(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第4691105号
(P4691105)
(45)発行日　平成23年6月1日(2011.6.1)　(24)登録日　平成23年2月25日(2011.2.25)

(51)Int.Cl.　　F I
G06F 11/16 (2006.01)　　G06F 11/16 310D
G06F 11/18 (2006.01)　　G06F 11/18 310C

請求項の数 28　(全 10 頁)

(21)出願番号　特願2007-533793 (P2007-533793)
(86)(22)出願日　平成17年9月29日 (2005.9.29)
(65)公表番号　特表2008-515064 (P2008-515064A)
(43)公表日　平成20年5月8日 (2008.5.8)
(86)国際出願番号　PCT/US2005/035375
(87)国際公開番号　WO2006/039595
(87)国際公開日　平成18年4月13日 (2006.4.13)
審査請求日　平成19年3月23日 (2007.3.23)
(31)優先権主張番号　10/953,887
(32)優先日　平成16年9月29日 (2004.9.29)
(33)優先権主張国　米国 (US)

(73)特許権者　591003943
インテル・コーポレーション
アメリカ合衆国　９５０５２　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・２２００
(74)代理人　100104156
弁理士　龍華　明裕
(72)発明者　マッカジー、シャブヘンドゥ
アメリカ合衆国、０１７０２　マサチューセッツ州、フレーミングハム、ゲーツ　ストリート　４３
(72)発明者　エメル、ジョエル
アメリカ合衆国、０１７２０　マサチューセッツ州、アクトン、ブラックホース　ドライブ　２０

最終頁に続く

(54)【発明の名称】冗長マルチスレッド環境でのチェッカ命令の実行

(57)【特許請求の範囲】
【請求項１】
コンピュータが、互いに離れて実行される先行スレッドおよび追跡スレッド内に、それぞれのスレッドからの出力値を指定するチェッカ命令を生成し、前記先行スレッド内に、前記出力値を指定する選択された命令を残して前記チェッカ命令を追加し、前記追跡スレッド内の前記選択された命令を前記生成した前記チェッカ命令で置き換える段階と、
前記コンピュータが、前記先行スレッドおよび前記追跡スレッドからの対応するチェッカ命令を待つ段階と、
前記コンピュータが、前記先行スレッドおよび前記追跡スレッドからの前記対応するチェッカ命令を比較する段階と
を備える方法。
【請求項２】
前記先行スレッド内に前記チェッカ命令を追加する場合に、前記選択された命令の前に前記チェッカ命令を挿入する
請求項１に記載の方法。
【請求項３】
前記チェッカ命令は前記先行スレッドおよび前記追跡スレッドの対応するパイプラインでパイプライン処理される
請求項１または２に記載の方法。
【請求項４】

前記コンピュータが、前記先行スレッドおよび前記追跡スレッドからの前記対応するチェッカ命令の比較が一致した場合に、前記チェッカ命令をコミットする段階をさらに備える請求項１から３のいずれか一項に記載の方法。
【請求項５】
前記コンピュータが、前記先行スレッドおよび前記追跡スレッドからの前記対応するチェッカ命令の比較が一致した場合に、前記先行スレッドの前記選択された命令に含まれる前記出力値を記憶する段階
をさらに備える請求項４に記載の方法。
【請求項６】
前記先行スレッドおよび前記追跡スレッドは、単一のプロセッサによって実行される
請求項１から５のいずれか一項に記載の方法。
【請求項７】
前記先行スレッドおよび前記追跡スレッドは、複数のプロセッサによって実行される
請求項１から５のいずれか一項に記載の方法。
【請求項８】
前記コンピュータは、前記チェッカ命令によって、前記先行スレッドの命令をホールドすることなく、前記追跡スレッドの前記対応するチェッカ命令が現れるまでリタイアポインタをホールドする
請求項１から７のいずれか一項に記載の方法。
【請求項９】
スレッドからの出力値を指定するチェッカ命令が当該出力値を指定する選択された命令を残してスレッド内に追加された先行スレッドを実行する回路である先行スレッド回路と、
スレッドからの出力値を指定するチェッカ命令でスレッド内の前記選択された命令が置き換えられた追跡スレッドを実行する回路である追跡スレッド回路と、
互いに離れて実行される前記先行スレッドおよび前記追跡スレッドからの対応するチェッカ命令を比較して、前記先行スレッドおよび前記追跡スレッドからの対応するチェッカ命令をコミットするコミットユニットと
を備える装置。
【請求項１０】
前記先行スレッド回路および前記追跡スレッド回路はパイプラインを有する
請求項９に記載の装置。
【請求項１１】
前記先行スレッドおよび前記追跡スレッドは、単一のプロセッサによって実行される
請求項９または１０に記載の装置。
【請求項１２】
前記先行スレッドおよび前記追跡スレッドは、複数のプロセッサによって実行される
請求項９または１０に記載の装置。
【請求項１３】
前記先行スレッド回路および前記追跡スレッド回路に結合されるバッファ
をさらに備える請求項９から１２のいずれか一項に記載の装置。
【請求項１４】
前記先行スレッドの前記チェッカ命令および前記追跡スレッドの前記チェッカ命令は、前記バッファにおいて対応するチェッカ命令を待つ
請求項１３に記載の装置。
【請求項１５】
前記選択された命令が指定する出力値は、前記先行スレッドおよび前記追跡スレッドのそれぞれの前記対応するチェッカ命令の比較が一致する場合に記憶される
請求項９から１４のいずれか一項に記載の装置。
【請求項１６】

前記コミットユニットは、前記対応するチェッカ命令が一致しない場合にエラーを生成する
請求項９から１５のいずれか一項に記載の装置。
【請求項１７】
前記先行スレッドにおいて、前記チェッカ命令は前記選択された命令の前にバイナリトランスレータにより配置される
請求項９から１６のいずれか一項に記載の装置。
【請求項１８】
前記選択された命令はストア命令である
請求項９から１７のいずれか一項に記載の装置。
【請求項１９】
前記チェッカ命令は、前記先行スレッドの命令をホールドすることなく、前記追跡スレッドの前記対応するチェッカ命令が現れるまでリタイアポインタをホールドする
請求項９から１８のいずれか一項に記載の装置。
【請求項２０】
第１プロセッサと、
第２プロセッサへの第１インタフェースと、
入／出力装置への第２インタフェースと、
前記第２インタフェースに結合されるオーディオ入出力装置と
を備え、
前記第１プロセッサは、
スレッドからの出力値を指定するチェッカ命令が当該出力値を指定する選択された命令を残してスレッド内に追加された先行スレッドを実行する回路である先行スレッド回路と、
スレッドからの出力値を指定するチェッカ命令で、スレッド内の前記選択された命令が置き換えられた追跡スレッドを実行する回路である追跡スレッド回路と、
互いに離れて実行される前記先行スレッドおよび前記追跡スレッドからの対応するチェッカ命令を比較して、前記先行スレッドおよび前記追跡スレッドからの対応するチェッカ命令をリタイアさせるリタイアユニットと
を有するシステム。
【請求項２１】
前記先行スレッド回路および前記追跡スレッド回路はパイプラインを含む
請求項２０に記載のシステム。
【請求項２２】
前記選択された命令が指定する出力値は、前記先行スレッドおよび前記追跡スレッドのそれぞれの前記対応するチェッカ命令の比較が一致する場合に記憶される
請求項２０または２１に記載のシステム。
【請求項２３】
前記先行スレッド回路および前記追跡スレッド回路に結合されるバッファ
をさらに備える請求項２０から２２のいずれか一項に記載のシステム。
【請求項２４】
前記リタイアユニットは、前記対応するチェッカ命令が一致しない場合にエラーを生成する
請求項２０から２３のいずれか一項に記載のシステム。
【請求項２５】
前記先行スレッド内において、前記チェッカ命令は前記選択された命令の前にバイナリトランスレータにより配置される
請求項２０から２４のいずれか一項に記載のシステム。
【請求項２６】
前記選択された命令はストア命令である

請求項２０から２５のいずれか一項に記載のシステム。
【請求項２７】
前記第１インタフェースおよび前記第２インタフェースはポイントツーポイントインタフェースである
請求項２０から２６のいずれか一項に記載のシステム。
【請求項２８】
前記チェッカ命令は、前記先行スレッドの命令をホールドすることなく、前記追跡スレッドの前記対応するチェッカ命令が現れるまでリタイアポインタをホールドする
請求項２０から２７のいずれか一項に記載のシステム。
【発明の詳細な説明】
【背景技術】
【０００１】
現在の冗長実行システムは一般に、自己検査式であり且つハードウェアで実装されるチェッカ回路を採用している。チェッカ回路に類似するのは、２つのスレッド（たとえば、ストアアドレス及びデータ）からの結果を比較する比較命令である。両方のスレッドで比較命令を複製し、複製を介して自己検査の効果を得ることが可能であり得る。
【特許文献１】米国特許第６４６３５７９号明細書
【発明の開示】
【発明が解決しようとする課題】
【０００２】
不都合なことに、比較命令を複製することにより、アーキテクチャは冗長マルチスレッド（ＲＭＴ）の性能利点を失うことになる。ＲＭＴの性能利点は、先行スレッドが追跡スレッドのキャッシュミス及び分岐予測をプリフェッチできるように十分に離れた先行スレッド及び追跡スレッドを有することから生じる。比較命令が複製される場合、キューの追加が必要なだけでなく（より高いオーバーヘッドを招く）、アーキテクチャは、両方向で求められる同期により２つのスレッドを十分離れた状態で保持することができなくなり得る。したがって、必要なのは、ＲＭＴの性能利点を犠牲にすることなくより低い故障率を実現できる命令である。
【課題を解決するための手段】
【０００３】
本発明の各種特徴が、同様の参照番号が概して全図面を通して同じパーツを指す添付図面に示す好ましい実施形態の以下の説明から明らかになろう。図面は必ずしも一定の縮尺で描かれておらず、それに代えて本発明の原理を示すことに重点が置かれている。
【発明を実施するための最良の形態】
【０００４】
以下の説明では、限定ではなく説明のために、特定の構造、アーキテクチャ、インタフェース、技法等の特定の詳細を記して、本発明の各種態様の完全な理解を提供する。しかし、本発明の各種態様を他の例ではこれら特定の詳細から離れて実施可能なことが、本開示の恩恵を受ける当業者には明白であろう。特定の場合では、本発明の説明を不必要な詳細で曖昧にしないように、既知の装置、回路、及び方法についての説明を省いている。
【０００５】
冗長マルチスレッド環境でのチェッカ命令の方法及び装置について記載する。以下の説明では、説明のために、多くの特定の詳細を記して本発明の完全な理解を提供する。しかし、本発明をこれら特定の詳細なしで実施可能なことが当業者には明白であろう。
【０００６】
図１は、冗長マルチスレッドアーキテクチャの一実施形態のブロック図である。冗長マルチスレッドアーキテクチャでは、故障は、プログラムの２つのコピーを別個のスレッドとして実行することによって検出することができる。
【０００７】
各スレッドには同一の入力が提供され、出力が比較されて、エラーが発生したか否かが

判断される。冗長マルチスレッドは、本明細書では「複製範囲（sphere of replication）」と呼ぶ概念に関連して説明することができる。複製範囲とは、論理的又は物理的な冗長動作の境界である。

【０００８】

複製範囲１００内の構成要素（たとえば、先行スレッド１０５を実行しているプロセッサ及び追跡スレッド１１０を実行しているプロセッサ）が冗長実行の対象である。これとは対照的に、複製範囲１００外の構成要素（たとえば、メモリ１１５）は冗長実行の対象ではない。故障保護は他の技法、たとえばメモリ１１５の誤り修正符号によって提供される。他の装置は複製範囲１００外であってもよく、且つ／又は他の技法を使用して故障保護を複製範囲１００外の装置に提供することができる。

【０００９】

複製範囲１００に入るデータは、データを複製し且つデータのコピーを先行スレッド１０５及び追跡スレッド１１０に送る入力複製エージェント１２０を通して入る。同様に、複製範囲１００を出るデータは、データを比較し且つエラーが発生したか否かを判断する出力比較エージェント１２５を通って出る。複製範囲１００の境界の変更は、性能とハードウェア量との間のトレードオフである。たとえば、メモリ１１５を複製すると、ストア命令の出力比較を避けることによりメモリへのより高速でのアクセスが可能になるが、システムでのメモリ量を２倍にすることによってシステムコストが増大する。

【００１０】

本発明の一実施形態は、ＲＭＴのソフトウェア実装でのチェッカ回路を検査するメカニズムを提案する。ＲＭＴはコミットされた命令の出力を比較する（命令毎の比較を必要とする）ため、これはソフトウェアで実装することもできる。ＲＭＴのソフトウェア実装があらゆる命令を比較する場合、相当なオーバーヘッドが発生する。しかしそれに代えて、ＲＭＴは、ストア命令のみの比較及びロード命令のみの複製を可能にし、これはＲＭＴ実装のソフトウェアオーバーヘッドを大幅に低減することができる。

【００１１】

図２は、チェッカ命令を生成する一方法を示す。まず、大半のコンピュータでのようにコンパイラが命令を生成する。コンパイラから、コンピュータはここで、一連のストア命令であり得るがこれに限定されないバイナリプログラムを有する（２００）。次に、バイナリトランスレータがバイナリプログラム中の各ストア命令の前にチェッカ命令を挿入することができる（２０５）。バイナリトランスレータは当該技術分野において既知の任意のバイナリトランスレータであってもよい。バイナリプログラムの変換後、システムは先行スレッド及び追跡スレッドの両方にバイナリプログラムを作成する。先行スレッドのバイナリプログラムは、チェッカ命令をストア命令に追加する（２１０）。追跡スレッドのバイナリプログラムは、ストア命令を先行スレッドのピアチェッカ命令で置き換える（２１５）。

【００１２】

図３は、チェッカ命令の一実施態様を示す。ＲＭＴが比較を要求すると、プロセッサはチェッカ命令を先行スレッド及び追跡スレッドの両方において発することができる（３００）。各チェッカ命令は各スレッドから６４ビット量を伝達することができる。チェッカ命令は、各パイプラインの終わりにあるバッファに達するまで各スレッドの個々のパイプラインを独立して移行することができる（３０５）。チェッカ命令は、ピアチェッカ命令をバッファで待つ（３１０）。これら２つのチェッカ命令は次いで、伝達している６４ビット量の比較を行うことができる（３１５）。一致しない場合、両方がエラーを報告することができる。一致する場合、プロセッサにチェッカ命令をコミットさせることができる（３２０）。パイプラインは、ＣＭＰ中の異なるプロセッサからであっても、又はＳＭＴプロセッサと同じマルチスレッドプロセッサからであってもよい。

【００１３】

この実施態様では、チェッカ命令は、命令キューからの処理命令からの先行する命令を中断しない。むしろ、リタイアポインタのみを、追跡スレッドからの対応するチェッカ命

令が現れるまで中断するだけである。また、環境がＲＭＴ環境ではない場合、チェッカ命令はＮＯＰとして扱われることができる。

【００１４】

図４は、２つのパイプラインを通って流れるチェッカ命令の一例のブロック図である。ストア命令Ｒ１→［Ｒ２］が、レジスタＲ１中の値をレジスタＲ２中のアドレスによってポイントされるメモリロケーションに記憶するものと仮定する。このストア命令は、チェッカ命令と共に先行スレッド及び追跡スレッドの両方で複製することができる。先行スレッドのストア命令は、チェッカ命令及びストア命令の両方を以下に示すように含むことができる。

```
Checkerinst R1
Checkerinst R2
Store: R1→[R2]
```

したがって、先行スレッドは、パイプライン４００を移行する時にストア命令とともにチェッカ命令を含むことができる。

【００１５】

追跡スレッドのストア命令は、以下に示すようにチェッカ命令のみを含むことができる。

```
Checkerinst R1
Checkerinst R2
```

したがって、追跡スレッドはパイプライン４０５を通って移行するストア命令を有さない。

【００１６】

パイプライン４００中の先行スレッドからのcheckerinst R1は、バッファ４１０においてピアチェッカ命令を待つ。パイプライン４０５中の追跡スレッドからのcheckerinst R１は、バッファ４１５においてピアチェッカ命令を待つ。チェッカ命令は常にその相手方すなわちピアを探すか、又は待つ。ミラースレッドがある場合、チェッカ命令はバッファ４１０、４０５においてそのスレッドを探すか、又は待ち、ミラースレッドがそこにあることを確認してから、チェッカ命令を比較する。

【００１７】

先行スレッドからのcheckerinst R1及び追跡スレッドからのcheckerinst R1は、コミット順によりペアになることができ、レジスタ指示子とＲ１の値を比較して、レジスタがいずれのエラーも有さなかったことを保証する。エラーが見つからない場合、チェッカ命令はコミットされる（４２０）。チェッカ命令がコミットされると、Ｒ１の値が記憶される。Ｒ１の値は、コミットポイントを通ってから記憶される。したがって、システムは、従来のように検査をストア毎に行うのではなくすべてのストアを同時に検査することができる。

【００１８】

図５は、マルチスレッドプロセッサ用の環境を提供することができるシステムのブロック図である。図５に示すシステムは或る範囲のシステムを表すことを意図される。代替のシステムは、より多数、より少数、且つ／又は異なる構成要素を備えることができる。

【００１９】

システム５００は、情報を伝達するバス５１０又は他の通信装置、及びバス５１０に結合されて情報を処理するプロセッサ（複数可）５２０を備える。システム５００は、メモリコントローラ５３０を介してバス５１０に結合されて、情報及びプロセッサ（複数可）５２０が実行する命令を記憶するランダムアクセスメモリ（ＲＡＭ）又は他のダイナミックメモリ並びにスタティックメモリ、たとえばハードディスク又は他の記憶装置５３５（メモリと呼ぶ）をさらに備える。メモリ５３５はまた、プロセッサ（複数可）５２０による命令の実行中に一時変数又は他の中間情報を記憶するためにも使用することができる。メモリコントローラ５３０は、１つ又は複数の種類のメモリ及び／又は関連するメモリ装置を制御する１つ又は複数の構成要素を備えることができる。システム５００は、バス５

１０に結合されてプロセッサ（複数可）５２０のためにスタティックな情報及び命令を記憶する読み取り専用メモリ（ＲＯＭ）及び／又は他のスタティック記憶装置５４０も備える。

【００２０】

システム５００はまた、バス５１０を介して入／出力（Ｉ／Ｏ）インタフェース５５０にも結合することができる。Ｉ／Ｏインタフェース５５０はＩ／Ｏ装置５５５へのインタフェースを提供し、Ｉ／Ｏ装置５５５はたとえば、コンピュータユーザに情報を表示する陰極線管（ＣＲＴ）又は液晶ディスプレイ（ＬＣＤ）、英数字及び他のキーを含む英数字入力装置、並びに／或いはマウス、トラックボール、又はカーソル方向キー等のカーソル制御装置を含むことができる。システム５００は、ローカルエリアネットワーク等のネットワークへのアクセスを有無線で提供するネットワークインタフェース５６０をさらに備える。

【００２１】

命令は、メモリ５３５に磁気ディスク、読み取り専用メモリ（ＲＯＭ）集積回路、ＣＤ－ＲＯＭ、ＤＶＤ等の記憶装置から、有無線のリモート接続（たとえば、ネットワークインタフェース８６０を介してネットワーク経由で）等を介して提供される。

【００２２】

これより図６を参照すると、システム６００は概して、プロセッサ、メモリ、及び入／出力装置が複数のポイントツーポイントインタフェースによって相互接続されるシステムを示す。システム６００はいくつかのプロセッサも備えることができ、明瞭にするために、そのうちの２つプロセッサ６０５、６１０のみを示す。プロセッサ６０５、６１０はそれぞれ、ローカルメモリコントローラハブ（ＭＣＨ）６１５、６２０を備えてメモリ６２５、６３０と接続することができる。プロセッサ６０５、６１０は、ポイントツーポイントインタフェース回路６４０、６４５を使用して、ポイントツーポイントインタフェース６３５を介してデータを交換することができる。プロセッサ６０５、６１０はそれぞれ、ポイントツーポイントインタフェース回路６６５、６７０、６７５、６８０を使用して、個々のポイントツーポイントインタフェース６５５、６６０を介してチップセット６５０とデータを交換することができる。チップセット６５０はまた、高性能グラフィックスインタフェース６９０を介して高性能グラフィックス回路６８５とデータを交換することもできる。

【００２３】

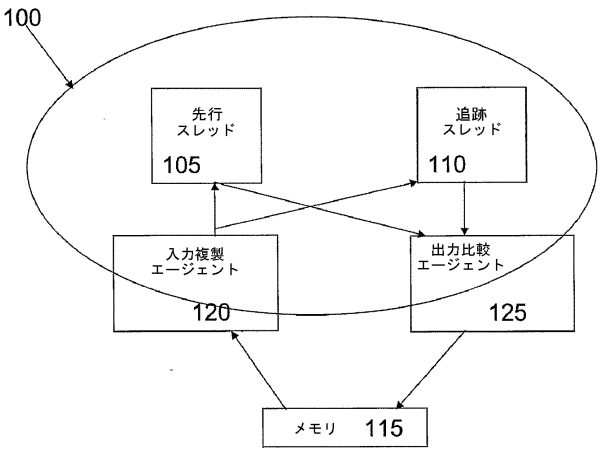
チップセット６５０は、バスインタフェース６９５を介してバス６１６とデータを交換することができる。いずれのシステムでも、いくつかの実施形態では低性能グラフィックスコントローラ、ビデオコントローラ、及びネットワーキングコントローラを含む様々な入／出力Ｉ／Ｏ装置６１４がバス６１６上にあり得る。いくつかの実施形態では、別のバスブリッジ６１８を使用して、バス６１６とバス６２０との間でデータを交換できるようにすることができる。いくつかの実施形態では、バス６２０は、小型コンピュータシステムインタフェース（ＳＣＳＩ）バス、統合ドライブエレクトロニクス（ＩＤＥ）バス、又はユニバーサルシリアルバス（ＵＳＢ）バスであることができる。追加のＩ／Ｏ装置をバス６２０に接続することができる。これらは、キーボード、マウスを含むカーソル制御装置６２２、オーディオＩ／Ｏ６２４、モデム及びネットワークインタフェースを含む通信装置６２６、及びデータ記憶装置６２８を含むことができる。ソフトウェアコード６３０は、データ記憶装置６２８に記憶することができる。いくつかの実施形態では、データ記憶装置６２８は、固定磁気ディスク、フロッピーディスクドライブ、光学ディスクドライブ、光磁気ディスクドライブ、磁気テープ、又はフラッシュメモリを含む不揮発性メモリであることができる。

【００２４】

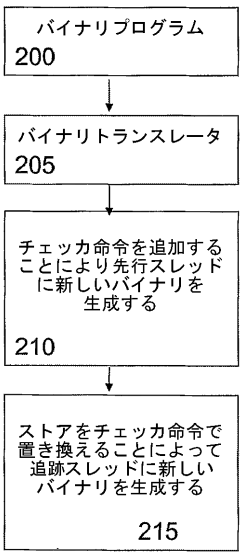
本明細書全体を通して、「命令」という語は、命令、マクロ命令、命令バンドル、又はプロセッサ動作を符号化するために使用される任意の数の他のメカニズムを全体的に指すために使用される。

【００２５】
以下の説明では、限定ではなく説明のために、特定の構造、アーキテクチャ、インタフェース、技法等の特定の詳細を記して、本発明の各種態様の完全な理解を提供する。しかし、本発明の各種態様を他の例ではこれら特定の詳細から離れて実施可能なことが、本開示の恩恵を受ける当業者には明白であろう。特定の場合では、本発明の説明を不必要な詳細で曖昧にしないように、既知の装置、回路、及び方法についての説明を省いている。
【図面の簡単な説明】
【００２６】
【図１】マルチスレッドアーキテクチャの一実施形態のブロック図である。
【図２】チェッカ命令を生成する一方法を示すフローチャートである。
【図３】いずれかのスレッドでのチェッカ命令の一実施態様を示すフローチャートである。
【図４】チェッカ命令の一実施形態のブロック図である。
【図５】マルチスレッドプロセッサの環境を提供することができるシステムのブロック図である。
【図６】マルチスレッドプロセッサの環境を提供することができる代替のシステムのブロック図である。

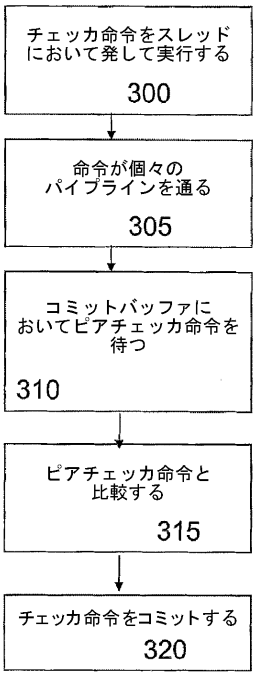
【図１】



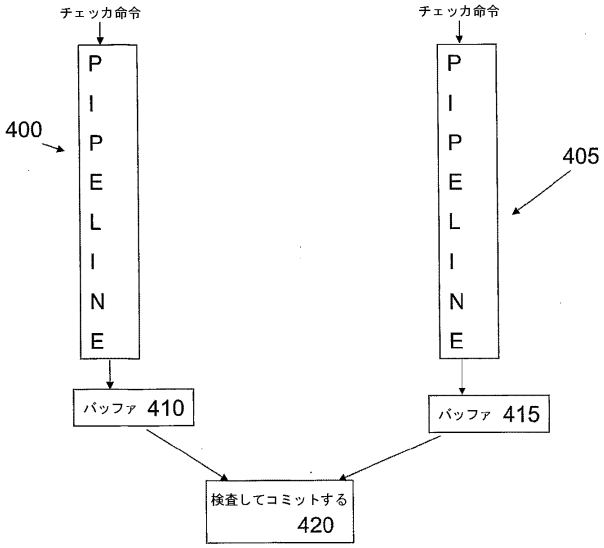
【図２】



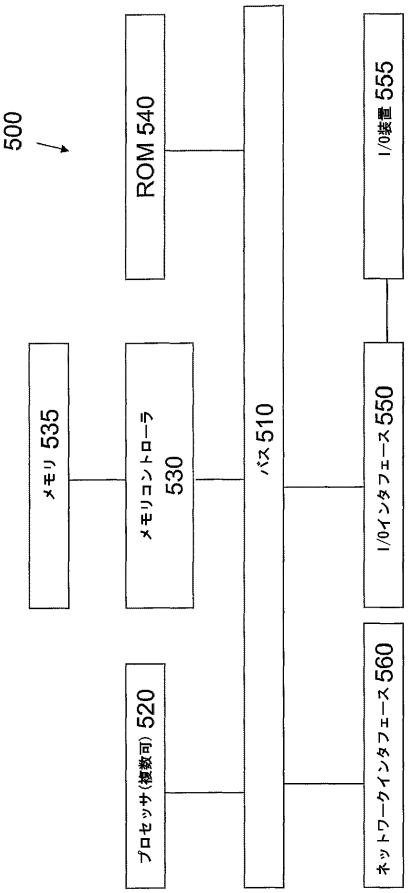
【図３】


チェッカ命令をスレッドにおいて発して実行する
300
命令が個々のパイプラインを通る
305
コミットバッファにおいてピアチェッカ命令を待つ
310
ピアチェッカ命令と比較する
315
チェッカ命令をコミットする
320

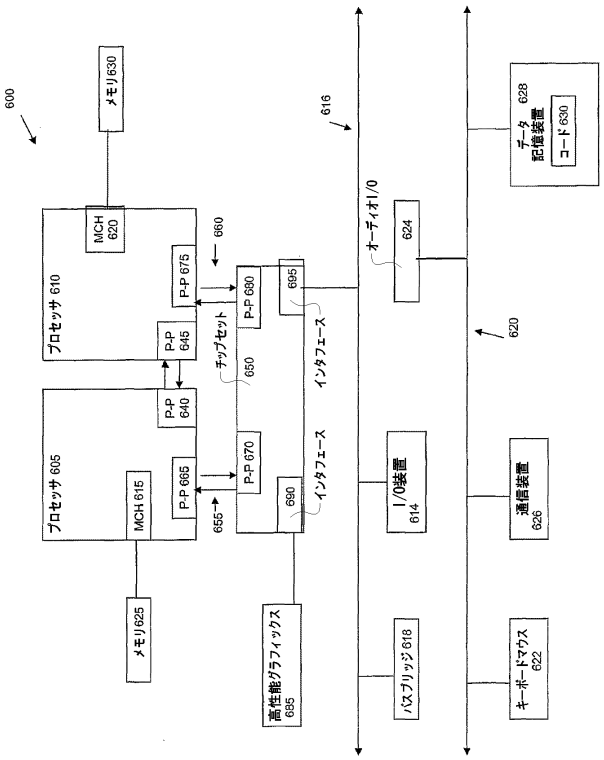
【図４】


チェッカ命令
チェッカ命令
400
405
PIPELINE
PIPELINE
バッファ 410
バッファ 415
検査してコミットする
420

【図５】


500
ROM 540
メモリ 535
メモリコントローラ 530
プロセッサ(複数可) 520
バス 510
I/O装置 555
I/Oインタフェース 550
ネットワークインタフェース 560

【図６】


600
メモリ630
MCH 620
プロセッサ 610
P-P 675
P-P 645
P-P 640
P-P 665
プロセッサ 605
MCH 615
メモリ625
660
チップセット
650
655
P-P 690
P-P 670
695
690
インタフェース
インタフェース
高性能グラフィックス 685
616
オーディオI/O
624
620
I/O装置 614
バスブリッジ 618
データ記憶装置 628
コード 630
通信装置 626
キーボードマウス 622

フロントページの続き

(72)発明者　ラインハルト、スティーブン
アメリカ合衆国、４８１０５　ミシガン州、アナーバー、キルバン　パーク　シーアイアール.　３１５８
(72)発明者　ウィーバー、クリストファー
アメリカ合衆国、０１７５２　マサチューセッツ州、モールバラ、アップルブライアー　レーン　１００９

審査官　漆原　孝治

(56)参考文献　特開平１１－０１５６６１（ＪＰ，Ａ）
T.N.Vijaykumar et al., Transient-fault recovery using simultaneous multithreading, Proceedings of the 29th annual international symposium on Computer architecture, IEEE Computer Society, ２００２年, p87-98

(58)調査した分野(Int.Cl., ＤＢ名)
G06F 11/16
G06F 11/18