

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第6963552号

(P6963552)

(45) 発行日 令和3年11月10日 (2021. 11. 10)

(24) 登録日 令和3年10月19日 (2021. 10. 19)

(51) Int. Cl.	F I
GO 6 F 9/46 (2006. 01)	GO 6 F 9/46 4 3 0
GO 6 F 9/312 (2006. 01)	GO 6 F 9/312 L
	GO 6 F 9/312 W

請求項の数 15 (全 26 頁)

(21) 出願番号	特願2018-531614 (P2018-531614)	(73) 特許権者	507364838
(86) (22) 出願日	平成28年12月9日 (2016. 12. 9)		クアルコム, インコーポレイテッド
(65) 公表番号	特表2018-538628 (P2018-538628A)		アメリカ合衆国 カリフォルニア 9 2 1
(43) 公表日	平成30年12月27日 (2018. 12. 27)		2 1 サン ディエゴ モアハウス ドラ
(86) 国際出願番号	PCT/US2016/065740		イブ 5 7 7 5
(87) 国際公開番号	W02017/116652	(74) 代理人	100108453
(87) 国際公開日	平成29年7月6日 (2017. 7. 6)		弁理士 村山 靖彦
審査請求日	令和1年11月25日 (2019. 11. 25)	(74) 代理人	100163522
(31) 優先権主張番号	62/271, 475		弁理士 黒田 晋平
(32) 優先日	平成27年12月28日 (2015. 12. 28)	(72) 発明者	グレゴリー・マイケル・ライト
(33) 優先権主張国・地域又は機関	米国 (US)		アメリカ合衆国・カリフォルニア・9 2 1
(31) 優先権主張番号	15/252, 323		2 1・サン・ディエゴ・モアハウス・ドラ
(32) 優先日	平成28年8月31日 (2016. 8. 31)		イヴ・5 7 7 5
(33) 優先権主張国・地域又は機関	米国 (US)	審査官	三坂 敏夫

最終頁に続く

(54) 【発明の名称】 ブロックアトミック実行モデルを利用するプロセッサベースシステムにおける部分的に実行された命令ブロックのリプレイ

(57) 【特許請求の範囲】

【請求項 1】

命令ブロック中のすべての命令がコミットされるか、または命令のどれもコミットされない命令ブロックに命令がグループ化されるブロックアトミック実行モデルを利用するプロセッサにおいて実行された命令ブロックの実行リプレイを制御するための部分的リプレイコントローラであって、

前記プロセッサによる前記命令ブロックの実行中に、起こり得るアーキテクチャ状態修正に関連付けられた命令の検出または前記プロセッサにおける例外の発生にตอบสนองして、命令ブロックに関して記録/リプレイ状態をアクティブ状態に設定するように構成された検出回路であって、命令は、前記プロセッサにおいてアーキテクチャ状態を修正する場合、起こり得るアーキテクチャ状態修正に関連付けられ、前記命令ブロックの外の機能またはプロセスとの観察可能な対話をもつ、検出回路と、

前記命令ブロックに関する前記記録/リプレイ状態がアクティブ状態であることにตอบสนองして、前記プロセッサによる前記命令ブロックの再実行中に、

以前生成されたデータが次のロード/記憶命令に関して記録されているかどうかを判断するために、前記命令ブロック中で実行されるべき前記次のロード/記憶命令に対応する記録/リプレイログファイル中のエントリ状態を検査し、

前記以前生成されたデータが、前記次のロード/記憶命令に関して前記記録/リプレイログファイルに記録されていないことにตอบสนองして、前記実行された次のロード/記憶命令の生成データを前記記録/リプレイログファイルに記録し、

10

20

前記以前生成されたデータが、前記次のロード/記憶命令に関して前記記録/リプレイログファイルに記録されていることに応答して、前記次のロード/記憶命令に関して前記記録/リプレイログファイルに記録されている、前記以前生成されたデータを使って、前記次のロード/記憶命令を実行する

ように構成された記録/リプレイ回路と
を備える部分的リプレイコントローラ。

【請求項 2】

前記検出回路は、

前記プロセッサによる前記命令ブロックの前記実行中に、起こり得る副作用を有するロード/記憶命令を含む、前記命令ブロック中の前記命令を検出し、

前記起こり得る副作用を有する前記ロード/記憶命令を含む前記命令の検出に応答して、前記命令ブロックに関して前記記録/リプレイ状態を前記アクティブ状態に設定するように構成された副作用検出回路を備える、請求項1に記載の部分的リプレイコントローラ。

【請求項 3】

前記検出回路は、

前記プロセッサによる前記命令ブロックの前記実行中に、書込みシステム構成レジスタ命令を含む、前記命令ブロック中の前記命令を検出し、

前記書込みシステム構成レジスタ命令を含む前記命令の検出に応答して、前記命令ブロックに関して、前記記録/リプレイ状態を前記アクティブ状態に設定するように構成された副作用検出回路を備える、請求項1に記載の部分的リプレイコントローラ。

【請求項 4】

前記検出回路は、

前記プロセッサによる前記命令ブロックの前記実行中に、予想される例外を有する命令を含む、前記命令ブロック中の前記命令を検出し、

前記予想される例外を有する前記命令を含む前記命令の検出に応答して、前記命令ブロックに関して前記記録/リプレイ状態を前記アクティブ状態に設定するように構成された副作用検出回路を備え、

前記予想される例外を有する前記命令は、ウォッチポイント、ブレイクポイント、デバッグステップポイント、または静的ブロック例外を引き起こす動作に関連付けられた命令を含む、請求項1に記載の部分的リプレイコントローラ。

【請求項 5】

前記予想される例外を有する前記命令は、静的ブロック例外を引き起こす動作に関連付けられた命令を含み、前記静的ブロック例外は、命令ページ障害、無効ブロックヘッダ、および無効命令符号化からなるグループから構成される、請求項4に記載の部分的リプレイコントローラ。

【請求項 6】

前記命令ブロック中のすべての命令が実行されたかどうかを判断し、

前記命令ブロック中の前記命令すべてが実行されたと判断したことに応答して、前記命令ブロック中の前記実行された命令による外部生成値をコミットするようにさらに構成された、請求項1に記載の部分的リプレイコントローラ。

【請求項 7】

前記記録/リプレイ回路は、前記実行された次のロード/記憶命令の前記生成データについての前記記録/リプレイ状態を前記記録/リプレイログファイルに記録するようにさらに構成される、請求項1に記載の部分的リプレイコントローラ。

【請求項 8】

前記記録/リプレイ回路は、前記実行された次のロード/記憶命令によってアドレス指定された仮想アドレスを、前記実行された次のロード/記憶命令の前記生成データと関連付けて、前記記録/リプレイログファイルに記録するようにさらに構成される、請求項1に記載の部分的リプレイコントローラ。

10

20

30

40

50

【請求項 9】

前記命令ブロックに関する前記記録/リプレイ状態がアイドル実行状態であることに応答して、前記検出回路は、前記プロセッサによる前記命令ブロックの実行中に、前記起こり得るアーキテクチャ状態修正に関連付けられた前記命令ブロック中の前記命令を検出するように構成される、請求項1に記載の部分的リプレイコントローラ。

【請求項 10】

前記記録/リプレイ状態が前記アイドル実行状態であるときに前記命令ブロックの前記実行中に例外が発生したことに応答して、前記命令ブロックを再実行させるように構成された、請求項9に記載の部分的リプレイコントローラ。

【請求項 11】

前記記録/リプレイ状態が前記アクティブ状態であるときに前記命令ブロックの実行中に例外が発生したことに応答して、前記記録/リプレイ回路は、

前記命令ブロックに関する前記記録/リプレイ状態をロック状態に設定し、

前記記録/リプレイログファイルへの前記実行された次のロード/記憶命令の前記生成データの記録を中断し、前記次のロード/記憶命令について記録されている前記以前生成されたデータを使って前記次のロード/記憶命令を実行するのを中断するようにさらに構成される、請求項1に記載の部分的リプレイコントローラ。

【請求項 12】

前記命令ブロックについての前記記録/リプレイ状態が以前のロック状態からアクティブ状態に戻ったことに応答して、前記記録/リプレイ回路は、前記実行された次のロード/記憶命令の前記生成データを前記記録/リプレイログファイルに記録し、前記次のロード/記憶命令に関して記録されている、前記以前生成されたデータを使って前記次のロード/記憶命令を実行するのを再開するようにさらに構成される、請求項11に記載の部分的リプレイコントローラ。

【請求項 13】

前記記録/リプレイ状態が前記アクティブ状態にあるときの前記命令ブロックの前記実行中の前記例外の前記発生に応答して、前記記録/リプレイ回路は、前記例外の前記発生に先立って、前記命令ブロック中の実行された命令による外部生成値をコミットするようにさらに構成される、請求項12に記載の部分的リプレイコントローラ。

【請求項 14】

命令ブロック中のすべての命令がコミットされるか、または命令のどれもコミットされない命令ブロックに命令がグループ化されるブロックアトミック実行モデルを利用するプロセッサにおいて命令ブロックをリプレイする方法であって、

前記プロセッサによる前記命令ブロックの実行中に、起こり得るアーキテクチャ状態修正に関連付けられた命令の検出または前記プロセッサにおける例外の発生に応答して、命令ブロックに関して記録/リプレイ状態をアクティブ状態に設定するステップであって、命令は、前記プロセッサにおいてアーキテクチャ状態を修正する場合、起こり得るアーキテクチャ状態修正に関連付けられ、前記命令ブロックの外の機能またはプロセスとの観察可能な対話をもつ、ステップと、

前記命令ブロックに関する前記記録/リプレイ状態がアクティブ状態であることに応答して、前記プロセッサによる前記命令ブロックの再実行中に、

以前生成されたデータが次のロード/記憶命令に関して記録されているかどうかを判断するために、前記命令ブロック中で実行されるべき前記次のロード/記憶命令に対応する記録/リプレイログファイル中のエントリ状態を検査するステップと、

前記以前生成されたデータが、前記次のロード/記憶命令に関して前記記録/リプレイログファイルに記録されていないことに応答して、前記実行された次のロード/記憶命令の生成データを前記記録/リプレイログファイルに記録するステップと、

前記以前生成されたデータが、前記次のロード/記憶命令に関して前記記録/リプレイログファイルに記録されていることに応答して、前記次のロード/記憶命令に関して前記記録/リプレイログファイルに記録されている、前記以前生成されたデータを使って、前記

10

20

30

40

50

次のロード/記憶命令を実行するステップとを含む方法。

【請求項 15】

コンピュータ実行可能命令を記憶したコンピュータ可読記憶媒体であって、前記コンピュータ実行可能命令は、命令ブロック中のすべての命令がコミットされるか、または命令のどれもコミットされない命令ブロックに命令がグループ化されるブロックアトミック実行モデルを利用するプロセッサによって実行されると、前記プロセッサに、

前記プロセッサによる前記命令ブロックの実行中に、起こり得るアーキテクチャ状態修正に関連付けられた命令の検出または前記プロセッサにおける例外の発生にตอบสนองして、命令ブロックに関して記録/リプレイ状態をアクティブ状態に設定することであって、命令は、前記プロセッサにおいてアーキテクチャ状態を修正する場合、起こり得るアーキテクチャ状態修正に関連付けられ、前記命令ブロックの外の機能またはプロセスとの観察可能な対話をもつ、ことと、

前記命令ブロックに関する前記記録/リプレイ状態がアクティブ状態であることにตอบสนองして、前記プロセッサによる前記命令ブロックの再実行中に、

以前生成されたデータが次のロード/記憶命令に関して記録されているかどうかを判断するために、前記命令ブロック中で実行されるべき前記次のロード/記憶命令に対応する記録/リプレイログファイル中のエントリ状態を検査することと、

前記以前生成されたデータが、前記次のロード/記憶命令に関して前記記録/リプレイログファイルに記録されていないことにตอบสนองして、前記実行された次のロード/記憶命令の生成データを前記記録/リプレイログファイルに記録することと、

前記以前生成されたデータが、前記次のロード/記憶命令に関して前記記録/リプレイログファイルに記録されていることにตอบสนองして、前記次のロード/記憶命令に関して前記記録/リプレイログファイルに記録されている、前記以前生成されたデータを使って、前記次のロード/記憶命令を実行することと

を行わせる、コンピュータ可読記憶媒体。

【発明の詳細な説明】

【技術分野】

【0001】

優先出願

本出願は、その内容全体が参照により本明細書に組み込まれる、2015年12月28日に出願された「REPLAY OF PARTIALLY-EXECUTED INSTRUCTION BLOCKS IN A PROCESSOR-BASED SYSTEM EMPLOYING A BLOCK-ATOMIC EXECUTION MODEL」と題する米国仮特許出願第62/271,475号の優先権を主張する。

【0002】

本出願は、その内容全体が参照により本明細書に組み込まれる、2016年8月31日に出願された「REPLAY OF PARTIALLY-EXECUTED INSTRUCTION BLOCKS IN A PROCESSOR-BASED SYSTEM EMPLOYING A BLOCK-ATOMIC EXECUTION MODEL」と題する米国特許出願第15/252,323号の優先権も主張する。

【0003】

本開示の技術は概して、プロセッサベースシステムにおける命令の実行に関し、より詳細には、命令ブロック中のすべての命令がコミットされるか、または命令のどれもコミットされない命令ブロックに命令がグループ化されるブロックアトミック実行モデルを利用するプロセッサベースシステムに関する。

【背景技術】

【0004】

マイクロプロセッサは、多種多様なアプリケーションにおいて計算タスクを実施する。従来のマイクロプロセッサアプリケーションは、ソフトウェア命令を実行する、「プロセッサコア」としても知られる1つまたは複数のプロセッサを含む中央処理ユニット(CPU)を含む。ソフトウェア命令は、CPUに、データに基づいて動作を実施するよう命令する。そ

10

20

30

40

50

のようなデータの例は、命令フェッチデータ中に符号化された即値、レジスタに記憶されたデータ、メモリ中のあるロケーションからのデータ、および入出力(I/O)デバイスなどの外部デバイスからのデータを含む。CPUは、命令に従って動作を実施して、結果を生じる。結果は次いで、レジスタもしくはメモリに記憶されるか、または出力としてI/Oデバイスに与えられ得る。

【0005】

いくつかのCPUは、「ブロックアトミック」実行モデルを利用する。ブロックアトミック実行モデルでは、命令のセット(たとえば、128個の命令)が命令ブロックにグループ化される。たとえば、図1は、ブロックアトミック実行モデルを利用するプロセッサによって実行されるべき複数の命令102(1) ~ 102(N)からなる例示的な命令ブロック100を示す。ブロックアトミック実行モデルに従って動作するプロセッサは、命令ブロック100を単一のエンティティとして論理的にフェッチし、実行し、コミットする。ブロックアトミック実行モデルは、アウトオブオーダープロセッサ(OoP)の複雑さを低減させるという利点を有する。たとえば、ブロックアトミック実行モデルを利用するOoPは、各命令の実行の後、プロセッサの正確な状態を折り返し報告する必要がない。この点において、図1に示す例示的な命令ブロック100において、それ以外の場合にはグローバルレジスタに記憶されることになる、命令102(1)、102(2)、102(4)、102(N-2) ~ 102(N)の実行からの中間生成結果104(1)、104(2)、104(4)、104(N-2) ~ 104(N)は、レジスタにセーブされる必要がない。一例として命令102(1)を取り上げると、その中間生成結果104(1)が、ピアツーピア方式で命令ブロック100中の消費者命令102(2)に、中間生成結果104(1)をセーブする必要なく与えられ得る。こうすることにより、より少ないレジスタ読出しおよび書込み動作が可能になる。ただし、ブロックアトミック実行モデルでは、中間生成結果はグローバルレジスタに記憶されないため、命令ブロック100の(たとえば、メモリ、I/Oデバイスなどへの)外部実行結果がコミットされ得る前に、命令すべてが実行されなければならない。たとえば、図1に示すように、命令ブロック100の中の命令102(4)の実行中に例外108(たとえば、ページ障害、またはデバッグブレイクポイントもしくはセットポイントなど、正確な例外)が発生した場合、中間生成結果104(1)、104(2)、104(4)が保存されないため、残りの命令102(N-2) ~ 102(N)は実行されず、実行は次の命令102(N-2)において始まり得る。命令ブロック100は、例外108が解決された後、最初から再実行される。

【0006】

したがって、ブロックアトミック実行モデルを利用するプロセッサは、複雑さの低減という利点を有するが、ブロックアトミック実行モデルを利用するプロセッサは、外部結果がコミットされる前に命令ブロック中のすべての命令を実行しなければならないという欠点を有する。一例として、これは、マルチスレッドCPUが存在する場合、デバッグをより難しくする可能性があり、というのは、命令ブロックの実行中に発生した例外またはブレイクポイントにつながったいかなる挙動を再生することも概して、可能でないからである。別のスレッド中の前の書込み動作から読み取られた可能性がある、命令ブロックの中間生成結果は、記憶されない。これは、例外が発生するのに先立ってI/O動作がすでに実行されている、I/Oデバイス副作用などの副作用動作に関する問題点を提示する。たとえば、図1の命令ブロック100中のI/O動作命令102(4)の実行が始動されたが、例外108の発生に先立って完了されていない場合、命令102(4)におけるI/O動作から読み取られた中間データは、命令ブロック100の再実行中に、もはや利用可能でない可能性がある。したがって、この例では、命令102(4)の最初の実行中に読み取られた中間データは、レジスタr2に再生可能に記憶されない場合がある。

【発明の概要】

【課題を解決するための手段】

【0007】

本明細書で開示する態様は、ブロックアトミック実行モデルを利用するプロセッサベースシステムにおける、部分的に実行された命令ブロックのリプレイを含む。この点において、一態様では、ブロックアトミック実行モデルを利用する部分的リプレイコントローラ

10

20

30

40

50

が、中央処理ユニット(CPU)のプロセッサ中に設けられる。ブロックアトミック実行モデルにおいて、命令は、外部生成結果がコミットされる前にプロセッサ(たとえば、アウトオブオーダープロセッサ(OoP))中で全体が実行される命令ブロックにグループ化される。こうすることにより、外部生成結果の値に、ならびにしたがってそのような外部的に生成された結果に依存するプロセッサのアーキテクチャ状態および他のプロセスに影響し得るすべてのロード/記憶依存を解決する。例外によりロードされた命令ブロックの再実行中に、ロード/記憶動作からの生成結果が再生できないことを避けるために、部分的リプレイコントローラは、命令ブロックの再実行中にロード/記憶命令の結果を記録/リプレイするように構成される。この点において、いくつかの態様では、命令ブロック中の命令の実行中に、命令が、起こり得るアーキテクチャ状態修正(たとえば、起こり得る副作用)に関連付けられているものとして命令ブロック中で検出されるか、または例外が発生した場合、命令ブロックは再実行される。命令ブロックの再実行中、部分的リプレイコントローラは、ロード/記憶命令からの生成結果を記録するように構成される。したがって、命令ブロックの再実行中に例外が発生した場合、例外が発生する前の、実行されたロード/記憶命令についての以前記録された生成結果は、例外が解決された後の命令ブロックの再実行中にリプレイされ得る。このようにして、起こり得るアーキテクチャ状態修正に関連付けられた動作につながるか、または例外が発生する前の命令の実行は、副作用を繰り返すことなく、以前生成された結果を伴って決定論的に繰り返され得る。

【0008】

ただし、命令ブロックの再実行中に例外が発生しない場合、このことは、命令ブロック中のすべての命令全体が実行され、生成結果が記録されたことを意味することに留意されたい。この事例における命令ブロックは再実行されないので、以前実行されたロード/記憶命令についての記録された生成結果をリプレイする必要はない。

【0009】

この点において、1つの例示的な態様では、プロセッサ中で実行された命令ブロックの実行リプレイを制御するための部分的リプレイコントローラが提供される。部分的リプレイコントローラは、起こり得るアーキテクチャ状態修正に関連付けられた命令の検出、またはプロセッサにおける例外の発生にตอบสนองして、命令ブロックに関して記録/リプレイ状態をアクティブ状態に設定するように構成された検出回路を備える。部分的リプレイコントローラは、記録/リプレイ回路も備える。命令ブロックに関する記録/リプレイ状態がアクティブ状態であることにตอบสนองして、記録/リプレイ回路は、以前生成されたデータが次のロード/記憶命令に関して記録されているかどうかを判断するために、命令ブロック中で実行されるべき次のロード/記憶命令に対応する記録/リプレイログファイル中のエントリ状態を検査するように構成される。また、命令ブロックに関する記録/リプレイ状態がアクティブ状態であることにตอบสนองして、記録/リプレイ回路は、以前生成されたデータが、次のロード/記憶命令に関して記録/リプレイログファイルに記録されていないことにตอบสนองして、実行された次のロード/記憶命令の生成データを記録/リプレイログファイルに記録するようにも構成される。また、命令ブロックに関する記録/リプレイ状態がアクティブ状態であることにตอบสนองして、記録/リプレイ回路は、以前生成されたデータが、次のロード/記憶命令に関して記録/リプレイログファイルに記録されていることにตอบสนองして、次のロード/記憶命令に関して記録/リプレイログファイルに記録されている、以前生成されたデータを使って、次のロード/記憶命令を実行するようにも構成される。

【0010】

別の例示的な態様では、プロセッサ中で実行された命令ブロックの実行リプレイを制御するための部分的リプレイコントローラが提供される。部分的リプレイコントローラは、起こり得るアーキテクチャ状態修正に関連付けられた命令の検出、またはプロセッサにおける例外の発生にตอบสนองして、記録/リプレイ状態を、命令ブロックに関してアクティブ状態に設定するための手段を備える。部分的リプレイコントローラは、命令ブロックに関して記録/リプレイ状態をアクティブ状態に設定したことにตอบสนองして、以前生成されたデータが次のロード/記憶命令に関して記録されているかどうかを判断するために、命令プロ

ック中で実行されるべき次のロード/記憶命令に対応する記録/リプレイログファイルを記憶するための手段におけるエントリ状態を検査するための手段も備える。部分的リプレイコントローラは、以前生成されたデータが、次のロード/記憶命令に関して、記録/リプレイログファイルを記憶するための手段に記録されていないことに応答して、実行された次のロード/記憶命令の生成データを記録/リプレイログファイルを記憶するための手段に記録するための手段も備える。部分的リプレイコントローラは、以前生成されたデータが、次のロード/記憶命令に関して、記録/リプレイログファイルを記憶するための手段に記録されていることに応答して、次のロード/記憶命令に関して記録/リプレイログファイルを記憶するための手段に記録されている、以前生成されたデータを使って、次のロード/記憶命令を実行するための手段も備える。

10

【0011】

別の例示的な態様では、プロセッサにおいて命令ブロックをリプレイする方法が提供される。この方法は、起こり得るアーキテクチャ状態修正に関連付けられた命令の検出、またはプロセッサにおける例外の発生に応答して、命令ブロックに関して記録/リプレイ状態をアクティブ状態に設定するステップを含む。命令ブロックに関する記録/リプレイ状態がアクティブ状態であることに応答して、方法はまた、以前生成されたデータが次のロード/記憶命令に関して記録されているかどうかを判断するために、命令ブロック中で実行されるべき次のロード/記憶命令に対応する記録/リプレイログファイル中のエントリ状態を検査するステップと、以前生成されたデータが、次のロード/記憶命令に関して記録/リプレイログファイルに記録されていないことに応答して、実行された次のロード/記憶命令の生成データを記録/リプレイログファイルに記録するステップと、以前生成されたデータが、次のロード/記憶命令に関して記録/リプレイログファイルに記録されていることに応答して、次のロード/記憶命令に関して記録/リプレイログファイルに記録されている、以前生成されたデータを使って、次のロード/記憶命令を実行するステップとを含む。

20

【0012】

別の例示的な実施形態では、コンピュータ実行可能命令を記憶した非一時的コンピュータ可読媒体が提供される。コンピュータ実行可能命令は、プロセッサによって実行されると、プロセッサに、起こり得るアーキテクチャ状態修正に関連付けられた命令の検出、またはプロセッサにおける例外の発生に応答して、命令ブロックに関して記録/リプレイ状態をアクティブ状態に設定させる。命令ブロックに関する記録/リプレイ状態がアクティブ状態であることに応答して、コンピュータ実行可能命令はまた、プロセッサによって実行されると、プロセッサに、以前生成されたデータが次のロード/記憶命令に関して記録されているかどうかを判断するために、命令ブロック中で実行されるべき次のロード/記憶命令に対応する記録/リプレイログファイル中のエントリ状態を検査させる。命令ブロックに関する記録/リプレイ状態がアクティブ状態であることに応答して、コンピュータ実行可能命令はまた、プロセッサによって実行されると、プロセッサに、以前生成されたデータが、次のロード/記憶命令に関して記録/リプレイログファイルに記録されていないことに応答して、実行された次のロード/記憶命令の生成データを記録/リプレイログファイルに記録させる。命令ブロックに関する記録/リプレイ状態がアクティブ状態であることに応答して、コンピュータ実行可能命令はまた、プロセッサによって実行されると、プロセッサに、以前生成されたデータが、次のロード/記憶命令に関して記録/リプレイログファイルに記録されていることに応答して、次のロード/記憶命令に関して記録/リプレイログファイルに記録されている、以前生成されたデータを使って、次のロード/記憶命令を実行させる。

30

40

【図面の簡単な説明】

【0013】

【図1】ブロックアトミック実行モデルを利用する中央処理ユニット(CPU)によって実行されるように構成された例示的な命令ブロックを示す図であって、命令ブロック中で実行された入力/出力(I/O)動作の後に、命令ブロック中の命令の実行中に例外が発生した、図

50

である。

【図2】各プロセッサが、外部メモリおよびI/Oデバイスへのアクセスを含む機能を実施するためのソフトウェア命令を実行するように構成される、例示的な多重プロセッサ(「マルチプロセッサ」)CPUのブロック図である。

【図3】図2のマルチプロセッサCPU中に含まれ得るプロセッサの例示的な詳細を示すブロック図であって、プロセッサは、起こり得るアーキテクチャ状態修正に関連付けられた、命令ブロック中の命令を検出したことに応答して、および/または命令ブロック中の命令の実行中の例外の発生に応答して、命令ブロックの再実行中にロード/記憶命令の結果を記録および/またはリプレイするように構成された部分的リプレイコントローラを含む、図である。

10

【図4】起こり得るアーキテクチャ状態修正に関連付けられた命令の検出に応答して、および/またはプロセッサにおける例外の発生に応答して、命令ブロックの再実行中にロード/記憶命令の結果を記録および/またはリプレイするために記録/リプレイアクティブ状態に入るための、図3のプロセッサ中の部分的リプレイコントローラによって実施され得る例示的なプロセスを示すフローチャートである。

【図5】命令ブロックの実行および再実行中に、図3の部分的リプレイコントローラのさらなる例示的な検討を容易にするための、起こり得るアーキテクチャ状態修正に関連付けられた命令を含む、図3のプロセッサによって実行される例示的な命令ブロックを示す図である。

【図6】起こり得るアーキテクチャ状態修正に関連付けられた、命令ブロック中の命令を検出したことに応答して、命令ブロックの再実行中にロード/記憶命令の結果を記録し、リプレイするための、図3の部分的リプレイコントローラによって実施され得る例示的なプロセスを示すフローチャートである。

20

【図7】プロセッサにおける命令ブロックの実行中の、図3のプロセッサ中の部分的リプレイコントローラの例示的な動作を示す例示的な状態機械である。

【図8】本明細書で開示する例による、命令ブロックのアイドル実行状態中に、起こり得るアーキテクチャ状態修正に関連付けられた、命令ブロック中の命令を検出したことおよび/または例外の発生に応答して、命令ブロックの再実行中にロード/記憶命令の結果を記録および/またはリプレイするように構成された部分的リプレイコントローラを含むマルチプロセッサCPUを含む例示的なプロセッサベースシステムのブロック図である。

30

【発明を実施するための形態】

【0014】

ここで図面を参照して、本開示のいくつかの例示的な態様について説明する。「例示的」という語は、「例、事例、または例示として機能すること」を意味するために本明細書で使用される。本明細書で「例示的」と記載されている任意の態様は、必ずしも他の態様よりも好ましいまたは有利であると解釈されるべきではない。

【0015】

本明細書で開示する態様は、ブロックアトミック実行モデルを利用するプロセッサベースシステムにおける、部分的に実行された命令ブロックのリプレイを含む。この点において、一態様では、ブロックアトミック実行モデルを利用する部分的リプレイコントローラが、中央処理ユニット(CPU)のプロセッサ中に設けられる。ブロックアトミック実行モデルにおいて、命令は、外部生成結果がコミットされる前にプロセッサ(たとえば、アウトオブオーダープロセッサ(OoP))中で全体が実行される命令ブロックにグループ化される。こうすることにより、外部生成結果の値に、ならびにしたがってそのような外部的に生成された結果に依存するプロセッサのアーキテクチャ状態および他のプロセスに影響し得るすべてのロード/記憶依存を解決する。例外によりロードされた命令ブロックの再実行中に、ロード/記憶動作からの生成結果が再生できないことを避けるために、部分的リプレイコントローラは、命令ブロックの再実行中にロード/記憶命令の結果を記録/リプレイするように構成される。この点において、いくつかの態様では、命令ブロック中の命令の実行中に、命令が、起こり得るアーキテクチャ状態修正(たとえば、起こり得る副作用)に関

40

50

連付けられているものとして命令ブロック中で検出されるか、または例外が発生した場合、命令ブロックは再実行される。命令ブロックの再実行中、部分的リプレイコントローラは、ロード/記憶命令からの生成結果を記録するように構成される。したがって、命令ブロックの再実行中に例外が発生した場合、例外が発生する前の、実行されたロード/記憶命令についての以前記録された生成結果は、例外が解決された後の命令ブロックの再実行中にリプレイされ得る。このようにして、起こり得るアーキテクチャ状態修正に関連付けられた動作につながるか、または例外が発生する前の命令の実行は、副作用を繰り返すことなく、以前生成された結果を伴って決定論的に繰り返され得る。

【 0 0 1 6 】

この点において、図2は、本明細書ではプロセッサ204(1) ~ 204(P)と呼ばれる複数のプロセッサコア204(1) ~ 204(P)を有するCPU202を含む例示的なプロセッサベースシステム200のブロック図であり、「P」は、CPU202に含まれるプロセッサ204の数である。非限定的例として、プロセッサ204(1) ~ 204(P)は、入力データの利用可能性に基づいて命令のアウトオブオーダー実行を実施するように構成されたアウトオブオーダープロセッサ(OoP)であってよい。これは、先行命令がプログラム中の次の命令のためのデータの取出しを完了するのを待つ間、アイドルになることを避けるために、プログラム中の本来の順序に従って命令を実行しなければならないインオーダープロセッサとは反対である。この例における、CPU202中の1つまたは複数のプロセッサ204(1) ~ 204(P)は、部分的リプレイコントローラ206(1) ~ 206(P)を含む。この例では、CPU202中の各プロセッサ204(1) ~ 204(P)は、部分的リプレイコントローラ206(1) ~ 206(P)を含むが、そのようなことが求められるわけではない。

【 0 0 1 7 】

図3において開始し、以下でより詳しく論じるように、プロセッサ204(1) ~ 204(P)中に含まれる部分的リプレイコントローラ206(1) ~ 206(P)は各々、命令ブロック中の命令の実行中に、起こり得るアーキテクチャ状態修正(たとえば、起こり得る副作用)に関連付けられた命令を検出したことに応答して、命令ブロックの再実行中にロード/記憶命令の結果を記録および/またはリプレイするように構成される。部分的リプレイコントローラ206(1) ~ 206(P)の態様について論じる前に、図2のCPU202の他の構成要素の記述について最初に、以下で論じる。

【 0 0 1 8 】

この点において、図2への参照を続けると、この例における各プロセッサ204(1) ~ 204(P)は、アドレスバス210上へメモリアクセス要求を(「アドレス」において)オフプロセッサで与える必要なく、キャッシュされたデータへのアクセスをオンプロセッサで提供するためのキャッシュメモリ(「キャッシュ」)208(1) ~ 208(P)(たとえば、レベル2(L2)キャッシュ)を含み得る。オフプロセッサでのメモリアクセス要求向けに、プロセッサ204(1) ~ 204(P)は、システムメモリ214にアクセスするための、メモリコントローラ212へのアドレスバス210上の「アドレス」を与えるように構成される。要求側プロセッサ204(1) ~ 204(P)によって書き込まれるべき「データ」が、データバス216を介してシステムメモリ214に与えられる。システムメモリ214から読み取られた「データ」は、システムメモリ214によってデータバス216上で提供されて、要求側プロセッサ204(1) ~ 204(P)に与えられる。入力/出力(I/O)デバイス218も、I/Oデバイス218へのアクセスをプロセッサ204(1) ~ 204(P)に提供するように、図2のプロセッサベースシステム200中のアドレスバス210およびデータバス216に結合される。プロセッサベースシステム200は、システムオンチップ(SoC)220として設けられてもよく、その場合、図2に示し、上述したプロセッサ204(1) ~ 204(P)および関連構成要素は、チップ上で一体化される。

【 0 0 1 9 】

例示的な部分的リプレイコントローラを含み、データ構造をサポートする、CPU202中の各プロセッサ204(1) ~ 204(P)の例示的な詳細が、「プロセッサ204」として全体的に図3に示される。そこに示されるように、プロセッサ204は、命令キャッシュ304に記憶された命令302を、復号/ディスパッチ完了ユニット306に与えるためにフェッチすることができる

命令メモリ管理ユニット(MMU)300を含む。命令キャッシュ304は、実行されるべき命令302を、図2のシステムメモリ214などのメインメモリからロードすることができる。この例では、命令302は、実行されるべき命令ブロック308の中にグループ化される。復号/ディスパッチ完了ユニット306は、復号された命令ブロック308に関する命令302を、実行されるように、ディスパッチ/完了バス310に与える。前に説明したように、プロセッサ204はブロックアトミック実行モデルを利用するので、所与の命令ブロック308中のすべての命令302は、実行された命令302の結果がコミットされる前に全体的に実行する。この例では、浮動小数点レジスタファイル314中に記憶された生成結果を用いて実行されるべき、命令ブロック308中の浮動小数点命令302Fを受信するように、浮動小数点ユニット312が設けられる。同様に、汎用レジスタ(GPR)ファイル318中に記憶された生成結果を用いて実行されるべき実行命令ブロック308中の整数命令302Iを受信するように、整数ユニット316が設けられる。命令ブロック308からロード/記憶命令302Lを受信するように、ロード/記憶ユニット320が設けられる。特定のロード/記憶命令302Lに依存して、実行されたロード/記憶命令302Lからの生成結果が、アドレスバス210およびデータバス216に結合されたバスインターフェースユニット324を通した外部メモリへのアクセスのために、汎用レジスタファイル318、またはメモリキュー322中に記憶され得る。外部メモリへのアクセスを求める、実行されたロード/記憶命令302Lはまた、実行されたロード/記憶命令302Lに基づいて、外部メモリへ読み取られ、または外部メモリに書き込まれるべきデータを管理するためにデータMMU326に与えられる。

【 0 0 2 0 】

図3への参照を続けると、図1の例示的な命令ブロック100中の命令102(1) ~ 102(N)などの命令が、部分的リプレイコントローラ206が設けられずに、プロセッサ204によって命令ブロック308として実行されていた場合、および命令ブロック308中の命令302の実行中に例外が発生していた場合、命令ブロック308中の残りの命令302は実行されない。中間生成結果は保存されず、実行は、例外が解決された後、命令ブロック308中の次の命令302において始まる場合がある。この点において、命令ブロック308中の命令302は、例外が解決された後、最初から再実行されることになる。これにより、プロセッサ204によって実行されるデバッグ命令302が難しくなる場合があり、というのは、概して、命令ブロック308中の命令302の実行中に発生した例外につながったどの挙動も再生することは可能でないからである。別のスレッド中の以前の書込み動作から読み取られた可能性がある、命令ブロック308からの、実行された命令302の中間生成結果は、記憶されない。このことは、I/Oデバイス218(図2参照)へのI/O動作が、例外に先立ってすでに実行されているI/Oデバイス副作用などの副作用を有し得るか、または実際に有するロード/記憶命令302Lに関する問題点も提示する。この点において、命令ブロック308のロード/記憶命令302Lが、始動されたが、例外に先立って完了されなかった場合、中間データが記憶されなかったので、I/O動作から読み取られた中間データは、命令ブロック308中の命令302を再実行するためにもはや利用可能でない可能性がある。

【 0 0 2 1 】

この点において、部分的リプレイコントローラ206は、起こり得るアーキテクチャ状態修正(たとえば、起こり得る副作用)に関連付けられた、命令ブロック308中の命令302の検出に回答して、命令ブロック308中の命令302の再実行中にロード/記憶命令302Lの結果を記録および/またはリプレイするように、プロセッサ204中に設けられる。部分的リプレイコントローラ206は、ロード/記憶ユニット320を含むロード/記憶システム328の一部として、またはロード/記憶ユニット320とは別個に設けられてよい。命令302は、プロセッサ204(1) ~ 204(P)を含むCPU202中のアーキテクチャ状態を修正する動作を伴う場合、起こり得るアーキテクチャ状態修正に関連付けられ、命令ブロック308の外の機能またはプロセスとの観察可能な対話をもち得る。たとえば、起こり得るアーキテクチャ状態修正に関連付けられた命令302は、命令ブロック308および/または他のプロセッサ204(1) ~ 204(P)の外の他のプロセスおよびスレッドに影響し得る外部生成結果に影響し得る。そのような命令302は、外部生成結果の値に影響し得るロード/記憶依存を有する場合があり、したがっ

て、プロセッサ204(1)～204(P)および他のプロセスのアーキテクチャ状態は、そのような外部的に生成された結果に依存する。

【0022】

たとえば、命令ブロック308は、可能性としてはアーキテクチャ状態を修正し、したがって副作用を引き起こし得る命令302を有し得る。非限定的例は、グローバル変数を修正すること、システム構成レジスタに書き込むこと、予想される例外(たとえば、ブレイクポイント、セットポイント、またはウォッチポイント)を発生させること、およびメモリまたはI/Oデバイスにデータを書き込むことを含む。アーキテクチャ状態も、I/Oデバイスなどのデバイスからデータを読み取ることなど、副作用を有する動作から修正されてよく、ここで、データは、キューからポップオフされ、したがって、後続読取り動作において再生されない。命令302が、起こり得るアーキテクチャ状態修正に関連付けられている場合、アーキテクチャ状態修正によって影響される挙動または他のプロセスは、評価の順序に依存する。

10

【0023】

副作用を有するか、または引き起こさせる、起こり得るアーキテクチャ状態修正に関連付けられた命令302について、命令302のどの再実行中でも命令302が動作するための条件を再現または再生するのは難しい場合がある。したがって、命令ブロック308全体が実行される前に例外が発生することになっていた場合、起こり得るアーキテクチャ状態修正に関連付けられた命令302の動作を実施するのに使われる中間生成結果は、例外が解決された後、もはや利用可能でないか、または同じ値をもたない場合がある。ただし、部分的リプレイコントローラ206が、起こり得るアーキテクチャ状態修正に関連付けられた、命令ブロック308中の命令302の検出に応答して、命令ブロック308の再実行中にロード/記憶命令302Lの結果を記録および/またはリプレイするように構成されていると、中間生成結果は、命令ブロック308の再実行中に記録されてよい。したがって、例外が発生することになっていた場合、そのような中間結果は、プロセッサ204のアーキテクチャ状態に影響する、命令ブロック308からの、外部的に生成された値の完全性を保証するために、例外が解決された後、命令ブロック308の別の再実行中にリプレイされてよい。

20

【0024】

たとえば、例外は、命令ブロック308の実行境界において発生するブロック例外、および命令ブロック308内の命令302の境界において発生する命令例外を含む、正確な例外であってもよい。例外の例は、算術的例外、すなわち、ゼロによる除算、除算オーバーフロー、IEEE浮動小数点、特権違反(非MMU)、データページ障害(マップされないページ、特権違反、書き込み保護)、訂正不可能な正確な誤り訂正符号(ECC)エラー、および他の正確なハードウェア障害(TBD)を含む。例外はまた、他の例として、デバッグモードにおけるデバッグブレイクポイントまたはウォッチポイントであってもよい。デバッグモード中またはウォッチポイントにおいて、命令ブロック308からの、以前生成されたデータを記録し、リプレイすることができることは、実行中にすべてのロードまたは記憶されたデータの状態を見ることができるためには、特に有用であり得る。

30

【0025】

ただし、図3のプロセッサ204は、たとえば命令ブロック308の実行中に発生する例外により、命令ブロック308がプロセッサ204によって再実行されなければならない場合に、命令ブロック308中のロード/記憶命令302Lからの生成データを再生することができないという状況を避けるために、部分的リプレイコントローラ206を含む。以下でより詳しく論じるように、部分的リプレイコントローラ206は、ブロックアトミック実行モデルを利用するプロセッサ204における命令例外を再生可能にさせるように構成される。部分的リプレイコントローラ206は、部分的ブロックリプレイ(PBR)モデルが利用されるようにし、ここで、十分な状態が、例外につながる命令ブロック308の実行が決定論的に再生されるように、例外ハンドラまたはデバッガに報告され得る。そのようなシナリオにおいて、命令ブロック308の実行は、汎用レジスタおよびメモリに対するいかなる副作用も誘発しなかったという意味で、原子的なままである。部分的ブロックリプレイはまた、副作用動作(強

40

50

く順序づけられたロードおよび記憶ならびにシステムレジスタ書込み)動作を、命令ブロック308構造に収めさせる。部分的リプレイコントローラ206によって可能にされたPBRモデルは、命令ブロック308中の命令302に対するブロック原子性を、副作用で弱化させる。例外の後で命令ブロック308を再開させるのに十分な状態がセーブされ、実行を決定論的に繰り返すと副作用動作につながるが、続いて副作用自体を繰り返すことはない。

【 0 0 2 6 】

やはり以下でより詳しく論じるように、部分的リプレイコントローラ206は、プロセッサ204中での実際の例外の発生に応答して、命令ブロック308の再実行中にロード/記憶命令302Lの結果を記録および/またはリプレイするようにも構成され得る。たとえば、そのような例外は、命令ページ障害、無効ブロックヘッダ、および無効命令符号化などの静的ブロック例外を含み得る。したがって、プロセッサ204が例外から回復した後、部分的リプレイコントローラ206は、命令ブロック308の再実行中にロード/記憶命令302Lの結果を記録および/またはリプレイするようにも構成されてよい。これは、命令ブロック308の再実行中に例外が再度発生するか、または別の例外が発生した場合、別のスレッドにおける以前の書込み動作から読み取られている場合がある、命令ブロック308の中間生成結果が、別の例外の発生により命令ブロック308が再実行されなければならない場合は記憶され、リプレイされ得るようにするためである。そうでない場合、I/O動作などのロード/記憶動作を実施するために読み取られた中間データは、中間データが記憶されていないので、命令ブロック308を再実行するためにもはや利用可能でない場合がある。また、例外が正確な例外である場合、例外が発生する前に実行された命令ブロック308中の命令302からの生成結果は、部分的コミット動作においてプロセッサ204によってコミットされてもよい。したがって、プロセッサ204が例外から回復した後、部分的リプレイコントローラ206は、ロード/記憶命令302Lの結果を記録および/またはリプレイするための命令ブロック308の最初から、実行を開始するように構成されてよい。以前実行されたロード/記憶命令302Lの以前記録された結果が、命令ブロック308の再実行中にリプレイされる。ただし、一例では、部分的コミット動作の一部として、命令ブロック308の以前の実行中にコミットされたロード/記憶命令302Lからのどの生成データも、命令ブロック308全体が実行するとき、フルコミットの後続の部分的再コミット動作中にコミットし直されない。これは、これらの以前実行されたロード/記憶命令302Lの結果がすでに、以前コミットされているからである。したがって、部分的コミット機能性を利用すると、実行効率を増大させることができ、というのは、命令ブロック308は、例外から回復した後、開始命令302から再実行されなくてよいからである。部分的リプレイコントローラ206は、コミットされている、以前実行されたロード/記憶命令302Lからの結果を追跡するように構成されてよく、そうすることによって、ロード/記憶命令302Lのリプレイから生成されたそのような結果はコミットし直されない。

【 0 0 2 7 】

この点において、図4は、命令ブロック308の再実行中にロード/記憶命令302Lの結果を記録および/またはリプレイするために、アクティブ状態において記録/リプレイ状態に入るための、図3のプロセッサ204中の部分的リプレイコントローラ206によって実施され得る例示的なプロセス400を示すフローチャートである。この点において、図4に示すように、プロセス400は、部分的リプレイコントローラ206が、起こり得るアーキテクチャ状態修正に関連付けられた命令302、または命令ブロック308の実行中にプロセッサ204における例外の発生を検出することによって開始する(図4のブロック402)。図3の例に示すように、部分的リプレイコントローラ206は、命令ブロック308の実行中に、起こり得るアーキテクチャ状態修正に関連付けられた命令302、またはプロセッサ204における例外の発生を検出するように構成されている検出回路332を含み得る。プロセッサ204によって実行された命令302は、ロード/記憶ユニット320に与えられ、ユニット320は、部分的リプレイコントローラ206の検出回路332に、命令302が、起こり得るアーキテクチャ状態修正に関連付けられているかどうかを検出させる。

【 0 0 2 8 】

図4への参照を続けると、命令ブロック308の実行中の、起こり得るアーキテクチャ状態修正に関連付けられた命令302の検出、または例外の発生にตอบสนองして、部分的リプレイコントローラ206は、命令ブロック308に関して、記録/リプレイ状態330をアクティブ状態に設定するように構成される(図4のブロック404)。たとえば、起こり得るアーキテクチャ状態修正に関連付けられた命令302の検出、またはプロセッサ204における例外の発生の前に、記録/リプレイ状態330はアイドル実行状態に設定されていてよい。記録/リプレイ状態330が命令ブロック308に関してアクティブ状態であることにตอบสนองして、命令ブロック308はプロセッサ204によって再実行される。例外の発生にตอบสนองして、アクティブ状態が記録/リプレイ状態330として設定される場合、命令ブロック308は、CPU202によって(たとえば、CPU202中のオペレーティングシステムによって)(図2)例外が解決された後、再実行される。

10

【0029】

記録/リプレイ状態330がアクティブ状態であることにตอบสนองして命令ブロック308が再実行されると、部分的リプレイコントローラ206中の記録/リプレイ回路334は、次のロード/記憶命令302Lが実行される前に命令ブロック308中で実行されるべき各次のロード/記憶命令302Lに対応する、記録/リプレイログファイル338におけるエントリ状態336を検査する(図4のブロック406)。記録/リプレイ回路334は、エントリ状態336を検査して、次のロード/記憶命令302Lについての以前生成されたデータが記録されたかどうかを判断し、これは、命令ブロック308の実行中に起こり得るアーキテクチャ状態修正が検出されたか、または例外が発生したよりも前に、命令ブロック308の前の実行中に、検査される次のロード/記憶命令302Lがすでに実行されたことを意味する(図4のブロック406)。エントリ状態336が、実行されるべき次のロード/記憶命令302Lがすでに以前実行されたことを示す場合、次のロード/記憶命令302Lについての生成結果は、命令ブロック308における、この次のロード/記憶命令302Lの第1の実行インスタンスなので、記録/リプレイログファイル338に記録される。ただし、エントリ状態336が、命令ブロック308の、以前の実行インスタンスにおいて次のロード/記憶命令302Lがまだ実行されていないことを示す場合、記録/リプレイ回路334は、記録/リプレイ状態330がアクティブ状態であることにตอบสนองして、実行された次のロード/記憶命令302Lの生成データを記録/リプレイログファイル338に記録するように構成される(図4のブロック408)。これは、命令ブロック308が再度再実行されなければならない場合、実行された次のロード/記憶命令302Lの生成データが、リプレイされるべき記録/リプレイログファイル338中で利用可能になるように、そのようになっている。この点において、記録/リプレイ回路334は、記録/リプレイログファイル338中に次のロード/記憶命令302Lに関して記録されている、以前生成されたデータを使って、次のロード/記憶命令302Lを実行するようにも構成される(図4のブロック410)。

20

30

【0030】

図2のCPU202中のプロセッサ204(1)~204(P)中に設けられ得る部分的リプレイコントローラ206のさらなる検討を容易にするために、図5および図6が、命令ブロック308の実行および再実行中の図3の部分的リプレイコントローラ206のさらなる例示的な検討を容易にするために与えられる。この点において、図5は、起こり得るアーキテクチャ状態修正に関連付けられた命令を含む、図3のプロセッサ204によって実行され得る例示的な命令ブロック308(A)を示す。図6は、起こり得るアーキテクチャ状態修正に関連付けられた、命令ブロック308(A)中の命令502を検出したことにตอบสนองして、図5の命令ブロック308(A)の再実行中にロード/記憶命令302Lの結果を記録し、リプレイするために、部分的リプレイコントローラ206によって実施され得る例示的なプロセス600のさらなる詳細を示すフローチャートである。図5の命令ブロック308(A)とともに、図6のプロセス600への参照が行われる。

40

【0031】

この点において、図5を参照すると、命令ブロック308(A)は、記録/リプレイ状態330がアイドルに設定された新規インスタンスにおける実行(図6のブロック602)のために、図3のプロセッサ204によってフェッチされる。この例ではI/Oデバイスロード/記憶命令である第1の命令502(1)が、図3のロード/記憶ユニット320に与えられると、部分的リプレイコ

50

ントローラ206の検出回路332は、第1の命令502(1)が、起こり得るアーキテクチャ状態修正に関連付けられているかどうかを検出する(図6のブロック604)。この例では、ロードされるべきデータのソースがI/Oデバイスからであるので、第1の命令502(1)は、起こり得るアーキテクチャ状態修正に関連付けられた、起こり得る副作用を有する。したがって、第1の命令502(1)の実行後に例外が発生することになっていた場合、I/Oデバイスからポップされたデータは、I/Oデバイスキュー中にそれ以上保存されなくてよく、したがって、命令ブロック308(A)の再実行中に第1の命令502(1)が再実行される前は再生可能でない場合がある。したがって、この事例では、部分的リプレイコントローラ206は、第1の命令502(1)を実行することなしでの、第1の命令502(1)についての起こり得る副作用の検出に应答して、命令ブロック308(A)に関して、記録/リプレイ状態330をアクティブに設定する。これは、命令ブロック308(A)の再実行中に、第1の命令502(1)の次の実行インスタンスにおいて再生可能でない結果の副作用を引き起こさせる可能性がある第1の命令502(1)の実行からの結果が生成されないようにするためである(図6のブロック606)。その後、プロセッサ204は、命令ブロック308(A)を再実行させる。

【 0 0 3 2 】

図5の命令ブロック308(A)の再実行中、第1の命令502(1)は再度、実行のために図3のプロセッサ204中のロード/記憶ユニット320に与えられる。部分的リプレイコントローラ206中の記録/リプレイ回路334は、以前生成されたデータが第1の命令502(1)に関して記録されたかどうかを判断するために、記録/リプレイログファイル338中のエントリ340(1) ~ 340(E)におけるエントリ状態336を検査する(図6のブロック608)。この例では、以前生成されたデータは第1の命令502(1)に関して記録されているのではなく、というのは、第1の命令502(1)は、起こり得る副作用の検出により、命令ブロック308(A)の初回実行中に実行されなかったからである。したがって、部分的リプレイコントローラ206の記録/リプレイ回路334は、第1の命令502(1)からの生成結果504(1)を、ロード/記憶ユニット320が記録/リプレイログファイル338中の第1の命令502(1)を実行した結果として記録し、そうすることによって、生成された結果504(1)は、たとえば例外により命令ブロック308(A)が再度再実行される場合、リプレイすることができる(図6のブロック610)。その後、プロセッサ204は、第2の命令502(2)がロード/記憶命令ではないような副作用にかかわらず、第2の命令502(2)を実行する。ただし、図5の例示的命令ブロック308(A)中の第3の命令502(3)は、起こり得る副作用を有するロード/記憶動作である。ただし、記録/リプレイ状態330はすでにアクティブであり、このことは、部分的リプレイコントローラ206が、実行されたロード/記憶命令302Lからの生成結果504(1) ~ 504(Q)を記録中であることを意味するので、命令ブロック308(A)は再実行されない。第3の命令502(3)における、グローバルレジスタr3から取得された生成結果504(3)は、記録/リプレイログファイル338に記録される。命令502(4) ~ 502(Q)の実行は、ロード/記憶命令302Lの生成結果504(1) ~ 504(Q)を、記録/リプレイログファイル338中の部分的リプレイコントローラ206によって記録させ続ける(図6のブロック608、610)。

【 0 0 3 3 】

ただし、図5の例に示すように、命令502(4)と命令502(Q-2)との間で例外508が発生している。例外508が解決された後、命令ブロック308(A)はプロセッサ204中で再実行される。ただし、命令502(1) ~ 502(4)は、命令ブロック308(A)の以前の実行インスタンスにおいてすでに実行されているので、部分的リプレイコントローラ206中の記録/リプレイ回路334は、生成された結果504(1)、504(3) ~ 504(4)が記録/リプレイログファイル338に記録されているかどうかを判断するために、記録/リプレイログファイル338を調べるように構成される(図6のブロック608)。この例では、生成結果504(1)、504(3) ~ 504(4)は記録/リプレイログファイル338に記録されるので、記録/リプレイ回路334は、この例では命令ブロック308(A)の以前の再実行中に記録/リプレイログファイル338に記録された、以前生成された結果504(1)、504(3) ~ 504(4)を使って、以前生成された結果504(1) ~ 504(Q)を実行または「リプレイ」するように構成される(図6のブロック612)。したがって、例外508が発生した後に命令502(1)、502(3) ~ 502(4)に関して例外または副作用が発生していた場合、以

前生成された結果504(1)、504(3)～504(4)は、命令ブロック308(A)の再実行が正しい結果および生成値を生成することになるように、記録/リプレイログファイル338に保存されていた。

【0034】

この例では、例外508が発生する前に、命令ブロック308(A)の以前の再実行中に命令502(Q-2)～502(Q)は実行しなかったため、部分的リプレイコントローラ206は、命令502(Q-2)、502(Q)についての生成結果504(Q-2)、504(Q)が、記録/リプレイログファイル338に記録されたかどうかを判断するために、記録/リプレイログファイル338を調べるように構成される(図6のブロック608)。この例では、それらの命令は例外508の発生前には記録されなかったため、記録/リプレイ回路334は、命令502(Q-2)、502(Q)をリプレイするのとは反対に、命令502(Q-2)、502(Q)についての生成結果504(Q-2)、504(Q)を記録/リプレイログファイル338に記録するように構成される(図6のブロック610)。したがって、別の例外508が発生した場合、命令502(Q-2)、502(Q)についての生成結果504(Q-2)、504(Q)は、記録/リプレイログファイル338からリプレイされ得る。図5の命令ブロック308(A)中ですべての命令502(1)～502(Q)が実行された後(図6のブロック614)、実行された命令502(1)～502(Q)から生じた外部生成値がコミットされる(図6のブロック616)。

【0035】

どのタイプのロード/記憶命令302Lが、検出回路332によって、起こり得る副作用を有するものとして検出されるように構成されるかに対する設計選択であることに留意されたい。たとえば、検出回路332は、実行された命令ブロック308中のあらゆるロード/記憶命令302Lを、ロード/記憶命令302Lのタイプにかかわらず、起こり得る副作用を有するものとして検出するように構成される場合がある。たとえば、プロセッサ204が多重スレッドコードを実行中である場合、例外が発生した後で別のスレッドがそのようなロード命令のソースロケーションにおけるデータを変えるかもしれないので、可能性としてはどのロード命令も、生成不可能な結果を生じ得る。ただし、実行された命令ブロック308はより頻繁に再実行され得るので、実行された命令ブロック308中のあらゆるタイプのロード/記憶命令302Lにおける起こり得る副作用の検出により、プロセッサ204の性能が低下する場合がある。記録/リプレイ状態330は、任意のタイプのロード/記憶命令302Lを有する、実行された命令ブロック308に基づいて、アクティブに設定されることになる。これは、デバッグモードにおいて、またはデバッガブレイクポイントに応答して有用であり得るが、というのは、部分的リプレイコントローラ206は、デバッグモード中のロード/記憶命令302Lからのすべてのデータの状態を見直すことができるようにするはずだからである。ただし、別の例として、通常または非デバッグモード中、検出回路332は、命令ブロック308中のある特定のタイプのロード/記憶命令302Lのみを、他の再生可能性問題にかかわらず、I/O動作を伴う、起こり得る副作用を有するものとして検出するように構成されてよい。たとえば、検出回路332は、プロセッサ204の外部のロケーションもしくはデバイスからデータをロードし、またはそのようなロケーションもしくはデバイスに記憶する、命令ブロック308中のロード/記憶命令302Lを検出するようにのみ構成されてよい。

【0036】

部分的リプレイコントローラ206の動作は、命令ブロック308が、再実行の状態に基づく動作で複数回再実行され得るので、状態機械としての実装に好適であり得る。この点において、図7は、命令ブロックの再実行中にロード/記憶命令の結果を記録および/またはリプレイするために、図3のプロセッサ204中の部分的リプレイコントローラ206によって利用され得る状態機械700である。この点において、図7を参照すると、プロセッサ204における命令ブロック308の新規実行インスタンス(図7の702)に応答して、プロセッサ204によってアクセス可能な記録/リプレイ状態330は、アイドル状態(「IDLE」)に設定され、これは、命令ブロック308に関して記録/リプレイ動作が発生している最中ではないことを意味する。そのようなスレッド、または例外もしくは割込みについてのアーキテクチャ状態は、命令ブロック308境界に対応する。プロセッサ204は、命令ブロック308中の命令302をアイドル状態で実行する。命令ブロック308中の命令302すべてが、命令ブロック308の初回

実行中に、起こり得るアーキテクチャ状態修正を有するものとして検出されるわけではなく、いかなる例外も発生しない場合、命令302の実行は最終的に、外部生成結果がコミットされて全体が完了する(図7の704)。

【 0 0 3 7 】

図7への参照を続けると、検出回路332が、実行されるべき、起こり得るアーキテクチャ状態修正に関連付けられた命令302を検出すると(図7の706)、部分的リプレイコントローラ206は、記録/リプレイ状態330をアクティブ状態(「ACTIVE」)に変える。プロセッサ204は次いで、命令ブロック308を開始命令302から再実行させる。これは、命令ブロック308を再実行することができるようにし、起こり得る副作用を有するロード/記憶命令302Lについての生成結果を記録/リプレイログファイル338に記録することができるようにするためである。スレッドがアクティブであるとき、以下で論じるように、このことは、部分的リプレイコントローラ206が記録および/またはリプレイされている最中であることを意味する。PBR状態の記録は、システムレジスタのロードおよび記憶に加えて、読取りおよび書込みが、以降のリプレイに関して記録されることを意味する。

【 0 0 3 8 】

アクティブ状態において、部分的リプレイコントローラ206は、以前論じたように、記録/リプレイログファイル338にアクセスするように構成される。部分的リプレイコントローラ206は、命令ブロック308の再実行中に、実行されたロード/記憶命令302Lからの生成結果で、記録/リプレイログファイル338を更新または記録するように構成される。このようにして、起こり得るアーキテクチャ状態修正(たとえば、起こり得る副作用)に関連付けられたロード/記憶命令302Lの実行後に例外が発生した場合(図7の708)、生成結果は、ロード/記憶ユニット320によって記録/リプレイログファイル338に記憶される。命令ブロック308の再実行中に例外が発生した場合、部分的リプレイコントローラ206は、CPU202中のオペレーティングシステムにおける例外処理によって解決されるべき例外向けに、記録/リプレイ状態330をロック状態(「LOCKED」)に設定する。例外が解決された後、記録/リプレイ状態330は、命令ブロック308が再実行されるようにアクティブ状態(「ACTIVE」)に設定し戻される(図7の710)。以前上で論じたように、生成結果は、そのような記録された生成結果が、たとえば、例として、別のスレッドまたはI/Oデバイスからポップされたデータからの変化により異なる場合に命令ブロック308が以前記録された生成結果を用いて問題なく再実行され得るように、例外が解決された後、命令ブロック308の再実行中にリプレイされ得る。

【 0 0 3 9 】

たとえば、実行されたロード/記憶命令302Lについて記録/リプレイログファイル338に記録されたエントリ状態336は、非限定的例として、なし/ヌル、ロード(通常)、記憶(通常)、取消し、ロード-コミット(副作用)、記憶-コミット(副作用)、システムレジスタ読取り、および実行されるロード/記憶命令302Lのタイプによっては、システムレジスタ書込み-コミットであり得る。実行されたロード/記憶命令302Lから記録された仮想アドレス(VA)342は、ロード/記憶命令302LによってアクセスされるメモリまたはI/Oデバイスのアドレスであり、これは、プログラムをデバッグするために有用である。実行されたロード/記憶命令302Lから記録されたデータ344は、ロード/記憶命令302Lを実行した結果としての生成結果である。

【 0 0 4 0 】

一例では、記録/リプレイ状態330がアクティブ状態(「ACTIVE」)にあるときに例外が発生すると、例外の発生の前に実行した命令302からのどの生成結果もコミットされないことに留意されたい。命令ブロック308は、例外がロック状態(「LOCKED」)から解決されると再実行される(図7の710)。ただし、代替態様では、部分的リプレイコントローラ206は、例外の発生前に実行した命令ブロック308中の命令302からの生成結果をコミットするように構成される場合もある。この事例では、例外が解決され、記録/リプレイ状態330がアクティブ状態(「ACTIVE」)に戻ったときに残りの命令302が再実行されると、以前実行しなかった命令302からの生成結果のみがコミットされる。

【 0 0 4 1 】

図7への参照を続けると、記録/リプレイ状態330は、命令ブロック308中の命令302すべてが実行されるまで、アクティブ状態(「ACTIVE」)のままである。その後、命令ブロック308中の実行された命令302からの外部生成結果がコミットされる(または、部分的コミット機能性のケースでは、以前コミットされなかった命令302からの生成結果がコミットされる)(図7の712)。先に論じたように、一例では、部分的コミット動作が利用される場合、部分的コミット動作の一部として命令ブロック308の前の実行中にコミットされたロード/記憶命令302Lからの生成データは、命令ブロック308が全体的に実行したとき、フルコミットの後続の部分的再コミット動作中にコミットし直されない。部分的リプレイコントローラ206は、次の命令ブロック308を実行するために、記録/リプレイ状態330をアイドル状態(「IDLE」)に戻して設定する。記録/リプレイ状態330がアイドル状態(「IDLE」)にあるときに例外が発生した場合、CPU202中のオペレーティングシステムは、現在のアーキテクチャ状態をセーブし、命令ブロック308中のさらなる命令302が実行されることがないように、例外を解決するためにロック状態「LOCKED」に切り替わる(図7の714)。ロック状態「LOCKED」において例外が解決されると、オペレーティングシステムは、例外が発生する前に存在したアーキテクチャ状態を復元し、記録/リプレイ状態330をアイドル状態(「IDLE」)に戻すように切り替える(図7の716)。

10

【 0 0 4 2 】

要するに、図3のプロセッサ204中の部分的リプレイコントローラ206によって与えられるPBRモデルは、副作用を伴って動作を実施する命令ブロック308が、後になってどの種類の例外もとれ得るときに必要とされるだけである。ただし、PBRモデルは、いかなる正確な例外も、プログラム終了またはブロックアトミック実行モデル用のデバッグへの転送を引き起こさせることが予想されるとき、デバッグを強化するのにも使うことができる。性能上の理由のため、PBRモデルは任意選択で、スーパーバイザサービスを呼び出させ、次いで、通常実行を再開させ得る非エラー例外(たとえば、ページ障害)に関与しなくてよい。

20

【 0 0 4 3 】

部分的リプレイコントローラ206を利用することによって命令ブロック308のアーキテクチャ状態が保存され得る例示的な例外カテゴリの要約を、以下に示す。ただし、現在の命令ブロック308が、部分的リプレイコントローラ206によって可能にされたPBRモデルを用いて実行している場合、割込みが、正確な部分的ブロック状態とともに受け取られ得る。以前論じるとともに以下の表に示すように、命令ブロックが、副作用に遭遇するか、または例外が発生する前に生成データを記憶している場合、部分的リプレイコントローラは、例外が発生する前に命令ブロック中で実行された命令からの生成結果を破棄するように構成されてもよい。プロセッサが例外から回復した後、部分的リプレイコントローラは次いで、ロード/記憶命令の結果を記録および/またはリプレイするための命令ブロックの開始命令から再度、実行を開始するように構成されることになる。代替として、部分的リプレイコントローラは、例外が発生する前に命令ブロック中で実行された命令からの生成結果をコミットするように構成されてもよい。プロセッサが例外から回復した後、部分的リプレイコントローラは次いで、再実行中に、命令ブロック中の以前実行されなかったロード/記憶命令の結果を記録および/またはリプレイするために、実行されなかった、命令ブロック中の次の命令から開始するように構成される。

30

40

【 0 0 4 4 】

【表 1】

確度	要因	部分的 ブロック リプレイ	ブロックが 副作用または記憶を実施したか	アーキテクチャ 状態	再生可能性
正確	ブロック	N/A	No	ブロックの開始	Yes
	命令	No	No	ブロックの開始	No
		Yes	No	ブロックの開始	Yes
			Yes(通常記憶は破棄される)	GPR/通常メモリ: ブロックの開始 副作用:副作用の後	Yes
			Yes(通常記憶は実行ポイントまで部分的にコミットされる)	GPR:ブロックの開始 通常メモリまたは副作用:例外の ポイントにおける	Yes

10

20

【0045】

さらなる態様では、起こり得るアーキテクチャ状態修正に関連付けられた命令の検出、またはプロセッサにおける例外の発生にตอบสนองして、記録/リプレイ状態を、命令ブロックに関してアクティブ状態に設定するための手段を含む、プロセッサ中で実行された命令ブロックの実行リプレイを制御するための部分的リプレイコントローラが提供されてもよい。たとえば、図3の部分的リプレイコントローラ206、検出回路332または記録/リプレイ回路334は、設定するためのそのような手段の例であり、ここで部分的リプレイコントローラ206は、起こり得るアーキテクチャ状態修正に関連付けられた命令の検出、またはプロセッサ204における例外の発生にตอบสนองして、記録/リプレイ状態330をアクティブ状態に設定するように構成される。部分的リプレイコントローラ206はまた、命令ブロックに関して記録/リプレイ状態をアクティブ状態に設定したことに対応して、以前生成されたデータが次のロード/記憶命令に関して記録されているかどうかを判断するために、命令ブロック中で実行されるべき次のロード/記憶命令に対応する記録/リプレイログファイルを記憶するための手段におけるエントリ状態を検査するための手段を含んでもよい。たとえば、エントリ状態を検査するためのこの手段は、記録/リプレイログファイル338にアクセスすることによって、図3の部分的リプレイコントローラ206または記録/リプレイ回路334によって提供されてもよい。部分的リプレイコントローラ206はまた、以前生成されたデータが、次のロード/記憶命令に関して、記録/リプレイログファイルを記憶するための手段に記録されていないことに対応して、実行された次のロード/記憶命令の生成データを記録/リプレイログファイルを記憶するための手段に記録するための手段を含んでもよい。たとえば、記録/リプレイログファイルを記憶するための手段は、次のロード/記憶命令に関して記録/リプレイログファイル338に記録されていない以前生成されたデータを記憶す

30

40

50

るために、図3の部分的リプレイコントローラ206または記録/リプレイ回路334によって提供されてもよい。部分的リプレイコントローラ206はまた、以前生成されたデータが、次のロード/記憶命令に関して、記録/リプレイログファイルを記憶するための手段に記録されていることに応答して、次のロード/記憶命令に関して記録/リプレイログファイルを記憶するための手段に記録されている、以前生成されたデータを使って、次のロード/記憶命令を実行するための手段を含んでもよい。以前生成されたデータを使って次のロード/記憶命令を実行するための手段は、図3の部分的リプレイコントローラ206または記録/リプレイ回路334によって提供されてもよい。

【0046】

たとえば、図3のプロセッサ204中の部分的リプレイコントローラ206などの、および本明細書で開示した例のうちのいずれかによる、命令ブロック中の命令の実行中に、起こり得るアーキテクチャ状態修正に関連付けられた、命令ブロック中の命令を検出したことに応答して、および/または例外の発生に応答して、命令ブロックの再実行中にロード/記憶命令の結果を記録および/またはリプレイするように構成された部分的リプレイコントローラを含むプロセッサは、任意のプロセッサベースのデバイスで提供されるか、またはその中に統合されてよい。例として、限定はしないが、セットトップボックス、エンターテインメントユニット、ナビゲーションデバイス、通信デバイス、固定ロケーションデータユニット、モバイルロケーションデータユニット、モバイルフォン、セルラーフォン、スマートフォン、タブレット、ファブレット、コンピュータ、ポータブルコンピュータ、デスクトップコンピュータ、携帯情報端末(PDA)、モニタ、コンピュータモニタ、テレビ、チューナ、ラジオ、衛星ラジオ、音楽プレーヤ、デジタル音楽プレーヤ、ポータブル音楽プレーヤ、デジタルビデオプレーヤ、ビデオプレーヤ、デジタルビデオディスク(DVD)プレーヤ、ポータブルデジタルビデオプレーヤおよび自動車が含まれる。

【0047】

この点において、図8は、1つまたは複数のプロセッサ804を含むCPU802を含むプロセッサベースシステム800の例を示す。プロセッサ804は各々、命令ブロック中の命令の実行中に、起こり得る副作用を検出したことに応答して、命令ブロックの再実行中にロード/記憶命令の結果を記録および/またはリプレイするように構成される部分的リプレイコントローラ806を含み得る。部分的リプレイコントローラ806は、例として、図3の部分的リプレイコントローラ206であってよい。CPU802は、一時的に記憶されているデータに迅速にアクセスするために、プロセッサ804に結合されたキャッシュメモリ808を有し得る。CPU802は、システムバス810に結合され、プロセッサベースシステム800内に含まれる周辺デバイスを相互接続することができる。CPU802中のプロセッサ804は、システムバス810を介してアドレス情報、制御情報、およびデータ情報を交換することによって、これらの他のデバイスと通信することができる。図8には示されていないが、複数のシステムバス810が設けられてよく、各システムバス810は異なるファブリックを構成する。たとえば、CPU802は、スレーブデバイスの例としてメモリシステム814内のメモリコントローラ812にバストランザクション要求を通信することができる。この例では、メモリコントローラ812は、メモリシステム814の中のメモリアレイ816にメモリアクセス動作を提供するように構成される。

【0048】

他のデバイスもシステムバス810に接続され得る。図8に示すように、これらのデバイスは、例として、メモリシステム814、1つまたは複数の入力デバイス818、1つまたは複数の出力デバイス820、1つまたは複数のネットワークインターフェースデバイス822、および1つまたは複数のディスプレイコントローラ824を含むことができる。入力デバイス818は、限定はしないが、入力キー、スイッチ、音声プロセッサなどを含む、任意のタイプの入力デバイスを含むことができる。出力デバイス820は、限定はしないが、オーディオインジケータ、ビデオインジケータ、他の視覚インジケータなどを含む、任意のタイプの出力デバイスを含むことができる。ネットワークインターフェースデバイス822は、ネットワーク826との間のデータの交換を可能にするように構成された任意のデバイスであり得る。

ネットワーク826は、限定はしないが、有線ネットワークまたはワイヤレスネットワーク、プライベートネットワークまたは公衆ネットワーク、ローカルエリアネットワーク(LAN)、ワイヤレスローカルエリアネットワーク(WLAN)、ワイドエリアネットワーク(WAN)、BLUETOOTH(登録商標)ネットワーク、およびインターネットを含む、任意のタイプのネットワークであってよい。ネットワークインターフェースデバイス822は、所望のいかなるタイプの通信プロトコルもサポートするように構成され得る。

【0049】

CPU802はまた、システムバス810を通じてディスプレイコントローラ824にアクセスして、1つまたは複数のディスプレイ828に送られる情報を制御するように構成され得る。ディスプレイ828としては、陰極線管(CRT)、液晶ディスプレイ(LCD)、プラズマディスプレイなどを含むがそれらに限定されない、任意のタイプのディスプレイがあり得る。ディスプレイコントローラ824は、表示されることになる情報を1つまたは複数のビデオプロセッサ830を介してディスプレイ828に送信し、ビデオプロセッサ830は、表示されることになる情報を処理して、ディスプレイ828に適したフォーマットにする。

【0050】

命令ブロック中の命令の実行中に、起こり得る副作用を検出したことに応答して、命令ブロックの再実行中にロード/記憶命令の結果を記録および/またはリプレイするように構成された部分的リプレイコントローラが、ソフトウェアベースのシステムにおいて提供されてもよい。部分的リプレイコントローラは、ソフトウェア命令なしで部分的リプレイコントローラの機能を提供するハードウェアのみの回路において実装される必要はない。図3の部分的リプレイコントローラ206などの部分的リプレイコントローラは、コンピュータ実行可能命令を記憶した非一時的コンピュータ可読媒体で提供されてもよく、命令は、図2におけるそのようなプロセッサ204(1)~204(P)のようなプロセッサによって実行されると、プロセッサに、起こり得るアーキテクチャ状態修正に関連付けられた命令の検出、またはプロセッサにおける例外の発生に応答して、命令ブロックに関して記録/リプレイ状態をアクティブ状態に設定させる。命令ブロックに関する記録/リプレイ状態がアクティブ状態であることに応答して、コンピュータ実行可能命令は、プロセッサによって実行されると、プロセッサに、以前生成されたデータが次のロード/記憶命令に関して記録されているかどうかを判断するために、命令ブロック中で実行されるべき次のロード/記憶命令に対応する記録/リプレイログファイル中のエントリ状態を検査させ、以前生成されたデータが、次のロード/記憶命令に関して記録/リプレイログファイルに記録されていないことに応答して、実行された次のロード/記憶命令の生成データを記録/リプレイログファイルに記録させ、以前生成されたデータが、次のロード/記憶命令に関して記録/リプレイログファイルに記録されていることに応答して、次のロード/記憶命令に関して記録/リプレイログファイルに記録されている、以前生成されたデータを使って、次のロード/記憶命令を実行させる。

【0051】

「コンピュータ可読媒体」は、命令の1つまたは複数のセットを記憶する、単一の媒体または複数の媒体(たとえば、集中型もしくは分散型データベース、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)を含むものとされるべきである。「コンピュータ可読媒体」という用語はまた、プロセッサによる実行のための命令のセットを記憶し、符号化し、または運ぶことが可能である媒体であって、プロセッサに、本明細書で開示した態様の方法論のうちのどの1つまたは複数も実施させるどの媒体も含むものとされるものとする。「コンピュータ可読媒体」という用語はしたがって、固体状態メモリ、光学媒体、および磁気媒体を含むが、それらに限定されないものとされるものとする。

【0052】

本明細書で開示する態様に関して説明した様々な例示的な論理ブロック、モジュール、回路、およびアルゴリズムが、電子ハードウェア、メモリの中もしくは別のコンピュータ可読媒体の中に記憶されるとともにプロセッサもしくは他の処理デバイスによって実行される命令、または両方の組合せとして実装され得ることを、当業者はさらに諒解されよう

。本明細書で説明するマスターデバイスおよびスレーブデバイスは、例として、任意の回路、ハードウェア構成要素、集積回路(IC)、またはICチップにおいて利用され得る。本明細書で開示するメモリは、任意のタイプおよびサイズのメモリであってよく、任意のタイプの所望の情報を記憶するように構成され得る。この互換性を明確に示すために、様々な例示的な構成要素、ブロック、モジュール、回路、およびステップについて、上記では概してそれらの機能性に関して説明した。そのような機能性がどのように実装されるのかは、特定の適用例、設計選択、および/またはシステム全体に課される設計制約によって決まる。当業者は、説明された機能性を特定の適用例ごとに様々な方法で実装してよいが、そのような実施態様の決定は、本開示の範囲からの逸脱を引き起こすものと解釈されるべきではない。

10

【0053】

本明細書で開示される態様とともに説明した様々な例示的論理ブロック、モジュール、および回路は、プロセッサ、デジタル信号プロセッサ(DSP)、特定用途向け集積回路(ASIC)、フィールドプログラマブルゲートアレイ(FPGA)もしくは他のプログラマブル論理デバイス、個別ゲートもしくはトランジスタ論理、個別ハードウェア構成要素、または本明細書で説明した機能を実施するように設計されたそれらの任意の組合せで実装または実施され得る。プロセッサは、マイクロプロセッサであってよいが、代替として、プロセッサは、任意の従来のプロセッサ、コントローラ、マイクロコントローラ、または状態機械であってもよい。プロセッサはまた、コンピューティングデバイスの組合せ、たとえば、DSPおよびマイクロプロセッサの組合せ、複数のマイクロプロセッサ、DSPコアと連携した1つもしくは複数のマイクロプロセッサ、または任意の他のそのような構成として実装されてもよい。

20

【0054】

本明細書で開示する態様は、ハードウェアにおいて具現され、また、ハードウェアに記憶された命令において具現される場合があり、命令は、たとえば、ランダムアクセスメモリ(RAM)、フラッシュメモリ、読取り専用メモリ(ROM)、電氣的プログラマブルROM(EPROM)、電氣的消去可能プログラマブルROM(EEPROM)、レジスタ、ハードディスク、リムーバブルディスク、CD-ROM、または当技術分野において知られている任意の他の形態のコンピュータ可読媒体内に存在する場合がある。例示的な記憶媒体は、プロセッサが記憶媒体から情報を読み取り、記憶媒体に情報を書き込むことができるように、プロセッサに結合される。代替として、記憶媒体は、プロセッサに一体化される場合がある。プロセッサおよび記憶媒体は、ASICに存在する場合がある。ASICは、リモート局の中に存在する場合がある。代替として、プロセッサおよび記憶媒体は、個別構成要素としてリモート局、基地局、またはサーバの中に存在してよい。

30

【0055】

本明細書の例示的な態様のいずれかにおいて説明した動作ステップは、例について説明するためのものであることにも留意されたい。説明した動作は、図示した順序以外の数多くの異なる順序で実施されてよい。さらに、単一の動作ステップにおいて記載した動作は、実際にはいくつかの異なるステップにおいて実施される場合がある。さらに、例示的な態様において論じられた1つまたは複数の動作ステップは組み合わせられる場合がある。フローチャート図に示した動作ステップは、当業者には容易に明らかであるように、多数の異なる変更を受ける場合があることを理解されたい。当業者にはまた、情報および信号が様々な異なる技術および技法のいずれを使用して表されてもよいことが理解されよう。たとえば、上記の説明全体にわたって参照される場合があるデータ、命令、コマンド、情報、信号、ビット、シンボル、およびチップは、電圧、電流、電磁波、磁場もしくは磁気粒子、光場もしくは光学粒子、またはそれらの任意の組合せによって表される場合がある。

40

【0056】

本開示の上記の説明は、あらゆる当業者が本開示を作成または使用することを可能にするように構成されている。本開示に対する種々の変更が、当業者に容易に明らかになり、

50

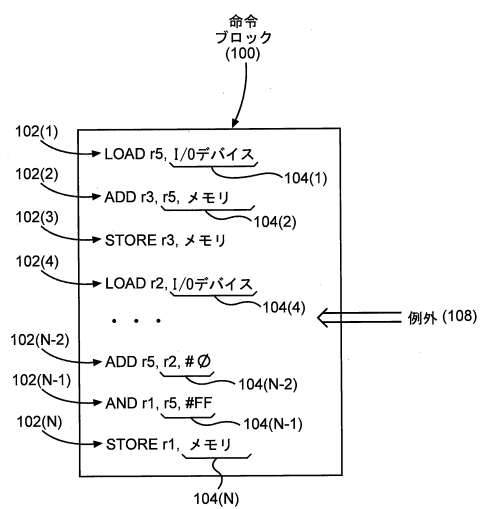
本明細書において規定される一般原理は、本開示の趣旨または範囲から逸脱することなく他の変形形態に適用される場合がある。したがって、本開示は、本明細書において説明される例および設計に限定されるものでなく、本明細書において開示される原理および新規の特徴と一致する最も広い範囲を与えられるべきである。

【符号の説明】

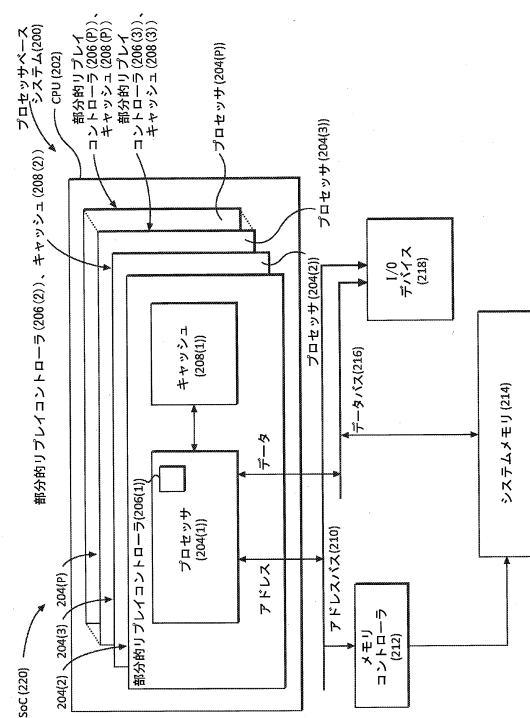
【 0 0 5 7 】

100	命令ブロック	
200	プロセッサベースシステム	
202	CPU	
204	プロセッサ、プロセッサコア	10
206	部分的リプレイコントローラ	
208	キャッシュメモリ、キャッシュ	
210	アドレスバス	
212	メモリコントローラ	
214	システムメモリ	
216	データバス	
218	入力/出力(I/O)デバイス	
220	システムオンチップ(SoC)	
300	命令メモリ管理ユニット(MMU)	
304	命令キャッシュ	20
306	復号/ディスパッチ完了ユニット	
308	命令ブロック、実行命令ブロック	
310	ディスパッチ/完了バス	
312	浮動小数点ユニット	
314	浮動小数点レジスタファイル	
316	整数ユニット	
318	汎用レジスタ(GPR)ファイル	
320	ロード/記憶ユニット	
322	メモリキュー	
324	バスインターフェースユニット	30
326	データMMU	
328	ロード/記憶システム	
332	検出回路	
334	記録/リプレイ回路	
338	記録/リプレイログファイル	
700	状態機械	
800	プロセッサベースシステム	
802	CPU	
804	プロセッサ	
806	部分的リプレイコントローラ	40
808	キャッシュメモリ	
810	システムバス	
812	メモリコントローラ	
814	メモリシステム	
818	入力デバイス	
820	出力デバイス	
822	ネットワークインターフェースデバイス	
824	ディスプレイコントローラ	
826	ネットワーク	
828	ディスプレイ	50

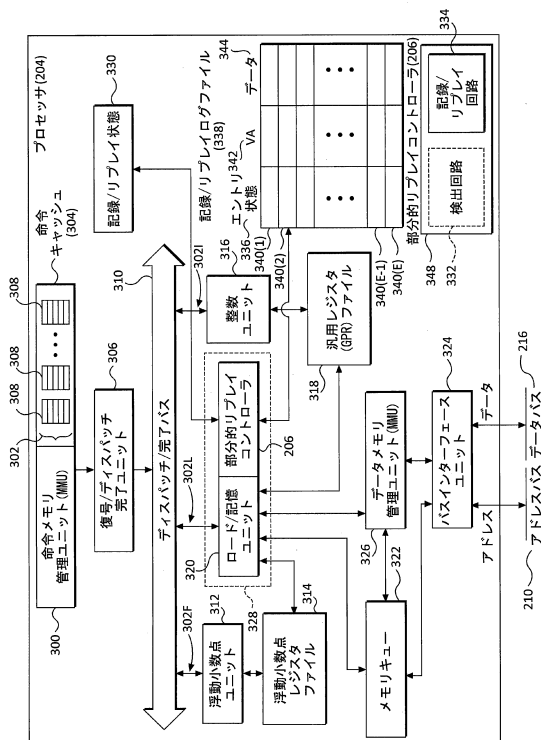
【 図 1 】



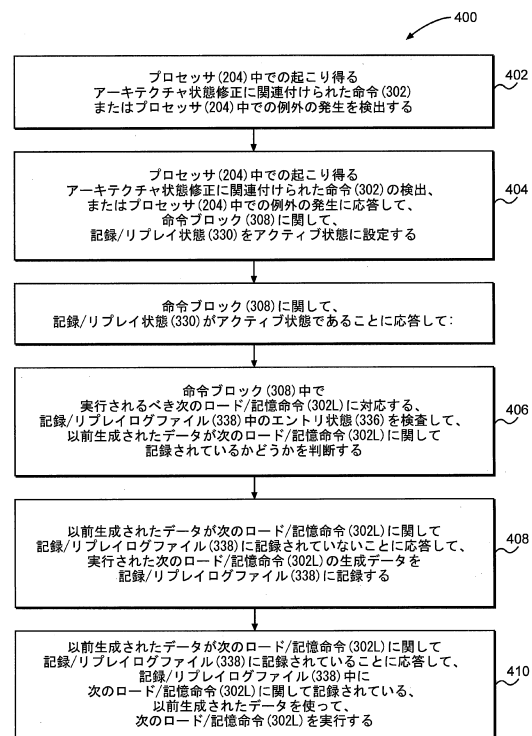
【圖 2】



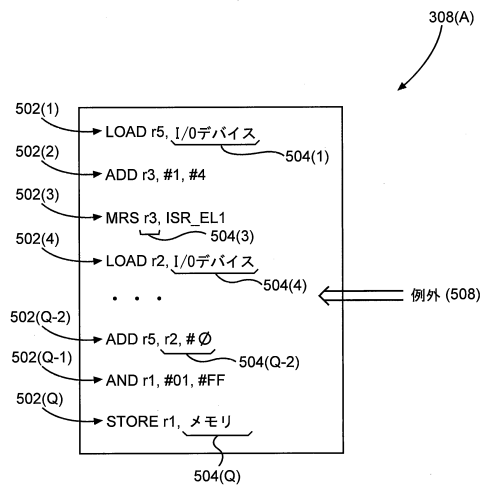
【 図 3 】



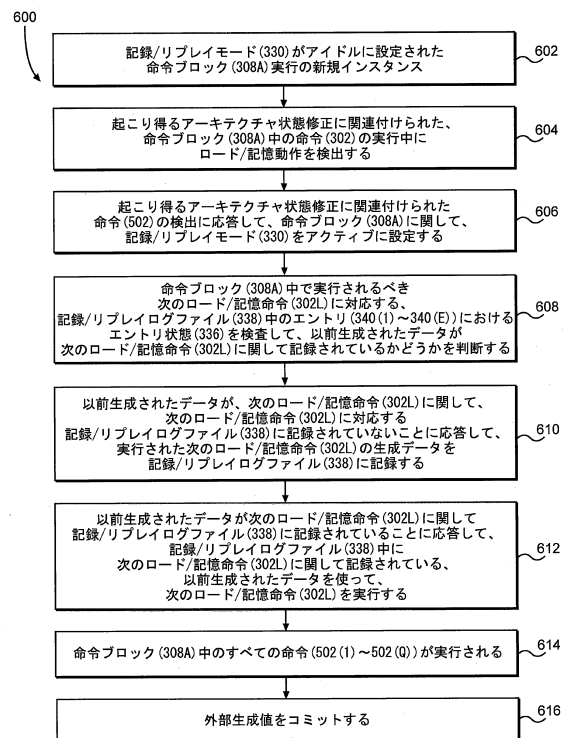
【 図 4 】



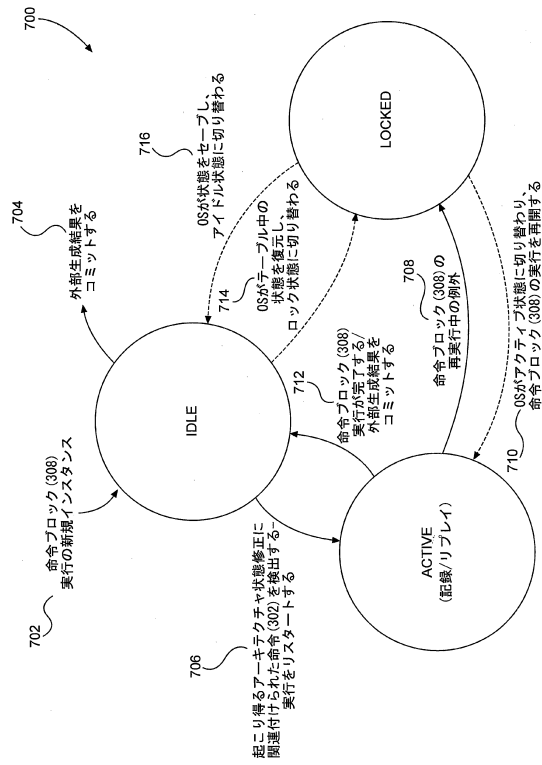
【 図 5 】



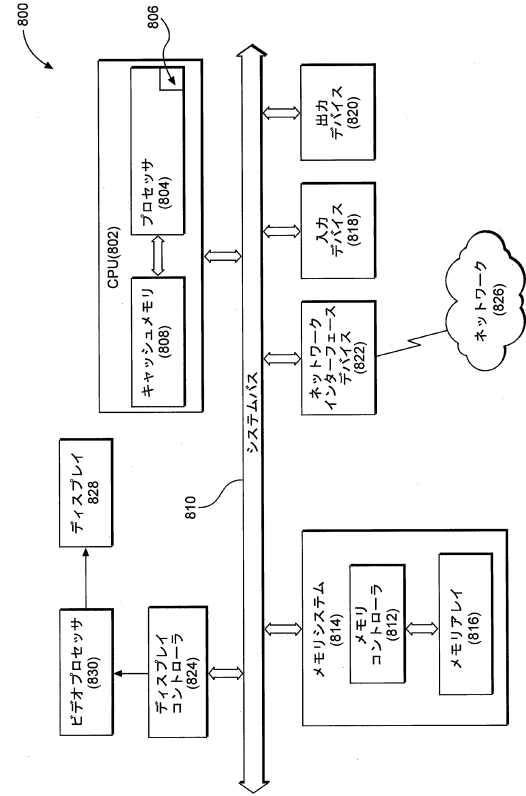
【 図 6 】



【図 7】



【図 8】



フロントページの続き

(56)参考文献 特表平09-509515(JP,A)
特表平11-500551(JP,A)
特表2009-501366(JP,A)

(58)調査した分野(Int.Cl., DB名)

G 0 6 F	9 / 4 5 5	-	9 / 5 4
G 0 6 F	9 / 3 1 2		