

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2014-194754  
(P2014-194754A)

(43) 公開日 平成26年10月9日(2014.10.9)

(51) Int.Cl.		F I		テーマコード (参考)
<b>G06F 9/52</b>	<b>(2006.01)</b>	G06F 9/46	472A	
<b>G06F 9/46</b>	<b>(2006.01)</b>	G06F 9/46	430	

審査請求 有 請求項の数 25 O L 外国語出願 (全 29 頁)

(21) 出願番号 特願2014-26130 (P2014-26130)  
 (22) 出願日 平成26年2月14日 (2014.2.14)  
 (31) 優先権主張番号 13/803, 658  
 (32) 優先日 平成25年3月14日 (2013.3.14)  
 (33) 優先権主張国 米国 (US)

(特許庁注：以下のものは登録商標)

1. Itanium

(71) 出願人 591003943  
 インテル・コーポレーション  
 アメリカ合衆国 95054 カリフォル  
 ニア州・サンタクララ・ミッション カレ  
 ッジ ブレーバード・2200  
 (74) 代理人 110000877  
 龍華国際特許業務法人  
 (72) 発明者 ラッシュ、ウィリアム シー、  
 アメリカ合衆国 95054 カリフォル  
 ニア州・サンタクララ・ミッション カレ  
 ッジ ブレーバード・2200 インテル  
 ・コーポレーション内

最終頁に続く

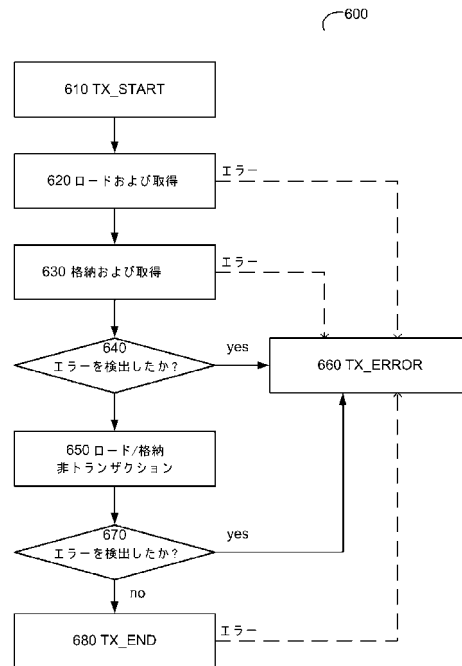
(54) 【発明の名称】 トランザクショナルメモリ実装用のシステムおよび方法

(57) 【要約】 (修正有)

【課題】 実行効率の良いトランザクショナルメモリ実装用のシステムおよび方法を提供する。

【解決手段】 プロセッサは、ブロック610で開始したオペレーションのトランザクションモード中にエラーをブロック670で検出すると、ブロック660によりエラー回復ルーチンを実行し；そうでなければ該プロセッサは、ブロック680でブロック670で参照した非トランザクショナルメモリアクセスオペレーションにより読み出しおよび/または改変されたメモリ位置の状態に関係なく、当該トランザクションを完了することができる。当該プロセッサは、ライトオペレーションの結果を対応するメモリまたはキャッシュ位置に対しコミットし、当該トランザクションに予め割り当てられていたバッファを開放することができる。ブロック670により参照されているオペレーションを完了すると終了する。

【選択図】 図6



## 【特許請求の範囲】

## 【請求項 1】

プロセッサにより、メモリアクセストランザクションを開始するステップと、

第 1 メモリ位置に対する、メモリアクセス追跡論理に関連付けられた第 1 バッファを使用するトランザクショナルリードオペレーション、および、第 2 メモリ位置に対する、前記メモリアクセス追跡論理に関連付けられた第 2 バッファを使用するトランザクショナルライトオペレーションの少なくとも 1 つを実行するステップと、

第 3 メモリ位置に対する非トランザクショナルリードオペレーション、および、第 4 メモリ位置に対する非トランザクショナルライトオペレーションの少なくとも 1 つを実行するステップと、

前記メモリアクセス追跡論理により、前記プロセッサ以外のデバイスによる前記第 1 メモリ位置および前記第 2 メモリ位置の少なくとも 1 つに対するアクセスを検出したことに応じて、前記メモリアクセストランザクションをアポートするステップと、

トランザクションアポート条件の検出の失敗に応じて、前記第 3 メモリ位置の状態および前記第 4 メモリ位置の状態に関わり無く、前記メモリアクセストランザクションを完了するステップと、を備える方法。

10

## 【請求項 2】

前記第 1 バッファおよび前記第 2 バッファが、1 つのバッファからなる、請求項 1 に記載の方法。

## 【請求項 3】

前記第 1 メモリ位置および前記第 2 メモリ位置が、1 つのメモリ位置である、請求項 1 に記載の方法。

20

## 【請求項 4】

前記第 3 メモリ位置および前記第 4 メモリ位置が、1 つのメモリ位置である、請求項 1 に記載の方法。

## 【請求項 5】

前記第 1 バッファおよび前記第 2 バッファの少なくとも 1 つが、データキャッシュ内の一エントリによって与えられる、請求項 1 に記載の方法。

## 【請求項 6】

第 2 ライトオペレーションを実行するステップは、前記第 2 ライトオペレーションのコミットをするステップを含む、請求項 1 から 5 の何れか一項に記載の方法。

30

## 【請求項 7】

前記メモリアクセストランザクションを完了するステップは、前記第 2 バッファから、より高いレベルのキャッシュエントリおよびメモリ位置の一方へデータをコピーするステップを含む、請求項 1 から 5 の何れか一項に記載の方法。

## 【請求項 8】

割込み、バッファオーバーフローおよびプログラムエラーの少なくとも 1 つの検出に応じて、前記メモリアクセストランザクションをアポートするステップを更に含む、請求項 1 から 5 の何れか一項に記載の方法。

## 【請求項 9】

前記アポートを行うステップが、前記第 1 バッファおよび前記第 2 バッファの少なくとも 1 つを解放するステップを含む、請求項 1 から 5 の何れか一項に記載の方法。

40

## 【請求項 10】

前記メモリアクセストランザクションを開始するステップが、更に、保留中のライトオペレーションのコミットをするステップを含む、請求項 1 から 5 の何れか一項に記載の方法。

## 【請求項 11】

前記メモリアクセストランザクションを開始するステップが、割込みを禁止するステップを含む、請求項 1 から 5 の何れか一項に記載の方法。

## 【請求項 12】

50

前記メモリアクセストランザクションを開始するステップが、データのプリフェッチを禁止するステップを含む、請求項 1 から 5 の何れか一項に記載の方法。

【請求項 1 3】

前記メモリアクセストランザクションを完了する前に、ネスト形メモリアクセストランザクションを開始するステップと、

前記メモリアクセス追跡論理に関連付けられた第 3 バッファを使用する第 2 トランザクショナルリードオペレーション、およびメモリアクセス追跡論理に関連付けられた第 4 バッファを使用する第 2 トランザクショナルライトオペレーションの少なくとも 1 つを実行するステップと、

前記ネスト形メモリアクセストランザクションを完了するステップと、を更に備える、請求項 1 から 5 の何れか一項に記載の方法。

10

【請求項 1 4】

トランザクションアポート条件の検出に応じて、前記メモリアクセストランザクションおよび前記ネスト形メモリアクセストランザクションをアポートするステップを更に備える、請求項 1 3 に記載の方法。

【請求項 1 5】

メモリアクセス追跡論理と、

前記メモリアクセス追跡論理に関連付けられた第 1 バッファと、

前記メモリアクセス追跡論理に関連付けられた第 2 バッファと、

前記第 1 バッファおよび前記第 2 バッファに通信式に結合されたプロセッサコアと、を備える処理システムであって、

20

前記処理システムにおいて、前記プロセッサコアは、複数のオペレーションを行うように構成され、

前記複数のオペレーションは、

メモリアクセストランザクションを開始ステップと、

第 1 メモリ位置に対する、第 1 バッファを使用するトランザクショナルリードオペレーション、および、第 2 メモリ位置に対する、第 2 バッファを使用するトランザクショナルライトオペレーションの少なくとも 1 つを実行するステップと、

第 3 メモリ位置に対する非トランザクショナルリードオペレーション、および、第 4 メモリ位置に対する非トランザクショナルライトオペレーションの少なくとも 1 つを実行するステップと、

30

前記メモリアクセス追跡論理により、前記プロセッサ以外のデバイスによる前記第 1 メモリ位置および前記第 2 メモリ位置の少なくとも 1 つに対するアクセスを検出したことに応じて、前記メモリアクセストランザクションをアポートするステップと、

トランザクションアポート条件の検出の失敗に応じて、前記第 3 メモリ位置の状態および前記第 4 メモリ位置の状態に関わり無く、前記メモリアクセストランザクションを完了するステップと、を含む、処理システム。

【請求項 1 6】

データキャッシュを更に備え、

前記第 1 バッファおよび前記第 2 バッファの少なくとも 1 つが、前記データキャッシュの中に備わっている、請求項 1 5 に記載の処理システム。

40

【請求項 1 7】

エラー回復ルーチンのアドレスを格納するためのレジスタを更に備える、請求項 1 5 に記載の処理システム。

【請求項 1 8】

前記メモリアクセストランザクションのステータスを格納するためのレジスタを更に備える、請求項 1 5 に記載の処理システム。

【請求項 1 9】

前記第 1 バッファおよび前記第 2 バッファが、1 つのバッファからなる、請求項 1 5 に記載の処理システム。

50

**【請求項 2 0】**

第 3 バッファおよび第 4 バッファが、1 つのバッファからなる、請求項 1 5 に記載の処理システム。

**【請求項 2 1】**

前記第 1 メモリ位置および前記第 2 メモリ位置が、1 つのメモリ位置である、請求項 1 5 に記載の処理システム。

**【請求項 2 2】**

前記第 3 メモリ位置および前記第 4 メモリ位置が、1 つのメモリ位置である、請求項 1 5 に記載の処理システム。

**【請求項 2 3】**

割り込み、バッファオーバーフロー、およびプログラムエラーの少なくとも 1 つの検出に応じて、前記メモリアクセストランザクションをアポートするように前記プロセッサコアが、更に構成されている、請求項 1 5 に記載の処理システム。

**【請求項 2 4】**

コンピュータに、

メモリアクセストランザクションを開始する手順と、

第 1 メモリ位置に対する、メモリアクセス追跡論理に関連付けられた第 1 バッファを使用するトランザクショナルリードオペレーション、および、第 2 メモリ位置に対する、前記メモリアクセス追跡論理に関連付けられた第 2 バッファを使用するトランザクショナルライトオペレーションの少なくとも 1 つを実行する手順と、

第 3 メモリ位置に対する非トランザクショナルリードオペレーション、および、第 4 メモリ位置に対する非トランザクショナルライトオペレーションの少なくとも 1 つを実行する手順と、

前記メモリアクセス追跡論理により、前記コンピュータ以外のデバイスによる前記第 1 メモリ位置および前記第 2 メモリ位置の少なくとも 1 つに対するアクセスを検出したことに応じて、前記メモリアクセストランザクションをアポートする手順と

トランザクションアポート条件の検出の失敗に応じて、前記第 3 メモリ位置の状態および前記第 4 メモリ位置の状態に関わり無く、前記メモリアクセストランザクションを完了する手順と、を実行させるプログラム。

**【請求項 2 5】**

メモリと、

前記メモリと結合された処理システムと、を備え、

前記処理システムが、請求項 1 から 1 4 のいずれか一項に記載の方法を行うように構成される装置。

**【発明の詳細な説明】****【技術分野】****【0 0 0 1】**

本発明は、一般にコンピュータシステムに関し、特にトランザクショナルメモリ実装用のシステムおよび方法に関する。

**【背景技術】****【0 0 0 2】**

複数のプロセスを同時に実行するには、共有資源（例えば、複数のプロセッサからアクセス可能なメモリ）に対し同期メカニズムを実装することが必要である。そのような同期メカニズムの一例は、セマフォア式ロッキングであり、セマフォア式ロッキングによりプロセスの実行が直列化され、その結果、システム全体の性能が悪影響を受ける可能性がある。更に、セマフォア式ロッキングでは、デッドロック（複数のプロセスがそれぞれ、他のプロセスが資源のロックを解除するのを待っているときに起きる状態）を生じる恐れがある。

**【図面の簡単な説明】****【0 0 0 3】**

【図1】本願開示の一または複数態様によるコンピュータシステムの実施例のハイレベルコンポーネント図である。

【図2】本願開示の一または複数態様によるプロセッサのブロック図である。

【図3a】本願開示の一または複数態様によるプロセッサマイクロアーキテクチャの要素を概略的に示す図である。

【図3b】本願開示の一または複数態様によるプロセッサマイクロアーキテクチャの要素を概略的に示す図である。

【図4】本願開示の一または複数態様によるトランザクショナルメモリアクセスを実施するコンピュータシステムの実施例の態様を幾つか示す図である。

【図5】本願開示の一または複数態様によるトランザクションモード命令の使用を説明する、コードフラグメントの例を示す図である。

【図6】本願開示の一または複数態様によるトランザクショナルメモリアクセスを実行する方法の流れ図である。

【図7】本願開示の一または複数態様によるコンピュータシステムの実施例のブロック図である。

【発明を実施するための形態】

【0004】

本発明は限定されるのではなく例示されており、添付図面と関連付けて検討される場合、以下の詳細な説明を参照しながら本開示内容を十分に理解できるであろう。

【0005】

コンピュータシステムによりトランザクショナルメモリアクセスを実施するための方法およびシステムを本願明細書において説明する。「トランザクショナルメモリアクセス」とは、1つのプロセッサが複数のメモリアクセス命令をアトミックオペレーションとして、それらの命令が一括で成功か失敗かの何れかになるように実行することを指す。後者の状況では、メモリが変更されずに一連のオペレーションのうち最初のオペレーションの実行前に存続している状態であり、且つまたは他の修正動作を行ってもよい。特定の実装において、推論的に、即ちアクセスされているメモリをロックせずに、トランザクショナルメモリアクセスを実行することもあり、その結果、同時に実行中の複数のスレッドおよび/またはプロセスによる一つの共有資源へのアクセスを同期するための有効なメカニズムを設けることになる。

【0006】

トランザクショナルメモリアクセスを実施するには、トランザクション開始命令と、トランザクション終了命令をプロセッサ命令セットに入れればよい。オペレーションのトランザクションモードでは、プロセッサは、それぞれのリードバッファおよび/またはライトバッファを介して、推論的に複数のメモリリードオペレーションおよび/またはメモリライトオペレーションを行うことができる。対応するメモリ位置にデータを記憶せずに、ライトバッファは、メモリライトオペレーションの結果を保持する。バッファに関連付けられるメモリ追跡論理は、指定されたメモリ位置への別の装置によるアクセスを検出すると、エラーの状態をプロセッサへ信号送信する。エラー信号を受信すると、プロセッサは、そのトランザクションをアポートして、エラー回復ルーチンへ制御を移す。あるいは、トランザクション終了命令が到達した時点で、プロセッサがエラーをチェックしてもよい。トランザクションアポート条件が無ければ、プロセッサは、ライトオペレーションの結果を対応するメモリまたはキャッシュ位置に記憶する。オペレーションのトランザクションモードでは、トランザクションが無事完了したかアポートしたかに関わりなく、プロセッサは、メモリリードオペレーションおよび/またはライトオペレーションを1回または複数回実行するが、それは、これらのオペレーションの結果が他のデバイス（例えば、他のプロセッサコアまたは他のプロセッサ）に直ちにできるように迅速に記憶される。トランザクションの範囲内で非トランザクショナルメモリアクセスを行うことが可能であることで、プロセッサのプログラミングに関し融通性が更に良くなり、所定のプログラミングタスクを完了するのに必要なトランザクション数を潜在的に減らすことにより、実行効率

10

20

30

40

50

全体が向上する。

【0007】

上に参照した方法およびシステムの種々の態様は、本願明細書では以下に、制限ではなく一例として詳細に説明する。

【0008】

以下の説明では、本発明を徹底的に理解できるように、特定の複数種類のプロセッサおよびシステム構成、特定のハードウェア構造、特定の設計上、つまりマイクロ設計上の細部、特定のレジスタ構成、特定の命令型、特定のシステムコンポーネント、特定の寸法/高さ、特定のプロセッサパイプラインステージ、およびオペレーション等の実施例のような数多くの具体的細部を記述している。しかしながら、本発明を実施するために、これらの具体的細部を使用する必要がないことは、当業者には明らかとなるであろう。他の例では、特定および代替プロセッサアーキテクチャ、特定の論理回路/記述されたアルゴリズム用のコード、特定のファームウェアコード、特定の相互接続オペレーション、特定の論理構成、特定の製造技術および材料、特定のコンパイラの実行、コードでの特定のアルゴリズム表現、特定のパワーダウンおよびゲーティング技術/論理、その他コンピュータシステムの特定のオペレーション上の詳細など、周知のコンポーネントまたは方法については、本発明を不必要に不明瞭にすることを避けるために、詳述しなかった。

【0009】

以下の実施態様はプロセッサに関して記載されているが、他の実施態様は他の種類の集積回路および論理デバイスに適用することができる。類似の技術および本発明の実施態様による開示内容は、より高いパイプラインスループットおよび改良された性能の恩恵を受け得る他の種類の回路または半導体デバイスに適用することができる。データ操作を実行するいかなるプロセッサまたはマシンに対しても、本発明の実施態様の開示内容は、適用することができる。しかしながら、本発明は、512ビット、256ビット、128ビット、64ビット、32ビットまたは16ビットのデータオペレーションを行うプロセッサまたはマシンに限定されるのではなく、データ操作または管理が行われるいかなるプロセッサおよびマシンにも適用することができる。また以下の説明は実施例を提供し、添付図面は例示のため様々な実施例を示す。しかし、これらの実施例は、本発明の実施態様の可能性のある実施の全ての完全なリストを提供するよりはむしろ、本発明の実施態様の実例を提供するように意図されているにすぎないため、それらの実施例は、限定の意味で解釈してはならない。

【0010】

下記の実施例では、実行ユニットと論理回路を例にとって命令の処理と分散を説明してあるが、本発明の他の実施態様は、機械読取り可能な有形媒体上に記憶されたデータまたは命令により実現可能であり、機械による動作時には、同機械は、少なくとも一実施態様に一致する機能を果たす。ある一実施態様では、本発明の複数の実施態様に関連した複数の機能が機械にて実行可能な命令の中で実現される。命令を用いてプログラムを受ける汎用または特殊用途のプロセッサに本発明のステップを行わせるために、複数の命令を使用することがある。本発明の実施態様は、コンピュータプログラム製品またはソフトウェアとして提供することも可能であり、そのようなコンピュータプログラム製品またはソフトウェアは、本発明の実施態様による一または複数のオペレーションを行うようにコンピュータ(または他の電子デバイス)をプログラムするために使用することができる命令が記憶された機械またはコンピュータ読取り可能な媒体を含むものでもよい。あるいは、本発明の実施態様のオペレーションは、同オペレーションを実施するための一定の機能を有する論理を具備する特定のハードウェアコンポーネント、またはプログラムドコンピュータコンポーネントと一定の機能を有するハードウェアコンポーネントとの何らかの組合せにより、実施してもよい。

【0011】

本発明の実施態様を実施するための論理をプログラムするために使用される命令は、DRAM、キャッシュ、フラッシュメモリまたは他の記憶装置等の当該システム内のメモリ

に記憶することができる。更に、それらの命令は、ネットワークを介して分散するか、他のコンピュータ読取り可能媒体により分散することができる。機械読取り可能媒体は、機械（例えば、コンピュータ）で読み取ることができる形態で情報を記憶または送信するための何らかのメカニズムを具備してよく、そのような媒体は、一例として、フロッピー（登録商標）ディスク、光ディスク、コンパクトディスク・リードオンリーメモリ（CD ROM）、光磁気ディスク、リードオンリーメモリ（ROM）、ランダムアクセスメモリ（RAM）、消去可能プログラム可能リードオンリーメモリ（EPROM）、電氣的消去可能プログラム可能リードオンリーメモリ（EEPROM）、磁気または光カード、フラッシュメモリ、または電気形態、光形態、音響形態または他の形態の伝播信号（例えば、搬送波、赤外線信号、デジタル信号など）によりインターネット経由で情報を送信する際に使用される有形の機械読取り可能な記憶装置であるが、これらに限定されるものではない。従って、コンピュータ読取り可能媒体は、機械（例えば、コンピュータ）で読取り可能な形態で電子命令または情報を記憶または送信するのに適した、何らかの種類の有形の機械読取り可能媒体を含む。

10

20

30

40

50

#### 【0012】

本願明細書における「プロセッサ」は、符号化算数演算、論理演算、または入出力オペレーションのための命令を実行可能なデバイスを指す。実例の一つでは、プロセッサは、フォンノイマンアーキテクチャモデルに従い、算数論理演算装置（ALU）、コントロールユニット、および複数のレジスタを具備する。更に別の態様では、プロセッサは、一または複数のプロセッサコアを含むものでよく、従って、通常単一命令パイプラインを処理するシングルコアプロセッサ、または複数の命令パイプライン同時に処理することができるマルチコアプロセッサでよい。また別の態様では、プロセッサは、単一の集積回路または複数の集積回路として実装してもよいし、さもなければ、マルチチップモジュールのコンポーネントでもよい（例えば、マルチチップモジュールでは、個々のマイクロプロセッサダイが単一の集積回路パッケージに含まれ、従って個々のマイクロプロセッサチップが一つのソケットを共有する）。

#### 【0013】

図1は、本願開示の一態様または複数の態様によるコンピュータシステムの実施例のハイレベルコンポーネント図を示す。コンピュータシステム100は、本願明細書に記載の実施態様により、データを処理するためのアルゴリズムを実行するための論理を含む実行ユニットを使用するためのプロセッサ102を含む。システム100は、カリフォルニア州サンタクララのインテル社から入手可能なPentium（登録商標）III、Pentium（登録商標）4、Xeon（登録商標）、Itanium、XScale（登録商標）および/またはStrongARM（登録商標）マイクロプロセッサを使用した処理システムを表しているが、他のシステム（他のマイクロプロセッサを有するPC、エンジニアリングワークステーション、セットトップボックス等）を使用してもよい。一実施態様では、サンプルシステム100は、ワシントン州レッドモンドのマイクロソフト社から入手可能なWINDOWS（登録商標）バージョンのオペレーティングシステムを実行するが、他のオペレーティングシステム（例えば、UNIX（登録商標）やLinux（登録商標））、埋込み型ソフトウェア、および/またはグラフィカルユーザインタフェースも使用することができる。このように、本発明の実施態様は、ハードウェア回路構成とソフトウェアの如何なる特定の組合せにも限定されるものではない。

#### 【0014】

実施態様は、コンピュータシステムに限定されるものではない。本発明の別の実施態様は、携帯用装置や埋込み型アプリケーションのような他の装置で使用することができる。携帯用装置の実施例には、携帯電話、インターネットプロトコルデバイス、デジタルカメラ、携帯情報端末（PDA）、携帯用PCがある。埋込み型アプリケーションとしては、マイクロコントローラ、デジタル信号プロセッサ（DSP）、システムオンチップ、ネットワークコンピュータ（NetPC）、セットトップボックス、ネットワークハブ、広域ネットワーク（WAN）スイッチ、あるいは少なくとも1つの実施態様による一または複

数の命令を実行することができる何らかの他のシステムが可能である。

【 0 0 1 5 】

この実例では、プロセッサ 1 0 2 は、一または複数の命令（例えば、トランザクショナルメモリアクセス命令）を実行するためのアルゴリズムを実行するための一または複数の実行ユニット 1 0 8 を含む。一実施態様は、シングルプロセッサデスクトップまたはサーバシステムに関して記述されているが、別の実施態様は、マルチプロセッサシステムに含まれる。システム 1 0 0 は、「ハブ」システムアーキテクチャの一例である。コンピュータシステム 1 0 0 は、データ信号を処理するためのプロセッサ 1 0 2 を含む。実例としてのプロセッサ 1 0 2 は、例えば、複雑命令セットコンピュータ（CISC）マイクロプロセッサ、縮小命令セットコンピューティング（RISC）マイクロプロセッサ、超長命令語（VLW）マイクロプロセッサ、命令セットの組合せを実行するプロセッサ、またはデジタル信号プロセッサのような他のプロセッサ装置を含む。プロセッサ 1 0 2 は、システム 1 0 0 内でプロセッサ 1 0 2 と他のコンポーネントとの間でデータ信号を送信するプロセッサバス 1 1 0 に結合されている。システム 1 0 0 の構成要素（例えば、グラフィックスアクセラレータ 1 1 2、メモリコントローラハブ 1 1 6、メモリ 1 2 0、入出力コントローラハブ 1 2 4、無線トランシーバ 1 2 6、フラッシュ B I O S 1 2 8、ネットワークコントローラ 1 3 4、オーディオコントローラ 1 3 6、シリアル拡張ポート 1 3 8、入出力コントローラ 1 4 0 等）は、当業者に周知の従来機能を果たす。

10

【 0 0 1 6 】

一実施態様において、プロセッサ 1 0 2 は、レベル 1（L1）の内部キャッシュ 1 0 4 を含む。そのアーキテクチャに依存して、プロセッサ 1 0 2 は、単一の内部キャッシュまたは複数レベルの内部キャッシュを有する。他の実施態様は、個別の実装と必要性に応じて、内部キャッシュと外部キャッシュの両方の組合せを含む。レジスタファイル 1 0 6 は、整数レジスタ、浮動小数点レジスタ、ベクトルレジスタ、バンクドレジスタ、シャドウレジスタ、チェックポイントレジスタ、ステータスレジスタ、および命令ポインタレジスタ等の様々なレジスタに異なる種類のデータを記憶することになる。

20

【 0 0 1 7 】

プロセッサ 1 0 2 は、整数および浮動小数点演算を行うための論理を含む実行ユニット 1 0 8 も具備する。プロセッサ 1 0 2 は、一実施態様において、マイクロコードを記憶するためのマイクロコード（ $\mu$ code）ROM を含み、実行時には、特定のマクロ命令のためのアルゴリズムを実行するか、あるいは複雑なシナリオを処理する。ここで、場合によって、マイクロコードは、プロセッサ 1 0 2 に対し論理バグ / 修正を処理するように更新することができる。一実施態様に関しては、実行ユニット 1 0 8 は、圧縮された命令セット 1 0 9 を処理するための論理を含む。汎用プロセッサ 1 0 2 の命令セットの中に圧縮された命令セット 1 0 9 を入れることによって、命令を実行するための関連回路構成と共に、多くのマルチメディア・アプリケーションにより使用されるオペレーションが、汎用プロセッサ 1 0 2 内のバックデータを使用しながら行われる。このように、多くのマルチメディア・アプリケーションは、加速されて、バックデータに対しオペレーションを実行するためのプロセッサのデータバスの全幅を使うことでより効率的に実行される。これによって、一または複数のオペレーションを行うために、一度に一データ要素づつ、より小さな単位のデータをプロセッサのデータバス越しに転送する必要が潜在的に無くなっている。

30

40

【 0 0 1 8 】

別の実施例では、実行ユニット 1 0 8 は、マイクロコントローラ、埋込み型プロセッサ、グラフィックスデバイス、DSP、および他の種類の論理回路でも使用することができる。システム 1 0 0 は、メモリ 1 2 0 を含む。メモリ 1 2 0 は、ダイナミックランダムアクセスメモリ（DRAM）デバイス、スタティックランダムアクセスメモリ（SRAM）デバイス、フラッシュメモリデバイス、またはその他メモリデバイスを含む。メモリ 1 2 0 は、命令および / またはプロセッサ 1 0 2 が実行するデータ信号により表されたデータを記憶する。

50



## 【0019】

システム論理チップ116は、プロセッサバス110およびメモリ120に結合されている。例示した実施態様では、システム論理チップ116は、メモリコントローラハブ(MCH)である。プロセッサ102は、プロセッサバス110を介してMCH116と通信することができる。MCH116は、命令およびデータの記憶兼グラフィックスコマンド、データおよびテクスチャの記憶用のメモリ120への高帯域メモリバス118を与える。MCH116は、プロセッサ102と、メモリ120と、システム100内の他のコンポーネントとの間で、データ信号を指示し、プロセッサバス110と、メモリ120と、システム入出力122との間で、データ信号の橋渡しをする。一部の実施態様では、システム論理チップ116により、グラフィックスコントローラ112へ結合するためのグラフィックスポートを設けることが可能になっている。MCH116は、メモリインタフェース118を介してメモリ120に結合されている。グラフィックスカード112は、アクセラレイティッド・グラフィックス・ポート(AGP)相互接続部114を介してMCH116に結合されている。

10

## 【0020】

システム100独自のハブインタフェースバス122を使用して、MCH116を出力コントローラハブ(ICH)130に結合する。ICH130は、ローカル入出力バスを介して、幾つかの入出力装置への直接接続を可能にする。ローカル入出力バスは、周辺機器をメモリ120、チップセット、およびプロセッサ102に接続するための高速入出力バスである。幾つかの例は、オーディオコントローラ、ファームウェアハブ(フラッシュBIOS)128、無線トランシーバ126、データ記憶装置124、ユーザ入力装置およびキーボードインタフェースを具備するレガシー入出力コントローラ、ユニバーサルシリアルバス(USB)のようなシリアル拡張ポート、およびネットワークコントローラ134である。データ記憶装置124は、ハードディスクドライブ、フロッピー(登録商標)ディスクドライブ、CD-ROM装置、フラッシュメモリデバイス、または他の大容量記憶装置が可能である。

20

## 【0021】

別の実施例のシステムでは、一実施態様による命令は、システムオンチップに対し使用することができる。一実施態様のシステムオンチップは、プロセッサとメモリとを含む。そのようなシステムのメモリは、フラッシュメモリである。フラッシュメモリは、プロセッサおよびその他システムコンポーネントと同じダイの上に配置することができる。更に、メモリコントローラまたはグラフィックスコントローラのような他の論理ブロックをシステムオンチップ上に配置することもできる。

30

## 【0022】

上記実施例のプロセッサ102は、トランザクショナルメモリアクセスを実行する。ある種の実行の際には、プロセッサ102は、一または複数回のメモリアクセス操作および/またはライト操作も実行するが、以下に更に詳細に述べるように、これらの操作は、トランザクショナル無事完了またはアポートに関係なく、結果が他の装置(例えば、他のプロセッサコア、または他のプロセッサ)から直ちにできるように、迅速に行われる。

40

## 【0023】

図2は、本発明の一実施態様によるトランザクショナルメモリアクセス命令および/または非トランザクショナルメモリアクセス命令を実行するための論理回路を含むプロセッサ200用のマイクロアーキテクチャのブロック図である。幾つかの実施態様では、一実施態様による命令を実行し、バイト、ワード、ダブルワード、クワッドワード等のサイズを有するデータエレメントにも、単精度整数データ型および倍精度整数データ型、ならびに浮動小数点データ型のようなデータ型のデータエレメントにも作用することが可能である。また、一実施態様において、間順フロントエンド201は、実行対象の命令を取り出し、後からプロセッサパイプラインで使用するよう命令を準備するプロセッサ200の部分である。フロントエンド201は、幾つかのユニットを含んでよい。一実施態様にお

50

いて、命令事前取出し部 2 2 6 は、メモリから命令を取り出し、それらの命令を命令デコーダ 2 2 8 へ供給し、命令デコーダは、命令を順次復号化または解釈実行する。例えば、一実施態様において、デコーダは受け取った命令を、機械が実行可能な「マイクロ命令」または「マイクロオペレーション」（マイクロOPまたは $\mu$ OPとも称する）と称する一または複数のオペレーションに復号化する。別の実施態様では、デコーダは、命令を解析して、一実施態様によるオペレーションを行うためのマイクロアーキテクチャにより使用されるオペコードおよび対応するデータ、ならびにコントロールフィールドとする。また、一実施態様では、トレースキャッシュ 2 3 0 は、復号化された $\mu$ OPを取得し、それらをアセンブルして、実行用の $\mu$ OPキュー 2 3 4 中のプログラムオーダードシーケンスまたはトレースとする。トレースキャッシュ 2 3 0 が、複雑命令に遭遇すると、マイクロコードROM 2 3 2 は、オペレーションを完了するために必要な $\mu$ OPを供給する。

10

#### 【0024】

一部の命令は、単一のマイクロOPに変換されるが、他の命令は、全オペレーションを完了するために、数個のマイクロOPを必要とする。また、一実施態様では、4以上のマイクロOPが命令を完了するために必要な場合、デコーダ 2 2 8 は、マイクロコードROM 2 3 2 にアクセスして、命令を実行する。一実施態様において、一つの命令は、命令デコーダ 2 2 8 での処理用の少数のマイクロOPに復号化される。また別の実施態様では、マイクロコードROM 2 3 2 内に一つの命令を記憶可能であり、若干数のマイクロOPが、そのオペレーションの遂行に必要とされるはずである。トレースキャッシュ 2 3 0 は、一実施態様による一または複数の命令を完了するためのマイクロコードシーケンスを、マイクロコードROM 2 3 2 から読み出すための正しいマイクロ命令ポインタを決定するためのエントリポイントプログラム可能論理アレイ(PLA)を指す。命令に対するマイクロOPの順序付けをマイクロコードROM 2 3 2 が完了した後、当該機械のフロントエンド 2 0 1 は、トレースキャッシュ 2 3 0 からのマイクロOPの取出しを再開する。

20

#### 【0025】

アウト・オブ・オーダー実行エンジン 2 0 3 は、命令が実行用に準備される場所である。アウト・オブ・オーダー実行論理は、命令がパイプラインを下り実行についてスケジュールされるときに、命令の流れを平滑化して再順序付けし、性能を最適化するための若干数のバッファを有する。アロケータ論理は、各 $\mu$ OPが順次実行する必要があるバッファおよび資源をマシンに割り当てる。レジスタ・リネーミング論理は、論理レジスタをレジスタファイル内のエントリにリネームする。また、命令スケジューラの前に、アロケータは、2本の $\mu$ OPキュー(メモリオペレーション用に1キューと、ノンメモリオペレーション用に1キュー)のうちの本キューの各 $\mu$ OP毎に1エントリを割り当てるが、ここで命令スケジューラは、メモリスケジューラ、高速スケジューラ 2 0 2、低速/汎用浮動小数点スケジューラ 2 0 4、および単純浮動小数点スケジューラ 2 0 6 である。 $\mu$ OPスケジューラ 2 0 2、2 0 4、2 0 6 は、それらの従属入力レジスタ・オペランド・ソースの準備性および実行資源の入手可能性に基づいて、オペレーションを完了する必要がある $\mu$ OPの実行の準備ができるかを決定する。一実施態様の高速スケジューラ 2 0 2 は、主クロックサイクルの1/2毎にスケジュールを行い、他のスケジューラは、一主プロセッサクロックサイクル毎に1回スケジュールを行うことができる。スケジューラは、ディスパッチポートについて調停を行い、実行のために $\mu$ OPをスケジュールする。

30

40

#### 【0026】

レジスタファイル 2 0 8、2 1 0 は、スケジューラ 2 0 2、2 0 4、2 0 6 と実行ブロック 2 1 1 内の実行ユニット 2 1 2、2 1 4、2 1 6、2 1 8、2 2 0、2 2 2、2 2 4 との間に位置する。整数オペレーションと浮動小数点オペレーション用にそれぞれ別のレジスタファイル 2 0 8、2 1 0 が存在する。一実施態様による各レジスタファイル 2 0 8、2 1 0 は、バイパスネットワークも含んでおり、同バイパスネットワークは、レジスタファイルに未だ書き込んでいない丁度終了した結果を、新しい従属 $\mu$ OPへバイパスするか、あるいは転送する。整数レジスタファイル 2 0 8 および浮動小数点レジスタファイル 2 1 0 も、互いにデータ通信を行うことが可能である。一実施態様において、整数レジス

50

タファイル 208 は、2つの異なるレジスタファイルに分割されており、一方のレジスタファイルは、下位 32 ビットデータ用で、二つめのレジスタファイルは、高位 32 ビットデータ用である。一実施態様による浮動小数点レジスタファイル 210 は、128 ビット幅のエントリを有する。これは、浮動小数点命令が、通常、幅 64 から 128 ビットのオペランドを有するからである。

#### 【0027】

実行ブロック 211 は、実行ユニット 212、214、216、218、220、222、224 を含み、これらの実行ユニットで、命令が実際に実行される。このセクションは、マイクロ命令を実行する必要がある整数および浮動小数点データオペランド値を記憶するレジスタファイル 208、210 を含む。また、一実施態様によるプロセッサ 200 は、若干数の実行ユニット、すなわちアドレス生成ユニット (AGU) 212、AGU 214、高速算術論理演算装置 216、高速算術論理演算装置 218、低速算術論理演算装置 220、浮動小数点算術論理演算装置 222、浮動小数点ムーブユニット 224 から成る。一実施態様において、浮動小数点実行ブロック 222、224 は、浮動小数点、MMX、SIMD および SSE、またはその他オペレーションを実行する。一実施態様による浮動小数点算術論理演算装置 222 は、除算、平方根、および剰余マイクロ OP を実行するための 64 ビット / 64 ビット浮動小数点除算器を含む。本発明の実施態様においては、浮動小数点値を含む命令は、浮動小数点ハードウェアを用いて処理すればよい。一実施態様では、算術論理演算装置のオペレーションは、高速算術論理演算装置実行ユニット 216、218 によりなされる。一実施態様による高速算術論理演算装置 216、218 は、クロックサイクルの半分の効果的レイテンシで、高速オペレーションを実行することができる。

また、一実施態様において、最も複雑な整数オペレーションは、低速算術論理演算装置 220 によりなされるが、これは、低速算術論理演算装置 220 が、乗算器、シフト、フラグ論理、および分岐処理のような長レイテンシ型のオペレーション用の整数実行ハードウェアを含んでいるからである。メモリロード / 記憶オペレーションは、AGU 212、214 により実行される。一実施態様において、整数算術論理演算装置 216、218、220 は、64 ビットデータオペランドに対する整数オペレーションの実施に関して説明する。また、別の実施態様では、16、32、128、256 等の様々なデータビットをサポートできるように、算術論理演算装置 216、218、220 を実装することができる。同様に、浮動小数点ユニット 222、224 は、複数ビットの様々な幅を有するオペランドの範囲をサポートするように実装することができる。また、一実施態様において、浮動小数点ユニット 222、224 は、SIMD およびマルチメディア命令と共に 128 ビット幅のパックデータオペランドに対し作用することが可能である。

#### 【0028】

一実施態様において、 $\mu$ OP スケジューラ 202、204、206 は、ペアレントロードが実行し終わる前に従属オペレーションをディスパッチする。 $\mu$ OP は、プロセッサ 200 において、推論式にスケジューラされて実行されるので、プロセッサ 200 は、メモリミス処理するための論理も含む。データキャッシュにおいてデータロードが失敗する場合は、一時的にデータが間違っただけの状態にスケジューラを放置するパイプラインにおいて一斉処理中の複数の従属オペレーションが存在し得る。リプライメカニズムは、間違っただけのデータを使用している命令を追跡して再実行する。従属オペレーションは繰り返されるはずであるが、無関係なオペレーションは、完了することができる。スケジューラと、プロセッサの一実施態様のリプライメカニズムは、テキストストリング比較オペレーションのために命令シーケンスを取得するようにも設計されている。

#### 【0029】

「レジスタ」という用語は、オペランドを識別する命令の一部として使用されるオンボードのプロセッサ記憶位置を指す。すなわち、レジスタは、(プログラマーの立場から) プロセッサの外部から使用することができるレジスタである。しかし、実施態様のレジスタは、効果の面で、特別な種類の回路に限定すべきではない。むしろ、実施態様のレジス

10

20

30

40

50

タは、データを記憶、提供し、本願明細書に記載の機能を果たすことができる。専用の物理レジスタ、レジスタのリネームを用いる動的に割り当てられた物理レジスタ、専用の物理レジスタと動的に割り当てられた物理レジスタとの組合せ等の、任意の数の異なる技術を使用するプロセッサ内の回路構成により、本願明細書記載のレジスタを実装することができる。また一実施態様において、整数レジスタは、32ビットの整数データを記憶する。また、一実施態様によるレジスタファイルは、パックデータ用に8つのマルチメディアSIMDレジスタも含む。以下の解説に関しては、レジスタは、カリフォルニア州サンタクララのインテル社によるMMX（登録商標）テクノロジーを用いて可能となるマイクロプロセッサにおける64ビット幅MMXレジスタ（場合によって、「mm」レジスタとも称される）のようなパックデータを保持するように設計されたデータレジスタとして見なす。これらのMMXレジスタは、整数形態と浮動小数点形態の両方で入手可能であり、SIMDおよびSSE命令を伴うパックデータ要素を用いて作用する。同様に、SSE2、SSE3、SSE4、またはこれらを超えるテクノロジー（一般に、「SSEx」と称する）に係る128ビット幅XMMレジスタは、そのようなパックデータオペランドを保持するためにも使用することができる。また、一実施態様において、パックデータおよび整数データを記憶する際、レジスタは、これら2つのデータ型の差別化を必要としない。更にまた、一実施態様において、整数および浮動小数点は、同じレジスタファイルに格納されるか、または異なるレジスタファイルに格納されるかのいずれかである。更に、一実施態様において、浮動小数点データおよび整数データは、異なるレジスタに記憶されるか、同じレジスタに記憶してよい。

10

20

#### 【0030】

図3aおよび図3bは、本願開示の一または複数態様によるプロセッサマイクロアーキテクチャの構成要素を概略的に例示している。図3aにおいて、プロセッサパイプライン400は、フェッチステージ402、長さ復号化ステージ404、復号化ステージ406、割付けステージ408、リネーミングステージ410、スケジューリング（ディスパッチまたは問題としても既知）ステージ412、レジスタリード/メモリアクセスステージ414、実行ステージ416、ライトバック/メモリアクセスステージ418、例外処理ステージ422、およびコミットステージ424を含む。

#### 【0031】

図3bにおいて、矢印は、複数のユニット間の結合を表しており、矢印の方向は、それらのユニット間のデータフローの方向を示している。図3bは、実行エンジンユニット450に結合されたフロントエンドユニット430を含むプロセッサコア490を示しており、共に、メモリアクセスユニット470と結合されている。

30

#### 【0032】

コア490は、縮小命令セットコンピューティング（RISC）コア、複雑命令セットコンピューティング（CISC）コア、超長命令語（VLIW）コア、またはハイブリッドまたは代替コア型でよい。また別の選択肢として、コア490は、例えばネットワークまたは通信コア、圧縮エンジン、グラフィックスコア等の特殊用途のコアでよい。ある特定の実装においては、コア490は、本願開示の一または複数態様によるトランザクショナルメモリアクセス命令および/または非トランザクショナルメモリアクセス命令を実行可能に構成することができる。

40

#### 【0033】

フロントエンドユニット430は、命令キャッシュユニット434に結合された分岐予測ユニット432を含み、これは、命令トランザクション・ルックアサイド・バッファ（TLB）436に結合されており、該命令トランザクション・ルックアサイド・バッファは、命令フェッチユニット438に結合されており、該命令フェッチユニットは、復号化ユニット440に接合されている。復号化ユニット、すなわちデコーダは、命令を復号化し、出力として、一または複数のマイクロオペレーション、マイクロコードエントリポイント、マイクロ命令、その他の命令、またはその他の制御信号を生成するが、これらは、元の命令から復号化されるか、もしくは別の方法で元の命令を反映しているか、または元

50

の命令から得たものである。デコーダは、様々に異なるメカニズムを使用して実装することができる。好適なメカニズムの実施例は、ルックアップテーブル、ハードウェア実装、プログラム可能論理アレイ (PLA)、マイクロコード・リード・オンリー・メモリ (ROM) 等であるが、これらに限定されるものではない。命令キャッシュユニット 434 は、メモリユニット 470 において、レベル 2 (L2) のキャッシュユニット 476 に更に結合されている。復号化ユニット 440 は、実行エンジンユニット 450 内のリネーム / アロケータユニット 452 に結合されている。

#### 【0034】

実行エンジンユニット 450 は、リタイアメントユニット 454 と一または一セットのスケジューラユニット 456 とに結合されたリネーム / アロケータユニット 452 を含む。スケジューラユニット 456 は、任意の数の異なるスケジューラを表しており、予約ステーション、中央命令ウインドウ等を含む。一または複数のスケジューラユニット 456 の一セットは、物理レジスタファイル (複数可) のユニット (複数可) 458 に結合されている。物理レジスタファイル (複数可) の複数ユニット 458 のうちの各ユニットは、物理レジスタファイル (複数可) に相当し、異なる物理レジスタファイルは、一または複数異なるデータ型、例えばスカラー整数、スカラー浮動小数点、パック整数、パック浮動小数点、ベクトル整数、ベクトル浮動小数点等、ステータス (例えば、次の実行対象の命令のアドレスである命令ポインタ) 等を保存する。上記物理レジスタファイル (複数可) のユニット (複数可) 458 は、リタイアメントユニット 454 と重複させて、レジスタエイリアシングとアウト・オブ・オーダー式の実行が行われる様々な様子を例示する。 (この例示は、例えば、リオダバッファ (複数可) およびリタイアメントレジスタファイル (複数可) を使用しながら ; フューチャーファイル (複数可) およびヒストリバッファ (複数可) とリタイアメントレジスタファイル (複数可) を使用しながら ; レジスタマップ (複数可) およびレジスタ・プールを使用しながら行う。) 一般に、アーキテクチャ上のレジスタは、プロセッサの外側から、つまりプログラマーの立場から見える。レジスタは、いかなる周知の特定の種類の回路にも限定されていない。

本願明細書において記載されているようにレジスタがデータを記憶して提供することができる限り、様々な種類のレジスタが好適である。制限するものではないが、好適なレジスタの例には、専用の物理レジスタ、レジスタエイリアシングを用いる動的に割り付けられた物理レジスタ、専用の物理レジスタと動的に割り当てられた物理レジスタとの組合せ等がある。リタイアメントユニット 454 と、物理レジスタファイル (複数可) のユニット (複数可) 458 とは、実行クラスタ (複数可) 460 と結合されている。実行クラスタ (複数可) 460 は、一または複数の実行ユニット 162 の一セットおよび一または複数のメモリアクセスユニット 464 の一セットを含む。実行ユニット 462 は、様々な型 (例えば、スカラー浮動小数点、パック整数、パック浮動小数点、ベクトル整数、ベクトル浮動小数点) のデータに対し様々な演算 (例えば、シフト、加算、減算、乗算) を行うことができる。一部の実施態様には、特定の機能または機能セットに専用の若干数の実行ユニットを含んでもよいが、他の実施態様では、一実行ユニット、または全てが全部の機能を果たす複数の実行ユニットを含んでもよい。スケジューラユニット (複数可) 456、物理レジスタファイル (複数可) のユニット (複数可) 458、および実行クラスタ (複数可) 460 は、できる限り複数であるかのように示した。これは、ある特定の実施態様では、ある特定の型のデータ / オペレーションに対する別々のパイプライン (例えば、それぞれスケジューラユニット、物理レジスタファイル (複数可) のユニット、および / または実行クラスタを有する、スカラー整数パイプライン、スカラー浮動小数点 / パック整数 / パック浮動小数点 / ベクトル整数 / ベクトル浮動小数点パイプラインおよび / またはメモリアクセスパイプライン、ただし、個別のメモリアクセスパイプラインの場合、特定の実施態様が実施され、その際、このパイプラインの実行クラスタは、メモリアクセスユニット (複数可) 464 を有する) を生成するからである。別々のパイプラインが使用される場合、それらのパイプラインの一または複数がアウト・オブ・オーダー問題 / 実行となり、残りは、間順となり得るということも理解されるはずである。

10

20

30

40

50

## 【 0 0 3 5 】

該一組のメモリアクセスユニット464は、メモリユニット470に結合されており、同メモリユニットは、データTLBユニット472を含む。同データTLBユニット472は、データキャッシュユニット474に結合されており、同データキャッシュユニットは、レベル2(L2)のキャッシュユニット476に結合されている。代表的な一実施態様において、メモリアクセスユニット464は、ロードユニット、格納アドレスユニット、および格納データユニットを含み、これらのユニットは各々メモリユニット470内のデータTLBユニット472に結合されている。L2キャッシュユニット476は、一または複数の他のレベルのキャッシュ、および最終的にはメインメモリに結合される。

## 【 0 0 3 6 】

例えば、アウト・オブ・オーダー問題/実行のコア・アーキテクチャでは、パイプライン400を以下のように実装すればよい：命令フェッチ438により、取出しステージ402および長さ復号化ステージ404を実行する；復号化ユニット440により復号化ステージ406を実行する；リネーム/アロケータユニット452により、割付けステージ408およびリネーミングステージ410を実行する；スケジューラユニット(複数可)456により、スケジュールステージ412を実行する；物理レジスタファイル(複数可)のユニット(複数可)458、およびメモリユニット470により、レジスタリード/メモリリードステージ414を実行する；実行クラスタ460により実行ステージ416を実行する；メモリユニット470および物理レジスタファイル(複数可)のユニット(複数可)458により、ライトバック/メモリライトステージ418を実行する；各種ユニットを、例外処理ステージ422に関係させてもよい；ならびにリタイアメントユニット454および物理レジスタファイル(複数可)のユニット(複数可)458により、コミットステージ424を実行する。

## 【 0 0 3 7 】

コア490は、一または複数の命令セットをサポート可能である(例えば、x86命令セット(新しいバージョンでは、一部の拡張が追加済み)；カリフォルニア州サニーバール(Sunnyvale, CA)のMIPSテクノロジーズ社によるMIPS命令セット；およびカリフォルニア州サニーバール(Sunnyvale, CA)のARMホールディングス社によるARM命令セット(NEON等の拡張が追加))。

## 【 0 0 3 8 】

特定の実装では、該コアは、マルチスレッディングをサポート(複数の並列セットのオペレーションまたはスレッドを実行)し、以下に挙げる様々な方法でこれを行う：時間スライスマルチスレッディング、同時マルチスレッディング(単一の物理コアが同時にマルチスレッディングする各スレッド毎に、同物理コアが論理コアを提供する)、およびこれらの組合せ(例えば、時間スライスフェッチおよび復号化、ならびにその後の、Intel(登録商標)ハイパースレッディングテクノロジー(Hyperthreading technology)等の同時マルチスレッディング)。

## 【 0 0 3 9 】

既に示した本プロセッサの実施態様は、個別の命令、データキャッシュユニット434/474、および共有L2キャッシュユニット476も含むが、別の実施態様は、命令とデータの両方用に単一の内部キャッシュ、例えば、レベル1(L1)の内部キャッシュまたは複数レベルの内部キャッシュを有する。一部の実施態様では、当該システムは、内部キャッシュと、コアおよび/またはプロセッサの外側にある外部キャッシュとの組合せを含んでよい。あるいは、キャッシュ全てをコアおよび/またはプロセッサの外側に配置してもよい。

## 【 0 0 4 0 】

図4は、本願開示の一または複数態様によるコンピュータシステム100の幾つかの態様を概略的に例示する。本願明細書において上で言及し、かつ図4で概略的に例示したように、プロセッサ102は、命令および/またはデータを格納するための一または複数のキャッシュ104を具備してよく、例えば、L1キャッシュおよびL2キャッシュを含む

10

20

30

40

50

。キャッシュ104は、一または複数のプロセッサコア123によりアクセス可能である。特定の実装においては、キャッシュ104は、ライト・スルー・キャッシュでもよく、キャッシュライトオペレーション毎に、システムメモリ120に対するライトオペレーションを発生させる。代わりに、キャッシュ104を、ライトバックキャッシュとしてもよく、その場合、キャッシュライトオペレーションは、システムメモリ120に対し、迅速に反映されない。特定の実装においては、キャッシュ104には、キャッシュ・コヒーレンシ・プロトコル、例えば、MESI (Modified - Exclusive - Shared - Invalid) プロトコルを導入し、一または複数のキャッシュに格納したデータの共有メモリに対する一貫性を与える。

#### 【0041】

また、特定の実装においては、プロセッサ102は更に、メモリ120から読み出したデータ/同メモリへ書き込むデータを保持するための、一または複数のリードバッファ127と、一または複数のライトバッファ129とを含む。これらのバッファは、同じサイズでも、異なる一定サイズでもよく、さもなければ、可変サイズでもよい。一例では、リードバッファとライトバッファは、同じ複数のバッファでもよい。また一例では、リードバッファおよび/またはライトバッファは、キャッシュ104の複数のキャッシュエントリでもよい。

#### 【0042】

プロセッサ102は更に、バッファ127および129に関連付けられたメモリ追跡論理131を含んでよい。同メモリ追跡論理は、(例えば、物理アドレスによって識別される)メモリ位置へのアクセスを追跡するように構成された回路構成を特徴とし、これらのメモリ位置は、前もってバッファ127および/または129に格納してあり、結果的に、対応するメモリ位置に対するバッファ127および/または129に格納されているデータのコヒーレンシを提供する。また、特定の実装においては、バッファ127および/または129は、これらに関連付けられたアドレスタグを有し、バッファに格納しているメモリ位置のアドレスを保持することができる。該メモリ追跡論理131を実装する回路構成は、コンピュータシステム100のアドレスバスに通信的に結合しており、従って、アドレスバス上で他のデバイス(例えば、他のプロセッサ、またはダイレクトメモリアクセス(DMA)コントローラ)により指定されたアドレスを読み出すこと、およびこれらのアドレスを、予めバッファ127および/または129に格納したメモリ位置を識別するアドレスと比較することによって、スヌーピングを実施することができる。

#### 【0043】

プロセッサ102は更に、本願明細書で以下により詳細に説明するようなトランザクションの異常終了の場合に実行すべきエラー回復ルーチンのアドレスを保持するためのエラー回復ルーチンアドレスレジスタ135を含んでよい。プロセッサ102は更に、本願明細書で以下により詳細に説明するようなトランザクションエラーコードを保持するためのトランザクションステータスレジスタ137を含んでよい。

#### 【0044】

プロセッサ102がトランザクショナルメモリアクセスを実施できるように、その命令セットは、トランザクション開始(TX\_\_START)命令と、トランザクション終了(TX\_\_END)命令とを含んでよい。TX\_\_START命令は、トランザクションが異常終了した場合に、プロセッサ102が実行すべきエラー回復ルーチンのアドレス、および/またはトランザクションを行うために必要なハードウェアバッファの数を含む一または複数のオペランドを含む。

#### 【0045】

特定の実装において、トランザクション開始命令により、プロセッサに、当該トランザクションを実行するために、リードバッファおよび/またはライトバッファを割り当てさせてもよい。また、特定の実装において、トランザクション開始命令により、更に、保留中の格納オペレーションの全てをプロセッサに処理させて、前に実行したメモリアクセスオペレーションの結果が同じメモリにアクセスする他のデバイスに確実に見えるようにし

10

20

30

40

50

てもよい。更にまた特定の実装において、トランザクション開始命令により更に、当該プロセッサに、データをプリフェッチするのを止めさせてもよい。また更に、特定の実装において、（トランザクションが保留になっている間に発生した割り込みがトランザクションを無効にする可能性があるため）トランザクションが成功する機会を改善するために、トランザクション開始命令により更に、当該プロセッサに、規定されたサイクル数の間、割り込みを無効にさせてもよい。

**【 0 0 4 6 】**

T X \_ S T A R T 命令の処理に応じて、該プロセッサ 1 0 2 は、対応する T X \_ E N D 命令またはエラーの状態の検出によって、終了させることができるオペレーションのトランザクションモードに入ることができる。オペレーションのトランザクションモードでは、プロセッサ 1 0 2 は、それぞれのリードバッファ 1 2 7 および / またはライトバッファ 1 2 9 を介してメモリリードオペレーションおよび / またはメモリライトオペレーションを推論式に（つまり、アクセス中のメモリに対しロックを取得せずに）複数回行うことができる。

10

**【 0 0 4 7 】**

オペレーションのトランザクションモードでは、該プロセッサは、各ロード取得オペレーション毎に、リードバッファ 1 2 7 を割り当てることができる（アクセスされているメモリ位置の内容を既に保持している場合、既存バッファは再利用することができるが、そうでない場合は新しいバッファが割り当てられる）。該プロセッサは更に、各格納取得オペレーション毎に、ライトバッファ 1 2 9 を割り当てることができる（アクセスされているメモリ位置の内容を既に保持している場合、既存バッファは再利用することができるが、そうでない場合は新しいバッファが割り当てられる）。ライトバッファ 1 2 9 は、対応するメモリ位置に対しデータを記憶せずに、ライトオペレーションの結果を保持することができる。メモリ追跡論理 1 3 1 は、指定されたメモリ位置に対する他のデバイスによるアクセスを検出すると、プロセッサ 1 0 2 に対しエラーの状態を信号送信することができる。このエラー信号を受信すると、プロセッサ 1 0 2 は、当該トランザクションをアポートし、対応する T X \_ S T A R T 命令により指定されるエラー回復ルーチンへ制御を移す。他に、T X \_ E N D 命令を受信すると、プロセッサ 1 0 2 は、対応するメモリまたはキャッシュ位置に対し、ライトオペレーションを行う。

20

**【 0 0 4 8 】**

オペレーションのトランザクションモードでは、該プロセッサは、当該トランザクションが無事完了またはアポートかに関わりなく、オペレーションの結果が他のデバイス（例えば、他のプロセッサコアまたは他のプロセッサ）に直ちにに見えるようになるように、一または複数のメモリリードオペレーションおよび / またはライトオペレーションを実行することもできる。トランザクションの範囲内で非トランザクショナルメモリアクセスを実行可能であることで、当該プロセッサのプログラミングの自由度は向上し、また、実行効率を更に改善することができる。

30

**【 0 0 4 9 】**

リードバッファ 1 2 7 および / またはライトバッファ 1 2 9 は、プロセッサ 1 0 2 の最低レベルのデータキャッシュ内で、複数のキャッシュエントリを割り当てることによって、実装することができる。トランザクションがアポートされるならば、リードバッファおよび / またはライトバッファは、無効および / または利用可のように、マークすればよい。本願明細書中上記で言及したように、実行のトランザクションモード中、読み出しおよび / または変更中のメモリに対する他のデバイスによるアクセスを検出すると、トランザクションをアポートすることができる。他のトランザクションアポート条件には、ハードウェアによる割り込み、ハードウェアバッファのオーバーフローおよび / または、トランザクションモードの実行中に検出されたプログラムエラーを入れてよい。更に、特定の実装においては、トランザクションモードの実行の際に検出したエラーの源を示すステータスを保持するために、ステータスフラグ、例えば、ゼロフラグ、キャリーフラグ、および / またはオーバーフローフラグ等を使用することができる。また代わりに、トランザクション

40

50



エラーコードを、トランザクションステータスレジスタ 137 に格納してもよい。

【0050】

実行が対応する TX\_\_END 命令に達し、バッファ 127 および / または 129 にバッファされたデータが、読み出しも変更もされない場合、トランザクションは、正常に完了する。TX\_\_END 命令に達すると、当該プロセッサは、オペレーションのトランザクションモード中にトランザクションアポート条件が発生しなかった旨の確認に応じて、ライトオペレーションの結果を対応するメモリまたはキャッシュ位置にコミットし、予めトランザクションに割り当て済みであるバッファ 127 および / または 127 を開放する。特定の実装においては、プロセッサ 102 は、非トランザクショナルメモリアクセスオペレーションにより読み出しおよび / または変更されたメモリ位置の状態に関係なく、トランザクショナルライトオペレーションをコミットすることができる。

10

【0051】

トランザクションアポート条件を検出すると、当該プロセッサは、当該トランザクションをアポートし、アドレスがエラー回復ルーチンアドレスレジスタ 135 に格納可能であるエラー回復ルーチンへ制御を移せばよい。トランザクションがアポートされるならば、予めトランザクションに割り当てておいたバッファ 127 および / または 129 は、無効および / または利用可のように、マークすればよい。

【0052】

特定の実装においては、プロセッサ 102 は、ネスト形トランザクションにサポートすることができる。ネスト形トランザクションは、別の（外側の）トランザクションの範囲内で実行される TX\_\_START 命令によって開始することができる。ネスト形トランザクションを実施しても、該ネスト形トランザクションの結果に対する外側のトランザクションの範囲内で見えるようにする以外には、外側のトランザクションの状態に対する影響は無い可能性がある。しかし、それらの結果は、外側のトランザクションもコミットするまで、他のデバイスから隠れたままにすることができる。

20

【0053】

ネスト形トランザクションを実施するには、TX\_\_END 命令は、対応する TX\_\_START 命令のアドレスを示すオペランドを含めばよい。更に、エラー回復ルーチンアドレスレジスタ 135 は、同時に有効にすることができる幾つかのネスト形トランザクション用のエラー回復ルーチンアドレスを保持するように拡張することができる。

30

【0054】

ネスト形トランザクションの範囲内で発生しているエラーは、外側のトランザクション全てを無効にすることができる。一連のネスト形トランザクションの範囲内の各エラー回復ルーチンは、対応する外側トランザクションのエラー回復ルーチンを呼び出す役目を果たすことができる。

【0055】

特定の実装においては、トランザクション開始命令およびトランザクション終了命令は、本願明細書中の上に詳述したように、幾つかのロード取得命令および / または格納取得命令をトランザクションモードで実行される一連の命令にグループ化することによって、プロセッサの命令セットの中に存在しているロード取得命令および / または格納取得命令の作用を改変するために使用することができる。

40

【0056】

実施例のコードフラグメントが図 5 に示してあり、トランザクションモード命令の使用を例示している。コードフラグメント 500 により、2つの口座間での送金を説明する。EBX に蓄えた金額は、Src Account から Dst Account へ振替えられる。コードフラグメント 500 は更に、非トランザクショナルメモリオペレーションを例示する。Some Statistic カウンタの内容は、レジスタへロードされ、インクリメントされ、読み出しおよび変更中のメモリのステータスを監視せずに、該メモリへ戻して格納される。Some Statistic カウンタのアドレスに対する格納オペレーションの結果は、直ちにコミットされ、それ故、直ちに他のデバイスの全てで見えるようにな

50

る。

【0057】

図6は、本願開示内容の一または複数態様によるトランザクショナルメモリアクセスの方法の実施例の流れ図を表す。当該方法600は、コンピュータシステムにより実施することが可能であり、同コンピュータシステムは、ハードウェア（例えば、回路構成、専用の論理、および/またはプログラム可能論理）、及びソフトウェア（例えば、コンピュータシステムに対して実行可能で、ハードウェアシミュレーションを行うための命令）、またはこれらの組合せを特徴とする。本方法600および/または、その機能、ルーチン、サブルーチン、もしくはオペレーションの各々は、本方法を実行するコンピュータシステムの一または複数の物理プロセッサによって実施される。2またはそれ以上の機能、ルーチン、サブルーチン、または方法600のオペレーションは、同じメモリにアクセスしている異なるプロセッサによって並列に、あるいは上記の順序とは異なる順序で行われる。一実施例において、図6により示したように、本方法600は、トランザクショナルメモリアクセスを実行するための、図1のコンピュータシステム100により実施することができる。

10

【0058】

図6を参照すると、ブロック610で、プロセッサが、メモリアクセストランザクションを開始することができる。本願明細書中上で言及したように、メモリアクセストランザクションは、専用のトランザクション開始命令によって開始することができる。トランザクション開始は、トランザクションが異常に終了する場合、プロセッサにより実行すべきエラー回復ルーチンのアドレスおよび/または、トランザクションの実施に必要なハードウェアバッファの数を含む一または複数のオペランドを含めばよい。また、特定の実装においては、トランザクション開始命令は、更に、トランザクションを実行するために、該プロセッサにリードバッファおよび/またはライトバッファ割り当てさせることができる。特定の実装において、また更に、トランザクション開始命令は、該プロセッサに保留中の格納オペレーション全てをコミットさせて、前に実行済みのメモリアクセスオペレーションの結果が、同じメモリにアクセスしている他のデバイスに見えるように確実にする。また更に特定の実装においては、トランザクション開始命令は更に、該プロセッサにデータをプリフェッチするのを止めさせることができる。

20

【0059】

ブロック620では、該プロセッサは、メモリ追跡論理に関連付けられた一または複数のハードウェアバッファを介して、一または複数のメモリリードオペレーション推論式に実行することができる。読み出し対象の各メモリブロックは、開始アドレスとサイズによるか、またはアドレス範囲によって識別することができる。メモリ追跡論理は、他のデバイスによって、指定されたメモリアドレスへのアクセスを検出し、エラー状態を当該プロセッサに対し信号送信する。

30

【0060】

ブロック630において、プロセッサは、メモリ追跡論理に関連付けた一または複数のハードウェアバッファを介して一または複数のメモリライトオペレーションを推論式に実行することができる。書き込み対象の各メモリブロックは、開始アドレスとサイズによるか、またはアドレス範囲によって識別することができる。ライトバッファが、メモリライトオペレーションの結果を保持し、対応するメモリ位置にデータをコミットしない。メモリ追跡論理は、他のデバイスによって、指定されたメモリアドレスへのアクセスを検出すると、エラーの状態をプロセッサへ信号送信する。

40

【0061】

ブロック640により概略的に示してあるように、ブロック630により参照されるメモリライトオペレーション中のエラーを検出すると、該プロセッサは、ブロック660で、TX\_\_START命令により指定されたエラー回復ルーチンを実行するが、さもなければ、ブロック670で、処理を継続することができる。

【0062】

50

ブロック670では、該プロセッサは、一または複数のメモリリードオペレーションおよび/またはライトオペレーションを実行し、直ちにコミットすることができる。それらのオペレーションが直ちにコミットされるので、それらの結果は、当該トランザクションが無事完了またはアボートかに関わりなく、他のデバイス（例えば、他のプロセッサコアまたは他のプロセッサ）に対し直ちにに見えるようになる。

【0063】

トランザクション終了命令に到達すると、当該プロセッサは、ブロック670で概略的に示されているように、トランザクションアボート条件がオペレーションのトランザクションモード中に生じていないことを確かめることができる。ブロック610で開始したオペレーションのトランザクションモード中にエラーをブロック670で検出すると、該プロセッサは、ブロック660により概略的に示すように、エラー回復ルーチンを実行し；そうでなければ該プロセッサは、ブロック680で概略的に示されるように、ブロック670で参照した非トランザクショナルメモリアクセスオペレーションにより読み出しおよび/または変更されたメモリ位置の状態に関係なく、当該トランザクションを完了することができる。当該プロセッサは、ライトオペレーションの結果を対応するメモリまたはキャッシュ位置に対しコミットし、当該トランザクションに予め割り当てられていたバッファを開放することができる。ブロック670により参照されているオペレーションを完了すると、本方法は、終了することができる。

10

【0064】

特定の実装において、トランザクションエラーも、オペレーションのトランザクションモードにおいて、幾つかの命令（ロードまたは格納命令など）の実行中に検出することができる。図6では、ブロック620および630から生じている破線は、オペレーションのトランザクションモードにおいて、実行される幾つかの命令からエラー回復ルーチンへの分岐を概略的に示している。

20

【0065】

特定の実装においては、トランザクションエラーも、トランザクション終了命令の実行中に検出することができる。（例えば、他のデバイスによる、トランザクショナルメモリに対するアクセスを報告する論理内に遅延が生じている場合）。図6では、ブロック680からの破線は、トランザクション終了命令からエラー回復ルーチンへの分岐を概略的に示している。

30

【0066】

図7は、本願開示の一または複数態様によるコンピュータシステムの実施例のブロック図である。図7に示すように、マルチプロセッサシステム700は、ポイントツーポイントの相互接続システムであり、また、ポイントツーポイント相互接続部750を介して結合されている第1プロセッサ770と第2プロセッサ780を含む。プロセッサ770および780の各プロセッサは、本願明細書において上でより詳細に説明したように、トランザクショナルメモリアクセスオペレーションおよび/または非トランザクショナルメモリアクセスオペレーションを実行することができるプロセッサ102の何らかのバージョンでよい。

40

【0067】

2つのプロセッサ770、780しか図示していないが、本発明の範囲がそのような制限されるものではない。他の実施態様では、一または複数の追加のプロセッサが、所定のプロセッサ内に存在することが可能である。

【0068】

図示したプロセッサ770、780は、集積メモリコントローラユニット772、782をそれぞれ含む。プロセッサ770は、そのバスコントローラユニットの一部として、ポイントツーポイント（P-P）インタフェース776、778も具備する；同様に、第2プロセッサ780は、P-Pインタフェース786、788を具備する。プロセッサ770、780は、P-Pインタフェース回路778、788を使用しながら、ポイントツーポイント（P-P）インタフェース750を介して情報を交換することができる。図7

50

に示すように、IMC 772、782は、プロセッサをそれぞれのメモリ、即ち、メモリ732およびメモリ734に結合させており、これらは、それぞれのプロセッサに局所的に取り付けられたメインメモリの部分である。

【0069】

プロセッサ770、780は、ポイントツーポイントインタフェース回路776、794、786、798を使用して、個々のP-Pインタフェース752、754経由で、チップセット790と互いに情報を交換することができる。チップセット790も、高性能グラフィックスインタフェース739を介して、高性能グラフィックス回路738と情報を交換することができる。

【0070】

共用キャッシュ(図示無し)は、何れのプロセッサ内にも、あるいは両プロセッサの外側にも、含めることが可能であるが、P-P相互接続部を介して、それぞれのプロセッサと接続されており、プロセッサが低パワーモードにある場合、これらのプロセッサの一方または両方のローカルキャッシュ情報を共用キャッシュに格納することができる。

【0071】

チップセット790は、インタフェース796を介して、第1バス716に結合することができる。更に一実施態様において、第1バス716は、周辺コンポーネント相互接続(PCI)バスもしくはPCI高速バスのようなバス、またはその他第三世代の入出力相互接続バスでよいが、本発明の範囲は、そのように限定されるものではない。

【0072】

図7に示したように、各種入出力装置714は、第1バス716を第2バス720に結合するバスブリッジ718と共に、第1バス716に結合することができる。一実施態様では、第2バス720は、低ピンカウント(LPC)バスでよい。一実施態様においては、様々なデバイスは、第2バス720に結合させることが可能であり、例えば、キーボードおよび/またはマウス722、通信装置727およびディスクドライブや、命令/コードおよびデータ730を含む可能性がある他の大容量記憶装置のような記憶装置ユニット728を含む。更に、オーディオ入出力724は、第2バス720と結合させることができる。なお、ここで、他のアーキテクチャは可能であるということを書き留めておく。例えば、図7のポイントツーポイントアーキテクチャに代えて、システムは、マルチドロップバスまたは他のそのようなアーキテクチャを実装することができる。

【0073】

以下の実施例では、本願開示の一または複数態様による様々な実装を示す。

【0074】

実施例1は、トランザクショナルメモリアクセスの方法である。本方法は、プロセッサにより、メモリアクセストランザクションを開始するステップと；第1メモリ位置に対する、メモリアクセス追跡論理に関連付けられた第1バッファを使用するトランザクショナルリードオペレーション、および第2メモリ位置に対する、該メモリアクセス追跡論理に関連付けられた第2バッファを使用するトランザクショナルライトオペレーションの少なくとも1つを実行するステップと；第3メモリ位置に対する非トランザクショナルリードオペレーション、および第4メモリ位置に対する非トランザクショナルライトオペレーションの少なくとも1つを実行するステップと；該メモリアクセス追跡論理による、該第1メモリ位置および該第2メモリ位置の少なくとも1つに対する、該プロセッサ以外のデバイスによるアクセスの検出に応じて、該メモリアクセストランザクションをアポートするステップと；トランザクションアポート条件の検出の失敗に応じて、該第3メモリ位置の状態と該第4メモリ位置の状態に関わり無く、該メモリアクセストランザクションを完了するステップとを含みなる。

【0075】

実施例2において、実施例1の方法における第1バッファおよび第2バッファが1つのバッファに相当する。

【0076】

10

20

30

40

50

実施例 3 において、実施例 1 の方法における第 1 メモリ位置および第 2 メモリ位置は、一メモリ位置により表すことができる。

【0077】

I 実施例 4 において、実施例 1 の方法における第 3 メモリ位置および第 4 メモリ位置は、一メモリ位置により表すことができる。

【0078】

実施例 5 において、実施例 1 の方法における第 1 バッファと第 2 バッファのうちの少なくとも 1 つが、データキャッシュ内のエントリによって与えられる。

【0079】

実施例 6 において、実施例 1 から 6 の何れかの方法における上記実行オペレーションは、第 2 ライトオペレーションを行うことを含む。

10

【0080】

実施例 7 において、実施例 1 から 6 の何れかの方法における上記完了オペレーションは、上記第 2 バッファから、より高いレベルのキャッシュエントリおよびメモリ位置の一方へデータをコピーすることを含む。

【0081】

実施例 8 において、実施例 1 から 6 の何れかの方法は、割込み、バッファオーバーフローおよびプログラムエラーの少なくとも 1 つの検出に応じて、上記メモリアクセストラッキングをアポートするステップを更に含む。

【0082】

20

実施例 9 において、実施例 1 から 6 の何れかの方法における上記アポートのオペレーションが、上記第 1 バッファおよび上記第 2 バッファの少なくとも 1 つを解放するステップを含む。

【0083】

実施例 10 において、実施例 1 から 6 の何れかの方法における上記開始オペレーションが、保留中のライトオペレーションにコミットするステップを含む。

【0084】

実施例 11 において、実施例 1 から 6 の何れかの方法における上記開始オペレーションが、割込みを禁止するステップを含む。

【0085】

30

実施例 12 において、実施例 1 から 6 の何れかの方法における上記開始オペレーションが、データのプリフェッチを禁止するステップを含む。

【0086】

実施例 13 において、実施例 1 から 6 の何れかの方法は、更に、上記メモリアクセストラッキングを完了する前に、ネスト形メモリアクセストラッキングを開始するステップと；上記メモリアクセス追跡論理に関連付けられた第 3 バッファを使用する第 2 トランザクショナルリードオペレーション、およびメモリアクセス追跡論理に関連付けられた第 4 バッファを使用する第 2 トランザクショナルライトオペレーションの少なくとも 1 つを実行するステップと；上記ネスト形メモリアクセストラッキングを完了するステップを含む。

40

【0087】

実施例 14 において、実施例 13 の方法は、更に、トランザクションアポート条件の検出に応じて、上記メモリアクセストラッキングおよび上記ネスト形メモリアクセストラッキングをアポートするステップを含む。

【0088】

実施例 15 は、処理システムであり、同処理システムは、メモリアクセス追跡論理と；上記メモリアクセス追跡論理に関連付けられた第 1 バッファと；上記メモリアクセス追跡論理に関連付けられた第 2 バッファと；上記第 1 バッファおよび上記第 2 バッファに通信式に結合されたプロセッサコアとを含む。上記処理システムにおいて、上記プロセッサコアは、複数オペレーションを行うように構成されており、上記複数オペレーションは、メ

50

モリアクセストランザクションを開始ステップと；第1メモリ位置に対する、第1バッファを使用するトランザクショナルリードオペレーション、および第2メモリ位置に対する、第2バッファを使用するトランザクショナルライトオペレーションの少なくとも1つを実行するステップと；第3メモリ位置に対する非トランザクショナルリードオペレーション、および第4メモリ位置に対する非トランザクショナルライトオペレーションの少なくとも1つを実行するステップと；上記メモリアクセス追跡論理による、上記第1メモリ位置および上記第2メモリ位置の少なくとも1つに対する、上記プロセッサ以外のデバイスによるアクセスの検出に応じて、上記メモリアクセストランザクションをアポートするステップと；トランザクションアポート条件の検出の失敗に応じて、上記第3メモリ位置の状態と上記第4メモリ位置の状態に関わり無く、上記メモリアクセストランザクションを完了するステップと、を含む。

10

## 【0089】

実施例16は、処理システムであり、同処理システムは、メモリアクセス追跡手段と；上記メモリアクセス追跡手段に関連付けられた第1バッファと；上記メモリアクセス追跡手段に関連付けられた第2バッファと；上記第1バッファおよび上記第2バッファに通信的に結合されたプロセッサコアとを含む。上記処理システムにおいて、上記プロセッサコアは、複数オペレーションを行うように構成されており、上記複数オペレーションは、メモリアクセストランザクションを開始ステップと；第1メモリ位置に対する、第1バッファを使用するトランザクショナルリードオペレーション、および第2メモリ位置に対する、第2バッファを使用するトランザクショナルライトオペレーションの少なくとも1つを実行するステップと；第3メモリ位置に対する非トランザクショナルリードオペレーション、および第4メモリ位置に対する非トランザクショナルライトオペレーションの少なくとも1つを実行するステップと；上記メモリアクセス追跡手段による、上記第1メモリ位置および上記第2メモリ位置の少なくとも1つに対する、上記プロセッサ以外のデバイスによるアクセスの検出に応じて、上記メモリアクセストランザクションをアポートするステップと；トランザクションアポート条件の検出の失敗に応じて、上記第3メモリ位置の状態と上記第4メモリ位置の状態に関わり無く、上記メモリアクセストランザクションを完了するステップと、を含む。

20

## 【0090】

実施例17において、実施例15および16の何れかの処理システムは、更に、データキャッシュを含み、上記第1バッファおよび上記第2バッファの少なくとも1つが、上記データキャッシュの中に備わっている。

30

## 【0091】

実施例18において、実施例15および16の何れかの処理システムは、エラー回復ルーチンのアドレスを格納するためのレジスタを更に含む。

## 【0092】

実施例19において、実施例15および16の何れかの処理システムは、上記メモリアクセストランザクションの状態を格納するためのレジスタを更に含む。

## 【0093】

実施例20において、実施例15および16の何れかの処理システムの上記第1バッファおよび上記第2バッファは、1つのバッファからなる。

40

## 【0094】

実施例21において、実施例15および16の何れかの処理システムの上記第3バッファおよび上記第4バッファは、1つのバッファからなる。

## 【0095】

実施例22において、実施例15および16の何れかの処理システムの上記第1メモリ位置および上記第2メモリ位置は、1つのメモリ位置である。

## 【0096】

実施例23において、実施例15および16の何れかの処理システムの上記第3メモリ位置および上記第4メモリ位置は、1つのメモリ位置である。

50

## 【 0 0 9 7 】

実施例 2 4 において、実施例 1 5 および 1 6 の何れかの処理システムのプロセッサコアは、更に、割込み、バッファオーバーフロー、およびプログラムエラーの少なくとも 1 つの検出に応じて、上記メモリアクセストランザクションをアポートするように構成することができる。

## 【 0 0 9 8 】

実施例 2 5 において、実施例 1 5 の処理システムのプロセッサコアは、更に、次のことを行うように構成することができる。上記メモリアクセストランザクションを完了する前に、ネスト形メモリアクセストランザクションを開始する；上記メモリアクセス追跡論理に関連付けられた第 3 バッファを使用する第 2 トランザクショナルリードオペレーション、およびメモリアクセス追跡論理に関連付けられた第 4 バッファを使用する第 2 トランザクショナルライトオペレーションの少なくとも 1 つを実行する；上記ネスト形メモリアクセストランザクションを完了する。

10

## 【 0 0 9 9 】

実施例 2 6 において、実施例 1 6 の処理システムのプロセッサコアは、更に、次のことを行うように構成することができる。上記メモリアクセストランザクションを完了する前に、ネスト形メモリアクセストランザクションを開始する；上記メモリアクセス追跡手段に関連付けられた第 3 バッファを使用する第 2 トランザクショナルリードオペレーション、およびメモリアクセス追跡手段に関連付けられた第 4 バッファを使用する第 2 トランザクショナルライトオペレーションの少なくとも 1 つを実行する；上記ネスト形メモリアクセストランザクションを完了する。

20

## 【 0 1 0 0 】

実施例 2 7 において、実施例 2 5 および 2 6 の処理システムのプロセッサコアは、トランザクションアポート条件の検出に回答して、上記メモリアクセストランザクションおよびネスト形メモリアクセストランザクションをアポートするように構成することができる。

## 【 0 1 0 1 】

実施例 2 8 は、メモリと上記メモリに結合された処理システムとを含む装置である。同装置において、該処理システムは、実施例 1 から 1 4 の何れかの方法を行うように構成されている。

30

## 【 0 1 0 2 】

実施例 2 9 は、コンピュータ読取り可能非一時的記憶媒体である。同記憶媒体は、実行可能命令を具備し、プロセッサが上記実行可能命令を実行するとき、同プロセッサに：メモリアクセストランザクションを開始させ、第 1 メモリ位置に対する、メモリアクセス追跡論理に関連付けられた第 1 バッファを使用するトランザクショナルリードオペレーション、および第 2 メモリ位置に対する、上記メモリアクセス追跡論理に関連付けられた第 2 バッファを使用するトランザクショナルライトオペレーションの少なくとも 1 つを実行させ、第 3 メモリ位置に対する非トランザクショナルリードオペレーション、および第 4 メモリ位置に対する非トランザクショナルライトオペレーションの少なくとも 1 つを実行させ、上記メモリアクセス追跡論理による、上記第 1 メモリ位置および上記第 2 メモリ位置の少なくとも 1 つに対する、上記プロセッサ以外のデバイスによるアクセスの検出に応じて、上記メモリアクセストランザクションをアポートし；トランザクションアポート条件の検出の失敗に応じて、上記第 3 メモリ位置の状態と上記第 4 メモリ位置の状態に関わり無く、上記メモリアクセストランザクションを完了する。

40

## 【 0 1 0 3 】

詳細な説明の一部は、コンピュータ・メモリの範囲内でデータ・ビット上のオペレーションのアルゴリズムおよび象徴的表象に関して示される。これらのアルゴリズムの説明および表現は、データ処理技術に熟練した人々によって他の当業者に最も効果的にそれらの仕事の要旨を伝えるために用いる手段である。一アルゴリズムはここで、また一般に、所望の結果に至る首尾一貫した一連のオペレーションであると考えられている。オペレーシ

50

ョンは、物理量の物理的操作を必要とするオペレーションである。通常、必然的ではないが、これらの量は格納、転送、合成、比較、その他操作を行うことができる電気信号か磁気信号の形をとる。これらの信号をビット、値、要素、シンボル、文字、用語、数字等で参照することは、主に一般的な使用法の理由から便利であることが時々証明されている。

#### 【0104】

しかし、これらの全て、および類似の用語は、適切な物理量に関連付けられたものであり、それら物理量に対する便宜上のラベルに過ぎない。他に特に言及しなければ、上記説明から明白であるように、全体にわたって説明、用語（「暗号化」、「解読」、「記憶」、「供給」、「誘導・派生」、「入手」、「受け取り」、「認証」、「削除」、「実行」、「要求」、「通信」等）を使用した考察では、計算システムのレジスタおよびメモリの範囲内で、物理（e.g. 電子）量として表されるデータを操作し、計算システムのメモリもしくはレジスタ、その他情報記憶装置、伝送または表示装置の範囲内で物理量として同様に表される別のデータへ変換させる計算システムまたは類似の電子計算装置の動作およびプロセスに言及しているという事実を認識されたい。

10

#### 【0105】

本願明細書で使用されている「実施例」または「典型的」なる文言は、典型もしくは実例、または事例となっていることを意味する。本願明細書に「実施例」または「典型的」と記載されている如何なる態様またはデザインも、必ずしも他の態様またはデザインよりも好適または有利であるかのように解釈されるべきものではない。むしろ、「実施例」または「典型的」なる文言の使用により、具体的なやり方で概念を示すことを意図している。本願で使用されているように、「または」なる言葉は、排他的な「または」ではなく、包括的「または」を意味するように意図した。すなわち、別に指定が無いが、または前後関係から明らかでなければ、「Xは、AまたはBを含む」は、任意のありのままの包括的順序を意味するように意図している。即ち、XがAを含み、XがBを含み、またはXがAとBの両方を含むとすると、上記の事例の何れであっても、「Xは、AまたはBを含む」という条件が満たされる。その上、本願および添付のクレームで使用されている冠詞"a"および"a n"は、一般に、別に指定しない限り、または前後関係から単数形で指示が出されていることが明らかにならない限りは、「一または複数」を意味すると解釈すべきである。更に、全体にわたる「実施態様(an embodiment)」もしくは「一実施態様(one embodiment)」または「実装(an implementation)」もしくは「一実装(one implementation)」なる語の使用は、そのように説明されていなければ、同じ実施態様または実装を意味するには意図されていない。また、本願明細書において使用されている「第1」、「第2」、「第3」、「第4」等の語は、異なる要素をラベルのように差別化するためのものであり、必ずしも数の記号どおりに序数の意味を持っているわけではない。

20

30

#### 【0106】

本願明細書に記載の実施態様は、本願明細書に記載のオペレーションを行うための装置にも関係する可能性がある。この装置は、求められている目的に応じて特別に構築されるか、またはコンピュータに格納されているコンピュータプログラムによって選択的に活性化されるか再構成される汎用コンピュータを含むものでよい。そのようなコンピュータプログラムは、非一時的コンピュータ読取り可能記憶媒体に格納することが可能であり、この非一時的コンピュータ読取り可能記憶媒体は、任意の種類ディスク、限定されるものではないが、例えば、フロッピー（登録商標）ディスク、光ディスク、CD ROMおよび磁気光ディスク、リードオンリーメモリ（ROMs）、ランダムアクセスメモリ（RAMs）、EPROMs、EEPROMs、磁気または光カード、フラッシュメモリ、または電子命令を格納するのに好適な任意の種類媒体である。「コンピュータ読取り可能記憶媒体」という文言は、一または複数命令セットを格納する単一媒体または複数媒体（例えば、集中もしくは分散データベース、ならびに/または関連付けられたキャッシュおよびサーバー）を含むように解釈すべきである。「コンピュータ読取り可能媒体」なる語も

40

50



、マシンによる実行用の命令セットを格納し、符号化し、あるいは担持することができる任意の媒体であって、本実施態様の技法のうちの任意の一または複数の技法をマシンに行わせる、任意の媒体を含むように解釈すべきである。従って、「コンピュータ読取り可能記憶媒体」なる語は、限定されるものではないが、固体メモリ、光学媒体、磁気媒体、当該マシンによる実行用の命令セットを格納することができる任意の媒体であって、当該マシンに対し、本実施態様の任意の一または複数の技法を行わせる媒体を含むように解釈すべきである。

【0107】

本願明細書に記載のアルゴリズムおよびディスプレイは、任意の特定コンピュータまたはその他装置に本質的に関係しているわけではない。本願明細書に記載の教示内容に従って、様々な汎用システムをプログラムと共に使用することができるし、あるいは、必要な方法オペレーションを行うためのより専門化された装置を構成するには、都合が良いことが証明された。多種多様なこれらのシステム用に必要な構造は、下の説明から明らかになるだろう。加えて、本実施態様は、何らかの特別なプログラミング言語を参照しながら、記述されたものではない。本願明細書に記載した実施態様の教示内容を実践するために、様々なプログラミング言語が使用できることが理解されるであろう。

10

【0108】

以上の説明には、一部の実施態様について、十分な理解が得られるように、特定のシステム、コンポーネント、方法等の実施例のような多数の具体的細部が記載されている。しかし、少なくとも一部の実施態様が、これらの具体的細部無しで実施可能であることは当業者には明らかであろう。また、他の例では、周知のコンポーネントまたは方法が詳細に記述されていないし、あるいは本実施態様を不必要に曖昧にすることを回避するために、単純なブロック図の形式で提示されている。このように、上記に記載の具体的細部は、単に典型的なだけである。特定の実装は、これらの典型的な詳細から変化することができて、本実施例の範囲内であるために、まだ意図されることができる。

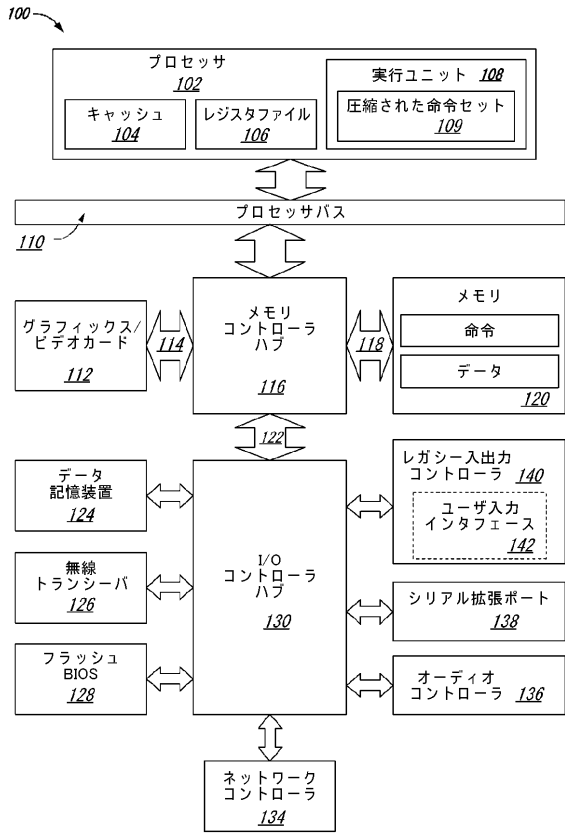
20

【0109】

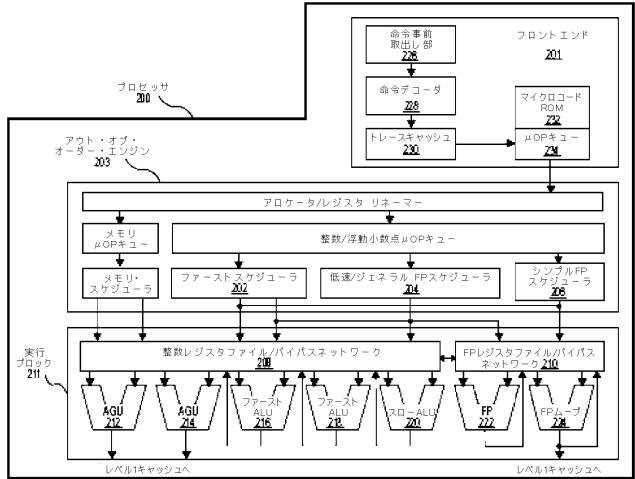
上記説明は例示的なものであり限定的ではないことが意図されていることは、理解されるはずである。前記説明を読み理解すれば、多くの他の実施態様は、当業者にとって明白なものとなる。従って、本実施態様の範囲は、この種の請求が受ける等価物の全範囲とともに、本実施態様の範囲は、添付の特許請求の範囲を参照して決定しなければならない。

30

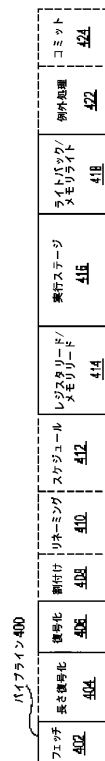
【図1】



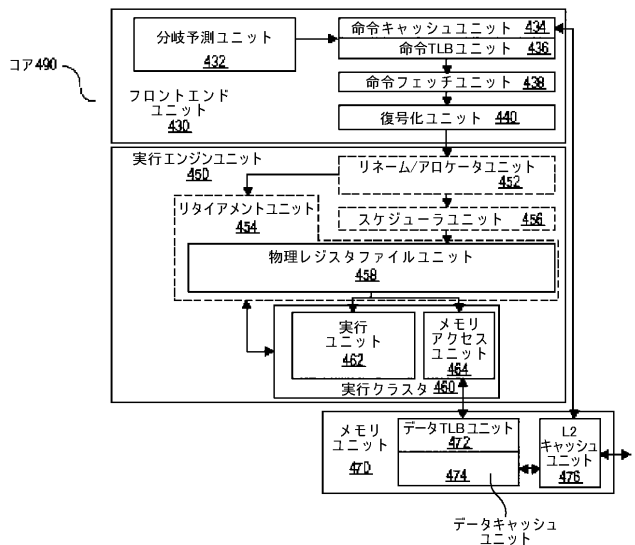
【図2】



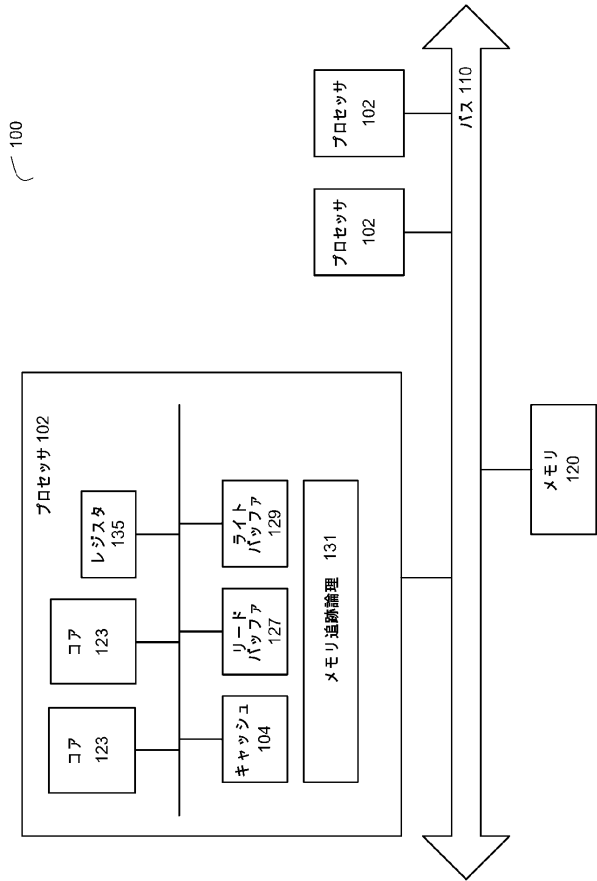
【図3a】



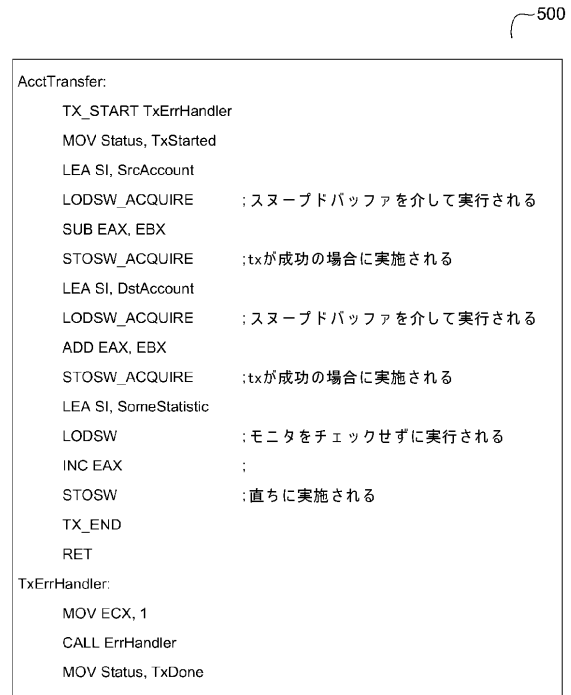
【図3b】



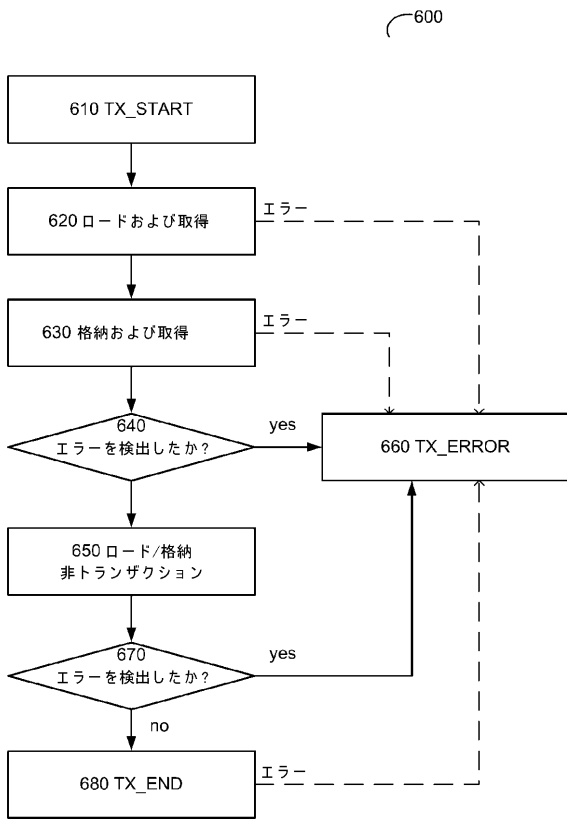
【 図 4 】



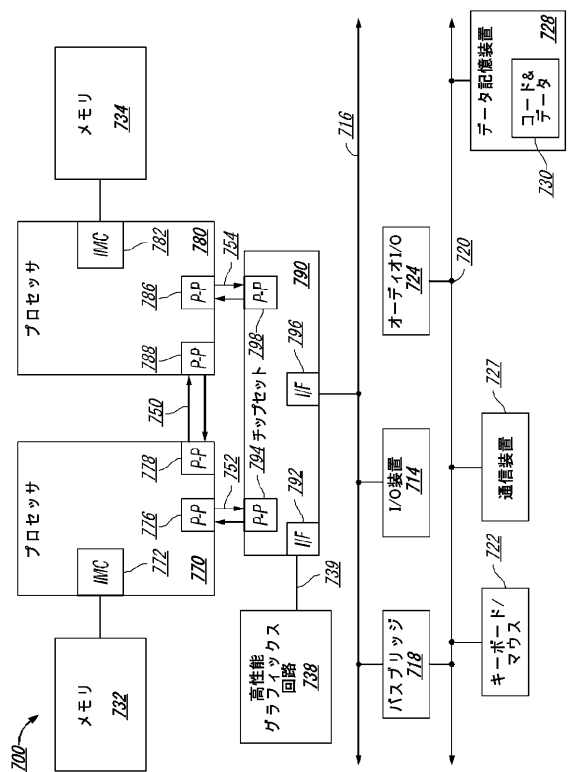
【 図 5 】



【 図 6 】



【 図 7 】



---

フロントページの続き

- (72)発明者 ハーン、スコット ディー .  
アメリカ合衆国 9 5 0 5 4 カリフォルニア州・サンタクララ・ミッション カレッジ ブーレ  
バード・2 2 0 0 インテル・コーポレーション内
- (72)発明者 トール、ブレット エル .  
アメリカ合衆国 9 5 0 5 4 カリフォルニア州・サンタクララ・ミッション カレッジ ブーレ  
バード・2 2 0 0 インテル・コーポレーション内
- (72)発明者 ヒントン、グレン ジェイ .  
アメリカ合衆国 9 5 0 5 4 カリフォルニア州・サンタクララ・ミッション カレッジ ブーレ  
バード・2 2 0 0 インテル・コーポレーション内

【外国語明細書】  
2014194754000001.pdf