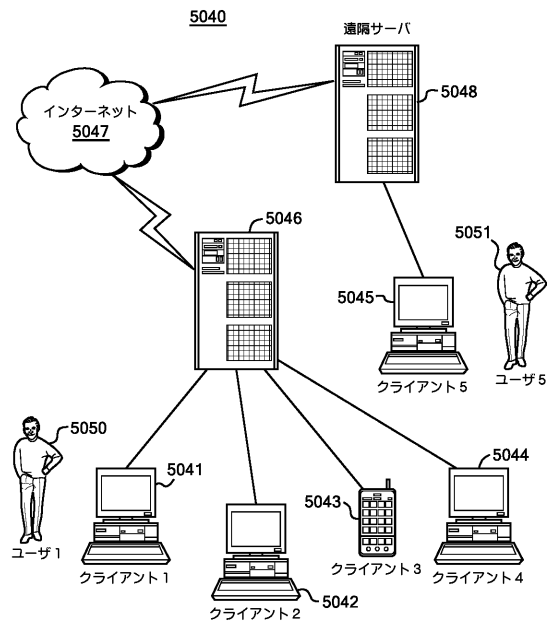
【 図 1 6 】



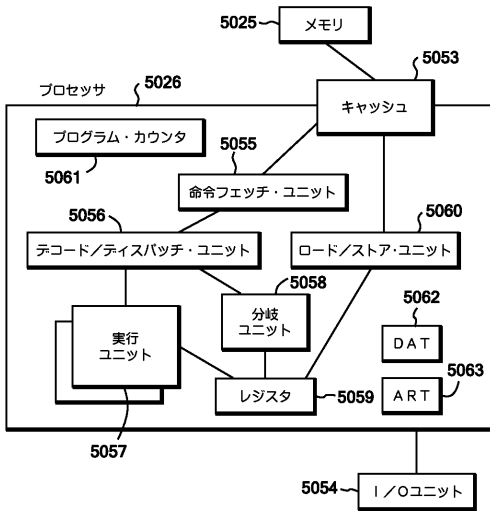
【 図 1 7 】



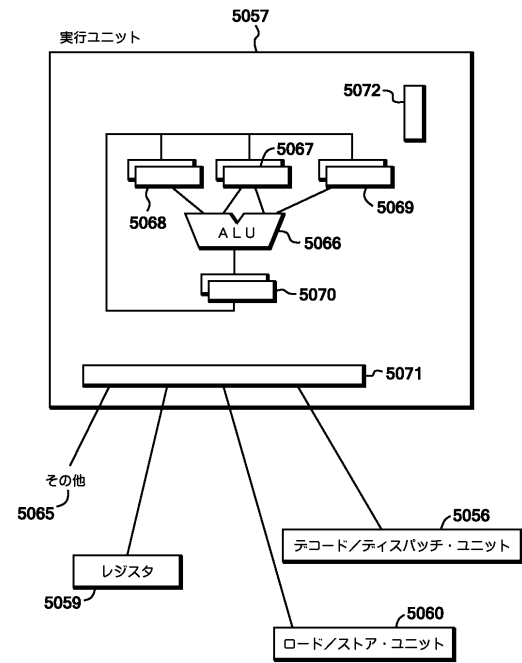
【 図 1 8 】



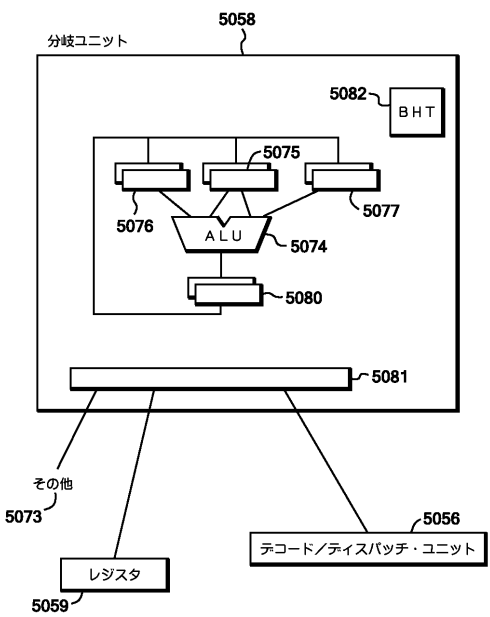
【 図 1 9 】



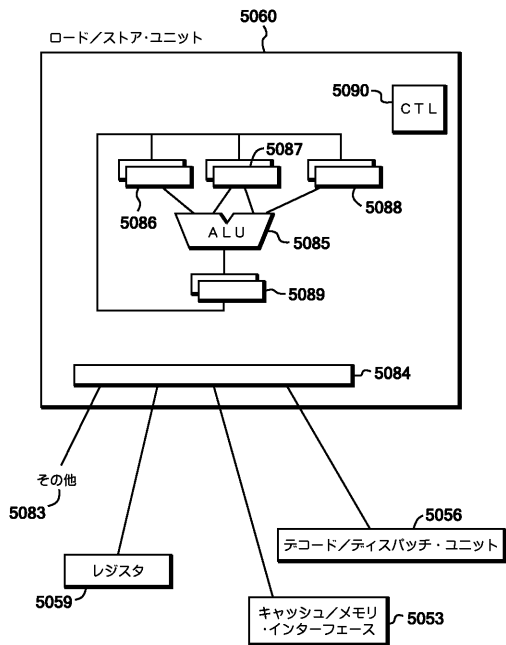
【 図 2 0 】



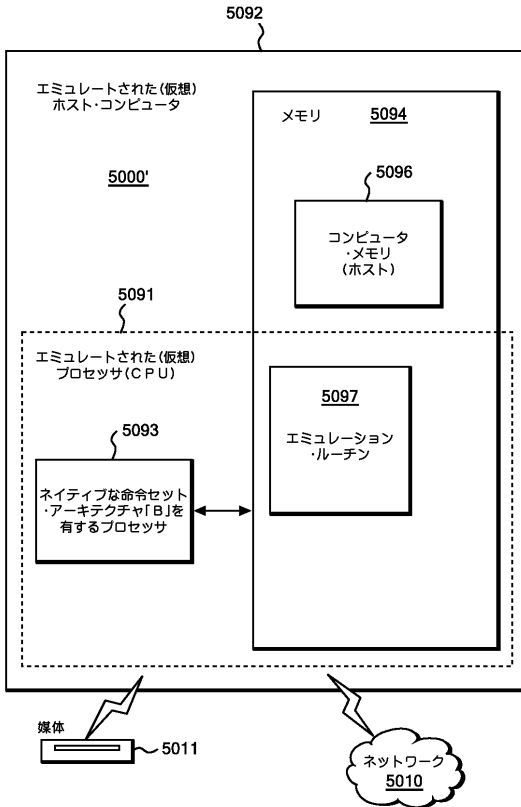
【 図 2 1 】



【 図 2 2 】



【図 2 3】



## 【手続補正書】

【提出日】平成27年1月21日(2015.1.21)

## 【手続補正 1】

【補正対象書類名】特許請求の範囲

【補正対象項目名】全文

【補正方法】変更

【補正の内容】

## 【特許請求の範囲】

## 【請求項 1】

コンピュータに、

非トランザクション・ストア操作を指定するためのオペコード、第1のオペランド、及び前記第1のオペランドの位置を指し示すための第2のオペランドを含むマシン命令をプロセッサにより取得する手順と、

前記プロセッサにより、前記第1のオペランドを、前記第2のオペランドにより指定された位置に非トランザクション的に配置する手順とを実行させ、

前記第2のオペランドに格納された情報は、前記マシン命令と関連したトランザクションのアポートにもかかわらず保持され、前記非トランザクション的に配置する手順は前記プロセッサのトランザクション実行モードの終了まで遅延される、コンピュータ・プログラム。

## 【請求項 2】

前記トランザクション実行モードの終了は、前記マシン命令と関連した最外トランザクションの終了又はアポート条件によって生じる、請求項1に記載のコンピュータ・プログラム。

## 【請求項 3】

前記プロセッサがトランザクション実行モードにあるかどうかを判断する手順と、

前記プロセッサが前記トランザクション実行モードにあることに基づいて、前記トランザクションが制約付きトランザクションであるか又は制約なしトランザクションであるかを判断する手順と、

前記トランザクションが制約なしトランザクションであることに基づいて、前記マシン命令の実行を続行する手順と、

をさらに前記コンピュータに実行させる、請求項 1 に記載のコンピュータ・プログラム。

【請求項 4】

前記トランザクションが制約付きトランザクションであることに基づいて、プログラム例外をもたらす、前記マシン命令の実行を終了させる、請求項 3 に記載のコンピュータ・プログラム。

【請求項 5】

前記トランザクションがトランザクション実行モードにないことに基づいて、前記マシン命令を STORE 命令として実行する、請求項 3 に記載のコンピュータ・プログラム。

【請求項 6】

前記第 2 のオペランドは、前記マシン命令のインデックス・フィールドにより指定されるレジスタの内容、前記マシン命令のベース・フィールドにより指定されるレジスタの内容、及び少なくとも 1 つの変位フィールドの内容の結合から形成される、請求項 1 に記載のコンピュータ・プログラム。

【請求項 7】

前記マシン命令は NON TRANSACT ION A L S T O R E 命令を含み、

TRANSACT I O N B E G I N 命令の実行に基づいて前記トランザクションを開始することであって、前記トランザクションは、選択されたトランザクションの完了まで主メモリへのトランザクション・ストアのコミットを効果的に遅延させる、開始する手順と、

前記選択されたトランザクションの実行を完了する TRANSACT I O N E N D 命令の実行に基づいて、主メモリへの前記トランザクション・ストアをコミットする手順と、

前記選択されたトランザクションの実行を終了するアボートに基づいて、前記トランザクション・ストアを廃棄する手順と、

をさらに前記コンピュータに実行させ、

前記非トランザクション的に配置することは、前記コミットすること又は前記廃棄することに関係なく実行される、請求項 1 に記載のコンピュータ・プログラム。

【請求項 8】

前記第 1 のオペランドは、前記マシン命令により指定されるレジスタの内容を含み、前記内容はユーザ指定のものである、請求項 1 に記載のコンピュータ・プログラム。

【請求項 9】

前記トランザクションは、制約なしトランザクションのネスト化における制約なしトランザクションである、請求項 1 に記載のコンピュータ・プログラム。

【請求項 10】

コンピューティング環境内で命令を実行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、

を含み、前記コンピュータ・システムは方法を実行するように構成され、前記方法は、

プロセッサにより、実行のためのマシン命令を取得することであって、前記マシン命令は、コンピュータ・アーキテクチャによるコンピュータ実行のために定められたものであり、且つ、

非トランザクション・ストア操作を指定するためのオペコードと、

第 1 のオペランドと、

前記第 1 のオペランドの位置を指し示すための第 2 のオペランドと、

を含む、取得することと、

前記プロセッサにより、前記マシン命令を実行することと、  
を含み、前記実行することは、

前記第 1 のオペランドを、前記第 2 のオペランドにより指定された位置に非トランザクシヨンの配置することを含み、前記第 2 のオペランドに格納された情報は、前記マシン命令と関連したトランザクシヨンのアポートにもかかわらず保持され、前記非トランザクシヨンの配置することは前記プロセッサのトランザクシヨンの実行モードの終了まで遅延される、コンピュータ・システム。

【請求項 11】

前記方法は、

前記プロセッサがトランザクシヨンの実行モードにあるかどうかを判断することと、

前記プロセッサが前記トランザクシヨンの実行モードにあることに基づいて、前記トランザクシヨンの制約付きトランザクシヨンの実行モードであるか又は制約なしトランザクシヨンの実行モードであるかを判断することと、

前記トランザクシヨンの制約なしトランザクシヨンの実行モードであることに基づいて、前記マシン命令の実行を続行することと、  
をさらに含む、請求項 10 に記載のコンピュータ・システム。

【請求項 12】

前記トランザクシヨンの制約付きトランザクシヨンの実行モードであることに基づいて、プログラム例外をもたらす、前記マシン命令の実行を終了させる、請求項 11 に記載のコンピュータ・システム。

【請求項 13】

前記トランザクシヨンの実行モードでないことに基づいて、前記マシン命令を STORE 命令として実行する、請求項 11 に記載のコンピュータ・システム。

【請求項 14】

前記第 1 のオペランドは、前記マシン命令により指定されるレジスタの内容を含み、前記内容はユーザ指定のものであり、前記第 2 のオペランドは、前記マシン命令のインデックス・フィールドにより指定されるレジスタの内容、前記マシン命令のベース・フィールドにより指定されるレジスタの内容、及び少なくとも 1 つの変位フィールドの内容の結合から形成される、請求項 10 に記載のコンピュータ・システム。

【請求項 15】

前記トランザクシヨンの制約なしトランザクシヨンのネスト化における制約なしトランザクシヨンの実行モードである、請求項 10 に記載のコンピュータ・システム。

【請求項 16】

コンピューティング環境内で命令を実行するための方法であって、前記方法は、

プロセッサにより、実行のためのマシン命令を取得することであって、前記マシン命令は、コンピュータ・アーキテクチャによるコンピュータ実行のために定められたものであり、且つ、

非トランザクシヨンのストア操作を指定するためのオペコードと、

第 1 のオペランドと、

前記第 1 のオペランドの位置を指し示すための第 2 のオペランドと、

を含む、取得することと、

前記プロセッサにより、前記マシン命令を実行することと、

を含み、前記実行することは、

前記第 1 のオペランドを、前記第 2 のオペランドにより指定された位置に非トランザクシヨンの配置することを含み、前記第 2 のオペランドに格納された情報は、前記マシン命令と関連したトランザクシヨンのアポートにもかかわらず保持され、前記非トランザクシヨンの配置することは前記プロセッサのトランザクシヨンの実行モードの終了まで遅延される、方法。

【請求項 17】

前記プロセッサがトランザクシヨンの実行モードにあるかどうかを判断することと、

前記プロセッサが前記トランザクション実行モードにあることに基づいて、前記トランザクションが制約付きトランザクションであるか又は制約なしトランザクションであるかを判断することと、

前記トランザクションが制約なしトランザクションであることに基づいて、前記マシン命令の実行を続行することと、  
をさらに含む、請求項 16 に記載の方法。

## 【 国際調査報告 】

INTERNATIONAL SEARCH REPORT		International application No. PCT/IB2012/056625
<b>A. CLASSIFICATION OF SUBJECT MATTER</b> IPC(8) - G06F 9/46 (2013.01) USPC - 714/15 According to International Patent Classification (IPC) or to both national classification and IPC		
<b>B. FIELDS SEARCHED</b> Minimum documentation searched (classification system followed by classification symbols) IPC(8) - G06F 9/46, G06F 12/00, G06F 12/08, G06F 9/30, G06F 11/00 (2013.01) USPC - 711/100, 159, 714/15, 16, 17 Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched CPC - G06F 9/46, G06F 12/00, G06F 12/08, G06F 9/30, G06F 11/00 (2013.01) Electronic data base consulted during the international search (name of data base and, where practicable, search terms used) PatBase, Google Patents, Proquest Dissertations & Theses: The Sciences and Engineering Collection; ProQuest Technology Collection, IEEE Xplore Digital Library, Google Scholar		
<b>C. DOCUMENTS CONSIDERED TO BE RELEVANT</b>		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2009/0172317 A1 (SAHA et al) 02 July 2009 (02.07.2009) entire document	1-20
A	US 2011/0191545 A1 (MILLER et al) 04 August 2011 (04.08.2011) entire document	1-20
A	US 2011/0283096 A1 ( ABERNATHY et al) 17 November 2011 (17.11.2011) entire document	1-20
A	US 2011/0209151 A1 (CHUNG et al) 25 August 2011 (25.08.2011) entire document	1-20
A	US 4,454,578 A (MATSUMOTO et al) 12 June 1984 (12.06.1984) entire document	1-20
A	US 2011/0145637 A1 (GRAY et al) 16 June 2011 (16.06.2011) entire document	1-20
<input type="checkbox"/> Further documents are listed in the continuation of Box C. <input type="checkbox"/>		
* Special categories of cited documents: "A" document defining the general state of the art which is not considered to be of particular relevance "E" earlier application or patent but published on or after the international filing date "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) "O" document referring to an oral disclosure, use, exhibition or other means "P" document published prior to the international filing date but later than the priority date claimed "T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art "&" document member of the same patent family		
Date of the actual completion of the international search 06 March 2013		Date of mailing of the international search report <b>19 MAR 2013</b>
Name and mailing address of the ISA/US Mail Stop PCT, Attn: ISA/US, Commissioner for Patents P.O. Box 1450, Alexandria, Virginia 22313-1450 Facsimile No. 571-273-3201		Authorized officer: Blaine R. Copenheaver PCT Helpdesk: 571-272-4300 PCT OSP: 571-272-7774

Form PCT/ISA/210 (second sheet) (July 2009)

## フロントページの続き

(81)指定国 AP(BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), EA(AM, AZ, BY, KG, KZ, RU, TJ, TM), EP(AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OA(BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG), AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC

(特許庁注：以下のものは登録商標)

1. イーサネット

(72)発明者 グライナー、ダン  
 アメリカ合衆国 9 5 1 4 1 - 1 0 0 3 カリフォルニア州 サンノゼ ベイリー・アベニュー  
 5 5 5

(72)発明者 ヤコビ、クリスチャン  
 アメリカ合衆国 1 2 6 0 1 - 5 4 0 0 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  
 アメリカ合衆国 1 2 6 0 1 - 5 4 0 0 ニューヨーク州 ポキプシー サウス・ロード 2 4 5  
 5

Fターム(参考) 5B033 BE01 DB07