

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号  
特許第5850532号  
(P5850532)

(45) 発行日 平成28年2月3日 (2016.2.3)

(24) 登録日 平成27年12月11日 (2015.12.11)

(51) Int. Cl.

G 0 6 F 9 / 3 8 ( 2 0 0 6 . 0 1 )

F I

G 0 6 F 9 / 3 8 3 5 0 A

請求項の数 20 (全 23 頁)

(21) 出願番号	特願2012-545216 (P2012-545216)	(73) 特許権者	390009531
(86) (22) 出願日	平成22年12月13日 (2010.12.13)		インターナショナル・ビジネス・マシーンズ・コーポレーション
(65) 公表番号	特表2013-515306 (P2013-515306A)		INTERNATIONAL BUSINESS MACHINES CORPORATION
(43) 公表日	平成25年5月2日 (2013.5.2)		アメリカ合衆国10504 ニューヨーク州アーモンク ニュー オーチャードロード
(86) 国際出願番号	PCT/EP2010/069496		New Orchard Road, Armonk, New York 10504, United States of America
(87) 国際公開番号	W02011/076602		
(87) 国際公開日	平成23年6月30日 (2011.6.30)	(74) 代理人	100108501
審査請求日	平成25年10月4日 (2013.10.4)		弁理士 上野 剛史
(31) 優先権主張番号	12/644, 923		
(32) 優先日	平成21年12月22日 (2009.12.22)		
(33) 優先権主張国	米国 (US)		
(31) 優先権主張番号	12/822, 960		
(32) 優先日	平成22年6月24日 (2010.6.24)		
(33) 優先権主張国	米国 (US)		
前置審査			最終頁に続く

(54) 【発明の名称】 アウトオブオーダー型マイクロプロセッサにおけるオペランド・ストア比較ハザードの予測及び回避

(57) 【特許請求の範囲】

【請求項 1】

アウトオブオーダー方式で実行されるロード及びストア動作を管理する方法であって、ロード命令の実行を開始するステップと、

前記ロード命令の実行の開始に基づいて、オペランド・ストア比較ハザードに遭遇したことを判断するステップと、

該判断に基づいて、ロード命令の命令アドレスとオペランド・ストア比較ハザードに遭遇したことを示すハザード表示フラグとを含むエントリをオペランド・ストア比較ハザード予測テーブル内に作成するとともに、オペランド・ストア比較ハザードの原因となったストア命令の命令アドレスとオペランド・ストア比較ハザードを起こしたことを示すハザード表示フラグとを含むエントリを前記テーブル内に作成するステップと、

前記ストア命令の実行によって、オペランド・ストア比較ハザードが発生したことを判断するステップと、

該判断に基づいて、オペランド・ストア比較ハザードの原因となったストア命令の命令アドレスとオペランド・ストア比較ハザードを起こしたことを示すハザード表示フラグとを含むエントリを前記テーブル内に作成するとともに、該オペランド・ストア比較ハザードによってヒットされたロード命令の命令アドレスとオペランド・ストア比較ハザードに遭遇したことを示すハザード表示フラグとを含むエントリを前記テーブル内に作成するステップと、

前記オペランド・ストア比較ハザード予測テーブルは、その後のロード命令及びストア

命令で参照され、これらの命令の命令アドレスが登録されているエントリと合致した場合は、そのような命令は、オペランド・ストア比較ハザードに遭遇するとの予想を付したロード命令又はオペランド・ストア比較ハザードを起こすとの予想を付したストア命令として、実行段階に送信されるステップと

を含む、前記方法。

【請求項 2】

前記ストア命令の実行によってオペランド・ストア比較ハザードが発生した場合、パイプラインをフラッシュするステップを含む、請求項 1 に記載の方法。

【請求項 3】

アウトオブオーダー方式で実行されるロード及びストア動作を管理する方法であって、ロード命令及びストア命令のうちの少なくとも 1 つの命令の実行を開始するステップと

、  
前記実行の開始に基づいて、オペランド・ストア比較ハザードに遭遇したことを判断するステップと、

前記判断に基づいて、オペランド・ストア比較ハザード予測テーブル内に、少なくとも、実行が開始された前記少なくとも 1 つの命令の命令アドレスと、実行が開始された前記少なくとも 1 つの命令に関連付けられたハザード表示フラグとを含むエントリを作成するステップであって、前記ハザード表示フラグは、実行が開始された前記少なくとも 1 つの命令が前記オペランド・ストア比較ハザードに遭遇したことを示す、前記作成するステップと

を含み、

前記方法は、

先行する前記ストア命令が実行を開始しており、後続の前記ロード命令が前記データセットを取得済みである記憶場所に、前記ストア命令がストアしたと判断するステップと、

前記判断にตอบสนองして、前記ストア命令により、ストア・ヒット・ロード・ハザードに遭遇したと判断するステップであって、前記ストア命令の実行によりストア・ヒット・ロード・ハザードを検出し、その結果パイプラインをフラッシュするステップを含む、前記遭遇したと判断するステップと、

前記ストア・ヒット・ロード・ハザードに遭遇したと判断するステップにตอบสนองして、前記ストア命令により、前記ロード命令が前記記憶場所から前記データセットを取得したと判断したときに前記ストア命令が比較した最も古いロードキュー・エントリ内に、ハザード表示フラグを設定するステップであって、前記ハザード表示フラグは、前記ロード命令が前記ストア・ヒット・ロード・ハザードに関連付けられることを示す、前記設定するステップと

をさらに含む、前記方法。

【請求項 4】

前記ストア命令により、前記オペランド・ストア比較ハザード予測テーブルを含む命令フェッチ・ユニットに、前記ストア命令が前記ハザード表示フラグを設定したことを通知するステップであって、前記命令フェッチ・ユニットは、前記通知することに対応して前記エントリを生成する、前記通知するステップ；

前記ストア命令が前記ハザード表示フラグを設定するステップにตอบสนองして、ストアキューにより、命令発行ユニットに、前記ストア命令の前記ストアキュー内のストアキュー・エントリ番号を通知するステップ；

前記ロード命令により、前記オペランド・ストア比較ハザード予測テーブルを含む前記命令フェッチ・ユニットに、前記ストア命令が前記ハザード表示フラグを設定したことを通知するステップであって、前記命令フェッチ・ユニットは、前記通知することに対応して前記エントリを生成する、前記通知するステップ；

のうちの少なくとも 1 つをさらに含む、請求項 3 に記載の方法。

【請求項 5】

前記実行するステップは、

前記ロード命令の前に前記ストア命令の実行を開始するステップであって、前記ストア命令に関連付けられたデータセットが遅延される、前記実行を開始するステップと、

前記データセットが前記ストア命令によりストアキュー内に書き込まれる前に前記ロード命令の実行を開始するステップと

をさらに含む、請求項 3 又は 4 に記載の方法。

【請求項 6】

前記データセットが前記ストア命令によりストアキュー内に書き込まれる前に実行されることに基づいて、前記ストア命令により、前記ロード命令が前記ストア命令に依存していると判断するステップと、

前記ロード命令が前記ストア命令に依存しているとの判断に基づいて、前記ロード命令により、ストア・データ転送を実行することができないと判断するステップと、

前記ストア・データ転送を実行することができないとの判断に基づいて、前記ロード命令により、前記ロード命令が転送不能ロード・ヒット・ストアハザードに遭遇したと判断するステップと

をさらに含む、請求項 5 に記載の方法。

【請求項 7】

前記ロード命令が前記転送不能ロード・ヒット・ストアハザードに遭遇したとの判断に基づいて、前記ロード命令により、前記ストア命令に関連付けられた前記ストアキュー内のストアキュー・エントリ内に、ハザード表示フラグを設定するステップであって、前記ハザード表示フラグは、前記ストア命令が前記転送不能ロード・ヒット・ストアハザード

に関連付けられることを示す、前記設定するステップと、  
前記ロード命令が前記転送不能ロード・ヒット・ストアハザードに遭遇したとの判断に基づいて、前記ロード命令により、前記ロード命令に関連付けられたロードキュー内のロードキュー・エントリ内に、ハザード表示フラグを設定するステップであって、前記ハザード表示フラグは、前記ロード命令が前記転送不能ロード・ヒット・ストアハザードに遭遇したことを示す、前記設定するステップと

をさらに含む、請求項 6 に記載の方法。

【請求項 8】

前記ロード命令により、前記オペランド・ストア比較ハザード予測テーブルを含む命令フェッチ・ユニットに、前記ロード命令が前記ロードキュー・エントリ内に前記ハザード表示フラグを設定したことを通知するステップであって、前記命令フェッチ・ユニットは、前記通知することに応答して前記エントリを生成する、前記通知するステップ；

前記ストア命令により、前記オペランド・ストア比較ハザード予測テーブルを含む前記命令フェッチ・ユニットに、前記ロード命令が前記ストアキュー・エントリ内に前記ハザード表示フラグを設定したことを通知するステップであって、前記命令フェッチ・ユニットは、前記通知することに応答して前記エントリを生成する、前記通知するステップ；

のうちの少なくとも 1 つをさらに含む、請求項 7 に記載の方法。

【請求項 9】

アウトオブオーダー方式で実行されるロード及びストア動作を管理する方法であって、ロード命令及びストア命令のうちの少なくとも 1 つの命令の実行を開始するステップと

前記実行の開始に基づいて、オペランド・ストア比較ハザードに遭遇したことを判断するステップと、

前記判断に基づいて、オペランド・ストア比較ハザード予測テーブル内に、少なくとも、実行が開始された前記少なくとも 1 つの命令の命令アドレスと、実行が開始された前記少なくとも 1 つの命令に関連付けられたハザード表示フラグとを含むエントリを作成するステップであって、前記ハザード表示フラグは、実行が開始された前記少なくとも 1 つの命令が前記オペランド・ストア比較ハザードに遭遇したことを示す、前記作成するステップと

を含み、

10

20

30

40

50

前記実行を開始するステップは、

前記ロード命令の前に前記ストア命令の実行を開始するステップであって、前記ストア命令に関連付けられたデータセットが入手可能でない状態が持続する、前記実行を開始するステップと、

前記ストア命令の実行の後に前記ロード命令の実行を開始するステップとを含む、前記方法。

【請求項 10】

前記データセットが入手可能でない状態が持続することに基づいて、前記ロード命令により、前記ロード命令が前記ストア命令に依存していると判断するステップと、

前記ロード命令が前記ストア命令に依存しているとの判断に基づいて、前記ロード命令により、ストア・データ転送を実行できないと判断するステップと、

前記ストア・データ転送を実行することができず、かつ、前記データセットが入手可能でない状態が持続するとの判断に基づいて、前記ロード命令により、前記ロード命令が持続的転送不能ロード・ヒット・ストアハザードに遭遇したと判断するステップと

をさらに含む、請求項 9 に記載の方法。

【請求項 11】

前記ロード命令が前記持続的転送不能ロード・ヒット・ストアハザードに遭遇したとの判断に基づいて、前記ロード命令により、前記ストア命令に関連付けられた前記ストアキュー内のストアキュー・エントリ内に、ハザード表示フラグを設定するステップであって、前記ハザード表示フラグは、前記ストア命令が前記持続的転送不能ロード・ヒット・ストアハザードに関連付けられること示す、前記設定するステップと、

前記ロード命令が前記持続的転送不能ロード・ヒット・ストアハザードに遭遇したとの判断に基づいて、前記ロード命令により、前記ロード命令に関連付けられたロードキュー内のロードキュー・エントリ内に、ハザード表示フラグを設定するステップであって、前記ハザード表示フラグは、前記ロード命令が前記持続的転送不能ロード・ヒット・ストアハザードに遭遇したことを示す、前記設定するステップと

をさらに含む、請求項 10 に記載の方法。

【請求項 12】

前記ロード命令により、前記オペランド・ストア比較ハザード予測テーブルを含む命令フェッチ・ユニットに、前記ロード命令が前記ロードキュー・エントリ内に前記ハザード表示フラグを設定したことを通知するステップであって、前記命令フェッチ・ユニットは、前記通知することに応答して前記エントリを生成する、前記通知するステップ；

前記ストア命令により、前記オペランド・ストア比較ハザード予測テーブルを含む前記命令フェッチ・ユニットに、前記ロード命令が前記ストアキュー・エントリ内に前記ハザード表示フラグを設定したことを通知するステップであって、前記命令フェッチ・ユニットは、前記通知することに応答して前記エントリを生成する、前記通知するステップ；

のうちの少なくとも 1 つを含む、請求項 11 に記載の方法。

【請求項 13】

アウトオブオーダー方式で実行されるロード及びストア動作を管理する方法であって、ロード命令及びストア命令のうちの少なくとも 1 つの命令の実行を開始するステップと

前記実行の開始に基づいて、オペランド・ストア比較ハザードに遭遇したことを判断するステップと、

前記判断に基づいて、オペランド・ストア比較ハザード予測テーブル内に、少なくとも、実行が開始された前記少なくとも 1 つの命令の命令アドレスと、実行が開始された前記少なくとも 1 つの命令に関連付けられたハザード表示フラグとを含むエントリを作成するステップであって、前記ハザード表示フラグは、実行が開始された前記少なくとも 1 つの命令が前記オペランド・ストア比較ハザードに遭遇したことを示す、前記作成するステップと

を含み、

前記方法は、

前記命令の前記命令アドレスについて、前記オペランド・ストア比較ハザード予測テーブルに問い合わせするステップであって、前記オペランド・ストア比較ハザード予測テーブルは、ロード命令に関する第1のエントリと、ストア命令に関する第2のエントリとを含み、前記第1及び第2のエントリは、互いに独立しており、前記ロード命令及び前記ストア命令がそれぞれ以前にオペランド・ストア比較ハザードに遭遇したことを示し、かつ、それぞれ前記ロード命令及び前記ストア命令のうちの1つの少なくとも命令アドレスと、前記オペランド・ストア比較ハザードに関連付けられたハザード表示フラグとを含む、前記問い合わせするステップと、

前記予測テーブルに問い合わせするステップに応答して、前記命令が、前記オペランド・ストア比較ハザード予測テーブル内の前記第1及び第2のエントリのうちの1つに関連付けられると判断するステップと、

前記判断することに基づいて、前記命令に関連付けられた前記第1及び第2のエントリのうちの1つの中に含まれる前記ハザード表示フラグを識別するステップと

をさらに含み、

前記命令が前記第1のエントリに関連付けられている場合、前記命令はロード命令であり、

前記方法は、

前記識別されたハザード表示フラグに基づいて前記命令をマーク付けするステップであって、前記マーク付けするステップは、前記命令の実行を、所与の実行段階に達した前記命令に関連付けられた前記ハザード表示フラグと合致したハザード表示フラグを含む前記予測テーブルの前記第2のエントリに関連付けられた少なくとも1つのストア命令に依存させる、前記マーク付けするステップ

をさらに含み、

前記命令が前記第2のエントリに関連付けられる場合、前記命令はストア命令である、前記方法。

#### 【請求項14】

前記命令に関連付けられた前記ハザード表示フラグは、前記命令が以前に転送不能ロード・ヒット・ストアハザードに遭遇したことを示し、

前記マーク付けするステップは、前記命令の実行を、ストアが転送可能である段階に達した前記命令に関連付けられたハザード表示フラグと合致した前記ハザード表示フラグを含む前記予測テーブルのエントリに関連付けられた全てのストア命令に依存させる、

請求項13に記載の方法。

#### 【請求項15】

前記命令に関連付けられた前記ハザード表示フラグは、前記命令が以前に持続的転送不能ロード・ヒット・ストアハザードに遭遇したことを示し、

前記少なくとも1つのストア命令に関連付けられた前記ハザード表示フラグは、前記少なくとも1つのストア命令が以前に持続的転送不能ロード・ヒット・ストアハザードに遭遇したことを示す、

請求項13に記載の方法。

#### 【請求項16】

前記命令に関連付けられた前記ハザード表示フラグは、前記命令が以前にストア・ヒット・ロード・ハザードに遭遇したことを示し、

前記少なくとも1つのストア命令に関連付けられた前記ハザード表示フラグは、前記少なくとも1つのストア命令が以前にストア・ヒット・ロード・ハザードに遭遇したことを示す、

請求項13に記載の方法。

#### 【請求項17】

前記マーク付けするステップに基づいて、前記命令に関連付けられた前記ハザード表示フラグと合致した前記ハザード表示フラグを含む前記予測テーブルのエントリに関連付け

10

20

30

40

50

られた全てのストア命令が、そのアドレス計算を実行し、そのデータをストアキュー内の対応するエントリ内に書き込むまで、前記ロード命令の前記実行を遅延させるステップをさらに含む、請求項 1 3 に記載の方法。

【請求項 1 8】

前記少なくとも 1 つのストア命令は、前記命令に関連付けられた前記ハザード表示フラグと合致した前記ハザード表示フラグを含む最新のストア命令であり、

前記マーク付けするステップに基づいて、前記少なくとも 1 つのストア命令が L 1 キャッシュ・ライトバックを実行するまで、書き込みビットを有するロード命令の前記実行を遅延させることをさらに含む、

請求項 1 5 に記載の方法。

10

【請求項 1 9】

請求項 1 ~ 1 8 のいずれか一項に記載の方法の各ステップを実行するように適合された手段を含むシステム。

【請求項 2 0】

コンピュータ・プログラムであって、コンピュータ・システムに、請求項 1 ~ 1 8 のいずれか一項に記載の方法の各ステップを実行させる、前記コンピュータ・プログラム。

【発明の詳細な説明】

【技術分野】

【0 0 0 1】

本発明は、一般に、マイクロプロセッサに関し、より具体的には、アウトオブオーダー方式 (out-of-order) で実行されるロード (load) 及びストア (store) 動作の管理に関する。

20

【背景技術】

【0 0 0 2】

機械命令をアウトオブオーダー方式で発行し、実行することができるマイクロプロセッサは、一般に、ロードがストアの前に実行されることを可能にする。この特徴は、ロードアドレス及びストア・アドレスの両方が同じ物理アドレスをもたないことを条件として、大きな性能の利点を可能にする。典型的なプログラムにおいては、ロードはストアより先に進み、それらの物理アドレスが合致する頻度は低い。しかしながら、この書き込み違反 (store violation) 状態の発見は、典型的には、命令実行パイプラインの終わりの方の段階であるので、回復の不利益は相当に厳しいものとなり得る。例えば、回復プロセスは、典型的には、違反を引き起こしたロード命令、及びロード命令以降のプログラム順における全てのより新しい命令を無効にすること、次いでロード命令を再発行することを要する。ストア比較ハザード (store-compare hazard) を管理するための従来のメカニズムは、一般に、これらのハザードをあまり有効に管理するものではない。

30

【先行技術文献】

【非特許文献】

【0 0 0 3】

【非特許文献 1】「z / Architecture Principles of Operation」、SA 2 2 - 7 8 3 2 - 0 7、英語版、International Business Machines 社、2 0 0 9 年 2 月発行

40

【発明の概要】

【発明が解決しようとする課題】

【0 0 0 4】

アウトオブオーダー方式で実行されるロード及びストア動作を管理するための方法及び情報処理システムを提供する。

【課題を解決するための手段】

【0 0 0 5】

一実施形態において、アウトオブオーダー方式で実行されるロード及びストア動作を管理する方法が開示される。この方法は、ロード命令及びストア命令のうちの少なくとも 1

50

つを実行することを含む。この実行に基づいて、オペランド・ストア比較ハザードが発生したという判断がなされる。この判断に基づいて、オペランド・ストア比較ハザード予測テーブル内にエントリが作成される。このエントリは、少なくとも、実行された命令の命令アドレスと、命令に関連付けられたハザード表示フラグ（ビットのような）とを含む。ハザード表示フラグは、命令がオペランド・ストア比較ハザードに遭遇したことを示す。

【0006】

別の実施形態において、アウトオブオーダー方式で実行されるロード及びストア動作を管理する方法が開示される。この方法は、命令をフェッチすることを含む。命令は、ロード命令及びストア命令のうちの1つである。命令はデコードされる。

【0007】

デコードに応答して、命令の命令アドレスについてオペランド・ストア比較ハザード予測テーブルに問い合わせる。オペランド・ストア比較ハザード予測テーブルは、ロード命令に関する第1のエントリと、ストア命令に関する第2のエントリとを含む。第1及び第2のエントリは互いに独立している。第1及び第2のエントリは、それぞれロード命令及びストア命令が以前にオペランド・ストア比較ハザードに遭遇したことを示す。第1及び第2のエントリは、少なくとも、それぞれロード命令及びストア命令の1つの命令アドレスと、オペランド・ストア比較ハザードに関連付けられたハザード表示フラグとを含む。予測テーブルに問い合わせることに応答して、命令が、オペランド・ストア比較ハザード予測テーブル内の第1及び第2のエントリの1つに関連付けられているという判断がなされる。この判断に基づいて、命令に関連付けられた第1及び第2のエントリのうちの1つの中に含まれるハザード表示フラグが識別される。命令が第1のエントリに関連付けられるとき、命令はロード命令である。識別されたハザード表示フラグに基づいて、命令をマーク付けする。このマーク付けは、命令の実行を、所与の実行段階に達した命令に関連付けられたハザード表示フラグと実質的に類似したハザード表示フラグを含む予測テーブルのエントリと関連した少なくとも1つのストア命令に依存させる。命令が第2のエントリに関連付けられるとき、この命令はストア命令である。

【0008】

さらに別の実施形態において、アウトオブオーダー方式で実行されるロード及びストア動作を管理するための情報処理システムが開示される。この情報処理システムは、メモリと、メモリに通信可能に結合されたプロセッサとを含む。プロセッサは、ロード命令及びストア命令のうちの少なくとも1つを実行することを含む方法を実行するように構成される。この実行に基づいて、オペランド・ストア比較ハザードに遭遇したという判断がなされる。この判断に基づいて、オペランド・ストア比較ハザード予測テーブル内にエントリが作成される。エントリは、少なくとも、実行された命令の命令アドレスと、命令に関連付けられたハザード表示フラグ（ビットのような）とを含む。ハザード表示フラグは、命令がオペランド・ストア比較ハザードに遭遇したことを示す。

【0009】

同様の参照番号が、個々の図面を通じて同一の又は機能的に類似した要素を指しており、以下の詳細な説明と共に本明細書内に組み込まれて、本明細書の一部を形成する添付図面は、種々の実施形態をさらに示し、本発明に従った種々の原理及び全ての利点を説明するのに役立つ。

【図面の簡単な説明】

【0010】

【図1】本発明の一実施形態による動作環境の一例を示す。

【図2】本発明の一実施形態によるロードキュー・エントリの一例を示す。

【図3】本発明の一実施形態によるストアキュー・エントリの一例を示す。

【図4】本発明の一実施形態によるオペランド・ストア比較ハザード予測テーブルの一例を示す。

【図5】本発明の種々の実施形態による、オペランド・ストア比較ハザード予測テーブル内のエントリを作成する種々の例を示す動作フロー図である。

【図6】本発明の種々の実施形態による、オペランド・ストア比較ハザード予測テーブル内のエントリを作成する種々の例を示す動作フロー図である。

【図7】本発明の種々の実施形態による、オペランド・ストア比較ハザード予測テーブル内のエントリを作成する種々の例を示す動作フロー図である。

【図8】本発明の種々の実施形態による、オペランド・ストア比較ハザードを予測し、回避する一例を示す動作フロー図である。

【図9】本発明の一実施形態による情報処理システムの一例を示すブロック図である。

【発明を実施するための形態】

【0011】

必要に応じて、本発明の詳細な実施形態を本明細書で開示するが、開示される実施形態は、種々の形態で具体化することができる本発明の例にすぎないことを理解すべきである。従って、本明細書で開示される特定の構造上及び機能上の詳細は、限定として解釈されるべきではなく、単に特許請求の範囲の基礎として、及び、事実上あらゆる適切に詳述される構造及び機能において本発明を様々に用いることを当業者に教示するための代表的な基礎として解釈されるべきである。さらに、本明細書で用いられる用語及び語句は、限定を意図するものではなく、寧ろ、本発明の分かりやすい説明を提供することを意図するものである。

【0012】

本明細書で用いられる「1つの(a)」又は「1つの(an)」という用語は、1つ以上として定義される。本明細書で用いられる「複数の(plurality)」という用語は、2つ以上として定義される。本明細書で用いられる「別の(another)」という用語は、少なくとも第2の又はそれ以上として定義される。本明細書で用いられる「含む(including)」及び「有する(having)」という用語は、備える(comprising)(すなわち、開放的語法(open language))として定義される。本明細書で用いられる「結合される(coupled)」という用語は、必ずしも直接及び機械的ではなく、接続されたものとして定義される。特に断らない限り、複数形及び単数形は同じである。

【0013】

概要

ロード命令及びストア命令をアウトオブオーダー方式で実行するマイクロプロセッサにおいて、依存しているロードとストアの間の並べ替え(reordering)が原因で、3つのオペランド・ストア比較ハザード(ストア・ヒット・ロード(store-hit-load)、転送不能ロード・ヒットストア(non-forwardableload-hit store)、及び持続的転送不能ロード・ヒットストア(persistent non-forwardableload-hit store))が発生することがある。例えば、アドレスAに対するストアに続いてアドレスAに対するロードが行われると仮定する。1つの状況においては、ロードは、ストアの前に実行することができる、即ち、ストアキュー(Store Queue、STQ)は、ストア・アドレス情報を含まない。従って、ロードが実行されるとき、ストアキューは競合を示さない。ひとたびロードが実行を終了すると、ストアが実行され、ロードキュー内の既に終了したロードに対する競合を検出し、パイプラインをフラッシュして、ロード及びあらゆる後続の命令を停止させる。これは、(既に以前推論的に実行されたロード及び全ての後続の命令を)再実行するのに大量の作業が必要であるため、非常に費用のかかる動作である。上記の状況は、ストア・ヒット・ロード(SHL)と呼ばれる。

【0014】

別の状況においては、ストアはそのアドレス計算を実行するが、例えばデータ生成命令が長い待ち時間を有する(例えば、分割する)ために、ストアに関するデータが遅延される。次いで、ストア・データがストアキュー(STQ)に書き込まれる前にロードが実行される。ロードは、それがストアに依存していることを検出するが、データが入手可能でないため、ストア・データ転送(store-data-forwarding)を実行することができない。従って、ロードは、拒絶され、後でストア・データが入手可能になった後に再試行する必要がある。この状況は、転送不能ロード・ヒット・ストア(nf-LHS)と呼ばれる。



## 【 0 0 1 5 】

さらに別の状況においては、上述した  $n f - L H S$  の状況に類似して、特定のストア命令（例えば、これらがライン交差している場合、又は長さが 8 バイトを上回る場合）、一般に、（特定の ストアキュー（STQ） 設計毎に）転送できないことがある。この状況において、ロードは、ストア・データが既に ストアキュー（STQ） 内に書き込まれているとしても、ストアが  $L 1$  キャッシュにライトバックされる（write back、書き戻される）まで、何度も繰り返して再循環させる必要がある。この状況は、「持続的  $n f - L H S$ 」と呼ばれる。

## 【 0 0 1 6 】

アウトオブオーダー処理を管理するための従来のメカニズムは、一般に、上述した 3 つのハザードを有効に処理するものではない。例えば、幾つかの従来のメカニズムは、ロードを全ての先行するストア命令に依存させることによって、命令レベル並列処理（Instruction-Level-Parallelism、ILP）を制限する。他の従来のメカニズムは、一般に、ロード命令が複数のストア命令に依存しており、これらの複数のストア命令がロード命令に通じる異なる分岐パス上にあり得る、一般的な場合を処理することができない。

## 【 0 0 1 7 】

しかしながら、本発明の種々の実施形態の 1 つの利点は、上述した 3 つのハザードが有効に回避されることである。少なくとも 1 つの実施形態において、どのロード及びストアが依存性を有するか、並びに、これらの依存性のタイプ（例えば、 $e$ -ビット又は  $w$ -ビット依存性）を予測する予測テーブルが作成される。次いで、命令のデコード後、 $e$ -ビットロードは、全ての先行する  $e$ -ビット・ストアに依存させられ、命令発行論理により、あたかも正規のレジスタ依存性であるかのように処理される。これにより、全ての  $e$ -ビット・ストアがそのアドレス計算を実行し、そのデータを ストアキュー（STQ） に書き込むまで、 $e$ -ビットロード命令の実行が有効に遅延される。このことにより、事実上、 $S H L$  及び  $n f - L H S$  ハザードが取り除かれることになる。 $w$ -ビット依存性に関して、ロードは、 $w$ -ビット・ストアとして予測された最後のストアの  $L 1$  キャッシュ・ライトバックに依存させられる。このことは、持続的  $n f - L H S$  ハザードを有効に防止する。

## 【 0 0 1 8 】

動作環境

図 1 は、本発明の 1 つ又は複数の実施形態による、1 つ又は複数の処理命令及びデータに適用可能な動作環境 100 の一例を示すブロック図である。プロセッサ 101 は、種々の実行ユニット、レジスタ、バッファ、メモリ、及び全て集積回路によって形成された他の機能ユニットを含む、スーパー scaler・プロセッサのような単一の集積回路プロセッサを含む。一実施形態において、プロセッサ 101 は、命令をアウトオブオーダー方式で発行し、実行することができる。

## 【 0 0 1 9 】

一実施形態において、プロセッサ 101 は、命令フェッチ・ユニット（IFU）102 と、命令デコード・ユニット（IDU）104 と、命令発行ユニット（ISU）106 と、ロード/ストア・ユニット（LSU）108 と、オペランドアドレス生成ユニット 110 と、固定小数点ユニット 112（又は、いずれかの他の実行ユニット）と、種々の他の実行ユニット、レジスタ、バッファ、メモリ、及び他の機能ユニットのような他のコンポーネント 114 とを含む。一実施形態において、命令フェッチ・ユニット（IFU） 102 は、オペランド・ストア比較（operand-store-compare、OSC）予測テーブル 116 を含む。OSC 予測テーブル 116 は、以下に詳述される。

## 【 0 0 2 0 】

この実施形態において、命令発行ユニット 106 は、発行キュー 118 を含む。この実施形態において、ロード/ストア・ユニット（LSU） 108 は、ロードキュー（LDQ）120 と、ストアキュー（STQ）122 と、 $L 1$  キャッシュ 124 とを含む。ロードキュー（LDQ） 120 及び ストアキュー（STQ） 122 の各々は、それぞれ、未処理

10

20

30

40

50

のロード命令及びストア命令に関連付けられた付加的な情報を追跡するエントリ 126、エントリ 128 をそれぞれ含む。本発明の種々の実施形態は、図 1 に示されるようなプロセッサ 101 の構成に限定されるものではないことに留意すべきである。本発明の実施形態は、図 1 に示される例とは異なり得る種々のアーキテクチャに適用可能である。

#### 【0021】

一般に、命令フェッチ・ユニット (IFU) 102 は、L1 キャッシュ 124 の一部とすることができる、I キャッシュ内にストアされた命令コードをフェッチする。これらのフェッチされた命令コードは、命令デコード・ユニット (IDU) 104 によって命令処理データにデコードされる。ひとたびデコードされると、命令はディスパッチされ、適切な発行キュー 118 内に一時的に入れられる。命令は、それらの全ての必要なオペランドが入手可能になるまで、発行キュー 118 内に保持される。発行キュー 118 から、命令を、実行のために、プロセッサ 100 の実行ユニット、例えばロード/ストア・ユニット (LSU) 108、固定小数点ユニット (FXU) 112 等などに日和見的に (opportunistically) 発行することができる。換言すれば、命令は、アウトオブオーダー方式で発行することができる。しかしながら、命令のいずれかを再発行する必要がある場合に備えて、これらの命令は、命令の実行が完了し、もしあれば、結果データがライトバックされるまで、発行キュー 118 内に保持される。

#### 【0022】

実行ユニット 108、112 の1つの中での実行中、命令は、実行ユニットに結合されたレジスタファイル内の1つ又は複数の設計済み (architected) 及び/又はリネーム・レジスタから、もしあれば、オペランドを受け取る。実行ユニットが1つの命令の実行を終えた後、実行ユニットはその結果を、命令で定められるように指定された宛先に書き込み、発行キューからその命令を取り除き、次いで、命令の完了をプログラム順にスケジュールすることができる。オペランドアドレス生成ユニット 110 は、ロード命令及びストア命令に関するオペランドアドレス情報を生成し、これらのアドレスをそれぞれのロードキュー (LDQ) 及びストアキュー (STQ) 122 に書き込む。固定小数点ユニット (FXU) 112 は、データ値をストアキュー (STQ) 122 に書き込む。

#### 【0023】

上述のように、ロード/ストア・ユニット (LSU) 108 は、命令発行ユニット (ISU) 106 からロード命令及びストア命令を受け取り、そのロード命令及びストア命令を実行する。一般に、各々のロード命令は、必要とされるデータのアドレスを指定するアドレス情報を含む。一実施形態において、ロード/ストア・ユニット (LSU) 108 は、ロード命令及びストア命令のアウトオブオーダー実行をサポートし、これにより高レベルの性能を達成することができる。一実施形態において、ロード/ストア・ユニット (LSU) 108 は、パイプライン型である。つまり、ロード/ストア・ユニット (LSU) 108 は、順に実行される一組の順序付けられたパイプライン・ステージを介して、ロード命令及びストア命令を実行する。

#### 【0024】

##### OSC ハザード管理

上述のように、3つのタイプのハザード (ストア・ヒット・ロード、転送不能ロード・ヒットストア、及び持続的転送不能ロード・ヒットストア) は、ロード命令及びストア命令をアウトオブオーダー方式で実行するプロセッサ内で発生し得る。従って、図 1 に関して上述した一般的な処理メカニズムに加えて、以下の実施形態の1つ又は複数をプロセッサ 100 内で実施し、これらのOSC ハザードを予測し、回避することもできる。

#### 【0025】

一般に、あらゆるロードにはロードキュー (LDQ) 内のエントリが割り当てられ、このエントリは、ロードが実行され、完了するまで各々のロードのアドレスを保存する。あらゆるストアにはストアキュー (STQ) 122 内のエントリが割り当てられ、同様にこのエントリは、ストアが完了し、データをL1 キャッシュ 124 に書き込むまで、ストア・アドレス計算の実行から、ストア・アドレスを保存する。しかしながら、検出されたハ

10

20

30

40

50

ガードのタイプに基づいて、ロードキュー (LDQ) エントリ及びストアキュー (STQ) エントリは、OSCハザードを予測し、回避するための付加的な情報を含むこともできる。

【0026】

例えば、一実施形態において、ロード/ストア・ユニット (LSU) 108がロード命令を実行し、このロードをストアキュー (STQ) 122内の対応するエントリと比較する。ロード命令は、ストア・データ転送を実行することができないと判断する。例えば、ストア・データがSTQに書き込まれる前にロードが実行される(nf-LHS)、又はデータが入手可能なときでさえ、ストア・データ転送が許可されない(持続的nf-LHS)。次に、ロード命令がnf-LHSハザードを検出した場合、ロード命令は、これが比較したストアキュー (STQ) エントリ内に、「e-フラグ」(例えば、実行フラグ)のようなOSCハザード表示ビットを設定する。或いは、ロード命令が持続的nf-LHSハザードを検出した場合、ロード命令は同様に、これが比較したストアキュー (STQ) エントリ内に、「w-フラグ」(例えば、書き込みフラグ)のようなOSCハザード表示ビットを設定する。ロード命令はまた、ロードキュー (LDQ)のそれ自体のエントリ内に、e-フラグ又はw-フラグなどの同じOSCハザード表示ビットを設定する。

【0027】

実行されたストア命令がSHLハザードを検出し、ロードキュー (LDQ) エントリに対してSHLフラッシュを実行すると、ストア命令は、それ自体のストアキュー (STQ) エントリ内に「e-フラグ」のようなOSCハザード表示ビットを設定し、命令が比較する(最も古い)ロードキュー (LDQ) エントリ内に、「e-フラグ」のようなOSCハザード表示ビットを設定する。結果として生じるフラッシュのためにこのロードキュー (LDQ) エントリは無効にされるが、「e-フラグ」はロードキュー (LDQ)内に保持されることに留意すべきである。プロセッサのパイプラインがフラッシュされた命令の再フェッチ及び再実行を開始するとき、同じロードが、今やフラッシュ前から設定された「e-フラグ」を有する同じロードキュー (LDQ) エントリに割り当てられる。

【0028】

図2及び図3はそれぞれ、本発明の一実施形態による、ロードキュー (LDQ) エントリ 226及びストアキュー (STQ) エントリ 328の一例を示す。図2に見られるように、オペランドアドレス情報204、有効ビット情報206、及び他の一般情報210などの一般情報に加えて、ロードキュー (LDQ) 120内のエントリ226はまた、1つ又は複数のOSCハザード表示ビット208も含む。上述のように、このOSCハザード表示ビット208は、ロード命令がnf-LHSハザードに遭遇したのか、又は持続的nf-LHSハザードに遭遇したのかに応じて、e-フラグ又はw-フラグとすることができる。上述のように、このOSCハザード表示ビット208は、ストア命令によって設定することもできる。他のタイプのフラグ等を用いて、ロード命令がnf-LHSハザード又は持続的nf-LHSハザードに遭遇したことを示すこともできることに留意すべきである。

【0029】

図3に見られるように、オペランドアドレス情報304、有効ビット情報306、データ値情報308、及び他の一般情報312のような一般情報に加えて、ストアキュー (STQ) 122内のエントリ328は、1つ又は複数のOSCハザード表示ビット310も含む。上述のように、このOSCハザード表示ビット310は、ロード命令がnf-LHSハザードに遭遇したのか、又は持続的nf-LHSハザードに遭遇したのかに応じて、e-フラグ又はw-フラグとすることができる。また、上述のように、このOSCハザード表示ビット310は、ストア命令がSHLハザードに遭遇した場合にe-フラグとすることができる。

【0030】

他のタイプのフラグ等を用いて、ロード命令がnf-LHSハザード又は持続的nf-LHSハザードに遭遇したこと、及び/又はストア命令がSHLハザードに遭遇したこと

10

20

30

40

50

を示すことができることに留意すべきである。また、ロード命令又はストア命令は、それぞれのキューのエントリ内に設定したのと同じビットを他のキューのエントリ内に設定するので、キュー120、122の1つの中のOSCハザード表示ビット208、310は、他のキュー120、122内の少なくとも1つのOSCハザード表示ビット208、310と合致する。また、OSCハザードを予測し、回避するために、OSCハザード表示ビット情報208、310をどのように使用するかについての考察が以下に与えられる。

#### 【0031】

ひとたびロード命令が完了すると、ロード命令は、ロードキュー(LDQ)内にe-フラグ又はw-フラグのようなOSCハザード表示ビット情報を有するかどうかを判断する。OSCハザード表示ビット情報を有する場合には、ロード命令はこれを命令フェッチ・ユニット(IFU)102に示す。次に、一実施形態において、命令フェッチ・ユニット(IFU)102は、この特定のロード命令に関するエントリをOSC予測テーブル116内に生成する。OSC予測テーブルは、ロードの命令アドレスに基づいてエントリを作成し、このロードに関する1つ又は複数のフラグを記憶する。例えば、OSC予測テーブル内のエントリは、ロードがe-フラグ及び/又はw-フラグに関連付けられているかどうかを示し、ここで、ロードが複数のストアキューに対して比較する場合、ロードは両方のフラグを有することができる。

#### 【0032】

ストア命令が完了し、L1キャッシュ124にライトバックされると、ストア命令は、該ストア命令がストアキュー(STQ)122内に、e-フラグ又はw-フラグのようなOSCハザード表示ビット情報を有するかどうかを判断する。OSCハザード表示ビット情報を有する場合、ストア命令は、これを命令フェッチ・ユニット(IFU)102に示す。次に、命令フェッチ・ユニット(IFU)102は、OSC予測テーブル116内に、ストア命令の命令アドレスと、ストアの命令アドレスの下1つ又は複数のフラグとを含む、この特定のストア命令に関するエントリを生成する。また、ストア命令がL1キャッシュ124にライトバックされるとき、ストアキュー(STQ)122は、命令発行ユニット(ISU)106に、ストアキュー(STQ)122内の所与のストア命令のストアキュー(STQ)エントリ番号(スタグ)を通知する。

#### 【0033】

図4は、OSC予測テーブル116内のロード命令エントリ及びストア命令エントリの一例を示す。特定的には、図4は、ロード命令エントリ402が、命令の命令アドレス404と、エントリ有効ビット406と、ロード「e」ビット408及びロード「w」ビット410のような1つ又は複数のハザード表示ビットとを含むことを示す。図4は、ストア命令エントリ403は、命令の命令アドレス412と、エントリ有効ビット414と、ストア「e」ビット416及びストア「w」ビット418のような1つ又は複数のハザード表示ビットとを含むことを示す。上述のように、ロード命令及びストアキュー(STQ)122から得られた情報に基づいて、OSCハザード表示ビット408、410、416、418が設定される。OSC予測テーブル116内の各々のロード及びストア・エントリは、互いに独立して作成される。換言すれば、OSC予測テーブル116内のロード・エントリは、OSC予測テーブル116内のストア・エントリ又はそれと関連した他のいずれのテーブルも参照せず、かつ、それらによっても参照されず、さらに言えば逆も同様である。

#### 【0034】

例えば、図4に示されるように、ロードキュー(LDQ)エントリ内に「e」依存性ビット(dependency bit)を設定し、対応するストア命令のストアキュー(STQ)エントリ内に「e」ビットを設定したロード命令は、これを命令フェッチ・ユニット(IFU)102に通知し、次に、予測テーブル116内にロード命令及びストア命令の各々についてのエントリ402、403を作成する。この例においては、ロード「e」ビット408がロード命令エントリ402内に設定され、ストア「e」ビット416がストア命令エントリ403内に設定される。種々の技術を用いて、命令アドレスに基づいた予測テーブル

10

20

30

40

50

を生成できることにも留意すべきである。例えば、オーバーフローに向けられた技術（最長時間未使用技術）、読み出し／書き込み競合（キューイング）、及び予測テーブルに対する類似の技術を用いることができる。

#### 【 0 0 3 5 】

ひとたび予測テーブル 1 1 6 がポピュレートされる（populate）と、命令が命令フェッチ・ユニット（IFU）1 0 2 によりフェッチされ、命令フェッチ・ユニット（IFU）1 0 2 は、命令デコード・ユニット（IDU）1 0 4 によりデコードするためにパイプライン内に送られる度に、並行して、OSC 予測テーブル 1 1 6 に問い合わせし、この情報を命令デコード・ユニット（IDU）1 0 4 に送る。この問い合わせは、所与のフェッチされた命令が「e」又は「w」ビットを含むロード及び／又はストアとして認識されるかどうかを判断するために、命令デコード・ユニット（IDU）1 0 4 により用いられる。例えば、命令フェッチ・ユニット（IFU）1 0 2 は、命令の命令アドレスを用いて予測テーブル 1 1 6 に問い合わせし、対応する命令アドレスを有するエントリを識別する。OSC 予測テーブル 1 1 6 から得られた情報は、命令デコード・ユニット（IDU）1 0 4 から命令発行ユニット（ISU）1 0 6 に渡される。

#### 【 0 0 3 6 】

命令発行ユニット（ISU）1 0 6 は、この依存性情報を用いて、以下のことを実行する。命令がストア・e - ビットを含む場合、命令発行ユニット（ISU）1 0 6 は、その命令をストア・e - ビット命令として記憶する。命令がストア・w - ビットを有する場合、命令発行ユニット（ISU）1 0 6 は、その命令に対して割り当てられたストアキュー（STQ）エントリ番号（「スタグ（stag）」）を記憶する。一実施形態において、命令発行ユニット（ISU）1 0 6 は、w - スタグと呼ばれる、最も新しいそうしたスタグを記憶する。命令がロード・e - ビットを有する場合、命令発行ユニット（ISU）1 0 6 は、この命令を、ストア・e - ビットでマーク付けされたあらゆる先行命令に依存するものとしてマーク付けする。この依存性は、所与の汎用レジスタのライターとリーダーの間で生成される依存性とよく似ている。命令発行ユニット（ISU）1 0 6 は、ロード命令が、ストア命令が正常に実行され、ストア・アドレス及びデータがストアキュー（STQ）エントリ内に書き込まれるまで実行されないことを保証する。その時点までロードを遅延させることにより、命令発行ユニット（ISU）1 0 6 は、ロードがストア転送済みデータをストアキュー（STQ）エントリから取得することができ、従って、SHL ハザード又はnf - LHS ハザードが回避されることを保証する。

#### 【 0 0 3 7 】

命令がロード・w - ビットを有する場合、ロード（即ち、w - スタグ）前の最後のストア・w - ビット・ストアのスタグがL 1 キャッシュ 1 2 4 にライトバックされるまで、命令発行ユニット（ISU）1 0 6 がロードの実行を許容しないように、ロードがマーク付けされる（上述のように、ストアキュー（STQ）1 2 2 は、いつライトバックが行われるかを、命令発行ユニット（ISU）1 0 6 に通知する）。ストアがL 1 キャッシュ 1 2 4 にライトバックするまでロードを遅延させることにより、持続的nf - LHS ハザードが防止される。事実上、命令発行ユニット（ISU）1 0 6 によって実行される上記のプロセスは、全てのe - ビットロードを、全ての先行するe - ビット・ストアの実行に依存させ、全てのw - ビットロードを全ての先行するw - ビット・ストアのライトバックに依存させる。

#### 【 0 0 3 8 】

例えば分岐の誤りのために、命令発行ユニット（ISU）1 0 6 内の最後にw - スタグを設定したストア命令がフラッシュされることもあることに留意すべきである。しかしながら、依然としてw - スタグを追跡することができる。例えば、一実施形態において、最新のフラッシュされていないw - ビット・ストアが、保存された最新のw - ビット・ストアとして指定される。この選択肢は、相当な追跡費用を必要とし得る。別の実施形態においては、最新のフラッシュされていないスタグが、保存された最新のw - ビット・ストアとなる。このストアを、実際に、w - ビット・ストアとしてマーク付けする必要はない。

この選択肢は、あまり追跡費用を必要とせず、そのストアがw - ビット予測を有していなかった場合でも、ストアをw - ストアとして処理する。このことは、僅かな性能の低下をもたらす。さらに別の実施形態においては、保存された最新のw - ビット・ストアが無効にされる。保存された最新のw - ビット・ストアが依然として無効状態にある間にディスパッチされたw - ビットロードは、スタグ依存性をマーク付けしない。

#### 【0039】

ロードでもストアでもある命令に対して、これらの命令が、それ自体へのw - ビット依存性をマーク付けしないことを確実にするように、ステップが取られる。これは特に、命令が複数の部分で構成される場合に問題となる。一実施形態においては、w - ビットをマーク付けしたストア部分が常にw - ビットをマーク付けしたロード部分の後にくるのを確実にすることによって、或いは、命令の全ての部分がディスパッチされるまで、保存された最新のw - ビット・ストアが更新されないことを確実にすることによって、この依存性の防止を達成することができる。一般に、それらの部分が互いに対して単にインオーダー方式に発行されるので、複数の部分からなる命令内でe - ビット依存性をマーク付けするのが安全である。しかしながら、必要に応じて、類似の方法を用いてこの依存性を防止することが可能である。

#### 【0040】

一実施形態において、e - ビットでマーク付けされたストアのベクトルが保持され（他のレジスタ依存性のような追跡メカニズムが可能である、発行キュー・エン트리毎に1ビット）、e - ビットでマーク付けされたロードは、ベクトル内の対応するビットが設定される全てのより古い有効な発行キュー・エントリに依存させられる。ベクトル内のビットは、命令がそれらにディスパッチされるときに書き込まれ、命令がe - ビット・ストアである場合に設定状態におかれ、他の場合には未設定状態におかれる。

#### 【0041】

代替的に、ロード及びストアは、e 及びw ビットのグループで追跡することができる（例えば、5 個のe - ビットe 1 , . . . , e 5 が存在し、e 1 - ストアとe 1 - ロードだけが対にされる、e 2 - ストアとe 2 - ロードが対にされるなど）。ハザード時には、ロードキュー（LDQ）及びストアキュー（STQ）内に設定するように、無作為のe - ビット（例えば、e 2 ）を選択することができる。この設計は、2 つの態様を別個に処理するが、ストアとしての依存性及びロードとしての依存性の両方を強制することによって、特定の命令がロードでもあり、ストアでもある（例えば、システムz 内のCS 命令）上述した方法に拡張することができる。

#### 【0042】

示されるように、本発明の上記の実施形態は、どのロード及びストアが依存性を有するか、及びこれらの依存性のタイプ（例えば、e - ビット又はw - ビット依存性）を予測する予測テーブルが作成されるという点で有利である。次に、命令のデコード後、e - ビットロードは、全ての先行するe - ビット・ストアに依存させられ、命令発行論理により、あたかも正規のレジスタ依存性であるかのように処理される。これは、全てのe - ビット・ストアがそのアドレス計算を実行し、そのデータをSTQ に書き込むまで、e - ビットロード命令の実行を有効に遅延させる。このことにより、事実上、SHL 及びnf - LHS ハザードが取り除かれる。w - ビット依存性に関して、ロードは、w - ビット・ストアとして予測された最後のストアのL 1 キャッシュ・ライトバックに依存させられる。このことは、持続的nf - LHS ハザードを有効に防止する。各ロード・エン트리及び各ストア・エントリは、OSC 予測テーブル内で互いに独立している。換言すれば、ロード命令エントリはストア命令エントリを参照せず、逆もまた同様である。このことは、依存性が複数のストア命令及び複数のロード命令の間に生成されることを許す。

#### 【0043】

さらに、一実施形態によると、ロードが設定されたe - ビット及びw - ビットの両方を有する場合、ロードは、全てのe - ビット・ストア及び全てのw - ビット・ストアがOSC 予測テーブル内に示されるまで遅延される。つまり、ロードは、全てのe - ビット・ス

10

20

30

40

50

トアがそのアドレス計算を実行し、そのデータをストアキュー (STQ) 内に書き込むまで遅延され、また、最後の w - ビット・ストアの L 1 キャッシュ・ライトバックに依存させられる。

#### 【 0 0 4 4 】

同様に、一例によると、ロードが OSC テーブル内に設定された e - ビットを有し、ストアが OSC テーブル内に設定された e - ビット及び w - ビットの両方を有する場合、ロードは、設定された e - ビットを有するストアがそのアドレス計算を実行し、そのデータをストアキュー (STQ) 内に書き込む (例えば、データが転送される) まで遅延される。ロードが OSC テーブル内に設定された w - ビットを有し、ストアが設定された e - ビット及び w - ビットの両方を有する場合、ロードは、設定された w - ビットを有するストアが L 1 キャッシュのライトバックを実行するまで遅延される。最後に、ロードが設定された e - ビット及び設定された w - ビットの両方を有し、ストアが設定された e - ビット及び設定された w - ビットの両方を有する場合、w - ビットが優先する。設定された w - ビットを有するロードは、設定された w - ビットを有するストアの実行まで遅延され、これは、設定された w - ビットを有するストアの最長遅延 (L 1 キャッシュのライトバックに依存する) に対応する。

#### 【 0 0 4 5 】

一実施形態によると、1つ又は複数のロード及び1つ又は複数のストアが、1つの複合命令内に含まれることがある。このタイプの複合命令の種々の例は、次の非特許文献 1 に見出すことができる。つまり、1つの命令をロード及びストアと見なすことができる。同じ命令が、OSC テーブル内に1つ又は複数のロード・エントリ及び1つ又は複数のストア・エントリを作成することができる。これらのロード及びストアは、OSC テーブル内のそれぞれの e - ビット及び w - ビットによって追跡することができる。

#### 【 0 0 4 6 】

#### 動作フロー図

図 5 は、OSC ハザードを予測し、防止するための OSC 予測テーブル 1 1 6 内にエントリを生成する一例を示す動作フロー図である。図 5 の動作フロー図は、ステップ 5 0 2 で開始し、直ちにステップ 5 0 4 に進む。ステップ 5 0 4 において、関連したストア命令に先立って、ロード命令の実行を開始する。ステップ 5 0 6 において、ロード命令は、ストア命令が将来書き込む記憶場所からデータを取得する。ステップ 5 0 8 において、ロード命令の実行を終了する。ステップ 5 1 0 において、ストア命令の実行を開始する。ステップ 5 1 2 において、ストア命令は、ストア命令が現在書き込んでいる記憶場所からロード命令が以前にデータを取得していると判断する。

#### 【 0 0 4 7 】

ステップ 5 1 4 において、ストア命令は、SHL ハザードに遭遇したことを判断する。次にステップ 5 1 6 において、ストア命令は、この命令 (ストア命令) が OSC ハザード状況の候補であることを示すように、e - フラグビットのようなフラグビットを設定する。この e - フラグビットは、ストア命令が比較する最も古いロードキュー (LDQ) エントリ内に設定される。ステップ 5 1 8 において、ストア命令は、ストアキュー (STQ) 1 2 2 内のストア命令に関連付けられたストアキュー (STQ) エントリ内に e - フラグビットを設定する。次にステップ 5 2 0 において、ストア命令は、ロード命令及びパイプラインからの全てのより新しいロード命令をフラッシュする。

#### 【 0 0 4 8 】

ステップ 5 2 2 において、ストア命令は、L 1 キャッシュ 1 2 4 にライトバックする。ステップ 5 2 4 において、ストア命令は、ストア命令が e - フラグ (又はロード命令によって設定されたような w - フラグ) ビット・ペンディングを有することを、命令フェッチ・ユニット (IFU) 1 0 2 に通知する。ステップ 5 2 6 において、命令フェッチ・ユニット (IFU) 1 0 2 は、OSC 予測テーブル 1 1 6 内に、ストア命令に関するエントリを生成する。このエントリは、ストア命令の命令アドレス、有効ビット、及びストア命令が e - フラグに関連付けられているという表示を含む。例えば、ビット又はフラグをエン

10

20

30

40

50

トリ内に設定し、命令がストア・e - ビットに関連付けられていることを示すことができる。ステップ527において、命令フェッチ・ユニット(IFU)102はまた、OSC予測テーブル116内に、ロード命令に関するエントリも生成する。例えば、ステップ520においてロードがSHLからフラッシュされた後、パイプラインが、以前と同じロードキュー(LDQ)エントリを用いるロードを再実行する。ロードが完了した後、ロードは、そのフラグ(例えば、e - ビット)をOSC予測テーブル116内に書き込む。このエントリは、ロード命令の命令アドレス、有効ビット、及びロード命令がe - フラグに関連付けられているという表示を含む。ストア・エントリ及びロード・エントリは互いに独立しており、決して互いに参照しない。ステップ528において、ストアキュー(STQ)122は、命令発行ユニット(ISU)106に、L1キャッシュ124にライトバックされたストア命令のストアキュー(STQ)エントリ番号(スタグ)を通知する。次に、ステップ530において、制御フローが終了する。

#### 【0049】

図6は、OSCハザードを予測し、防止するためのOSC予測テーブル116内にエントリを生成する別の例を示す動作フロー図である。図6の動作フロー図は、ステップ602で開始し、直ちにステップ604に進む。ステップ604において、ストア命令がそのストア命令に関するアドレス計算を実行する。ステップ606において、ストア命令に関するデータが遅延される。ステップ608において、ストア・データがストアキュー(STQ)122に書き込まれる前に、ロード命令の実行を開始する。ステップ610において、ロード命令は、ロード命令がストア命令に依存しており、ストア・データ転送を実行できないと判断する。

#### 【0050】

ステップ612において、ロード命令は、nf - LHS状況に遭遇したと判断する。ステップ614において、ロード命令は、ストア命令のストアキュー(STQ)エントリ内にe - フラグビットを設定する。ステップ616において、ロード命令は、対応するロードキュー(LDQ)エントリ内にe - フラグビットを設定する。ステップ618において、ロード命令が実行を終了する。ステップ620において、ロード命令は、ロード命令がロードキュー(LDQ)内にe - フラグビットを設定したという情報を命令フェッチ・ユニット(IFU)102に送る。ステップ622において、IFUが、OSCテーブル116内にロード命令に関するエントリを生成する。このエントリは、ロード命令の命令アドレス、有効ビット、及びロード命令がe - フラグビットに関連付けられているという表示を含む。例えば、命令がロード・e - ビットに関連付けられていることを示すビット又はフラグをエントリ内に設定することができる。ステップ623において、命令フェッチ・ユニット(IFU)102も同様に、OSC予測テーブル116内にストア命令に関するエントリを生成する。例えば、ストア命令がL1キャッシュ124にライトバックするとき(ステップ620の前又は後に起こり得る)、ストアキュー(STQ)内のフラグ(e - ビット)が命令フェッチ・ユニット(IFU)102に通信され、ストアに関するエントリがテーブル116内に作成される。このエントリは、ストア命令の命令アドレス、有効ビット、及びストア命令がe - フラグビットに関連付けられているという表示を含む。ロード及びストア・エントリは互いに独立しており、決して互いに参照しない。次に、ステップ624において、制御フローが終了する。

#### 【0051】

図7は、OSCハザードを予測し、防止するためのOSC予測テーブル116内にエントリを生成するさらに別の例を示す動作フロー図である。図7の動作フロー図はステップ702で開始し、直ちにステップ704に進む。ステップ704において、ストア命令がそのアドレス計算を実行する。ステップ706において、ロード命令の実行を開始する。ステップ708において、ロード命令は、ロード命令がストア命令に依存しており、ストア・データ転送を実行できないと判断する。

#### 【0052】

ステップ710において、ロード命令が持続的nf - LHS状況に遭遇したと判断する

10

20

30

40

50



。ステップ 712 において、ロード命令は、ストア命令のストアキュー (STQ) エントリ内に w - フラグビットを設定する。ステップ 714 において、ロード命令は、対応するロードキュー (LDQ) エントリ内に w - フラグビットを設定する。ステップ 716 において、ロード命令が実行を終了する。ステップ 718 において、ロード命令は、ロード命令がロードキュー (LDQ) 内に w - フラグビットを設定したことを命令フェッチ・ユニット (IFU) 102 に通知する。ステップ 720 において、IFU は、OSC 予測テーブル 116 内にロード命令に関するエントリを生成する。このエントリは、ロード命令の命令アドレス、有効ビット、及びロード命令が w - フラグビットに関連付けられているという表示を含む。例えば、命令がロード・w - ビットに関連付けられていること示すビット又はフラグをエントリ内に設定することができる。ステップ 721 において、IFU は、OSC 予測テーブル 116 内にストア命令に関するエントリを生成する。例えば、ストア命令が L1 キャッシュ 124 にライトバックするとき (ステップ 718 の前又は後に起こり得る)、STQ 内のフラグ (e - ビット) が命令フェッチ・ユニット (IFU) 102 に通信され、ストアに関するエントリがテーブル 116 内に作成される。このエントリは、ストア命令の命令アドレス、有効ビット、及びストア命令が w - フラグビットに関連付けられているという表示を含む。ロード及びストア・エントリは互いに独立しており、決して互いに参照しない。次に、ステップ 722 において、制御フローが終了する。

#### 【0053】

図 8 は、OSC ハザードを予測し、防止する一例を示す動作フロー図である。図 8 の動作フロー図はステップ 802 で開始し、直ちにステップ 804 に進む。ステップ 804 において、命令フェッチ・ユニット (IFU) 102 が命令をフェッチする。ステップ 806 において、命令フェッチ・ユニット (IFU) 102 は、並行して、命令の命令アドレスを用いて OSC 予測テーブル 116 に問い合わせする。ステップ 808 において、命令フェッチ・ユニット (IFU) 102 は、命令が予測テーブル 116 内のエントリを含むかどうかを判断する。この判断の結果が否定である場合には、ステップ 810 において、通常の処理が行われる。次いでステップ 812 において、制御フローが終了する。判断の結果が肯定である場合、ステップ 813 において、命令フェッチ・ユニット (IFU) 102 は、命令、及び OSC 予測テーブル 116 から取得された命令に関連付けられた OSC ハザード情報を命令デコード・ユニット (IDU) 104 に送る。ステップ 814 において、命令デコード・ユニット (IDU) 104 が命令をデコードする。

#### 【0054】

ステップ 808 における判断の結果が肯定である場合、ステップ 816 において、命令デコード・ユニット (IDU) 104 は、命令がストア・e - ビットを含む (例えば、命令が e - フラグビットを有するストアである) かどうかを判断する。判断の結果が肯定である場合、ステップ 808 において、命令発行ユニット (ISU) 106 は、ストア命令をストア・e - ビットとして記憶する。この命令が単一ビットしか含まない場合、次に制御フローはステップ 804 に戻る。しかしながら、命令が 1 つ又は複数のロード及び 1 つ又は複数のストアの両方を実行するために命令が複数のビットを含む場合、デコード・ユニット (IDU) 104 は、他のビットがロード「e」ビットである場合にはステップ 826 及び 828 を実行し、又は他のビットがロード「w」ビットである場合にはステップ 832 を実行する。

#### 【0055】

ステップ 816 における判断の結果が否定である場合、ステップ 820 において、命令デコード・ユニット (IDU) 104 は、命令がストア・w - ビットを含む (例えば、命令が w - フラグビットを有するストアである) かどうかを判断する。判断の結果が肯定である場合、ステップ 822 において、命令発行ユニット (ISU) 106 は、ストア命令に割り当てられたストアキュー (STQ) エントリ番号 (スタグ) を記憶する。次いで制御フローはステップ 804 に戻る。

#### 【0056】

ステップ 820 の判断の結果が否定である場合、ステップ 824 において、命令デコー

10

20

30

40

50

ド・ユニット ( I D U ) 1 0 4 は、命令がロード・e - ビットを含む (例えば、命令が e - フラグビットを有するロードである) かどうかを判断する。判断の結果が肯定である場合、ステップ 8 2 6 において、命令発行ユニット ( I S U ) 1 0 6 は、ストア・e - ビットでマーク付けされたあらゆる先行命令に依存するものとして、ロード命令をマーク付けする。ステップ 8 2 8 において、マーク付けされた結果として、ストアが成功裏に実行され、ストア・アドレス及びデータがそのストアキュー ( S T Q ) エントリ内に書き込まれるまで、ロード命令を実行することができない。次いで制御フローはステップ 8 0 4 に戻る。

#### 【 0 0 5 7 】

ステップ 8 2 4 の判断が否定である場合、ステップ 8 3 0 において、命令は、w - フラグビットを有するロードであると判断される。ステップ 8 3 2 において、命令発行ユニット ( I S U ) 1 0 6 は、ロードの前の最後のストア・w - ビット・ストアのスタグ (即ち、w - スタグ) が L 1 キャッシュにライトバックするまで実行されないように、この命令をマーク付けする。次に制御フローはステップ 8 0 4 に戻る。

#### 【 0 0 5 8 】

### 情報処理システム

図 9 は、本発明の一実施形態による情報処理システム 9 0 0 の詳細な図を示すブロック図である。この情報処理システム 9 0 0 は、本発明の 1 つ又は複数の実施形態を実施するように適合された、適切に構成された処理システムに基づくものである。いずれかの適切に構成された処理システムを、本発明の実施形態による情報処理システム 9 0 0 として同様に用いることができる。

#### 【 0 0 5 9 】

情報処理システム 9 0 0 は、コンピュータ 9 0 2 を含む。コンピュータ 9 0 2 は、図 1 のプロセッサのようなプロセッサ 1 0 1 を有する。プロセッサ 1 0 1 は、図 1 に示されるように、O S C 予測テーブル 1 1 6 を含む命令フェッチ・ユニット ( I F U ) 1 0 2 と、命令デコード・ユニット ( I D U ) 1 0 4 と、発行キュー 1 1 8 を含む命令発行ユニット ( I S U ) 1 0 6 と、ロードキュー ( L D Q )、ストアキュー ( S T Q ) 1 2 2、及び L 1 キャッシュ 1 2 4 を含むロード/ストア・ユニット ( L S U ) 1 0 8 と、オペランドアドレス生成ユニット 1 1 0 と、固定小数点ユニット ( F X U ) 1 1 2 と、種々の他のコンポーネント 1 1 4 とを含む。

#### 【 0 0 6 0 】

プロセッサ 1 0 1 は、主メモリ 9 0 6、マスストレージインタフェース 9 0 8、及びネットワークアダプタハードウェア 9 1 0 に接続される。システムバス 9 1 2 がこれらのシステムコンポーネントを相互接続する。マスストレージインタフェース 9 0 8 は、データストレージ装置 9 1 4 のようなマスストレージ装置を接続するために用いられる。データストレージ装置の 1 つの特定のタイプは、C D / D V D 9 1 6 (これに限定されるものではないが) などのコンピュータ可読媒体又はストレージ製品にデータをストアし、これからデータを読み出すために用いることができる、C D / D V D ドライブのような光ドライブである。別のタイプのストレージ装置は、例えばファイルシステムの動作をサポートするように構成されたデータストレージ装置である。

#### 【 0 0 6 1 】

一実施形態において、情報処理システム 9 0 0 は、従来の仮想アドレス指定メカニズムを用いて、プログラムが、あたかも、主メモリ 9 0 6 及びデータストレージ装置 9 1 6 のような複数のより小さいストレージ・エンティティにアクセスするのではなく、本明細書でコンピュータ・システム・メモリと呼ばれる、より大きい単一のストレージ・エンティティにアクセスするかのよう動作することを可能にする。「コンピュータ・システム・メモリ」という用語は、本明細書において、情報処理システム 9 0 0 の仮想メモリ全体を総称的に指すように用いられる。

#### 【 0 0 6 2 】

コンピュータ 9 0 2 に関して 1 つだけのプロセッサ 1 0 1 が示されるが、複数のプロセ

10

20

30

40

50

ッサを有するコンピュータ・システムを等しく有効に用いることもできる。本発明の種々の実施形態は、各々が、プロセッサ 101 からのオフロード処理のために用いられる完全にプログラムされた別個のマイクロプロセッサを含む、インタフェースをさらに組み込む。主メモリ内に含まれるオペレーティングシステム（図示せず）は、限定するものではないが、GNU/Linux、AIX、Solaris、及びHP-UXのような適切なマルチタスクオペレーティングシステムである。本発明の種々の実施形態は、いずれかの他の適切なオペレーティングシステムを用いることもできる。本発明の幾つかの実施形態は、オペレーティングシステム（図示せず）のコンポーネントの命令が、情報処理システム 900 内に配置されたいずれかのプロセッサ上で実行されるのを可能にする、オブジェクト指向フレームワークメカニズムのようなアーキテクチャを用いる。ネットワークアダプタハードウェア 910 は、1つ又は複数のネットワーク 918 へのインタフェースを与えるように用いられる。本発明の種々の実施形態は、今日のアナログ及び/又はデジタル技術を含む、又は将来のネットワーキングメカニズムを紹介する、いずれかのデータ通信接続と共に機能するように適合させることができる。

#### 【0063】

本発明の例示的な実施形態を完全に機能的なコンピュータ・システムとの関連で説明したが、当業者であれば、実施形態は、例えば、CD 916、CD-ROM、又は他の形態の記録可能媒体などのCD若しくはDVDを介して、或いは任意のタイプの電子伝送メカニズムを介して、プログラム製品として配布できることを認識するであろう。

#### 【0064】

限定されない例

本発明の特定の実施形態を開示したが、当業者であれば、本発明の趣旨及び範囲から逸脱することなく、特定の実施形態に対する変更を行うことができることを理解するであろう。従って、本発明の範囲は、特定の実施形態に限定されず、添付の特許請求の範囲は、本発明の範囲内のいずれか又は全てのこのような用途、修正及び実施形態を包含することが意図される。

#### 【0065】

本発明の種々の例示的な実施形態を、完全に機能的なコンピュータ・システムとの関連で説明したが、当業者であれば、種々の実施形態は、CD若しくはDVD、例えばCD 916、CD-ROM、又は他の形態の記録可能媒体を介して、又は任意の型の電子伝送メカニズムを介して、プログラム製品として配布することができることを認識するであろう。

#### 【符号の説明】

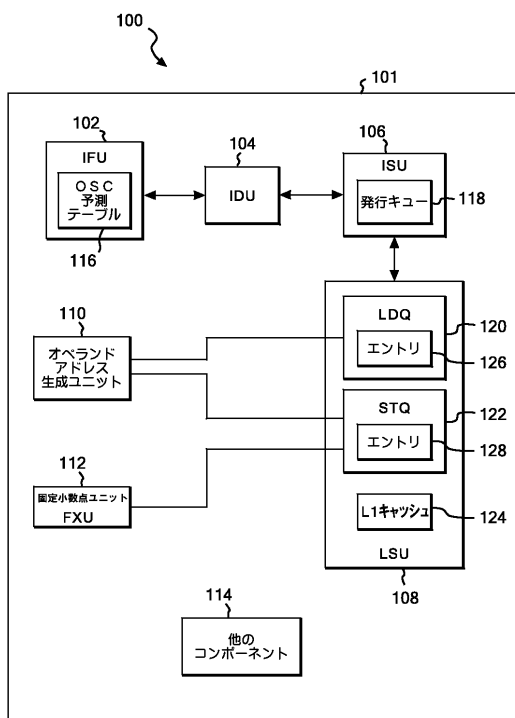
#### 【0066】

- 100：動作環境
- 101：プロセッサ
- 102：命令フェッチ・ユニット（IFU）
- 104：命令デコード・ユニット（IDU）
- 106：命令発行ユニット（ISU）
- 108：ロード/ストア・ユニット（LSU）
- 110：オペランドアドレス生成ユニット
- 112：固定小数点ユニット（FXU）
- 114：他のコンポーネント
- 116：オペランド・ストア比較（OSC）予測テーブル
- 118：発行キュー
- 120：ロードキュー（LDQ）
- 122：ストアキュー（STQ）
- 124：L1キャッシュ
- 126、128、226、328、402、403：エントリ
- 204、304：オペランドアドレス情報

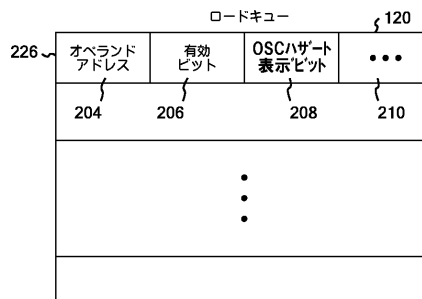
206、306：有効ビット情報  
 208、310：OSCハザード表示ビット  
 210、312：他の一般情報  
 308：データ値情報  
 404、412：命令アドレス  
 406、414：入力有効ビット  
 408：ロード「e」ビット  
 410：ロード「w」ビット  
 416：ストア「e」ビット  
 418：ストア「w」ビット  
 900：情報処理システム  
 902：コンピュータ  
 906：主メモリ  
 908：マスタストレージインタフェース  
 910：ネットワークアダプタハードウェア  
 912：システムバス  
 914：データストレージ  
 916：CD/DVD  
 918：ネットワーク

10

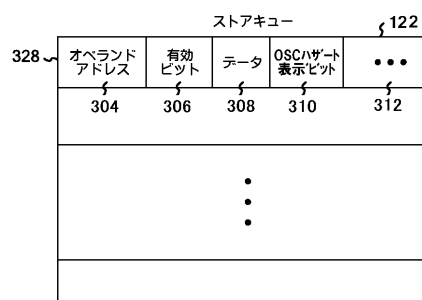
【図1】



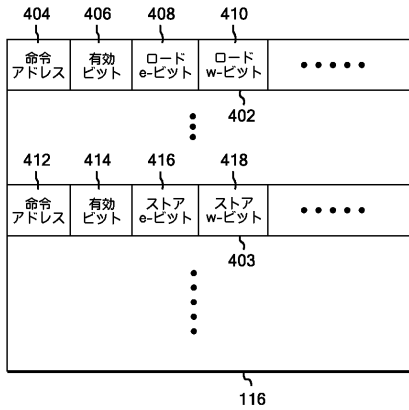
【図2】



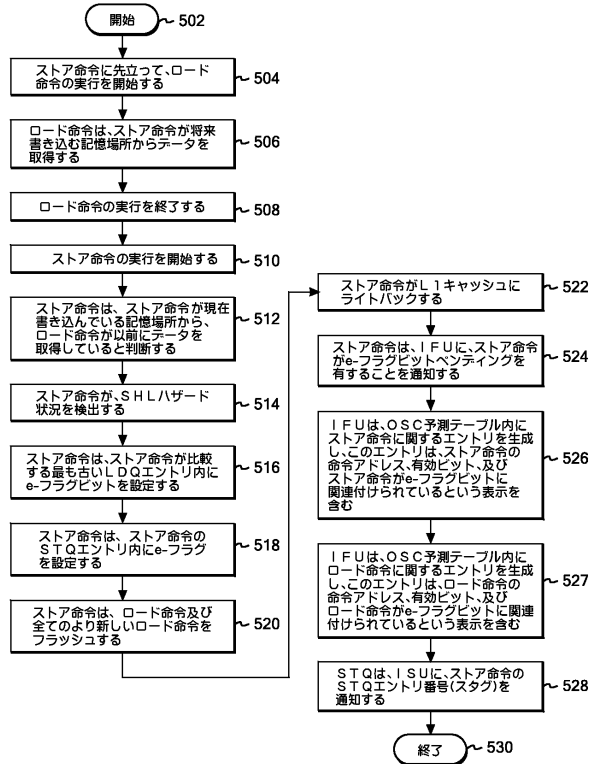
【図3】



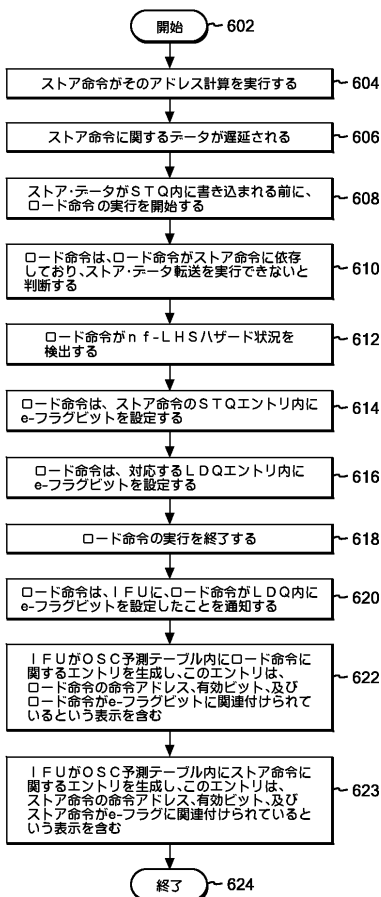
【図 4】



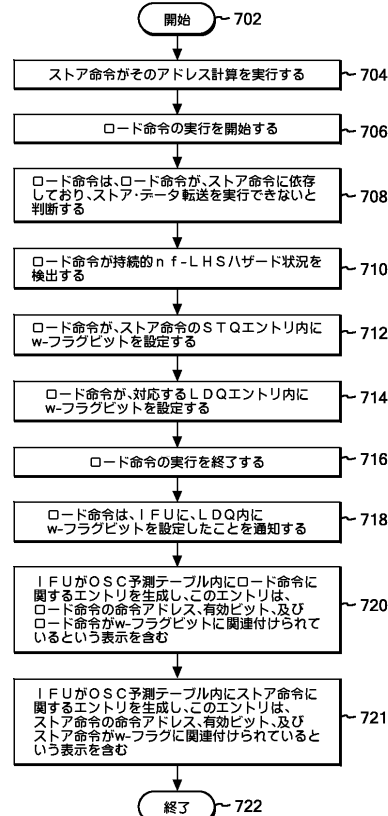
【図 5】



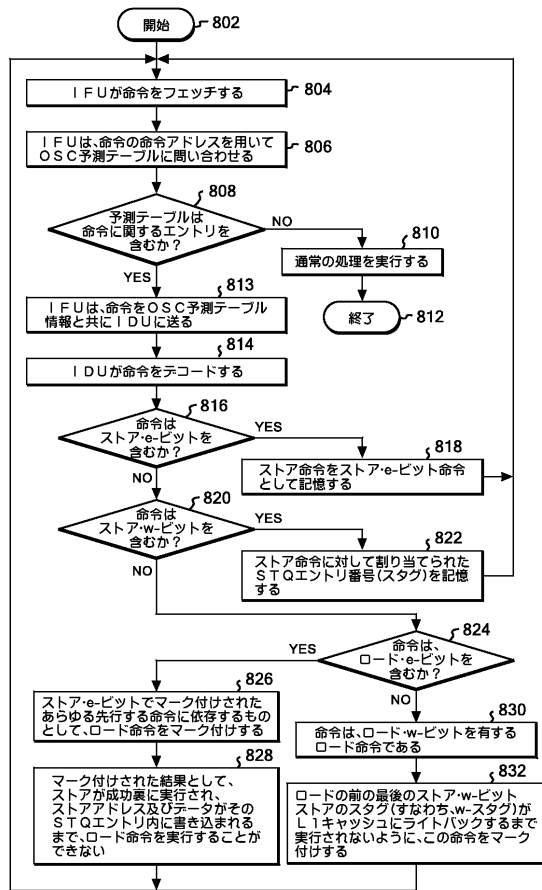
【図 6】



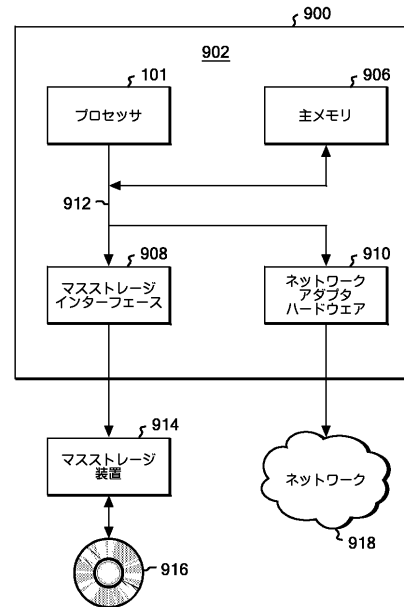
【図 7】



【図 8】



【図 9】



## フロントページの続き

- (74)代理人 100112690  
弁理士 太佐 種一
- (72)発明者 ジャコビ、クリスチャン  
ドイツ連邦共和国 7 1 0 3 2 ベープリングゲン シェーナイヒャー・シュトラッセ 2 2 0 ビー・オー・エー アイ・ピー・エム・ドイツ・リサーチ・アンド・デベロップメント有限会社
- (72)発明者 トンプト、ブライアン、ウィリアム  
アメリカ合衆国 7 8 7 5 8 テキサス州 オースティン バーネット・ロード 1 1 4 0 0 郵便受け 4 3 6 1
- (72)発明者 アレクサンダー、グレゴリー、ウィリアム  
アメリカ合衆国 7 8 7 5 8 テキサス州 オースティン バーネット・ロード 1 1 4 0 0 0 4 5 - 3 ディー - 0 3 0
- (72)発明者 アレクサンダー、カリー、ジェーソン  
アメリカ合衆国 1 2 6 0 1 ニューヨーク州 ポキプシー サウス・ロード 2 4 5 5 7 0 5 - 2 - エル 1 8
- (72)発明者 カラン、ブライアン、ウィリアム  
アメリカ合衆国 1 2 6 0 1 ニューヨーク州 ポキプシー サウス・ロード 2 4 5 5 郵便受け ピー 3 1 0
- (72)発明者 ミッチェル、ジェームズ、ラッセル  
アメリカ合衆国 1 2 6 0 1 ニューヨーク州 ポキプシー サウス・ロード 2 4 5 5 郵便受け ピー 3 1 8
- (72)発明者 シェ、ジョナサン、ティン  
アメリカ合衆国 1 2 6 0 1 ニューヨーク州 ポキプシー サウス・ロード 2 4 5 5 7 0 5 2 エル 0 4
- (72)発明者 プラスキー、ブライアン、ロバート  
アメリカ合衆国 1 2 6 0 1 ニューヨーク州 ポキプシー サウス・ロード 2 4 5 5 郵便受け ピー 3 1 2

審査官 三坂 敏夫

- (56)参考文献 特表 2 0 0 7 - 5 3 6 6 2 6 ( J P , A )  
特表 2 0 0 5 - 5 3 2 6 1 3 ( J P , A )  
米国特許第 0 6 4 6 3 5 2 3 ( U S , B 1 )  
Andreas MOSHOVOS et al. , "Streamlining Inter-operation Memory Communication via Data Dependence Prediction" , Microarchitecture, 1997. Proceedings. , Thirtieth Annual IEEE/ACM International Symposium on , 米国 , IEEE , 1 9 9 7 年 1 2 月 3 日 , pages:235-245  
Andreas MOSHOVOS et al. , "Speculative Memory Cloaking and Bypassing" , [online] , 米国 , CiteSeerX , 1 9 9 9 年 , pages:427-456 , [平成 2 6 年 9 月 2 2 日検索]、インターネット URL : <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.74.2941&rep=rep1&type=pdf>

- (58)調査した分野(Int.Cl. , D B 名)  
G 0 6 F 9 / 3 8