

(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2009-70378
(P2009-70378A)

(43) 公開日 平成21年4月2日(2009.4.2)

(51) Int.Cl.		F I			テーマコード (参考)
G06F 9/38	9/38	(2006.01)	G06F 9/38	350B	5B013
G06F 9/30	9/30	(2006.01)	G06F 9/30	350A	5B033

審査請求 未請求 請求項の数 20 O L (全 13 頁)

(21) 出願番号 特願2008-230625 (P2008-230625)
 (22) 出願日 平成20年9月9日 (2008.9.9)
 (31) 優先権主張番号 11/856170
 (32) 優先日 平成19年9月17日 (2007.9.17)
 (33) 優先権主張国 米国 (US)

(71) 出願人 390009531
 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
 INTERNATIONAL BUSINESS MACHINES CORPORATION
 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
 (74) 代理人 100108501 弁理士 上野 剛史
 (74) 代理人 100112690 弁理士 太佐 種一
 (74) 代理人 100091568 弁理士 市位 嘉宏

最終頁に続く

(54) 【発明の名称】 アウト・オブ・オーダー・プロセッサにおける述語型実行のための方法および装置

(57) 【要約】

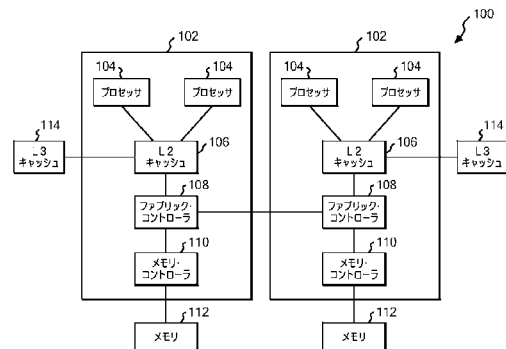
【課題】

アウト・オブ・オーダー・プロセッサにおいて予測実行するための方法および装置を提供する。

【解決手段】

アウト・オブ・オーダー・プロセッサにおいて述語付きコードを扱うための技術は、述語付きコード領域と関連した述語定義命令を検出することを含む。更に、述語付き定義命令の述語が解明されるまで、述語付きコード領域内の述語付き命令のリネーミングはストールされる。

【選択図】 図1



【特許請求の範囲】**【請求項 1】**

アウト・オブ・オーダー・プロセッサにおいて述語付きコードを処理するための方法であつて、

述語付き命令を含む述語付きコード領域に関連した述語定義命令を検出するステップと

、
前記述語定義命令の述語が解明されるまで、前記述語付きコード領域内に含まれた述語付き命令のリネーミングをストールするステップと、

を含む、方法。

【請求項 2】

10

前記述語定義命令を検出するステップは、前記述語付きコード領域に関連した述語定義命令を前記アウト・オブ・オーダー・プロセッサのリネーム・ステージにおいて検出するステップを更に含む、請求項 1 に記載の方法。

【請求項 3】

前記述語定義命令に関連したマスクをマスク・シフト・レジスタに格納するステップと

、
前記述語付き命令のうちのどれをリネームおよび実行すべきかを決定するために前記述語および前記格納されたマスクに関して論理機能を遂行するステップと、

を更に含む、請求項 2 に記載の方法。

【請求項 4】

20

前記述語が解明されたという表示を受け取るステップと、

選択されたパスに関連する述語付き命令をリネームおよび実行するステップと、

を更に含む、請求項 1 に記載の方法。

【請求項 5】

前記述語が解明されたという表示を受け取るステップと、

選択されなかったパスに関連する述語付き命令を廃棄するステップと、

を更に含む、請求項 1 に記載の方法。

【請求項 6】

前記述語が解明されたという表示を受け取るステップと、

選択されたパスに関連する述語付き命令をリネームおよび実行するステップと、

選択されなかったパスに関連する述語付き命令を廃棄するステップと、

を更に含む、請求項 1 に記載の方法。

30

【請求項 7】

前記述語は状態レジスタ値に基づいている、請求項 1 に記載の方法。

【請求項 8】

前記述語定義命令をデコードするステップと、

前記述語付き命令をデコードするステップと、

を更に含む、請求項 1 に記載の方法。

【請求項 9】

40

述語付き命令を含む述語付きコード領域に関連した述語定義命令をデコードするように構成されたデコーダと、

前記述語定義命令を検出するように、および、前記述語定義命令の述語が解明されるまで前記述語付き命令のリネーミングをストールするように構成されたパイプライン・ステージと、

を含むアウト・オブ・オーダー・プロセッサ。

【請求項 10】

前記パイプライン・ステージはリネーム・ステージに含まれる、請求項 9 に記載のアウト・オブ・オーダー・プロセッサ。

【請求項 11】

前記リネーム・ステージは、更に、

50

前記述語定義命令に関連するマスクをマスク・シフト・レジスタに格納し、
前記述語付き命令のうちのどれをリネームおよび実行すべきかを決定するために前記述語および前記格納されたマスクに関して論理機能を遂行する
ように構成される、請求項 10 に記載のアウト・オブ・オーダー・プロセッサ。

【請求項 12】

前記パイプライン・ステージは、更に、
前記述語が解明されたという表示を受け取り、
選択されたパスに関連する述語付き命令をリネームする
ように構成される、請求項 9 に記載のアウト・オブ・オーダー・プロセッサ。

【請求項 13】

前記パイプライン・ステージは、更に、
前記述語が解明されたという表示を受け取り、
選択されなかったパスに関連する述語付き命令を廃棄する
ように構成される、請求項 9 に記載のアウト・オブ・オーダー・プロセッサ。

【請求項 14】

前記パイプライン・ステージは、更に、
前記述語が解明されたという表示を受け取り、
選択されたパスに関連する述語付き命令をリネームし、
選択されなかったパスに関連する述語付き命令を廃棄する
ように構成される、請求項 9 に記載のアウト・オブ・オーダー・プロセッサ。

【請求項 15】

前記述語は状態レジスタ値に基づいている、請求項 9 に記載のアウト・オブ・オーダー・プロセッサ。

【請求項 16】

1 つまたは複数のコンピュータ可読記憶媒体上に保持される 1 つまたは複数の設計ファイルに含まれる、請求項 9 に記載のアウト・オブ・オーダー・プロセッサ。

【請求項 17】

メモリ・サブシステムと、
前記メモリ・サブシステムに接続されたアウト・オブ・オーダー・プロセッサと、
を含み、
前記アウト・オブ・オーダー・プロセッサは、
述語付き命令を含む述語付きコード領域に関連した述語定義命令をデコードするように構成されたデコーダと、

前記述語定義命令を検出するように、および、前記述語定義命令の述語が解明されるまで前記述語付き命令のリネーミングをストールするように構成されたりネーム・ステージと、

を含む、プロセッサ・システム。

【請求項 18】

前記リネーム・ステージは、更に、
前記述語定義命令に関連したマスクをマスク・シフト・レジスタに格納し、
前記述語付き命令のうちのどれを実行すべきかを決定するために前記述語および前記格納されたマスクに関して論理機能を遂行し、
前記論理機能の結果に基づいて前記述語付き命令の少なくとも 1 つをリネームする
ように構成される、請求項 17 に記載のプロセッサ・システム。

【請求項 19】

前記リネーム・ステージは、更に、
前記述語が解明されたという表示を受け取り、
選択されたパスに関連する述語付き命令をリネームする
ように構成される、請求項 17 に記載のプロセッサ・システム。

【請求項 20】

前記リネーム・ステージは、更に、
前記述語が解明されたという表示を受け取り、
選択されなかったパスに関連する述語付き命令を廃棄する
ように構成され、

前記述語は状態レジスタ値に基づいている、請求項 17 に記載のプロセッサ・システム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、一般的には、述語型の実行 (predicated execution) に関し、特に、アウト・オブ・オーダー (out-of-order) ・プロセッサにおける述語型の実行のための技術に関するものである。

10

【背景技術】

【0002】

現在、ブランチ命令は、コンピュータ・プログラム (プログラム) においてどれだけの数のオペレーションを同時に実行し得るかに関する 1 つの尺度である命令レベル並列度 (ILP) の活用にとって大きな障害と考えられる。一般に、コンパイラおよびハードウェアは、十分な ILP を達成するために、頻繁且つ正確なブランチ予測を行うことを要求される。一般に、ブランチ誤予測は、命令ストリームに取り込まれる不用なサイクルのためにパフォーマンスの低下を生じさせる。スーパースカラ・プロセッサおよび超長命令語 (VLIW) プロセッサにおけるブランチ誤予測は、各不用なサイクルが複数の命令によりスルーポイントを減少させることがあるスカラ・プロセッサにおけるブランチ誤予測よりもはるかに大きくパフォーマンスを低下させる。

20

【0003】

ブール・ソース・オペランド (命令の述語 (predicate) として知られている) の値に基づいた命令の条件付き実行と呼ばれる述語型の実行は、命令ストリームからブランチを除去するための技術を提供する。典型的な実施方法では、述語型の実行を使用するコンパイラは、条件付きのブランチを述語定義 (predicate defining) 命令に変換するために、および代替パスに沿った命令ストリームを述語付き命令に変換するために if 変換アルゴリズムを使用する。典型的な場合では、述語付き命令は、それらの述語値に関係なくフェッチされる。真の述語を有する命令は正常に実行されるが、偽の述語値を有する命令はヌル化にされ、従ってそのヌル化された命令はプロセッサ状態を修正することを妨げられる。一般に、述語型の実行は、コンパイラが複数の実行パスに沿って、低い命令フェッチ効率ではあるが、ハードウェアに ILP を施すことを可能にする。

30

【0004】

アウト・オブ・オーダー実行は、データ依存性を損なわない任意の順で命令が実行されることを可能にする。アウト・オブ・オーダー実行は、パイプライン処理およびスーパースカラ技術と関連して使用されてもよく、或いは、使用されなくてもよい。アウト・オブ・オーダー実行は、多くの高性能プロセッサにおいてプロセッサ・サイクルを利用するために使用され、それ以外には利用されないであろう。アウト・オブ・オーダー実行の主要な注目点は、オペレーションを遂行するために必要なデータが得られないときに生じるストールをプロセッサが回避することを可能にすることである。アウト・オブ・オーダー・プロセッサでは、命令はデータ順に、即ち、データ・オペランドがプロセッサのレジスタにおいて利用可能になる順に処理される。アウト・オブ・オーダー・プロセッサは、データが関連の命令にとって利用可能でないとき、実行する準備ができて他の命令でもってスロットを満たす。その後、アウト・オブ・オーダー・プロセッサは、命令がアウト・オブ・オーダーで実行されたことをアーキテクチャ状態に対してコミットする前に、命令がプログラム順に、即ち、オリジナル・プログラムにおける命令の順に処理されたように見えるよう、結果を再順序付ける。

40

【0005】

命令を成功裡に再順序付けするために、現在のアウト・オブ・オーダー・プロセッサは、

50

リネーミングと呼ばれる技術を使用する。リネーミングは、命令によって生成された各結果レジスタに一意的な物理的位置を割当てることを含む。実施方法は、一般に、予約ステーション、レジスタ更新ユニット(RUU)、物理的レジスタ・ファイル等を使用する。アウト・オブ・オーダー・プロセッサは、生成する命令の結果オペランドの可用性を追跡し、結果オペランドを得るべき一意的な物理的ロケーションを(従属的な消費命令(consumer instruction)に)指定する。叙述化(predication)は、各々がアウト・オブ・オーダー・プロセッサにおける一意的な物理的ロケーションにリネームされる複数の命令を同じアーキテクチャ上のレジスタに導入することが可能である。その結果、アウト・オブ・オーダー・プロセッサは、ソース・オペランドの正確なロケーションを(待機する消費命令にとっては)一意的に識別することができないであろう。従って、そのような状況を扱う明確なサポートがない場合、すべての作動時(in-flight)の生成命令(producer instruction)が完了するまで、アウト・オブ・オーダー・プロセッサはストールしており、その結果、非効率的なパフォーマンスを生じる。述語予測を使用するプロセッサは多くの米国特許に開示されている(例えば、米国特許第7,085,919号、同第6,513,109号、および同第6,442,679号参照)。

10

【特許文献1】米国特許第7,085,919号、

【特許文献2】米国特許第6,513,109号、

【特許文献3】米国特許第6,442,679号

【発明の開示】

【発明が解決しようとする課題】

20

【0006】

本発明の目的は、アウト・オブ・オーダー・プロセッサにおいて予測実行するための方法および装置を提供することにある。

【課題を解決するための手段】

【0007】

本発明の種々の実施態様によれば、アウト・オブ・オーダー・プロセッサにおいて述語付きコードを扱うための技術は、述語付きコード領域と関連した述語定義命令を検出することを含む。更に、述語付き定義命令の述語が解明されるまで、述語付きコード領域内の述語付き命令のリネーミングはストールされる。

【発明を実施するための最良の形態】

30

【0008】

当業者には明らかなように、本発明は、方法、システム、装置、或いはコンピュータ・プログラムとして具体化することが可能である。従って、本発明は、全体的にハードウェアの実施例、全体的にソフトウェアの実施例(ファームウェア、常駐ソフトウェア、マイクロコード等を含む)、或いは、「回路」、「モジュール」、または「システム」として本明細書において一般的に参照されるソフトウェアおよびハードウェアを組み合わせた実施例の形態を取ることが可能である。本発明は、例えば、コンピュータ使用可能プログラム・コードを有するコンピュータ使用可能記憶媒体上でコンピュータ・プログラムの形、例えば、その媒体において具現化された1つまたは複数の設計ファイルの形を取ることが可能である。

40

【0009】

任意の適当なコンピュータ使用可能記憶媒体またはコンピュータ可読記憶媒体が利用可能である。コンピュータ使用可能または可読記憶媒体は、例えば、電子的、磁氣的、光学的、電磁氣的、赤外線、或いは、半導体システム、装置、またはデバイスであってもよいが、それに限定されない。コンピュータ可読媒体の更に特定な例(非網羅的なリスト)は、下記のもの、即ち、ポータブル・コンピュータ・ディスク、ハードディスク、ランダム・アクセス・メモリ(RAM)、読取り専用メモリ(ROM)、消去可能プログラマブルROM(EPROMまたはフラッシュ・メモリ)、ポータブル・コンパクト・ディスクROM(CD-ROM)、光学的記憶装置、または磁気記憶装置を含む。

【0010】

50

オペランド・リネーミング前に述語値を予測するために、種々の技術を使用することが可能である。この場合、通常は、予測され且つ選定されたパス (predicted-taken path) 命令が処理されるが、予測され且つ選定されなかったパス (predicted non-taken path) 命令はプロセッサ・バッファに保持される。述語値の実際の説明時に、述語予測が適正であるとわかった場合、余分なアクションは取られず、プロセッサ・バッファに保持された予測され且つ選定されなかったパスからの命令が廃棄される。予測が不適正であるとわかった場合、すべての予測され且つ選定された命令およびディスパッチされたその後のいずれの命令も廃棄される (即ち、予測され且つ選定された命令およびその後のいずれの命令もアーキテクチャ上の状態を更新することを許されない)。しかる後、ルーチン処理は、プロセッサ・バッファに保持された予測され且つ選定されなかった命令でもって再開する。少なくともいくつかの実施態様では、述語付きコード領域から命令を再フェッチする必要を回避するために、ロジックを使用することも可能である。

10

20

30

40

50

【 0 0 1 1 】

一般に、消費命令がそのオペランドを得なければならないロケーションを、即ち、レジスタ・ファイルまたはバイパス・ロジックから正確に指定することが、プロセッサの適正なオペレーションには必要である。更に、アウト・オブ・オーダー実行は、一般に、そのオペランドを使い尽くすことを待っているすべての命令をウェイク・アップするために、レジスタ・オペランドの生成を明瞭に識別する技術を必要とする。残念ながら、叙述化 (predication) は条件付きの書込み機能を導入し、それ自体で、アウト・オブ・オーダー・プロセッサにおけるウェイク・アップ・プロセスおよびバイパス・ロジックを混乱させる。本発明の種々の実施態様によれば、述語付きコード領域 (選定されたパスおよび選定されなかったパスに関連した述語付き命令を含む) の処理は、対応する述語が説明されるまでストールされる。述語の説明時に、レジスタ・リネーミングが適正なパス (即ち、選定されたパス) に関連した述語付き命令に関して遂行され、不適正なパス (即ち、選定されなかったパス) に関連した述語付き命令は廃棄される。

【 0 0 1 2 】

本明細書において開示された技術は、ブランチ・パイプラインの実行ステージおよびブランチ・パイプラインのリネーム・ステージの間にデータ・パスを加えることによって具現化される。一般に、レジスタ・リネーム・ステージは、述語付きコード領域を認識し、説明された述語をブランチ・パイプラインの実行ステージから受け取り、選定されなかったパスに関連した述語付き命令を廃棄するように修正される。本実施例で使用されるように、述語定義命令は、述語の値をセットする命令を含み、その値に基づいてその後の命令 (述語付きコード領域に含まれる) の実行が述語付きにされる。例えば、インテル IA-64 アーキテクチャのような明確な叙述化のサポートを備えた命令セット・アーキテクチャ (ISA) では、比較命令が述語定義命令である。

【 0 0 1 3 】

本発明の 1 つの実施態様によれば、アウト・オブ・オーダー・プロセッサにおいて述語付きコードを扱うための技術は、述語付きコード領域に関連した述語定義命令を検出することを含む。述語付きコード領域内の述語付き命令のリネーミングは、述語定義命令の述語が説明されるまでストールされる。

【 0 0 1 4 】

本発明のもう 1 つの実施態様によれば、アウト・オブ・オーダー・プロセッサはデコードおよびパイプライン・ステージを含む。デコードは、述語付き命令を含む述語付きコード領域に関連した述語定義命令をデコードするように構成される。パイプライン・ステージは、述語定義命令を検出するように、および、述語定義命令の述語が説明されるまで述語付き命令のリネーミングをストールするように構成される。パイプライン・ステージは、述語の説明時に、選定されなかったパスに関連する述語付き命令を廃棄するように、または、選定されなかったパスに関連する述語付き命令を、アーキテクチャ状態を変更しないノー・オペレーション (NOP) 命令に変換するようにも構成され得る。

【 0 0 1 5 】

本発明の1つの実施例によれば、プロセッサ・システムはメモリ・サブシステムおよびアウト・オブ・オーダー・プロセッサを含み、アウト・オブ・オーダー・プロセッサはメモリ・サブシステムに接続されたデコーダおよびパイプライン・ステージを含む。デコーダは、述語付き命令を含む述語付きコード領域に関連した述語定義命令をデコードするように構成される。パイプライン・ステージは、述語定義命令を検出するように、および述語定義命令の述語が解明されるまで述語付き命令のリネーミングをストールするように、構成される。パイプライン・ステージは、述語の解明時に、選定されなかったパスに関連する述語付き命令を廃棄するように、または、選定されなかったパスに関連する述語付き命令を、アーキテクチャ状態を変更しないノー・オペレーション（NOP）命令に変換するようにも構成され得る。

10

【0016】

本明細書に開示された技術は、間接的な叙述化サポート（ARM Thumbアーキテクチャのような）を備えたアーキテクチャと同様に明示的な叙述化サポートを備えたISAに等しく適用可能である。1つの例として、既存のISAは、Thumbアーキテクチャのif-then命令と同様の述語定義命令（例えばemask命令）によって拡大し得る（例えば、米国特許第7,178,011号参照）。叙述化を含まない例示的な擬似コードは下記のように記述される。

```
isequal cr7, 0, r10
```

```
branch cr7, ELSE
```

```
sub r1=r1, r2
```

```
load r4=[r1]
```

```
jump MERGE
```

```
ELSE: add r1=r1, r2
```

```
load r3=[r1]
```

```
add r5=r3, r4
```

```
MERGE: store r5, [r1]
```

20

【0017】

上記の擬似コードは、次のような述語付きコードと置換することが可能である。

```
isequal cr7, 0, r10
```

```
emaskcr7, 11001b, 5
```

```
add r1=r1, r2
```

```
load r3=[r1]
```

```
sub r1=r1, r2
```

```
load r4=[r1]
```

```
add r5=r3, r4
```

```
MERGE: store r5, [r1]
```

30

【0018】

上記述語付きコードでは、emask命令が叙述化セマンティックスを提供する責任を負う。

特に、emask命令は、1番目、2番目、および5番目の述語付き命令が第1パスに属し、

3番目および4番目の述語付き命令が第2パスに属することによって、5つの後続する命令（述語付きコード領域内に含まれる）の実行を制御しなければならない実行マスクを指定する。戻された述語値（状態レジスタ7（cr7））に従って、第1パスの述語付き命令または第2パスの述語付き命令のいずれかが実行され、逆に、第2パスの述語付き命令または第1パスの述語付き命令のいずれかが廃棄されるか、ノー・オペレーション（NOP）命令に変換される。

40

【0019】

図1を参照すると、2つのチップ・レベル・マルチプロセッサ（CMP）102を含む例示のプロセッサ・システム100が示され、各マルチプロセッサ102は2つのプロセッサ104を含む。少なくとも1つの実施例では、プロセッサ104の各々は第1レベル（L1）キャッシュ・メモリ（個別に示されていない）を含み、そのL1キャッシュは共有第2レベル（L2）キャッシュ・メモリ（キャッシュ）106に接続される。L2キャ

50

ッシュ 106 は、第 3 レベル (L3) キャッシュ 114 および ファブリック・コントローラ 108 に接続される。ファブリック・コントローラ 108 は、メモリ・サブシステム 112 に接続されたメモリ・コントローラ 110 に接続される。メモリ・サブシステム 112 は、アプリケーションに適した量の揮発性および不揮発性メモリを含む。ファブリック・コントローラ 108 は、種々の CMP 102 の間およびプロセッサ 104 とメモリ・サブシステム 112 との間のコミュニケーションを容易にし、この場合、インターフェースとして機能する。

【0020】

図 2 を参照すると、本発明に従って構成されるプロセッサ 104 の 1 つの例示的な命令パイプライン 200 が示される。プロセッサ 104 は、例えば、同時的マルチスレッディング (SMT) モードまたはシングル・スレッド (ST) モードで作動し得る。プロセッサ 104 が SMT モードで作動しているとき、プロセッサ 104 は、複数のスレッドのためのプログラム・カウンタを格納するために、複数の個別の命令フェッチ・アドレス・レジスタを使用し得る。この場合、命令フェッチ (IF) ステージ 202 における命令フェッチは複数のスレッドの間で交互になる。ST モードでは、一般に、プロセッサ 104 は 1 つのプログラム・カウンタしか使用せず、各プロセッサ・サイクル当たり単一のスレッドのための命令をフェッチする。1 つの実施例では、プロセッサ 104 は、(命令キャッシュ (IC) ステージ 204 において) 各プロセッサ・サイクル当たり 8 個までの命令を命令キャッシュからフェッチするように構成されてもよい。SMT モードでは、複数の (例えば 2 個の) スレッドが命令キャッシュおよび命令変換機構を共用し得る。

10

20

【0021】

1 つの所与のプロセッサ・サイクルでは、一般に、すべてのフェッチされた命令が同じスレッドに関連付けられる。1 つの実施例では、プロセッサ 104 は、ブランチ予測 (BP) ステージ 206 においてブランチに関するフェッチされた命令をスキャンする。ブランチが BP ステージ 206 において検出される場合、プロセッサ 104 は、例えば、2 つのスレッドにより共用される 3 つのブランチ履歴テーブル (BHT) を使用してブランチ方向を予測し得る。1 つの実施例では、BHT のうちの 2 つが、ブランチ方向を予測するために 2 モードの且つパス相関のブランチ予測機構を使用する。適正な方向を予測するためにどの予測機構がより適当であるかを予測するように第 3 の BHT が構成され得る。フェッチされた命令が複数のブランチを含む場合、BP ステージ 206 はすべてのブランチを同時に予測するように構成されてもよい。ブランチ方向の予測に加えて、BP ステージ 206 は現命令グループ・サイクルにおいて選択されたブランチのターゲットを予測するようにも構成されてもよい。プロセッサ 104 は、サブルーチン・リターン・ターゲットを予測するためのリターン・スタックを、例えば、各スレッドに対して 1 つのリターン・スタックを使用し得る。

30

【0022】

プロセッサ 104 は、他のブランチのターゲットを予測するために共用ターゲット・キャッシュを使用し得る。選択されたブランチが存在する場合、プロセッサ 104 は、ブランチのターゲット・アドレスをプログラム・カウンタにロードするように構成される。選択されたブランチが存在しない場合、プロセッサ 104 は、次の順次命令のアドレスをプログラム・カウンタにロードするように構成される。プロセッサ 104 は、第 1 デコード (D0) ステージ 208 において、予測されたパスにおけるフェッチされた命令を複数のスレッドに対する個別の命令フェッチ・キューに入れるように構成される。スレッド優先順位に基づいて、プロセッサ 104 は、命令フェッチ・キューの 1 つから命令を選び、デコード・ステージ D1 (210)、デコード・ステージ D2 (212)、およびデコード・ステージ D3 (214) において 1 つのグループを形成するように構成される。述語付き命令を識別するためのロジックが、例えば、デコード・ステージ D1 (210) およびデコード・ステージ D2 (212) において使用されてもよい。1 つの実施例では、1 つのグループ内のすべての命令が同じスレッドから生じ、並行してデコードされる。プロセッサ 104 は、命令のグループがディスパッチされるような資源が使用可能になるまで、命令

40

50

を（転送（Xfer）ステージ 2 1 6 における）キュー内に入れる。

【 0 0 2 3 】

1 つの実施例では、各ディスパッチされた命令グループは、グローバル完了テーブル（GCT）におけるエントリに対応し、そのグループ内の各命令は、適切な発行キュー内のエントリに対応する。各ロードおよびストア命令は、アウト・オブ・オーダー実行ハザードを検知するために、ロード再順序付けキューおよびストア再順序付けキュー内のエントリをそれぞれ占有する。ディスパッチに必要なすべての資源が命令グループにおいて使用可能になるとき、その命令グループはグループ・ディスパッチ（GD）ステージ 2 1 8 においてディスパッチされる。命令は、IF ステージ 2 0 2 と GD ステージ 2 1 8 との間のパイプライン・ステージをプログラム順に流れる。ディスパッチの後、各命令は、命令にお

10

【 0 0 2 4 】

本発明の種々の実施態様によれば、述語付き命令のためのレジスタ・リネーミングは、述語定義命令に関連した述語が解明されるまで（即ち、述語定義命令が実行され、述語が解明されるまで）、MP ステージ 2 2 8 においてストールされる。1 つの実施例では、MP ステージ 2 2 8 は、述語定義命令に関連したマスクを格納するマスク・シフト・レジスタ（個別的に示されていない）を使用する。述語が解明され、MP ステージ 2 2 8 に戻されるとき、どの述語付き命令を実行しなければならないか決めるために述語およびマスクに関して論理的機能（例えば、排他的 NOR）が遂行される。レジスタ・リネーミングは

20

【 0 0 2 5 】

1 つの実施例では、プロセッサ 1 0 4 は、1 2 0 個の物理的汎用レジスタ（GPR）および 1 2 0 の物理的浮動小数点レジスタ（FPR）を実装している。複数のスレッドが実行されるとき、プロセッサ 1 0 4 は、それら複数のスレッドがレジスタ・ファイルを動的に共用するように構成される。ST モードにおけるアウト・オブ・オーダー・プロセッサ処理を容易にするために、プロセッサ 1 0 4 は、すべての物理的レジスタを単一スレッドにとって利用可能にして、より高い命令レベル並列度を促進するように構成される。レジスタ

30

【 0 0 2 6 】

パイプライン 2 2 0 ~ 2 2 6 を通して命令を追跡するためのロジックを単純化するために、プロセッサ 1 0 4 は、命令をグループとして追跡するように構成されてもよい。この場合、ディスパッチされた命令の各グループは、ディスパッチ時に GCT におけるエントリを占有する。この場合、そのグループがコミットされるとき、エントリは GCT から割振り解除される。GCT におけるエントリはプログラム順であり且つ所与のスレッドからのものであるが、連続したエントリが異なるスレッドに属することも可能である。1 つの命令に対する入力オペランドがすべて利用可能であるとき、それは発行の資格を有するようになる。発行ロジックは、発行キューにおける適格の命令の中から 1 つを選択し、発行（ISS）ステージにおける実行のためにそれを発行する。命令発行に関しては、通常、

40

50

2つのスレッドからの命令の間に区別は存在しない。

【0027】

命令は、それが発行されると、レジスタ・ファイル（RF）ステージにおいて1つまたは複数の関連入力物理的レジスタを読み取り、実行（EX）ステージにおいて実行し、書き戻し（WB）ステージにおいて関連出力物理的レジスタに結果を書き戻す。1つの実施例では、各浮動小数点パイプライン226が6サイクル実行パイプ（F1乃至F6ステージ）を有する。各ロード・ストア・パイプライン222において、加算器が、計算アドレス（EA）ステージにおいて読取るまたは書込むべきアドレスを計算し、データ・キャッシュがデータ・キャッシュ（DC）ステージにおいてアクセスされる。ロード命令に関しては、一旦データが戻されると、フォーマッタが、フォーマット（Fmt）ステージにおいてキャッシュ・ラインから適正なバイトを選び、しかる後、WBステージにおいてデータをレジスタに書込む。グループにおけるすべての命令が（例外を生じることなく）実行されてしまい且つそのグループが所与のスレッドのうちの最も古いグループであるとき、そのグループは完了（CP）ステージ232においてアーキテクチャの状態にコミットされる。1つの実施例では、2つの命令グループがサイクル毎に、即ち、2つのスレッドの各々から1つをコミットし得る。1つの実施例では、プロセッサ104は、64KバイトのL1命令キャッシュおよび32Kバイトのデータ・キャッシュ（個別に示されてなく、それぞれ、2ウェイおよび4ウェイ関連性有する）を含む。或る実施例では、第1レベルのデータ変換テーブル（個別に示されていない）が完全に連想的なものであり、128個のエントリを含んでいる。

10

20

【0028】

図3を参照すると、本発明の種々の実施態様に従って、アウト・オブ・オーダー・プロセッサにおいて述語付きコードを処理する例示的プロセス300が示される。ブロック302において、プロセス300が開始され、その時点で、制御が判断ブロック304に移る。ブロック304では、述語定義命令（既知の操作コード（OPコード）に相当する）が、例えば、プロセッサ・パイプライン200のMPステージ228において検出されない場合、制御はブロック316に移る。ブロック304において、述語定義命令が検出される場合、制御はブロック306に移る。ブロック306では、述語定義命令に関連したマスクが、その後の使用のためにマスク・シフト・レジスタ（例えば、MPステージ228に含まれる）に格納される。マスクは、述語付き命令の各々が2つの代替パス（即ち、選択されたパスまたは選択されなかったパス）のどちらに属するかを表わす。しかる後、判断ブロック308において、その述語定義命令に関連した述語値が使用可能であるかどうかが決される。

30

【0029】

図2に示されるように、述語が、ランチ・パイプライン220のEXステージの出力からMPステージ228に返送される。ブロック308において述語定義命令の述語が利用可能でない場合、制御はブロック310に移り、そこでは、述語付き命令のリネーミングがストールされる。ブロック310に続いて、制御はブロック308に移る。ブロック308において、述語定義命令の述語が使用可能である場合、制御はブロック312に移る。上述のように、述語定義命令を実行した結果（即ち、状態レジスタの値）が、ランチ・パイプライン220のEXステージの出力からMP（リネーム）ステージ228に返送される。ブロック312において、後続の述語付き命令のうちのどれがリネームされ且つ実行されるべきか、およびその述語付き命令のうちのどれが廃棄されまたはNOP命令に変換されるべきかを決定するために、マスクおよび述語に関して論理的機能（例えば、排他的NOR機能）が遂行される。次に、ブロック314において、（その論理的機能に従って）リネームされると思われる述語付き命令がリネームされ、（その論理的機能に従って）リネームされると思われる述語付き命令が廃棄されるかまたはNOP命令に変換される。しかる後、リネームされた命令が実行されてもよい。ブロック314に続いて、制御はプロセス316に移り、そこでは次の述語定義命令が検出されるまでプロセスは停止する。

40

50

【0030】

本明細書において開示された技術は、プロセッサ・コアおよび関連の命令セット・アーキテクチャに対する最小の修正でもって具現化することが可能である。従って、アウト・オブ・オーダ・プロセッサにおける述語付きコードの効率的な処理を容易にする技術が開示された。

【0031】

添付図面におけるフローチャートおよびブロック図は、本発明の種々の実施例に従って、システム、方法、およびコンピュータ・プログラムの可能な実施態様のアーキテクチャ、機能、およびオペレーションを示す。これに関して、フローチャートまたはブロック図における各ブロックはモジュール、セグメント、またはコード部分を表わし得る。そのコード部分は、指定された論理機能を具現化するための1つまたは複数の実行可能命令を含む。或る代替方法では、ブロックにおいて示された機能は、図面に示された順序を外れて生じ得るということに留意すべきである。例えば、連続して示された2つのブロックは、実は、実質的に同時に実行され得るし、或いは、関連する機能次第で、逆順に実行されてもよいことがある。更に、ブロック図および/またはフローチャートの各ブロック、並びにブロック図および/またはフローチャートにおけるブロックの組合せが、指定された機能または動作を遂行する特殊目的のハードウェア・ベースのシステム、或いは特殊目的のハードウェアおよびコンピュータ命令の組合せによって具現化され得ることも留意すべきであろう。

10

【0032】

本明細書において使用される用語は、特定の実施例のみを説明することを目的とするものであり、発明を限定することを意図するものではない。

20

【0033】

「特許請求の範囲」の各請求項におけるすべての手段またはステップおよび機能要素の対応する構造、材料、動作、および均等物は、他の請求項における他の要素と組み合わせるその機能を遂行するための任意の構造、材料、または動作を含むように意図される。本発明の説明は、図解および記述を目的として示されたが、網羅的であることまたは開示された形式に限定されることを意図したものではない。本発明の範囲および趣旨から逸脱することなく、多くの修正および変更が当業者には明らかであろう。実施例は、本発明の原理および実用的な応用例を最適に説明するために、および当業者が、特定の意図された用途に適するように様々な修正を含む様々な実施例に関して本発明を理解することを可能にするために、選択および記述された。

30

【0034】

従って、本発明を詳細に且つ好適な実施例を参照して説明したが、「特許請求の範囲」において定義された発明の範囲を逸脱することなく、修正および変更が可能であることは明らかであろう。

【図面の簡単な説明】

【0035】

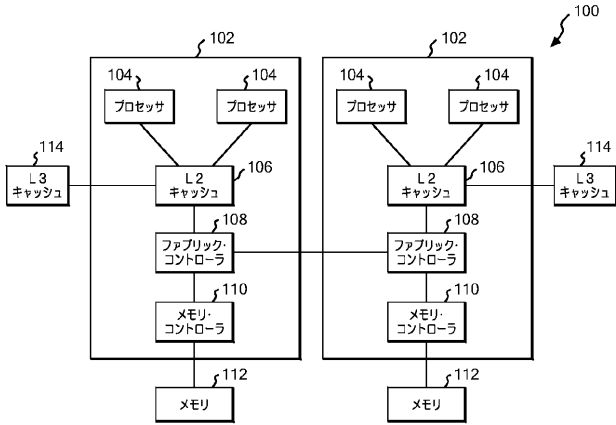
【図1】本発明の種々の実施態様に従って構成され得る例示的なプロセッサ・システムのブロック図である。

40

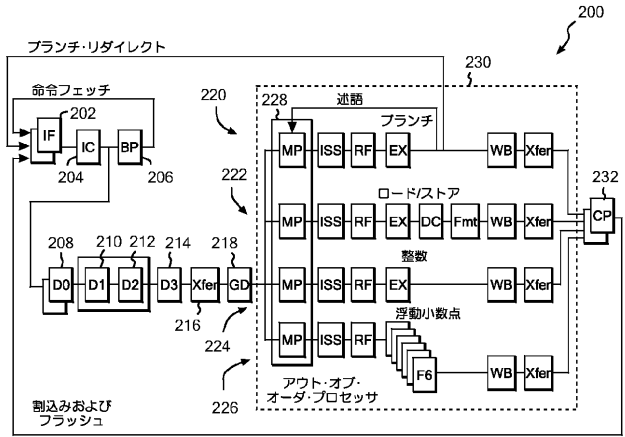
【図2】本発明の種々の実施例に従って、図1のプロセッサ・システムのプロセッサにおいて使用され得る例示的なプロセッサ命令パイプラインのブロック図である。

【図3】図2における例示的なプロセッサ命令パイプラインにおいて使用される例示的なプロセスのフローチャートである。

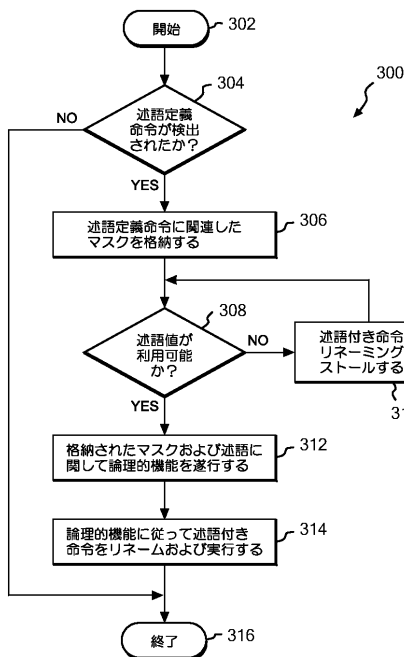
【 図 1 】



【 図 2 】



【 図 3 】



フロントページの続き

(74)代理人 100086243

弁理士 坂口 博

(72)発明者 ラム・ランガン

アメリカ合衆国 7 8 7 5 1、テキサス州オースチン、アベニュー ビー 4 3 1 0 ナンバー 2 0
1

(72)発明者 ウィリアム・エヴァン・スパイト

アメリカ合衆国 7 8 7 3 3、テキサス州オースチン、パディナ・ドライブ 2 4 0 7

(72)発明者 マーク・ダブリュー・スティーブソン

アメリカ合衆国 7 8 7 5 9、テキサス州オースチン、シエラ・オークス 1 0 9 2 3

(72)発明者 リキシン・ジャン

アメリカ合衆国 7 8 7 2 6、テキサス州オースチン、ダーク・スター 1 0 0 0 8

Fターム(参考) 5B013 CC08

5B033 AA13 BE00