

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第5615384号
(P5615384)

(45) 発行日 平成26年10月29日(2014.10.29)

(24) 登録日 平成26年9月19日(2014.9.19)

(51) Int.Cl.

F I

G06F 9/38 (2006.01)
 G06F 9/46 (2006.01)
 G06F 9/52 (2006.01)
 G06F 9/48 (2006.01)
 G06F 9/34 (2006.01)

G06F 9/38 380A
 G06F 9/46 430
 G06F 9/46 472B
 G06F 9/46 310Z
 G06F 9/34 350A

請求項の数 6 (全 20 頁) 最終頁に続く

(21) 出願番号 特願2012-555083 (P2012-555083)
 (86) (22) 出願日 平成23年2月22日(2011.2.22)
 (65) 公表番号 特表2013-520753 (P2013-520753A)
 (43) 公表日 平成25年6月6日(2013.6.6)
 (86) 国際出願番号 PCT/US2011/025778
 (87) 国際公開番号 W02011/106333
 (87) 国際公開日 平成23年9月1日(2011.9.1)
 審査請求日 平成26年2月24日(2014.2.24)
 (31) 優先権主張番号 12/711,851
 (32) 優先日 平成22年2月24日(2010.2.24)
 (33) 優先権主張国 米国(US)

早期審査対象出願

(73) 特許権者 591016172
 アドバンスト・マイクロ・ディバイズ・
 インコーポレイテッド
 ADVANCED MICRO DEVI
 CES INCORPORATED
 アメリカ合衆国、94088-3453
 カリフォルニア州、サニベイ、ピー・
 オウ・ボックス・3453、ワン・エイ・
 エム・ディ・プレイス、メイル・ストップ
 ・68 (番地なし)
 (74) 代理人 100108833
 弁理士 早川 裕司
 (74) 代理人 100111615
 弁理士 佐野 良太

最終頁に続く

(54) 【発明の名称】 ハードウェアトランザクショナルメモリにおける自動サスペンド及び再開

(57) 【特許請求の範囲】

【請求項1】

プロセッサを備える装置であって、

前記プロセッサは、複数の投機的メモリアクセス動作をトランザクショナルモードで単一のアトミックトランザクションとして実行するように構成されており、

前記プロセッサは、前記単一のアトミックトランザクションとして実行するに際して、前記投機的メモリアクセス動作によってアクセスされるデータを投機的データであるとしてマークするものであり、

前記プロセッサは、

前記プロセッサによって実行中のスレッドにおける制御転送を検出すること及び前記プロセッサが所与の特権レベルで実行中であることを検出することによって暗黙サスペンド条件が満たされていることを検出するように構成される検出ユニットと、

前記暗黙サスペンド条件が満たされている旨の表示を受信することに応答して前記トランザクショナルモードの実行をサスペンドするように構成されるトランザクションサスペンションユニットと、を備え、

前記トランザクショナルモードの実行がサスペンドされている間に実行されたメモリ動作によってデータがアクセスされるときに、前記プロセッサは、アクセスされる前記データについては、投機的であるとマークすることはせず、

前記プロセッサは、前記トランザクショナルモードがサスペンドされている間に前記単一のアトミックトランザクションをアボートする条件が満たされるか否かを検出するよう

10

20

に構成されており、

前記プロセッサは、前記トランザクショナルモードがサスペンドされている間に前記単一のアトミックトランザクションをアボートする条件が満たされていることを検出した場合には、前記投機的データを廃棄することを含めて、前記単一のアトミックトランザクションをアボートするように構成されている、装置。

【請求項 2】

前記プロセッサは、前記トランザクショナルモードがサスペンドされている間に前記単一のアトミックトランザクションをアボートする条件が満たされていることを検出した場合には、前記トランザクショナルモードを再開する条件が満たされていることを検出した後にのみ、前記単一のアトミックトランザクションをアボートするように構成されている、請求項 1 の装置。

10

【請求項 3】

前記プロセッサは、前記トランザクショナルモードの実行をサスペンドする際に、サスペンドフラグをセットするように構成されており、

前記プロセッサは、前記トランザクショナルモードがサスペンドされている間に前記単一のアトミックトランザクションをアボートする条件が満たされていることを検出した場合に、アボートフラグをセットするように構成されている、請求項 2 の装置。

【請求項 4】

プロセッサを備える装置であって、

前記プロセッサは、複数の投機的メモリアクセス動作をトランザクショナルモードで単一のアトミックトランザクションとして実行するように構成されており、

20

前記プロセッサは、前記単一のアトミックトランザクションとして実行するに際して、前記投機的メモリアクセス動作によってアクセスされるデータを投機的データであるとしてマークするものであり、

前記プロセッサは、

前記プロセッサによって実行中のスレッドにおける制御転送を検出すること及び前記プロセッサが所与の特権レベルで実行中であることを検出することによって暗黙サスペンド条件が満たされていることを検出するように構成される検出ユニットと、

前記暗黙サスペンド条件が満たされている旨の表示を受信することに応答して前記トランザクショナルモードの実行をサスペンドするように構成されるトランザクションサスペンションユニットと、を備え、

30

前記トランザクショナルモードの実行がサスペンドされている間に実行されたメモリ動作によってデータがアクセスされるときに、前記プロセッサは、アクセスされる前記データについては、投機的であるとマークすることはせず、

前記検出ユニットは、前記制御転送からのリターンに対応するリターン制御転送を検出することによるのを含めて、前記トランザクショナルモードを再開するための条件が満たされるか否かを検出するように構成されており、

前記プロセッサは、前記トランザクショナルモードを再開するための条件が満たされていることが検出されると、前記投機的メモリアクセス動作によってアクセスされるデータを投機的データであるとしてマークすることを再開するように構成されている、装置。

40

【請求項 5】

前記トランザクショナルモードがサスペンドされている間に前記単一のアトミックトランザクションをアボートする条件が満たされた場合に、前記トランザクションサスペンションユニットは、前記トランザクショナルモードを再開するための条件が満たされていることが検出された後に、前記投機的データを廃棄することを含めて、前記単一のアトミックトランザクションをアボートするように構成されている、請求項 4 の装置。

【請求項 6】

前記単一のアトミックトランザクションが再開された後に、前記単一のアトミックトランザクションをコミットする際には、前記プロセッサは、前記単一のアトミックトランザクションをサスペンドする前に投機的であるとマークされた投機的データと、前記単一の

50

アトミックトランザクションを再開した後に投機的であるとマークされた投機的データの両方を、非投機的であるとマークするように構成されている、請求項5の装置。

【発明の詳細な説明】

【背景技術】

【0001】

ハードウェアトランザクショナルメモリ(Hardware Transactional Memory)(HTM)は、並列プログラミングをサポートするためのコンピュータアーキテクチャにおけるメカニズムである。HTMによると、プログラマは一連の命令をトランザクションとして単に宣言すればよく、そしてHTMシステムは、トランザクションにおける命令がアトミックで且つ隔離された方法で実行されることを保証する。アトミック性(atomicity)は、トランザクションの全ての命令が他の全ての同時スレッドの実行に対して単一のアトミックブロックとして実行されることを意味する。隔離は、トランザクションが完了するまではトランザクションの中間結果が残りのシステムにさらされないことを意味する。HTMシステムは、トランザクションが対立しない限りはそれらが並列に動作することを可能にし得る。2つのトランザクションは、それら両方が同じメモリ区域にアクセスする場合に対立することがあり、この場合2つのトランザクションのいずれか一方が当該メモリ区域に書き込む。

【0002】

既存の多くのHTM設計は、プログラム制御転送の存在下、例えばシステムコール、例外、割り込み、信号送信、又は他のイベントに応答するオペレーティングシステムへの転送の存在下で、正確性及び/又はセキュリティの問題を提示する。伝統的に、そのようなプログラム制御転送は、HTMシステムに対して透過的に生じるであろう。そのような転送がトランザクションの実行中に生じると、オペレーティングシステムコードはトランザクションの一部として実行されるであろう。これにより正確性及び/又はセキュリティの問題が生じ得る。幾つかの従来のシステムにおいては、トランザクションがオペレーティングシステムにおける実行中に停止される(aborted)と、予期しない副次的影響が生じることがある。

【0003】

1つの例において、TCP/IPネットワークデバイスドライバを考える。ネットワークからパケットを受け取ると、割り込みがトリガーされてデバイスドライバコードを実行する。プロセッサはアプリケーショントランザクションを実行中に割り込みを受けたものと仮定する。プログラム制御はデバイスドライバへと転送されるであろうし、また従来のHTMシステムにおいては、従来のHTM設計は単純にトランザクションの開始及び終了の間の全ての命令がトランザクションの一部であるとみなすので、ドライバはトランザクションの一部として動作するであろう。

【発明の概要】

【発明が解決しようとする課題】

【0004】

この挙動は少なくとも2つの問題を生じさせる可能性がある。第一に、デバイスドライバコード実行中にトランザクションが停止されると、システム障害が発生することがある。例えば、デバイスドライバが、ネットワークインタフェースハードウェアを管理するために、メモリマッピングされたレジスタに書き込むように構成されており、且つレジスタ値を変更している途中に停止されると、インタフェースハードウェアは矛盾したままの状態になり得るので、ネットワークシステムの完全性を壊してしまう。また、上述した例では、デバイスドライバがトランザクションの一部として実行されている間に又はその後でトランザクションを停止することは、ネットワーク接続がパケットを失うことにつながる。例えば、ドライバコードがパケットを処理した後、ドライバは受信確認メッセージ(acknowledge message)をパケット送信機へ送る。このとき送信機は、再送信の必要性はないものとして、パケットを削除するであろう。しかし、トランザクションが後で停止されると、パケットを受信機で記憶するために使用されたメモリ書き込み動作に戻されるである

うし、そしてパケットは恒久的に失われているであろうから、TCP/IPシステムの信頼性のある通信保証を阻害することになる。

【 0 0 0 5 】

第二に、セキュリティ問題は、トランザクションの一部としてオペレーティングシステムコードに透過的にジャンプする別の副次的影響である。多くの最新のプロセッサは、オペレーティングシステムとアプリケーションコード実行を分離するために、セキュリティ特徴をサポートする。例えばx86アーキテクチャは、システムコールの境界(boundary of system calls)でコードセグメントセクタを切り換えることによって、オペレーティングシステム及びアプリケーションが異なるコードセグメント及び特権レベル(例えばユーザレベル特権対カーネルレベル特権(kernel-level privilege))を使用することを可能にしている。しかし、従来のHTMシステムは、多くの場合、コードセグメントセクタ等のプロセッサ状態の何らかの部分をチェックポイントにしないし管理しない。ユーザレベルコードに由来するトランザクションがオペレーティングシステムへのシステムコールを実行すると、特権レベルが高くなる。トランザクションが停止すると、従来のHTMは、特権レベルを低いレベルに戻すようには構成されないであろう。従って、結果としてセキュリティリークがもたらされることになる。悪意のあるプログラムは、トランザクションの停止の後にこのセキュリティホールについてオペレーティングシステム特権レベルを得ることが可能になり得る。

【課題を解決するための手段】

【 0 0 0 6 】

多重処理コアによって共有されるメモリにアクセスすると共にメモリアクセス動作をトランザクショナルモードで単一のアトミックトランザクションとして実行するように構成されるコンピュータプロセッサのための装置及び方法が開示される。装置は、暗黙サスペンド条件(implicit suspend condition)を決定することに応答してトランザクショナルモードをサスペンドするように構成される。トランザクションを実行することの一部として、プロセッサは、投機的メモリアクセス動作(speculative memory access operations)によってアクセスされるデータを投機的データであるとしてマークする。暗黙サスペンド条件を決定すること(実行中のスレッドにおける例えばオペレーティングシステムへの制御転送を検出することによるものを含む)に応答して、プロセッサはトランザクショナルモードの実行をサスペンドし、このことはサスペンドフラグをセットすることと投機的データのマーキングをサスペンドすることを含む。後で、プロセッサが再開条件(resumption condition)を検出した場合(例えば制御転送からのリターンに対応するリターン制御転送を検出することによる)には、プロセッサは、投機的データのマーキングを再開すること(resuming)によって、トランザクショナルモードの実行を再開する。

【 0 0 0 7 】

幾つかの実施形態においては、プロセッサは、トランザクションがサスペンドされている間にアトミックトランザクションが失敗したことを検出するように更に構成されてよい。幾つかの実施形態においては、プロセッサは、停止フラグをセットすると共にトランザクションを再開する試みがなされるときに停止フラグをチェックすることによって、サスペンドされたトランザクションが失敗したのを検出することに応答してよい。停止フラグがセットされているとチェックが決定した場合には、プロセッサはトランザクションを停止する。幾つかの実施形態においては、プロセッサは、停止フラグ及び/又はサスペンドフラグの値を実行コンテキストの一部としてセーブ/リストアするように構成されてよい。

【図面の簡単な説明】

【 0 0 0 8 】

【図1】図1は幾つかの実施形態に従いサスペンド/再開能力を伴うHTMメカニズムを実装するように構成されるコンピュータシステムを示すブロック図である。

【 0 0 0 9 】

【図2】図2は幾つかの実施形態に従いメモリトランザクションを実行しながらトランザ

10

20

30

40

50

クショナルモードの実行をサスペンドするための方法を示すフロー図である。

【 0 0 1 0 】

【図 3】図 3 はトランザクションの間にトランザクショナルモードの実行をサスペンドすることによるものを含め H T M を用いたアトミックトランザクションを実行するための方法を示すフロー図である。

【 0 0 1 1 】

【図 4】図 4 は幾つかの実施形態に従いサスペンドされたトランザクショナルモードの間に検出される停止条件を処理するための方法を示すフロー図である。

【 0 0 1 2 】

【図 5】図 5 は幾つかの実施形態に従い自動サスペンド / 再開能力を伴う H T M を実装するための特定の方法を示す図である。

10

【 0 0 1 3 】

【図 6】図 6 は幾つかの実施形態に従いサスペンションが実行され得る種々の制御転送をリストアップした表である。

【 0 0 1 4 】

【図 7】図 7 は幾つかの実施形態に従いここに説明される自動サスペンド / 再開機能を伴うハードウェアトランザクショナルメモリシステムを実装するように構成されるコンピュータシステムを示す図である。

【 0 0 1 5 】

ここで用いられる任意の表題は、組織化の目的だけのものであり、明細書の範囲又は特許請求の範囲を限定することを意図されるものではない。ここで用いられる「～してよい、～あってよい、～し得る、～することができる、～場合がある、～ことがある(may)」の語は、必須の意味(即ち「～なければならない」を意味する)ではなく許容の意味(即ち可能性があることを意味する)で用いられている。同様に、「含む(include, includes)」及び「含んでいる(including)」の語は、含んでいることを意味するが、これには限定されない。

20

【発明を実施するための形態】

【 0 0 1 6 】

従来の H T M 設計は、プログラム制御転送の存在下で正確性及び / 又はセキュリティの問題を提示する。ここで用いられる「制御転送」の用語は、プログラムフロー、例えば何らかの実行中ソフトウェアプログラムから何らかの監視プログラム又はエラー若しくは割り込みのハンドラーへのプログラムフロー及び逆のフロー、における変化を信号として送り、生じさせ又は表示する命令又はイベントを参照する。幾つかの実施形態においては、監視プログラムはオペレーティングシステムであってよい。種々の実施形態において、異なるオペレーティングシステムが用いられてよく(例えばLinux(登録商標)、Windows(登録商標)等、各オペレーティングシステムは、スケジューリングタスク、実行アプリケーション及び周辺機器制御、等のコンピュータの基本機能をサポートするソフトウェアとして実装されてよい。幾つかの実施形態においては、オペレーティングシステムは、基本機能を実装するためのカーネル(kernel)を含み得る。種々の実施形態において、オペレーティングシステムのカーネルはモノリシックであってよく、あるいはマイクロカーネルであってよい。

30

40

【 0 0 1 7 】

幾つかの実施形態においては、1つ以上のオペレーティングシステムは仮想であってよい(例えばオペレーティングシステムは仮想マシン内の「ゲスト」OSであってよい)。そのような実施形態においては、仮想オペレーティングシステムを展開するために用いられるハイパーバイザ(hypervisor)又は別のランタイムシステムへの制御転送は、プログラムと仮想オペレーティングシステムの間でプログラム制御を転送してよい。ここに説明される実施形態は、オペレーティングシステムからの / への転送制御を参照するが、実施形態は、仮想オペレーティングシステム、ハイパーバイザ、及び / 又は任意の他のランタイムシステムからの / への転送を含むように適合させられ得ることが理解されるべきである

50

。従って、特に断りのないかぎり、「制御転送」は、何らかの実行中コードとそれが実行中のオペレーティングシステムとの間でのプログラムフローの変化には限定されない。

【 0 0 1 8 】

種々の実施形態において、制御転送は、種々のイベント、例えばオペレーティングシステムへのシステムコールを実行中のプログラム、ソフトウェア及び／若しくはハードウェアの例外、ハードウェア割り込み、ソフトウェア信号、又は他のイベント、を検出することに応答して行われてよい。アプリケーションからオペレーティングシステムへの各制御転送は、対応するリターン制御転送を有してよく、リターン制御転送は、元の（サスペンド中の）制御転送が生じたプログラムへ制御をリターンする。例えば、プログラムがシステムコールを起動して(invokes)オペレーティングシステムへの転送を制御する場合、オペレーティングシステムは、何らかの機能を実行し次いでシステムコール起動直後に起動中のプログラム内の点へ制御をリターンすることによって、応答してよい。

10

【 0 0 1 9 】

種々の実施形態において、ハードウェアトランザクショナルメモリシステム(HTM)は、複数のメモリアクセス動作を単一のアトミックトランザクションとして実行するための機能を提供してよい。幾つかの実施形態においては、プログラムは、トランザクショナルモードの実行を開始するための開始命令及びトランザクショナルモードの実行を終了するためのそれに続く終了命令を実行してよい。HTMは、トランザクショナルモード(即ち開始命令及び終了命令の間)においてアプリケーションによって実行される幾つかの又は全てのメモリ動作が、システムにおける他のスレッド/プロセッサの実行に対してアトミックに(atomically)行われるのを確実にするように構成されてよい。

20

【 0 0 2 0 】

アトミックな実行を確実にするために、HTMは、トランザクションのメモリアクセス動作を、トランザクションが成功裏にコミットされるまで投機的な(speculative)ものとして取り扱うと共に停止条件をモニタリングするように構成されてよい。例えばHTMは、トランザクショナルモードの実行の間に投機的命令によって投機的なものとして書き込まれた又は読み出されたデータをマークしてよく、またトランザクションが成功裏に終了した場合(即ち停止条件が検出されなかった場合)には、HTMは、当該データを非投機的なものとしてマークしてよく且つ/又はそれを共有メモリ内へ組み込んでよい。一方、停止条件が検出された場合には、HTMは、投機的データをドロップ/無効化し且つ/又は投機的データ書き込みの影響を共有メモリへ後退させるように構成されてよい。

30

【 0 0 2 1 】

種々の実施形態に従うと、ハードウェアメカニズム(例えばプロセッサ上の)は、アクティブトランザクション内でプログラム制御転送を検出することによるものを含めサスペンション条件(suspension condition)を自動的に検出すると共にトランザクショナル実行モードをサスペンドすることによって応答するように構成されてよい。幾つかの実施形態においては、HTMは、次いで対応するリターン制御転送を検出して、必要であればトランザクショナルモードの実行を再開するように構成されてよい。

【 0 0 2 2 】

幾つかの実施形態においては、トランザクショナル実行モードがサスペンドされている間に、プロセッサは、メモリ動作によってアクセスされるデータを投機的であるとはマークしない。従ってそのような実施形態において、トランザクションが停止された場合、トランザクショナルモードがサスペンドされていた間に実行されたメモリアクセス動作は、後退させられないであろう。

40

【 0 0 2 3 】

幾つかの実施形態において、トランザクショナル実行がサスペンドされている間に停止条件が検出された場合、HTMは、トランザクショナルモードが再開した後までトランザクションを停止するのを遅らせるように構成されてよい。例えば、停止条件(例えばデータ衝突によって生じる)を検出することに応答して、HTMは、停止フラグをセットし、そしてサスペンドされたトランザクショナルモードで実行を継続してよい。トランザクシ

50

ヨナル実行モードを再開する場合、H T Mは、停止フラグをチェックし、フラグがセットされている場合には停止を行ってよい。

【 0 0 2 4 】

種々の実施形態において、トランザクショナル実行モードのサスペンション / 再開は、ソフトウェア実行に対して透過的な (transparent) 方法で H T Mによって行われてよい。即ち、H T Mは、サスペンド / 再開条件を自動的に検出すると共にトランザクショナル実行モードをサスペンド / 再開するように構成されているので、ソフトウェア (例えばプログラム及び / 又はオペレーティングシステム) は、トランザクショナル実行モードを適切なタイミングでサスペンドし又は再開することを H T Mに指示する特別な命令を実行するように修正される必要はない。

10

【 0 0 2 5 】

図 1 は幾つかの実施形態に従いサスペンド / 再開能力を伴う H T Mメカニズムを実装するように構成されるコンピュータシステムを示すブロック図である。図 1 において、システム 1 0 0 (幾つかの実施形態においてはトランザクションメモリシステムと称されることがある) は、コンピュータプロセッサ 1 0 2、共有メモリ 1 4 0 及び他のプロセッサ 1 5 0 を備える。これらの要素はバス 1 6 0 によって接続されていてよく、バス 1 6 0 は、ブロードキャスト及び / 又は点对点を含む種々の相互接続であってよい。種々の実施形態において、他のプロセッサ 1 5 0 は、共有メモリ 1 4 0 をコンピュータプロセッサ 1 0 2 と共有する 1 つ以上のプロセッサを含んでいてよい。幾つかの実施形態においては、プロセッサ 1 5 0 の各々は、コンピュータプロセッサ 1 0 2 として構成され得る。

20

【 0 0 2 6 】

図示される実施形態に従うと、コンピュータプロセッサ 1 0 2 は、上述したようにサスペンド / 再開能力を伴うハードウェアトランザクショナルメモリ機能を実装するための H T M 1 1 0 を含む。種々の実施形態において、H T M 1 1 0 は種々の方法で投機的データをマークしてよい。例えば H T M 1 1 0 は、データキャッシュ 1 3 0 等のデータキャッシュ内に投機的データを記憶すると共にマークすることによるものを含めキャッシュベースの投機的データバッファを用いてよい。他の実施形態においては、投機的データバッファは、他の構造において、例えばロードキュー、ストアキュー、又はロード / ストアキューにおいて実装されてよい。種々の実施形態において、各プロセッサは、多重の協働する投機的データバッファを実装してよい (例えばデータキャッシュ 1 3 0 内に及びロード / ストアキュー内に)。

30

【 0 0 2 7 】

システム 1 0 0 において、H T M 1 1 0 は、制御転送検出ユニット 1 1 2、トランザクションサスペンションユニット 1 1 4、衝突検出ユニット 1 1 6 及びオーバフロー検出ユニット 1 1 8 を含む種々のハードウェアユニットを含む。種々の実施形態において、これらの要素のうちの種々の要素が全部又は一部において組み合わせられてよく、あるいは更なる下位要素へと分解されてよい。

【 0 0 2 8 】

幾つかの実施形態においては、制御転送検出ユニット 1 1 2 はサスペンド条件を決定するように構成されてよい。特定の実施形態においては、このサスペンド条件は「暗黙的な (implicit)」サスペンド条件であってよく、暗黙サスペンド条件は、明示的な命令 (例えば「ポーズ (pause)」又は「サスペンド」) の存在以外の基準の幾つかのセットに基づいてサスペンド条件が決定されることを意味する。幾つかの実施形態においては、ユニット 1 1 2 は、プロセッサ 1 0 2 によって実行中のスレッドにおける制御転送を検出することに部分的に基づいて、サスペンド条件が存在することを決定するように構成されてよい。例えば、制御転送検出ユニットは、システムコール起動によって又はハードウェア割り込み (例えば I / O 割り込み) によって生じる制御転送を検出してよい。

40

【 0 0 2 9 】

種々の実施形態において、サスペンド状態を決定することはまた、プロセッサ 1 0 2 が現在トランザクショナル実行モードにあること、及び / 又はプロセッサがより高い特権レ

50

ベル（例えばカーネルレベル）でよりもむしろより低い特権レベル（例えばアプリケーションレベル）で実行中であること、を確認することを含んでいてよい。例えば幾つかのシステムにおいては、現在の特権レベル(current privilege level) (CPL) はレジスタ（例えばレジスタ120）内に記憶され、そして種々の特権レベルを表示する値を保持することができる。そのようなシステムにおいては、システムは、より高い特権レベルで、種々のメモリ領域、I/Oポート、及び/又は特殊命令、等の保護された資源への実行スレッドアクセスを許可する。例えばx86アーキテクチャにおいては、最高の特権を有するレベル0から最低の特権を有するレベル3までの4つの特権レベルがある。そのようなシステムにおいては、レベル0はカーネル/エグゼクティブのために用いられ、レベル3はアプリケーションプログラムのために用いられる。幾つかの実施形態においては、高められた特権レベルでソフトウェアが命令を実行する必要がある場合（例えばオペレーティングシステムコードを実行する場合）には、ソフトウェアは、1つ以上の命令を実行して特権のレベルを上げてよく（例えばレベル3からレベル0へ）、また高められた特権がもはや必要なくなった場合には、1つ以上の命令を実行して特権レベルを下げてよい（例えばレベル0からレベル3へ）。幾つかのシステムにおいては、オペレーティングシステムコードが、それぞれエントリ及びエグジットに際して特権レベルを上げること及び下げることが可能である。

【0030】

幾つかの実施形態においては、制御転送検出ユニット112は、a) オペレーティングシステムへの制御転送、b) プロセッサがトランザクショナルモードで実行中であること、及びc) 現在の特権レベルが上げられていないこと（例えばレベル3であること）、を検出することに応答してサスペンション条件を検出するように構成されてよい。サスペンド条件を検出することに応答して、制御転送検出ユニット112は、トランザクションサスペンションユニット114にサスペンド条件を通知してよい。それに応答して、トランザクションサスペンションユニット114は、トランザクショナルモードの実行をサスペンドすることによって、例えばレジスタ120内にサスペンドフラグ122をセットすることによって、応答するように構成されてよい。上述したように、トランザクショナルモードがサスペンドされている間、幾つかの実施形態においては、プロセッサは、メモリ動作によってアクセスされるデータの投機的なものとしてのマーキングをサスペンドするように構成される。しかし、トランザクショナルモードがサスペンドされるよりも前に投機的なものとしてマークされたデータは、投機的マークを伴って保持される。従って、特定の実施形態においては、停止が生じると、サスペンションの間に生じたメモリ値への修正は後退させられないであろう一方で、トランザクショナルモードがサスペンドされるよりも前に又はサスペンドされたよりも後に生じた修正は後退させられることになる。

【0031】

幾つかの実施形態においては、制御転送検出ユニット112は、再開条件を決定するように更に構成されてよく、サスペンド条件を決定するために検出された制御転送に対応するリターン制御転送を検出することによるものがこれに含まれる。例えば、サスペンド条件を決定するために検出された制御転送がオペレーティングシステムに対するシステムコールであったとすると、リターン転送コールはシステムコールからのリターンであってよい。

【0032】

幾つかの状況においては、サスペンド条件を最初に生じさせる制御転送（ここでは「サスペンディング」転送と称される）は、サスペンディング制御転送からのリターンが実行されるよりも前に、1つ以上の後続の制御転送によって追従されてよい。この場合、多重リターン制御転送が実行され得るが、トランザクショナルモードは、サスペンディング制御転送に対応するリターン制御転送が実行されるまでに再開されることはないであろう。

【0033】

幾つかの実施形態においては、制御転送検出ユニット112は、トランザクショナルモードの実行を再開することをトランザクションサスペンションユニットに通知することに

10

20

30

40

50

よって、再開条件を検出することに応答するように構成されてよい。幾つかの実施形態においては、トランザクショナルモードを再開することは、サスペンドフラグ 122 をアンセットすること (unsetting) 及び / 又は上述したように停止フラグ 124 をチェックすることを含み得る。

【0034】

図示される実施形態に従うと、HTM 110 は、他のメカニズム及び / 又はユニット、例えば衝突検出ユニット 116、を含み得る。種々の実施形態において、衝突検出ユニット 116 は、種々の停止条件を検出するように構成されてよい。例えば、衝突検出ユニットは、MESI 又は MOESI 等のキャッシュコヒーレンシプロトコルの一部として他のプロセッサ 150 から受け取られるキャッシュコヒーレンシメッセージ (プローブ) をモニタリングするように構成されてよい。1つ以上のそのようなプローブがデータ衝突を示す場合には、衝突検出ユニット 116 は、停止条件が存在すること及びトランザクション試行が停止され得ることを決定する。

【0035】

上述したように、幾つかの実施形態においては、HTM は、サスペンドされたトランザクショナルモードで実行中のトランザクション試行を停止することを、トランザクショナル実行モードが再開されるまで遅らせるように構成されてよい。そのような実施形態においては、サスペンドされたトランザクショナルモードの間に停止条件を検出することに応答して、衝突検出ユニットは、トランザクションをすぐに停止させ且つ / 又は後退させるよりもむしろ、停止フラグ 124 等の停止フラグをセットする (即ち予め定められた値を適切な記憶ロケーションに記憶させる) ように構成されてよい。トランザクショナルモードの実行が再開されると (例えばオペレーティングシステムカーネルから復帰すると)、HTM は、停止フラグ 124 がセットされていることを検出することができ、またトランザクション試行を停止することによって応答してよい。

【0036】

種々の他のハードウェアが他の停止条件を検出してよい。例えばオーバフロー検出ユニット 118 は、投機的バッファオーバフロー条件の形態にある停止条件を検出するように構成されてよい。バッファオーバフロー条件は、新たな投機的データをバッファリングするのに容量が十分でない投機的バッファ (例えばデータキャッシュ 130 内に実装されるもの) 内にデータの所与の部分がバッファリングされることが、トランザクションを実行するために必要な場合に、検出され得る。ここで用いられる「投機的バッファオーバフロー」は、検出されている最中のオーバフロー条件を参照し、投機的データが実際にはバッファから退去させられている状況は参照しない。

【0037】

データ衝突と同様に、サスペンドされたトランザクショナルモードでプロセッサが実行中にオーバフロー検出ユニット 118 がオーバフロー条件を検出した場合には、停止フラグ 124 がセットされてよく、これにより、HTM 110 は、トランザクショナル実行モードが再開されるときにトランザクション試行を停止する / 後退させることができる。

【0038】

図 2 は幾つかの実施形態に従いメモリトランザクションを実行しながらトランザクショナルモードの実行をサスペンドするための方法を示すフロー図である。方法 200 は、メモリトランザクションを利用するように構成されるコンピュータプログラムを実行する過程において HTM (例えば図 1 の HTM 110) によって実行されてよい。

【0039】

図示される実施形態に従うと、方法 200 は、210 に示されるようにプロセッサがトランザクショナルモードでの実行を開始するときに始まる。幾つかの実施形態においては、このことは、トランザクショナル実行を開始するように設計された 1つ以上の命令を実行しているソフトウェアプログラムに応答して行われてよい。210 に示されるようにトランザクショナル実行モードを開始することは、例えばシステム 100 のレジスタ 120 内にあるような 1つ以上のフラグをセットすることを含み得る。

【 0 0 4 0 】

一旦プロセッサがトランザクショナルモードで実行中になると、220に示されるように、プロセッサは、投機的メモリアクセス動作によってアクセスされるデータを投機的なものとしてマークするように構成されてよい。幾つかの実施形態においては、トランザクショナルモードで実行される全てのメモリアクセス動作は投機的であると考えられてよく、従って各そのようなメモリアクセス動作によってアクセスされるデータは投機的であるとしてマークされてよい。他の実施形態においては、トランザクション内のメモリアクセス動作の一部が、投機的であるとして明示的に指定されてよい。幾つかのそのような実施形態においては、これらの明示的に投機的な命令の各々によってアクセスされるデータのみが、220に示されるように投機的であるとしてマークされてよい。

10

【 0 0 4 1 】

種々の実施形態において、データを投機的であるとしてマークする処理は、H T Mによって使用されている投機的バッファの種類に依存するであろう。例えば、投機的データをデータキャッシュ内にバッファリングするH T Mに対しては、データを投機的であるとしてマークすることは、当該データのコピーをデータキャッシュ内に記憶することと、1つ以上の投機的ビットを用いて投機的データを含むキャッシュのキャッシュブロックをマークすることと、を含んでいてよい。投機的ビットをマークすることはまた、ロード/ストアキュー等の他の構造によって実装されるバッファ内で投機的データをマークするために用いられてもよい。

【 0 0 4 2 】

20

図示される実施形態に従うと、トランザクショナル実行の間におけるどこかの時点で、230に示されるように、H T M（又はその何らかの要素、例えば制御転送検出ユニット112）が、制御転送を検出することによるのを含めサスペンド条件を決定する。幾つかの実施形態においては、サスペンド条件は、プログラム制御がまさにオペレーティングシステムカーネルへ転送されようとしていることを示すものであってよい。例えば幾つかの実施形態においては、サスペンド条件は、作成されているシステムコールと、何らかの所与のアプリケーションレベルに対応するプロセッサの現在の特権レベル（例えば幾つかの実施形態においてこの所与の特権レベルは3（ユーザ又はアプリケーションレベル）であってよい）と、に応答して検出され得る。別の例においては、サスペンド条件は、ハードウェア割り込みと、C P Lがレベル3等のアプリケーションレベル特権を示していることと、を検出することに応答して検出され得る。従ってそのような実施形態においては、制御転送がトランザクショナル実行の間に検出されると、H T MはC P Lをチェックして制御転送がアプリケーションからオペレーティングシステムカーネルへのものであるかどうかを決定してよい。そうである場合には、サスペンド条件が存在すると決定され得る。

30

【 0 0 4 3 】

方法200においては、サスペンド条件を検出することに応答して、H T Mは、240に示されるようにトランザクショナル実行をサスペンドしてよい。トランザクショナル実行をサスペンドすることは、サスペンドフラグ（例えば何らかの記憶ロケーション内に記憶される値を参照するサスペンドフラグ122）をセットすることを含んでいてよく、サスペンドフラグは、トランザクショナル実行モードがサスペンドされる旨の信号をH T Mへ送ってよい。

40

【 0 0 4 4 】

一旦トランザクショナル実行モードがサスペンドされると、プロセッサは、250に示されるように、アクセスされたデータを投機的であるとしてマークすることなしにメモリ動作を実行してよい。即ち、制御が一旦オペレーティングシステムへ転送されると、トランザクショナル実行モードはサスペンドされ、そしてオペレーティングシステムによって実行されるメモリ動作は、投機的であるとして取り扱われることはなく、それにより停止の場合に後退させられることはない。上述したように、幾つかの実施形態においては、停止条件が検出されたとしても、プロセッサはサスペンドされたモードにある間はトランザクショナル試行を停止しなくてよい。そのような実施形態においては、プロセッサは、停止

50

フラグ（何らかの記憶ロケーションに記憶された値）をセットすること、及び一旦トランザクショナル実行モードが再開されたならばトランザクションを停止すること（後退させることを含む）、によって停止条件を検出することに応答してよい。

【0045】

図3はトランザクションの間にトランザクショナルモードの実行をサスペンドすることによるものを含めHTMを用いたアトミックトランザクションを実行するための方法を示すフロー図である。方法は、305に示されるように、トランザクショナル実行モードを開始することによって始まる。前述と同様に、プロセッサはトランザクション開始命令を実行することに応答してこのステップを行ってよい。

【0046】

方法300において、HTMがサスペンド条件を検出した場合（310からの肯定の出口によって示される）には、HTMは、例えば315に示されるようにサスペンドフラグをセットすることによって、トランザクショナルモードをサスペンドしてよい。310においてサスペンド条件を検出することは、図2の230においてサスペンド条件を検出することに類似してよく、またアクティブな（サスペンドされていない）トランザクショナルモードの間におけるオペレーティングシステムへの制御転送を検出することを含んでいてよい。オペレーティングシステムへの制御転送を検出することに応答して、HTMはまた、適切なりターン制御転送が後で識別され得るように、用いられる制御転送の種類を記録してもよい。サスペンド条件が検出されない場合には、310からの否定の出口で示されるように、トランザクションはサスペンドされない。

【0047】

方法300に従うと、プロセッサは次いで320に示されるように次のメモリアクセス命令を実行する。トランザクショナルモードの実行がサスペンドされていない場合（325からの否定の出口によって示される）には、HTMは、330に示されるように、メモリ動作によってアクセスされる任意の投機的データを投機的であるとしてマークする。例えば、動作が読み出し動作である場合には、HTMは、値が読み出されたメモリロケーションがトランザクションの読み出しセットの一部であることを記録してよい。従って、投機的メモリロケーションに記憶された値を別のプロセッサが修正したことを示す無効化プローブが受信されると、プロセッサは停止条件を検出する。

【0048】

上述したように、幾つかの実施形態においてはアクティブトランザクションにおける全てのメモリ動作が投機的であると考えられてよいが、他の実施形態においては、幾つかのデータ及び/又はメモリ動作は投機的であるとして明示的に識別されてよい一方では投機的であるとして明示的に識別されなくてよい。そのような実施形態においては、非投機的データにのみアクセスするメモリ動作は、いかなるデータをも投機的であるとしてマークすることはないであろう。例えば1つのそのような実施形態においては、方法300は、325からの否定的な出口が、既に投機的であるとして明示的にマークされたメモリ動作によってアクセスされるメモリ動作及び/又はデータを更に条件とするように増強されてよい。

【0049】

投機的データバッファを実装するためにキャッシュが用いられる実施形態においては、読み出しデータに対応するキャッシュブロックは、それがトランザクショナル実行モードで読み出されたことを示す「投機的フラグ」を用いてマークされてよい。幾つかの実施形態においては、各キャッシュブロックは、「読み出し投機的フラグ」及び「書き込み投機的フラグ」を備えていてよく、あるいはこれらに関連付けられていてよく、これらはメモリ区域（もしあれば）上で実行された投機的メモリアクセス動作の種類を示す。種々の実施形態において、これらのフラグは組み合わせられてよく又は別々であってよく、またデータキャッシュ、ロード/ストアキュー、及び/又は他のものによって実装されるような種々の投機的バッファにおいて用いられてよい。

【0050】

図示される実施形態に従うと、トランザクションがサスペンドされた場合（３２５からの肯定的な出口によって示される）、投機的データのマーキング（３３０におけるような）はＨＴＭによって行われたい。その代わりにＨＴＭは、３３５にあるように、再開条件が存在するかどうかを決定してよい。幾つかの実施形態においては、再開条件は、オペレーティングシステムからアプリケーションへのリターン制御転送を検出することに対応して検出され得る。幾つかの実施形態においては、そのようなリターン制御転送を検出することは、実行中の（又はまさに実行されようとしている）制御転送が、オペレーティングシステムへの元の（サスペンディング）制御転送に対応するリターンであることを決定することを含んでいてよい。加えて、再開条件を検出することは、プロセッサがレベル０等の特権モード（例えばカーネルレベル）で実行中であることを示すＣＰＬに更に依存して

10

【００５１】

方法３００において、３３５からの肯定の出口によって示されるように再開条件が検出されると、３４０に示されるようにトランザクショナルモードの実行が再開されてよい。トランザクショナルモードの実行を再開することは、サスペンドフラグをアンセットすることにより、投機的データのマーキングが必要に応じて再開すべきであることをＨＴＭに示すことを含んでいてよい。図４を参照して後で論じられるように、トランザクショナルモードを再開することはまた、サスペンションの間に停止条件が検出されたかどうかを決定すること（例えば停止フラグをチェックすることによって）を含み得る。

【００５２】

20

図３に図示される実施形態において、トランザクションが更なる命令を含む場合（３４５からの肯定の出口によって示される）には、プロセッサは方法３００を繰り返してよい（３４５から３１０までのフィードバックループによって示されるように）。そうでない場合には、３４５からの否定の出口によって示されるように、トランザクションは３５０に示されるようにコミットされて(committed)よい。種々の実施形態において、トランザクションをコミットすることは種々のステップを含み得る。例えば、トランザクションをコミットすることの一部として、マークされた投機的データが非投機的であるとしてマークされてよい。幾つかの実施形態においては、トランザクションをコミットすることは、投機的に書き込まれたデータを共有メモリへ書き込むことを含んでいてよく、あるいは投機的に書き込まれたデータをシステム内の他のプロセッサと共有することを含んでいてよい。尚、方法３００は、３１０でサスペンド条件を検出することと３３５で再開条件を検出することとの間に少なくとも１つのメモリアクセスを実行することを含むが、当業者であればこれが必ずしもサスペンドされたトランザクションの全ての場合ではないことを理解するであろう。

30

【００５３】

図４は幾つかの実施形態に従いサスペンドされたトランザクショナルモードの間に検出される停止条件を処理するための方法を示すフロー図である。前述したのと同様に、方法は、４０５に示されるようにトランザクショナル実行モードを開始することと、４１０に示されるようにサスペンド条件を検出することに対応してトランザクショナル実行モードをサスペンドすることと、によって開始する。一旦トランザクショナル実行モードがサスペンドされると、プロセッサは、オペレーティングシステムコードを非トランザクショナルモードで（即ちアクセスされたデータを投機的であるとしてマークすることなしに）実行することを開始してよい。

40

【００５４】

方法４００に従うと、ＨＴＭは、４１５に示されるように、トランザクション試行が失敗したことを示す停止条件を検出してよい。種々の実施形態において、種々の状況が停止条件を引き起こすであろうし、これには、上述したように、別のプロセッサとのデータ衝突、又は投機的バッファオーバーフロー条件が検出されること、が含まれる。例えばデータ衝突は、当該プロセッサによって投機的であるとしてマークされたメモリ区域に別のプロセッサがアクセスし且つそのメモリ区域に記憶されたデータをいずれかのプロセッサが修

50

正した場合に生じ得る。停止条件の別の例は、トランザクションの間にアクセスされる全ての投機的データをバッファリングするのに十分な容量を投機的データバッファ（例えばデータキャッシュ）が有していない場合等の投機的バッファオーバフロー条件が検出されることに応答して検出されてよい。過剰な投機的データ及び／又は非投機的データが同時にバッファリングされることに起因して、バッファが十分な容量を有することができない場合がある。

【 0 0 5 5 】

従来のシステムにおいては、H T Mは、停止条件を検出した場合、即座にトランザクションを停止するはずである。しかし、これにより、オペレーティングシステムソフトウェアの種々の構造が矛盾した状態のままになり得る。対照的に、種々の実施形態におけるH T Mは、サスペンドされたトランザクションを、それが再開するまでは停止することを遅らせることができる。例えば図4の方法400においては、415で停止条件を検出することに応答して、H T Mは420に示されるように停止フラグをセットし、そして425に示されるようにサスペンドされたトランザクショナルモードで実行を継続する。

【 0 0 5 6 】

実行はサスペンドされたトランザクショナルモードでオペレーティングシステムにおいて継続するが、種々の追加的な制御転送が行われてよい。後の時点で、H T Mは、サスペンディング制御転送に整合するリターン制御転送を検出すること及び制御をアプリケーションコードへリターンすることに応答して、再開条件を検出してよい（430に示されるように）。

【 0 0 5 7 】

図示される実施形態に従うと、アプリケーションコードへリターンし且つトランザクショナル実行を再開した後に、H T Mは停止フラグがセットされているかどうかを435で決定してよい。停止フラグがセットされていない場合には、サスペンドされた実行モードの間にトランザクションは失敗しなかったことになり、従ってH T Mは、サスペンドされたフラグをクリアしてよく、またトランザクショナルモードで実行を継続してよい。一方、方法400において420で停止フラグがセットされた場合には、H T Mは停止フラグがセットされていると435で決定することになり、これに応答して440でトランザクションを停止することになる。

【 0 0 5 8 】

上述したように、440でトランザクションを停止することは、投機的データを無効であるとしてマークすることを含んでいてよい。このことは、ここでは投機的データを「ドロップすること(dropping)」と称されることがある。投機的データが一旦ドロップされると、その投機的データは他のプロセッサによるアクセスのための共有メモリには記憶されない。例えば、投機的バッファ内の投機的データが無効であるとしてマークされると、プロセッサは、その後の命令実行に対してこのデータに頼らず、またこれを共有メモリへ転送しない。更に、無効化されたデータをシステム内の他のプロセッサと連携させるようなその後のトランザクショナル停止又はキャッシュコヒーレンシアクションは必要ない。

【 0 0 5 9 】

図5は幾つかの実施形態に従い自動サスペンド／再開能力を伴うH T Mを実装するための特定の方法を示している。方法502は、500に示されるようにトランザクショナル実行モードを開始することによって始まる。実行の間、H T Mは、505に示されるようにオペレーティングシステムカーネルへの制御転送を検出する。プロセッサが510からの肯定的な出口によって示されるようにサスペンドされたトランザクショナルモードで実行中の場合、トランザクショナル実行モードをサスペンドすることに関して追加的なアクションがとられる必要はない。一方、プロセッサが現在サスペンドされた状態にあるモードの実行中でない場合、H T Mは515に示されるように現在の特権レベルをチェックしてよい。特権レベルが515からの否定的な出口によって示されるように上げられている場合（例えばカーネルレベル0又は幾つかの実施形態においては非アプリケーションレベル）には、トランザクショナルモードの実行をサスペンドすることに関して更なるアクシ

ョンがとられる必要はない。

【 0 0 6 0 】

図示される実施形態に従うと、プロセッサがサスペンドされたトランザクショナルモードになく（ 5 1 0 からの否定的な出口）且つ現在の特権レベル（ C P L ）が 5 1 5 からの肯定的な出口によって示されるようにより低い特権レベル（例えばアプリケーションレベル 3）である場合には、 H T M は、制御転送イベントの表示を記録してよく（ 5 2 0 に示される）、またトランザクショナル実行モードをサスペンドしてよい（ 5 2 5 に示される）。

【 0 0 6 1 】

方法 5 0 2 においては、 5 2 5 でトランザクショナル実行モードがサスペンドされた後に、 H T M は 5 3 0 で別の制御転送を検出してよい。 5 3 0 で検出されたこの後続の制御転送が 5 0 5 のサスペンディング制御転送に対応するリターン制御転送である場合（ 5 3 5 からの肯定の出口で示される）には、プロセッサは実行をアプリケーションモードにリターンしてよく、また H T M は 5 4 0 に示されるようにトランザクショナル実行モードを再開（即ちアンサスペンド）してよい。一方、制御転送が 5 0 5 のサスペンディング制御転送に対応するリターン制御転送でない場合には、 H T M は、 5 3 5 から 5 3 0 のフィードバックループによって示されるように、対応するリターン制御転送が発見されるまでトランザクショナル実行モードを継続してよい。場合によっては、最終的にコミットされるまで所与のトランザクションは多重回数サスペンドし及び／又は再開してよい（例えば 5 4 0 から 5 0 5 の随意的なフィードバックループによって示されるように）。

【 0 0 6 2 】

種々の実施形態において、任意の数の制御転送がサスペンディング転送とその対応するリターン制御転送の間に実行されてよい。しかし、幾つかの実施形態においては、サスペンディング転送に対応するリターン制御転送が検出されるとき（ 5 3 5 からの肯定的な出口）だけが H T M 再開トランザクショナルモードであってよい。

【 0 0 6 3 】

図 6 は幾つかの実施形態に従いサスペンションが実行され得る種々の制御転送をリストアップした表である。リストアップされた制御転送の種類は、遠い制御転送 (far control transfers)、システムコール、及び例外／割り込みを含む。各々に対して転送の種類はそれぞれリストアップされたサスペンションイベントによって検出されてよく、また転送からのリターンはそれぞれリストされた再開イベントによって検出されてよい。例えば、オペレーティングシステムへのシステムコールは、 S Y S E N T E R 命令を検出することによって検出されてよく、またマッチングリターン制御転送は S Y S E X I T 命令によって示されることになる。

【 0 0 6 4 】

図 7 は幾つかの実施形態に従いここに説明される自動サスペンド／再開機能を伴うハードウェアトランザクショナルメモリシステムを実装するように構成されるコンピュータシステムを示している。コンピュータシステム 7 0 0 は種々のデバイスの任意のものであってよく、限定はされないが例えば、パーソナルコンピュータシステム、デスクトップコンピュータシステム、ラップトップ若しくはノートブック型のコンピュータ、メインフレームコンピュータシステム、ハンドヘルドコンピュータ、ワークステーション、ネットワークコンピュータ、コンシューマデバイス、アプリケーションサーバ、記憶デバイス、周辺機器デバイス（スイッチ、モデム、ルータ等）、又は一般的に任意の種類のコンピューティングデバイスが含まれる。

【 0 0 6 5 】

コンピュータシステム 7 0 0 は 1 つ以上のプロセッサ 7 5 0 を含んでいてよく、 1 つ以上のプロセッサ 7 5 0 の各々は多重コアを含んでいてよく、多重コアの任意のものは単スレッド又は多重スレッドにされていてよい。図 1 に関しても説明されるように、各プロセッサは、種々のフラグ及びオペランドを記憶するために用いられ得るレジスタ 7 5 2 を備えていてよい。プロセッサ 7 5 0 は、アクセスされたデータをキャッシュし且つ／又は

10

20

30

40

50

投機的データバッファを実装するように構成されるデータキャッシュ754を更に備えていてよい。例えばデータキャッシュ754は、各キャッシュブロックに関連付けられ且つデータを投機的に読み出すこと及び／又は書き込むことのために使用可能な1つ以上のビットを含み得る。プロセッサ750は、制御転送を検出することによるものを含めサスペンド／再開条件を検出するための制御転送検出ユニット756を更に含み得る。図7に従うと、プロセッサ750はまた、ここに説明されるようにトランザクショナル実行モードをサスペンド／再開するためのトランザクションサスペンションユニット758を含み得る。

【0066】

図示される実施形態においては、コンピュータシステム700はまた、オフチップL2又はL3キャッシュ等の1つ以上のオフチップキャッシュ760を含み得る。幾つかの実施形態においては、L2及び／又はL3キャッシュはオンチップで実装されてよい。コンピュータシステム700はまた、永続的記憶デバイス770（例えば光学的記憶装置、磁氣的記憶装置、ハードドライブ、テープドライブ、ソリッドステートメモリ等）を含んでいてよく、永続的記憶デバイス770は、772で示されるようなファイルシステムによって組織化されるデータを記憶してよい。コンピュータシステム700はまた、任意の数のネットワークを介してデータを送信し且つ受信するための780で示されるような1つ以上のネットワークインタフェースを含んでいてよい。コンピュータシステム700はまた、1つ以上のメモリ710（例えば1つ以上のキャッシュ、SRAM、DRAM、RDRAM、EDO-RAM、DDR-7-RAM、SDRAM、ラムバスRAM、EEPROM等）を含んでいてよい。種々の実施形態は、より少ない要素又は図7には図示されない付加的な要素（例えばビデオカード、オーディオカード、追加的なネットワークインタフェース、周辺デバイス、ネットワークインタフェース（ATMインタフェース、イーサネット（登録商標）インタフェース、フレームリレーインタフェース等））を含んでいてよい。

【0067】

1つ以上のプロセッサ750、単一又は複数の記憶装置740、オフチップキャッシュ760、永続的記憶デバイス770、ネットワークインタフェース780、及びシステムメモリ710は、システム相互接続740を介して結合されてよい。システムメモリ710の1つ以上は、プログラム命令720並びに種々のデータ構造及び変数730を含んでいてよい。プログラム命令720は、プラットフォームネイティブバイナリの任意のインタープリタ型言語、例えばJava（登録商標）バイトコード、若しくは任意の他の言語、例えばC/C++、フォートラン等、又はそれらの任意の組み合わせにおいてエンコードされてよい。

【0068】

プログラム命令720は、単一及び／又は多重スレッドの1つ以上のプログラム722を実装するように実行可能なプログラム命令を含んでいてよく、単一及び／又は多重スレッドの1つ以上のプログラム722は、ここに説明されるように命令の開始及びコミットを実行することによるようなアトミックメモリトランザクションを利用してよい。

【0069】

図示される実施形態に従うと、メモリ710は、Windows（登録商標）及び／又はLinux（登録商標）等のオペレーティングシステム724を実装するように実行可能なプログラム命令を備えていてよい。幾つかの実施形態においては、オペレーティングシステム724は、上げられた特権レベルで種々の機能を実行するように構成されるカーネルを含んでいてよい。例えばオペレーティングシステム724は、アプリケーション722が起動し得るシステムコールを定義するアプリケーションプログラミングインタフェース（API）を露出してよい。そのようなシステムコールを起動することに対応して、プロセッサ750は、アプリケーション722によって実行中のアクティブトランザクションを、ここに説明されるようにアプリケーション722及びオペレーティングシステム724に対して実質的に透過的な方法でサスペンドしてよい。

【 0 0 7 0 】

アプリケーション 7 2 2 等のソフトウェアプログラムは、コンピュータプログラム製品又はソフトウェアとして提供されてよく、これらは命令が記憶されたコンピュータ可読記憶媒体を含んでいてよく、命令は種々の実施形態に従う処理を行うようにコンピュータシステム（又は他の電子デバイス）をプログラムするために用いられてよい。コンピュータ可読記憶媒体は、機械（例えばコンピュータ）によって可読な形態（例えばソフトウェア、処理アプリケーション）で情報を記憶するための任意のメカニズムを含み得る。機械可読記憶媒体（持続的機械可読記憶媒体）は、限定はされないが、磁気記憶媒体（例えばフロッピー（登録商標）ディスク）；リードオンリメモリ（ROM）；ランダムアクセスメモリ（RAM）；消去可能プログラム可能メモリ（例えばEPROM及びEEPROM）；フラッシュメモリ；プログラム命令を記憶するのに適する電氣的な又は他の種類の媒体を含み得る。加えて、プログラム命令は、光学的な、音響的な、又は他の形態の伝搬信号（例えば搬送波、赤外線信号、デジタル信号等）を用いて通信されてよい。

10

【 0 0 7 1 】

上述したコンピュータ可読記憶媒体は、プログラムによって読み出される命令を記憶するための 1 つの実施形態において用いられてよく、またシステムプロセッサ 7 5 0 を備えるハードウェアを製造するために直接的に又は間接的に用いられてよい。例えば命令は、ベリログ(Verilog)又はVHDL等の高レベル設計言語(HDL)におけるハードウェア機能の行動レベル又はレジスタ転送レベル(RTL)の記述を記述する 1 つ以上のデータ構造を記述してよい。記述は合成ツールによって読み出されてよく、合成ツールは記述を合成してネットリスト(netlist)を作成することができる。ネットリストは一連のゲート（例えば合成ライブラリにおいて定義される）を備えていてよく、一連のゲートはプロセッサ 7 5 0 の機能を表す。ネットリストは次いで、マスクに適用されるべき幾何学的形状を記述するデータセットを作成するために配置されると共に経路付けられてよい。マスクは次いで、プロセッサ 7 5 0 に対応する単一又は複数の半導体回路を製造するために種々の半導体製造ステップにおいて使用されてよい。代替的には、キャリア媒体 3 0 0 上のデータベースが必要に応じてネットリスト（合成ライブラリを伴う又は伴わない）又はデータセットであってよい。

20

【 0 0 7 2 】

本開示の範囲は、ここで取り上げられている問題の一部又は全部を軽減するしないにかかわらず、ここに開示される（明示的に又は暗黙的に）任意の特徴又は特徴の組み合わせ、あるいはその任意の一般化を含む。従って、本出願（又はそれに対して優先権を主張する出願）の手續きの間に任意のそのような特徴の組み合わせに関して新たな特許請求の範囲が編成されることがある。特に、添付の特許請求の範囲を参照して、従属請求項からの特徴が独立請求項の特徴と組み合わせされることがあり、またそれぞれの独立請求項からの特徴は、添付の特許請求の範囲に列挙される特定の組み合わせにおけるだけでなく任意の適切な方法において組み合わせされることがある。

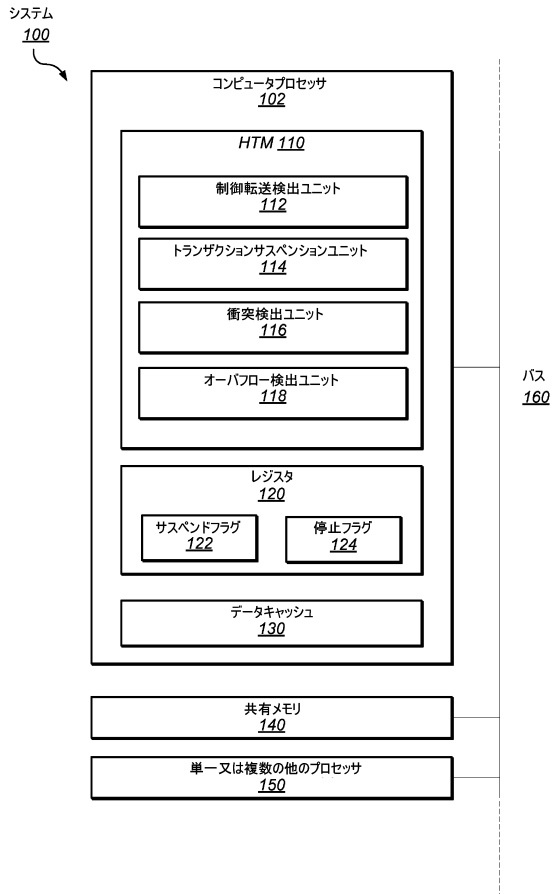
30

【 0 0 7 3 】

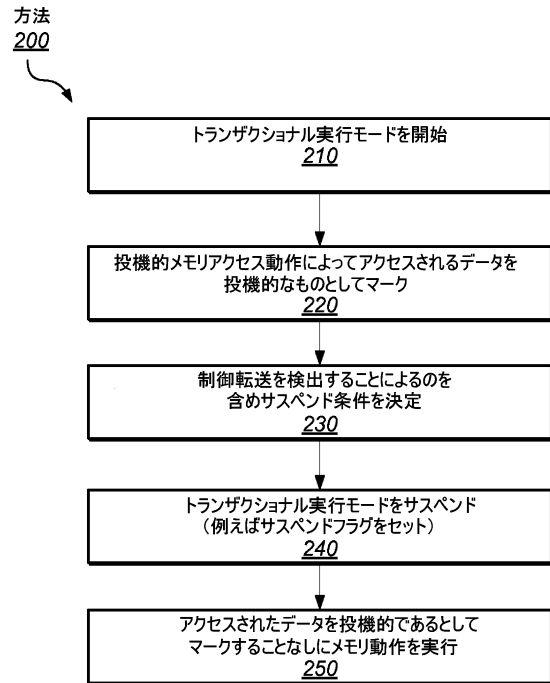
上記実施形態はかなり詳細に説明されてきたが、上記開示が一旦完全に理解されたならば、種々の変形及び修正が当業者にとって明らかなはずである。以下の特許請求の範囲は、全てのそのような変形及び修正を包含することが意図されている。

40

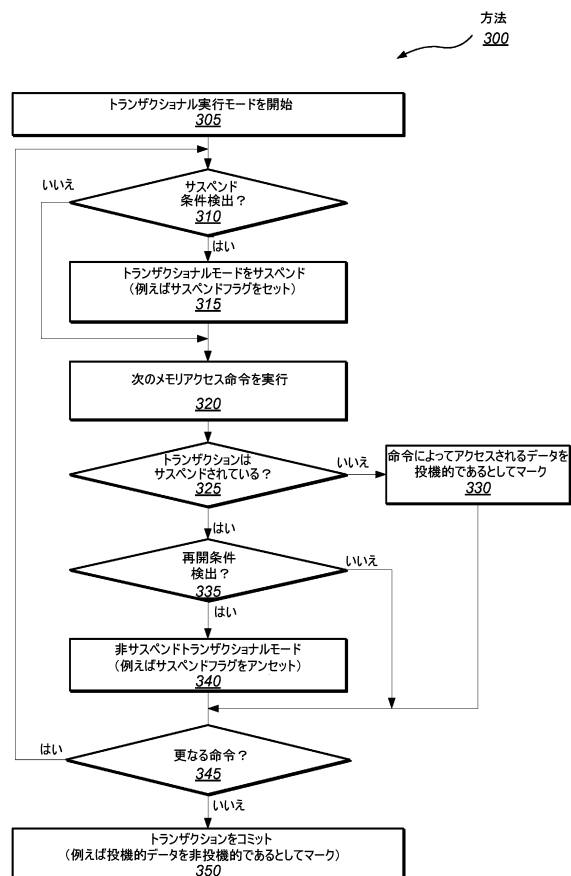
【図 1】



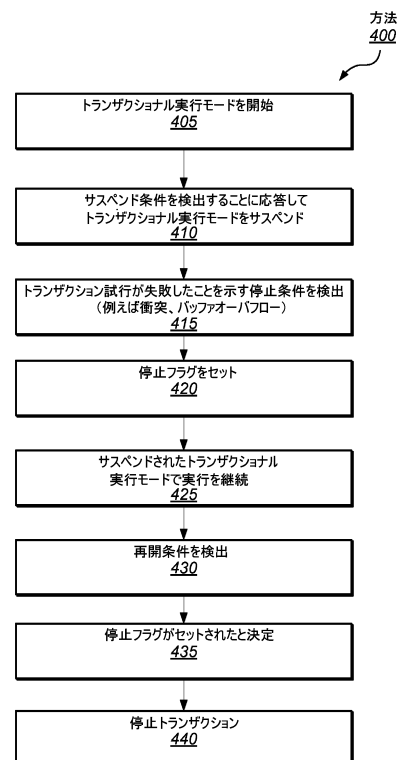
【図 2】



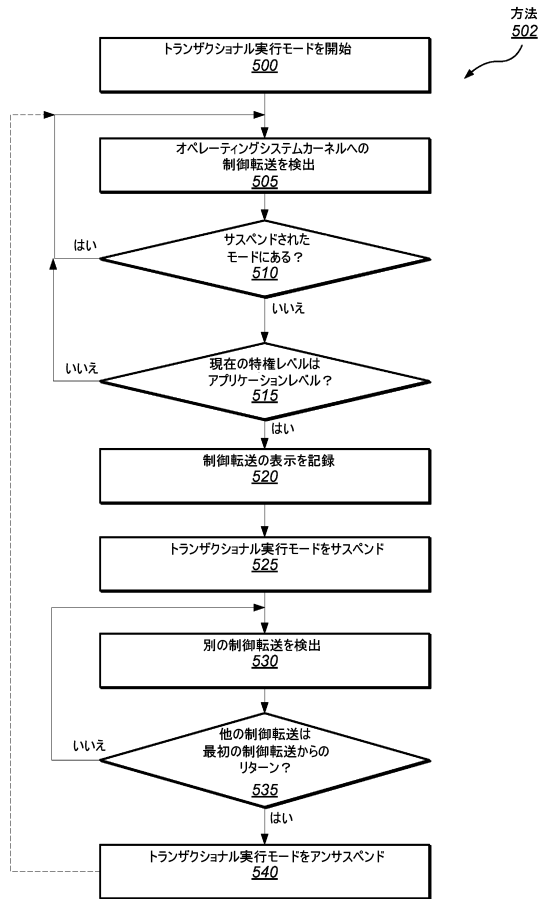
【図 3】



【図 4】



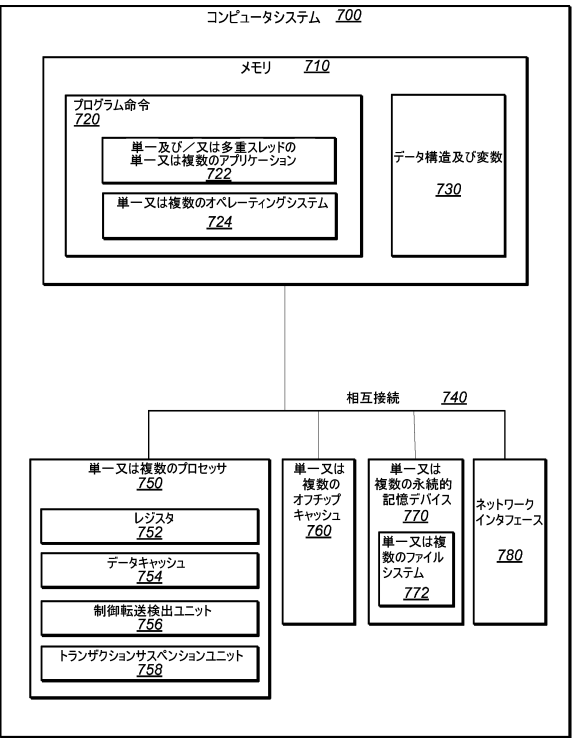
【 図 5 】



【 図 6 】

転送の種類	サスペンションイベント	再開イベント
遠い制御転送	CALL (コール)	RET
システムコール	SYSCALL, SYSEENTER	SYSRET, SYSEXIT
例外／割り込み	INT, INTn, 例外, 割り込み	IRET, RSM

【 図 7 】



フロントページの続き

(51)Int.Cl. F I
G 0 6 F 12/08 (2006.01) G 0 6 F 12/08 5 0 7 Z
 G 0 6 F 12/08 5 8 5

(74)代理人 100162156

弁理士 村雨 圭介

(72)発明者 ジェウン チュン

アメリカ合衆国 9 8 0 0 4 ワシントン州、ベルビュー、ノースイースト 1 0 番 ストリート
 1 1 0 0 0 , # 2 6 3

(72)発明者 デイビッド エス . クリスティ

アメリカ合衆国 7 8 7 3 9 テキサス州、オースティン、ニーダム レーン 6 2 0 1

(72)発明者 ミハエル ピー . ホーミュス

ドイツ 0 1 0 9 9 ドレスデン、ベッティナシュトラッセ 1 2

(72)発明者 シュテファン ディーステルホースト

ドイツ 0 1 1 5 9 ドレスデン、コネルトプラッツ 1 8

(72)発明者 マーティン ポーラック

ドイツ 0 1 0 9 9 ドレスデン、ブリースニッツシュトラッセ 3 5

審査官 清木 泰

(56)参考文献 特表 2 0 0 6 - 5 1 8 0 7 7 (J P , A)

特開 2 0 0 3 - 0 3 0 0 4 9 (J P , A)

特表 2 0 0 9 - 5 2 1 7 6 7 (J P , A)

特表 2 0 0 9 - 5 0 1 3 6 6 (J P , A)

特表 2 0 0 8 - 5 2 5 9 2 3 (J P , A)

特表 2 0 0 8 - 5 1 5 0 5 5 (J P , A)

特開 2 0 0 3 - 0 3 0 0 5 0 (J P , A)

米国特許出願公開第 2 0 0 9 / 0 0 3 1 3 1 0 (U S , A 1)

Craig Zilles, Lee Baugh, Extending Hardware Transactional Memory to Support Non-busy Waiting and Non-transactional Actions, TRANSACT: First ACM SIGPLAN Workshop on Languages, Compilers, and Hardware Support for Transactional Computing, ACM, 2 0 0 6 年 6 月, all 10 pages

Michelle J. Moravan, et al., Supporting Nested Transaction Memory in LogTM, Proceedings of the twelfth International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS'06), ACM, 2 0 0 6 年 1 0 月 2 1 日, Pages:359-370

C. Scott Ananian, et al., Unbounded Transactional Memory, Proceedings of 11th International Symposium on High-Performance Computer Architecture (HPCA-11), IEEE, 2 0 0 5 年 2 月, Pages:316-327

Maurice Herlihy, J. Eliot B. Moss, Transactional Memory: Architectural Support for Lock-Free Data Structures, Proceedings of the 20th Annual International Symposium on Computer Architecture (ISCA'93), IEEE, 1 9 9 3 年, Pages:289-300

Jayaram Bobba, et al., Performance Pathologies in Hardware Transactional Memory, Proceedings of the 34th annual international symposium on computer architecture (ISCA'97), ACM, 2 0 0 7 年, Pages:81-91

Nir Shavit, Dan Touitou, Software Transactional Memory, Proceedings of the fourteenth annual ACM symposium on Principles of distributed computing (PODC'95), ACM, 1 9 9 5 年, Pages:204-213

Sean LIE, Hardware Support for Unbounded Transactional Memory, Ph.D. Thesis (2004), Ma

ssachusetts Institute of Technology, 2004年, all 111 pages

(58)調査した分野(Int.Cl., DB名)

G06F 9/30 - 9/38

G06F 9/46 - 9/54

G06F12/00 - 12/06

G06F13/16 - 13/18

G06F12/08 - 12/12