

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第5485129号  
(P5485129)

(45) 発行日 平成26年5月7日 (2014.5.7)

(24) 登録日 平成26年2月28日 (2014.2.28)

(51) Int. Cl.

F I

G 0 6 F 9/38 (2006.01)

G 0 6 F 9/38 3 8 0 A

G 0 6 F 9/48 (2006.01)

G 0 6 F 9/38 3 3 0 A

G 0 6 F 9/46 3 1 0 Q

請求項の数 17 外国語出願 (全 16 頁)

(21) 出願番号 特願2010-283132 (P2010-283132)  
 (22) 出願日 平成22年12月20日 (2010.12.20)  
 (65) 公開番号 特開2011-129130 (P2011-129130A)  
 (43) 公開日 平成23年6月30日 (2011.6.30)  
 審査請求日 平成25年12月18日 (2013.12.18)  
 (31) 優先権主張番号 12/642, 970  
 (32) 優先日 平成21年12月21日 (2009.12.21)  
 (33) 優先権主張国 米国 (US)

早期審査対象出願

(73) 特許権者 509000482  
 チェバ ディー. エス. ピー. リミテッド  
 CEVA D. S. P. LTD.  
 イスラエル国 4 6 1 2 0 ヘルズリア  
 ピトゥアク マスキット ストリート 2  
 ビー. オー. ボックス 2 0 6 8  
 (74) 代理人 100068755  
 弁理士 恩田 博宣  
 (74) 代理人 100105957  
 弁理士 恩田 誠  
 (74) 代理人 100142907  
 弁理士 本田 淳  
 (72) 発明者 ジェフリー アラン ヤーコブ  
 イスラエル国 4 3 3 0 8 ラアナナ ラ  
 ムハル ストリート 1 0

最終頁に続く

(54) 【発明の名称】 コンピュータシステムにおいて割込みを処理するシステムおよび方法

(57) 【特許請求の範囲】

【請求項 1】

デジタル信号を処理する方法であって、

一連の命令に続いて割込みの処理を開始することであって、前記一連の命令は、一連の  
 パイプラインスロットで処理されるべきものであり、前記割込みは、前記一連のパイプ  
 ラインスロットに続くスロットにおいて処理されるべきものである、前記開始すること、

デコード段において、前記一連の命令における命令が分岐命令であるかどうかを検出す  
 ること、

前記一連の命令が分岐命令を含まない場合、割込みペナルティなしで前記一連の命令の  
 処理を完了すること、

分岐命令が検出され、かつ前記割込みが、検出された分岐命令に関連する遅延スロット  
 を占有する場合、前記検出された分岐命令、及び該分岐命令と、前記割込みに関連する割  
 込みアドレス命令との間の任意の命令を無効化し、前記検出された分岐命令に先行する前  
 記一連の命令における任意の命令の処理を完了すること

を備える方法。

【請求項 2】

前記遅延スロットは分岐命令を適切に処理するために必要である、請求項 1 に記載の方  
 法。

【請求項 3】

割込みサブルーチンに続くそれぞれのパイプラインスロットにおいて無効化された命令

を処理することを備える、請求項 1 に記載の方法。

【請求項 4】

前記無効化された命令をリターンレジスタに書き込むこと、前記無効化された命令を待ち行列に入れて、割込みサブルーチンに続くパイプラインスロットで順番に処理されるようにすることを備える、請求項 1 に記載の方法。

【請求項 5】

各無効化された命令の再処理に対して割込みペナルティを生じることを備える、請求項 1 に記載の方法。

【請求項 6】

少なくとも、前記命令の一つが分岐命令であると判断されるまで、連続する命令の処理を継続することを備える、請求項 1 に記載の方法。

10

【請求項 7】

前記割込みに先行し、かつ分岐命令ではなく、かつ分岐命令に関連する遅延スロットを占有しない各命令に対するパイプライン処理の全体を通じて中断することなく命令を処理することを備える、請求項 1 に記載の方法。

【請求項 8】

デジタル信号を処理するプロセッサであって、  
分岐検出器を備え、

前記プロセッサは、一連の命令に続いて割込みの処理を開始し、各命令はそれぞれの連続パイプラインスロットで処理され、前記分岐検出器は、命令が分岐命令であるかどうかを検出し、分岐命令が検出され、かつ前記割込みが、検出された分岐命令に関連する遅延スロットを占有する場合、前記プロセッサは、前記分岐命令、及び該分岐命令と、前記割込みに関連する割込みアドレス命令との間の前記一連の命令における任意の命令を無効化し、前記検出された分岐命令に先行する前記一連の命令における任意の命令の処理を完了し、分岐命令が検出されない場合、前記プロセッサは、割込みペナルティなしで前記一連の命令の処理を完了する、プロセッサ。

20

【請求項 9】

前記プロセッサは、割込みパイプラインスロットに続くそれぞれのパイプラインスロットにおいて無効化された命令を処理する、請求項 8 に記載のプロセッサ。

【請求項 10】

リターンレジスタを備え、無効化された命令が前記リターンレジスタに書き込まれ、待ち行列に入れられて、割込みサブルーチンに続くパイプラインスロットで前記プロセッサによって順番に処理されるようにする、請求項 8 に記載のプロセッサ。

30

【請求項 11】

割込みパイプラインスロットに続くパイプラインスロットにおいて各無効化された命令を再処理するために割込みペナルティが生じる、請求項 8 に記載のプロセッサ。

【請求項 12】

前記プロセッサは、少なくとも前記分岐検出器が分岐命令を検出するまで、連続する命令の処理を継続する、請求項 8 に記載のプロセッサ。

【請求項 13】

前記プロセッサが前記割込みに先行する命令は分岐命令ではなくかつ分岐命令に関連する遅延スロットを占有しないと判断する場合に、前記プロセッサは、パイプライン処理全体を通じて命令を中断することなく処理する、請求項 8 に記載のプロセッサ。

40

【請求項 14】

デジタル信号を処理するシステムであって、

一連の命令に続いて割込みの処理を開始するプロセッサであって、各命令はそれぞれの連続パイプラインスロットで処理される、前記プロセッサと、

命令が分岐命令であるかどうかを検出する分岐検出器であって、分岐命令が検出され、かつ前記割込みが、検出された分岐命令に関連する遅延スロットを占有する場合、前記プロセッサは、前記分岐命令、及び該分岐命令と、前記割込みに関連する割込みアドレス命

50

令との間の前記一連の命令における任意の命令を無効化し、前記検出された分岐命令に先行する前記一連の命令における任意の命令の処理を完了し、分岐命令が検出されない場合、前記プロセッサは、割込みペナルティなしで前記一連の命令の処理を完了する、前記分岐検出器と、

少なくとも前記一連の命令を実行してプログラムを実行する実行ユニットとを備えるシステム。

【請求項 15】

前記プロセッサは、割込みサブルーチンに続くそれぞれのパイプラインスロットにおいて無効化された命令を処理する、請求項 14 に記載のシステム。

【請求項 16】

前記プロセッサは、少なくとも前記分岐検出器が分岐命令を検出するまで、連続する命令の処理を継続する、請求項 14 に記載のシステム。

【請求項 17】

前記プロセッサが前記割込みに先行する命令は分岐命令ではなくかつ分岐命令に関連する遅延スロットを占有しないと判断する場合に、前記プロセッサは、パイプライン処理全体を通じて命令を中断することなく処理する、請求項 14 に記載のシステム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明の実施形態は、コンピュータシステムにおいて割込みを処理する方法およびシステムに関する。具体的には、本発明の実施形態は、割込みの処理に関連する計算ペナルティを軽減することに関する。

【背景技術】

【0002】

割込みは、プロセッサが命令を実行する順序を変更する事象である。割込みユニットは、一つ以上の優先度の高い命令を実行するために一連の命令を実行する現行の処理を中断してもよい。割込みサブルーチンが処理される時点で、プロセッサは割込みサブルーチンから戻って初期の一連の命令処理を再開してもよい。

【0003】

一連の命令は分岐命令を含んでいてもよい。分岐命令は、たとえば、分岐条件に基づいて複数の可能な方法の一つで処理を進行させてもよい。たとえば、一連の命令 1、2、3、4、5 において、5 は分岐命令であり、命令 5 は、処理を命令 6、7、8、... に従って順番に進行させてもよく、あるいは命令 100、101、102、... にジャンプさせてもよい。分岐命令、たとえば、6 または 100 に続く命令は、分岐命令が実行されるまで不明であってもよい。

【0004】

条件付き分岐の結果を予測するために分岐予測器が使用されてもよい。分岐において予測される命令は、プログラムメモリから前もって読み出されて容易にアクセスできるようプログラムキャッシュに一時的に記憶されてもよい。分岐予測は正しくても正しくなくてもよい。分岐予測器は、分岐結果を解析して、分岐予測が一つ以上の後続の計算サイクルにわたって正しいかそれとも正しくないかを判断してもよい。分岐を解析するために使用される計算サイクルの間に割込みが発生すると、分岐結果は不明であり再計算されなければならない。

【0005】

分岐命令は、一つ以上の後続の遅延スロットに関連していてもよい。割込みが分岐命令に続いて受け入れられるとき、割込みは遅延スロットの一つで処理されてもよい。予測が正しいか否かを判断するための分岐条件は、遅延スロットが処理される計算サイクルと同じ計算サイクル間に計算されてもよい。したがって、割込みが分岐命令の遅延スロットの間に発生して遅延スロットを交換することによって処理されるとき、分岐条件計算の結果は不明であるかもしれない。したがって、分岐命令および関連する遅延スロット内で処理

10

20

30

40

50

される命令は、適切に実行することができず、割込みが対処されて初期の命令が再開した後で再処理されなければならない。

【 0 0 0 6 】

典型的に、命令がデコードされるまで、たとえば、複数の計算サイクルにわたって命令を処理した後まで、命令が分岐命令であるかどうかは不明である。現行のシステムにおいて、まだデコードされていない命令は、デコーディングに先立って、万が一に備え、しかし、命令が分岐命令であるかどうか分かる前に、無差別に破棄される。破棄されたデータを再処理するために使用されるさらなる計算サイクルは、割込みペナルティと呼ばれることがある。結果的に、割込みを処理するときは割込みペナルティが必ず生じる。

【図面の簡単な説明】

10

【 0 0 0 7 】

【図 1 A】本発明の実施形態に従ったシステムの略図である。

【図 1 B】本発明の実施形態に従った図 1 A のシステムの一部の構成要素の拡大図である。

【図 2】従来のシステムにおいて割込みに対処するプロセッサ動作を示す表である。

【図 3】本発明の実施形態に従って割込みに対処するプロセッサ動作を示す表である。

【図 4】本発明の実施形態に従って割込みに対処するプロセッサ動作を示す表である。

【図 5】本発明の実施形態に従って割込みに対処するプロセッサ動作を示す表である。

【図 6】本発明の実施形態に従った方法のフローチャートである。

【発明を実施するための形態】

20

【 0 0 0 8 】

本発明として扱われる対象は、本明細書の結びの部分で特に指摘され、明確に主張される。しかし、本発明は、本発明の目的、特徴、および長所とともに、構成および操作方法に関して、以下の詳細な説明を参照し、添付図面を用いて読むと最もよく理解されるであろう。本発明の具体的な実施形態は、図面を参照して説明される。

【 0 0 0 9 】

説明を簡単かつ明瞭にするために、図に示される要素は必ずしも一定の縮尺で描かれていないことは理解されよう。たとえば、要素の一部の寸法は、明確にするために他の要素に比べて誇張されている場合がある。さらに、適当と考えられる場合、対応する要素または類似の要素を示すために図の中で参照番号が繰り返し使用されている場合がある。

30

【 0 0 1 0 】

以下の説明において、本発明の様々な態様が記載される。説明を目的として、本発明の完全な理解を与えるために特定の構成および詳細が記述される。しかし、本発明は本明細書で提示される具体的な詳細なしで実施されてもよいことも当業者にとって明らかであろう。さらに、本発明が不明瞭にならないように周知の特徴が省略または単純化されている場合がある。

【 0 0 1 1 】

特に明記しない限り、以下の議論から明らかなように、本明細書を通じて、「処理する」、「算定する」、「計算する」、「判断する」などの用語を利用する議論では、コンピュータシステムのレジスタおよび/またはメモリの内部の電子的量など、物理量として表わされるデータを操作しかつ/またはコンピュータシステムのメモリ、レジスタなどの他の情報記憶装置、伝送装置、または表示装置の内部で物理量として同様に表わされる他のデータに変換するコンピュータまたはコンピュータシステム、あるいは同様の電子コンピュータデバイスの作用および/または処理に言及していると理解される。

40

【 0 0 1 2 】

割込みとは、実行の変更が必要であることを示す事象を指していると言えよう。割込みは、通常であれば、処理において別の命令に割り当てられるはずであるプロセッサ・パイプライン・スロットを占有してもよい。割込みは、たとえば、複数の計算サイクルまたはパイプライン段にわたってそれぞれの連続パイプラインスロットで処理される一つ以上の命令を含む、割込みサブルーチンで処理されてもよい。割込みサブルーチンを処理した後

50

、プロセッサは、割込み時に一部が処理された処理命令あるいは待ち行列に入れられた処理命令を再開してもよい。命令が破棄、無効化、または「中止」されるとき、割込みサブルーチンから戻った後、命令はフェッチされ再び処理されて割込みペナルティを生じてもよい。

#### 【0013】

分岐命令は、処理を複数の可能な方法の一つで進行させる命令を指していると言えよう。最もふさわしい後続の命令を予測し前もって読み出すために、分岐予測器が採用されてもよい。予測が正しいときは予測された命令が処理されてもよく、または分岐予測が正しくない場合は、プロセッサが機能を停止して予測されていない正しい命令がプログラムメモリから読み出されてもよい。分岐命令は、複数の（たとえば、三つの）遅延スロットが後に続くプロセッサ・パイプライン・スロットで処理されてもよく、複数の遅延スロットはプロセッサパイプラインで読み出されて利用できる後続の命令を処理するために使用される。

#### 【0014】

割込みサブルーチン内の第1の命令が分岐の遅延スロットの一つで実行されるとき、プロセッサは割込みを処理した後に実行するための正しい一連の命令を保持していないかもしれない。これは、分岐予測と、分岐予測が正しいか正しくないかと、によって複雑化する。遅延スロットは分岐を適切に処理するために必要な場合があるので、割込みが遅延スロットの一つの間に発生するとき、分岐遅延スロット命令および分岐予測は割込みサブルーチンに続くパイプラインスロットで再処理されてもよい。

#### 【0015】

対照的に、非分岐命令は、単一のパイプラインスロットで、たとえば、後続の遅延スロットなしで独立に処理され解析されてもよい。それゆえ、割込みが非分岐命令に続いて実行されるとき、割込みは典型的に命令処理に影響を及ぼさない。

#### 【0016】

しかし、どの命令が分岐命令でどの命令が分岐命令でないかが直ちにはっきりしないかもしれない。命令が分岐命令であるかどうかの判断には、処理について複数の計算サイクルを必要とするかもしれない。一例において、命令は、これが分岐命令であるかどうかを判断するためにデコードされてもよい。

#### 【0017】

従来のシステムにおいて、予防保証のために、命令は、分岐命令であるかどうか、たとえば、まだデコードされていない部分的に処理された命令であるかどうかはまだ判断されていないとき、無差別に破棄される。したがって、従来のシステムにおいては、命令が実際にも分岐命令を含むか否かに関係なく、割込みに先立つ部分的に処理された命令を再処理および再読出しする割込みペナルティがある。

#### 【0018】

本発明の実施形態によると、割込みが一連の命令に続いて実行されるとき、命令は停止も、割込みも、破棄もされず、分岐命令が実際に検出されるまで処理されてもよい。割込みに先立つ命令は、少なくとも、命令が分岐命令を、たとえば、デコード段に含むか否かが判断されるまで処理されてもよい。命令が従来のシステムにおけるように分岐命令を含む場合、分岐命令スロットおよび分岐遅延スロットがあれば、これらは破棄されて、破棄された命令に対する割込みペナルティを生じてもよい。しかし、従来のシステムとは対照的に、分岐命令を含まない命令はいずれも引き続き処理されてよい。したがって、非分岐命令は、これらが割込み事象に受け継がれるとき計算ペナルティを生じないかもしれない。

#### 【0019】

本発明の実施形態によるシステムの標準図および拡大図の略図である図1Aおよび1Bを参照する。システムは、プロセッサ1、データ・メモリ・ユニット2、プログラム・メモリ・ユニット3、およびプログラムキャッシュ10を有するデバイス100を含んでいてもよい。

## 【 0 0 2 0 】

デバイス 1 0 0 は、コンピュータデバイス、セルラデバイス、または携帯電話、パーソナル・デジタル・アシスタント ( P D A )、ビデオ・ゲーム・コンソールなどの他の任意のデジタルデバイスを含んでいてもよい。デバイス 1 0 0 は、コンピュータプログラムを実行するための一連の命令を実行できる任意のデバイスを含んでいてもよい。デバイス 1 0 0 は、ユーザから入力を受信する (たとえば、ポインティングデバイス、クイックホイールまたはマウス、キー、タッチスクリーン、音声認識を用いるレコーダ/マイクロフォン、他の入力要素を介して) 入力デバイス 1 0 1 を含んでいてもよい。デバイス 1 0 0 は、プロセッサ 1 によって実行される一連の命令に従ってユーザーインタフェース上にプログラムを表示する出力デバイス 1 0 2 (たとえば、モニタまたはスクリーン) を含んでいてもよい。

10

## 【 0 0 2 1 】

プロセッサ 1 は、中央処理装置 ( C P U )、デジタル信号プロセッサ ( D S P )、マイクロプロセッサ、コントローラ、チップ、マイクロチップ、フィールド・プログラマブル・ゲート・アレイ ( F P G A )、特定用途向け集積回路 ( A S I C )、またはその他任意の集積回路 ( I C )、あるいはその他任意の適当な多目的プロセッサまたは特定プロセッサまたはコントローラを含んでいてもよい。

## 【 0 0 2 2 】

プロセッサ 1 は、データバス 4 を介してデータ・メモリ・ユニット 2 に、またプログラム・メモリ・バス 5 を介してプログラム・メモリ・ユニット 3 に結合されてもよい。

20

プログラム・メモリ・ユニット 3 は、典型的に、コンピュータプログラムを実行する一組の命令および規則を記憶し、その一方で、データ・メモリ・ユニット 2 は、典型的に、事前に生成された (すなわち、静的な) データおよび / またはスクラッチパッド (すなわち、動的な) データなど、プログラム命令を操作する間に生成されるデータを記憶するが、命令およびデータをメモリに記憶する他の配置が採用されてもよい。

## 【 0 0 2 3 】

命令がプログラム制御ユニット 8 によってより利用しやすいように、プログラムキャッシュ 1 0 がプログラム・メモリ・ユニット 3 から読み出されるプログラム命令の一時記憶装置を提供してもよい。プログラム・メモリ・ユニット 3 は長期メモリユニットであってもよいが、プログラムキャッシュ 1 0 は短期メモリユニットであってもよい。データ・メモリ・ユニット 2、プログラム・メモリ・ユニット 3、およびプログラムキャッシュ 1 0 は、たとえば、スタティック・ランダム・アクセス・メモリ ( S R A M )、ダイナミック R A M ( D R A M )、フラッシュメモリなどの適当なメモリユニットまたは記憶ユニットを含んでいてもよく、揮発性メモリまたは不揮発性メモリであってもよい。データ・メモリ・ユニット 2、プログラム・メモリ・ユニット 3、およびプログラムキャッシュ 1 0 は、独立したメモリユニットまたは集積化されたメモリユニットとして実装されてもよい。データ・メモリ・ユニット 2、プログラムキャッシュ 1 0、および / またはプログラム・メモリ・ユニット 3 は、集積化されてもよく (「オンチップ」)、独立していてもよい (すなわち、「オフチップ」)。階層記憶、すなわち、記憶の階層がある一実施形態において、プログラム・メモリ・ユニット 3 はオフチップであってもよく、プログラムキャッシュ 1 0 および / またはデータ・メモリ・ユニット 2 はオンチップ、たとえば、L - 1 キャッシュ (たとえば、プロセッサ 1 との一体化の程度が L - 2 キャッシュよりも比較的高く、典型的に、プロセッサ 1 のクロックレートで動作する) または L - 2 キャッシュ (たとえば、プロセッサ 1 との一体化の程度が L - 1 キャッシュよりも比較的低く、典型的に、プロセッサ 1 のクロックレートと異なるクロックレートで動作する) であってもよい。他のまたは追加のメモリアーキテクチャが採用されてもよい。

30

40

## 【 0 0 2 4 】

プロセッサ 1 は、プログラム制御ユニット 8 を含んでいてもよい。プログラム制御ユニット 8 は、命令をプログラム・メモリ・ユニット 3 から要求し、読み出して、ディスパッチしてもよく、一般に、プログラム・パイプライン・フローに関与してもよい。プログラ

50

ム制御ユニット 8 は、データ・メモリ・バス 4 に結合されたデータ・メモリ・コントローラ（図示せず）と、それぞれデータ・メモリ・ユニット 2 およびプログラム・メモリ・ユニット 3 からデータを読み出すためにプログラム・メモリ・バス 5 に結合されたプログラム・メモリ・コントローラ（図示せず）とを含んでいてもよい。

【 0 0 2 5 】

プログラム制御ユニット 8 は、フェッチユニット 1 2、割込みユニット 1 3、および分岐機構 1 4 を含んでいてもよい。フェッチユニット 1 2 は、命令をディスパッチする前に、命令をローカル待ち行列にフェッチするために使用されてもよい。分岐機構 1 4 は、分岐命令を検出してもよく、分岐命令の結果を予測してもよく、分岐命令の予測結果に基づいて、たとえば、分岐予測ミスがあるときに、フェッチユニット 1 2 を介してプログラム・メモリ・ユニット 3 から後続命令を要求してもよい。割込みユニット 1 3 は、割込み事象を読み出して対処してもよい。割込み信号は、関連優先順位を有していてもよい。場合によっては、たとえば、プロセッサ 1 が、高い優先事項、バンドル、または一連の命令、たとえば、命令ループを実行している間に、割込みユニット 1 3 は、入力割込みをマスク、先送り、またはディセーブルしてもよい。イネーブルされると、割込みユニット 1 3 は、たとえば、別のより高い優先順位の処理を開始するために現行の処理を中断してもよい。

【 0 0 2 6 】

プロセッサ 1 は、デコードユニット 6、ロード/記憶ユニット 7、一つ以上のレジスタファイル 9、および実行ユニット 1 1 を含んでいてもよい。命令がプログラム制御ユニット 8 によってディスパッチされた時点で、デコードユニット 6 は命令をデコードしてもよい。プロセッサ 1 は、タグを実装して、たとえば、要求通りの同じ計算サイクルでデコードされた命令に効果的にアクセスするためにレジスタファイル 9 を使用してもよい。実行ユニット 1 1 は命令を実行してもよい。ロード/記憶ユニット 7 は、データ・メモリ・ユニット 2 から / に対してロード操作および記憶操作を実施してもよい。

【 0 0 2 7 】

プロセッサ 1 は、たとえば、各命令に対して以下の連続パイプライン段を実行してもよい。

I F 1 - プログラム・メモリ・アドレス（プログラム制御ユニット 8 によって操作される）

I F 2 - プログラム・メモリ・フェッチ（フェッチユニット 1 2 によって操作される）

D 1 - 命令ディスパッチ（プログラム制御ユニット 8 によって操作される）

D 2 - 命令デコード（デコードユニット 6 によって操作される）

D 3 から D 4 - レジスタファイル読出し（レジスタファイル 9 を用いる）

E 1 から E 4 - 実行、データ・メモリ・アクセスおよびレジスタ・ファイル・ライトバック（実行ユニット 1 1、ロード/記憶ユニット 7、およびレジスタファイル 9 によって操作される）

しかし、本発明の実施形態は特定の順序に限定されるものではなく、他のあるいは追加のパイプライン段および操作デバイス構成部品が採用されてもよいことを当業者は理解されたい。

【 0 0 2 8 】

連続命令を備える処理において、将来処理される命令は事前に分かっており、プロセッサが命令をディスパッチする状態になる前に各命令がフェッチされるように、フェッチユニット 1 2 は命令を前もって読み出してもよい。フェッチされた命令は、プログラムメモリ 3 よりもアクセスが著しく速いプログラム・キャッシュ・メモリ 1 0 および / またはローカル待ち行列に一時的に記憶される。

【 0 0 2 9 】

しかし、条件付き分岐命令の後に続く命令は、命令がフェッチされる予定の時点でまだ分かっていない分岐条件に依存してもよい。たとえば、分岐命令は、分岐条件の結果に応

10

20

30

40

50

じて複数の異なる命令または処理経路のいずれかに進んでよい。

【 0 0 3 0 】

分岐機構 1 4 は、たとえば、命令を処理するデコード段 ( D 2 ) において命令が分岐命令であるか否かを識別してもよい。

命令が分岐命令であるとき、命令が読み出される予定の時点で見込まれる複数の後続命令の正しい一つが分かっていない場合、分岐機構 1 4 は分岐命令の後に続く可能性が最も高い命令を予測してもよい。予測された命令はプログラムメモリ 3 から読み出され、予測が正しければ、予測された命令は分岐遅延スロットを実行した後で処理される。

【 0 0 3 1 】

プロセッサ 1 は、分岐条件を判断してもよく、したがって、予測された命令が実際の分岐結果に一致していれば、たとえば、実行段 ( E 1 ) で分岐条件を判断してもよい。この判断には複数のサイクル (たとえば、3 サイクル) を要するかもしれない。

【 0 0 3 2 】

予測された分岐命令が実際の分岐結果に一致していれば、予測された命令は正しい命令であり採用されてもよい。

しかし、予測された分岐命令が実際の分岐結果に一致しなければ、実際の分岐命令はプログラム命令 3 からフェッチされてもよく、プログラムメモリ 3 から正しい命令をフェッチするために、たとえば、3 サイクル ( I F 1 ~ D 1 ) のさらなる計算ペナルティが生じるかもしれない。このペナルティは、プロセッサ 1 のパイプラインの種類によって異なるかもしれない。

【 0 0 3 3 】

割込みが分岐命令スロットに続いて遅延スロットまたは計算サイクルのいずれかの間に処理されるとき、プロセッサ 1 は割込みスロットの後に分岐を適切に処理するための正しい一連の命令を維持しないかもしれない。したがって、関連する遅延スロットにおいて処理される分岐命令および任意の命令は破棄されてもよい。第 1 の破棄または無効化された命令のプログラムアドレスは、リターンレジスタ (たとえば、レジスタファイル 9 内の) に書き込まれてもよく、割込みサブルーチンから戻った後にパイプラインスロットで再処理されてもよい。

【 0 0 3 4 】

しかし、命令が分岐命令でないとき、一連の後続命令は割込みの時点で既に分かっている。したがって、割込みが後に続く非分岐命令は、停止も、割込みも、破棄もされず、プロセッサ 1 のパイプラインによって最後まで処理されてもよい。

【 0 0 3 5 】

本発明の実施形態によると、割込みが既にプロセッサパイプライン内にある命令に続くとき、命令は、たとえば、D 2 段において、少なくとも分岐命令であるか否かが判断されるまで処理される。後続の遅延スロットにおいて割込みを有する分岐命令は分岐結果を適切に解析しないかもしれないので、プロセッサ 1 が一つ以上の命令が分岐命令でありかつ割込みが一つ以上の関連遅延スロットを占有すると判断するとき、分岐命令、分岐遅延スロット、およびこれらに関連する予測された命令が破棄または無効化される。割込みが対処された後に破棄されたデータを再処理するために、割込みペナルティが生じる。しかし、命令が単一パイプラインスロットにおいてのみ処理される非分岐命令であるとプロセッサ 1 が判断するとき、非分岐命令は最後まで中断することなく進行することが許可される。この場合、データが破棄されることはなく割込みペナルティが生じることはない。

【 0 0 3 6 】

本発明の実施形態は、従来のシステムより優れた長所を提供し、命令が分岐命令であるか否か分からない命令に割込みが続くとき (たとえば、命令が D 2 パイプライン段においてデコードされる前に)、命令は簡単に破棄され、命令の再処理に対して割込みペナルティが必ず生じる。

【 0 0 3 7 】

割込みを対処するプロセッサ動作を示す表である図 2 から 5 を参照する。図 2 から 5 に

10

20

30

40

50



において、表内の各行は独立したパイプラインスロット内の単一の命令を表わす。(列1に記載された)命令は、処理される順に、すなわち、(パイプライン段IF1において)命令が最初にプロセッサ・パイプライン・スロットに入る順に連続行または連続スロットで順序付けられる。各連続列は、各連続計算サイクルにおいて発生する命令で実行されるパイプライン段またはパイプライン動作を示す。すなわち、各行の命令が各連続列におけるプロセッサ・パイプライン・スロットに最初に入る時点で、プロセッサは、命令、たとえば、プログラム・メモリ・アドレス(IF1)、フェッチング(IF2)、ディスパッチング(D1)、デコーディング(D2)、レジスタファイル読出し(D3からD4)、実行、データ・メモリ・アクセス、およびレジスタ・ファイル・ライトバック(E1からE4)で連続した動作段を実行してもよい。しかし、割込みがパイプラインスロットで処理されるとき、命令が最後(たとえば、パイプライン段E4)まで処理させる前に、プロセッサは先行するパイプラインスロットの命令を無効化または「中止」してもよい。他のあるいは追加の動作が採用されてもよい。

10

#### 【0038】

図2は、従来のシステムにおいて割込みを対処するプロセッサ動作を示す。

表内の左上のセル(行1、列1)は、「A」と標記された命令を示す。命令Aは、プログラムの一連の命令、たとえば、A、A+1、A+2、A+3、... (行1および3~5、列1)における命令であってもよい。命令Aは、第1のアドレスで記憶され、たとえば、アドレス100(行1、列2)として任意に表わされてもよく、後続の命令A+1、A+2、A+3、... は異なる連続アドレスで記憶され、たとえば、101、102、103、... (行3~5、列2)として任意に表わされてもよい。

20

#### 【0039】

プログラム処理中に、命令Aは、プロセッサパイプライン、たとえば、パイプライン段IF1(行1、列3)に入ってもよい。パイプライン段IF1において、プロセッサ(たとえば、図1Aおよび1Bのプロセッサ1)は、プログラム・メモリ・アドレスにアクセスして、プログラムメモリ(たとえば、図1Aおよび1Bのプログラムメモリ3)におけるアドレス100に記憶された命令Aを検索してもよい。命令Aは、次の連続段IF2(行1、列4)、D1(行1、列5)、さらにD2(行1、列6)に進んでもよい。

#### 【0040】

パイプライン段D2(行1、列6)において、プロセッサは命令Aが分岐命令であるかどうかを判断するために命令Aをデコードしてもよい。この事例において、命令Aは分岐命令ではなく、命令は完全に処理されるまでプロセッサパイプラインの中を進行してもよい。

30

#### 【0041】

次の計算サイクル(列7の動作)において、割込みアドレス200(行6、列1)における割込み事象、すなわち、「割込み要求」(IRQ)(行2、列1)は、プロセッサによって検出されてもよい。割込みは、プロセッサによって実行されている現行のタスクよりも高い優先順位を有していてもよく、したがって、割込みはマスクされない。割込みの対処は、たとえば、パイプライン段IF1(行6、列7)における割込みアドレスのフェッチングを必要とする。プロセッサは、割込みを検索するためにパイプライン段IF1におけるプログラム・メモリ・アドレス200(行6、列2)にアクセスしてもよい。

40

#### 【0042】

割込みが最初に処理されるとき(たとえば、段IF1において：行6、列7)、命令A+1、A+2、およびA+3は、それぞれ、ディスパッチされ(D1：行3、列6)、フェッチされ(IF2：行4、列6)、アドレス指定される(IF1：行5、列6)が、命令A+1、A+2、およびA+3のいずれもデコードされていない(D2)。プロセッサは命令がデコード段(D2)において分岐命令であるかどうかを判断するので、割込みが最初に処理される時点では命令A+1、A+2、およびA+3が分岐命令であるか否かが分からない。命令A+1、A+2、およびA+3が分岐命令であれば、割込みは分岐命令を適切に処理するために必要な遅延スロットを占有することになる。従来の方法によると

50

、命令 A + 1、A + 2、および A + 3 は、分岐命令であるかどうか判断される前に割り込まれ、簡単に無効化され、破棄され、あるいは「中止」される（行 3 ~ 5、列 7）。

【 0 0 4 3 】

リターンレジスタ（たとえば、図 1 B のレジスタファイル 9 における）は、一連の無効化された命令の最初の命令のアドレス、たとえば、A + 1 をロードしてもよい。第 1 の無効化された命令は、命令 A + 1 として再ロード・レジスタ・アドレス 1 0 1 に書き込まれてもよい。割り込みサブルーチンから戻った後、または割り込みサブルーチンを完了した後、プロセッサは、プロセッサパイプラインにおける次の有効なスロットで無効化された命令の処理および実行を再開してもよい。

【 0 0 4 4 】

処理は、無効化された命令で既に行われた動作を繰り返す計算ペナルティを受ける。たとえば、割り込み（列 4、5、および 6 における）に先立って命令 A + 1、A + 2、A + 3 で既に行われた動作は、割り込みサブルーチンから戻った後に命令で再び実行される。この場合の割り込みペナルティは三計算サイクルである。

【 0 0 4 5 】

図 3 ~ 5 は、本発明の実施形態に従って割り込みに対処するプロセッサ動作を示す。図 3 ~ 5 の動作は、たとえば、図 1 A および 1 B のデバイス 1 0 0 によって実行されてもよい。

【 0 0 4 6 】

図 3 ~ 5 は、割り込みが最初に処理されるとき（行 6、列 7）命令 A + 1、A + 2、および A + 3 がまだデコードされておらず、したがって、命令が分岐命令であるかどうか分かっていないような、図 2 に関連して記載される同じ割り込みを示す。しかし、図 2 とは対照的に、図 3 ~ 5 の本発明の実施形態によると、割り込みが最初に処理されるとき、命令 A + 1、A + 2、および A + 3 は無効化されないが、少なくともこれらの一つが分岐命令であると判断されるまで処理される。命令 A + 1、A + 2、および A + 3 の一つが分岐命令であると判断され、割り込みが分岐を解析するために必要な遅延スロットの少なくとも一つを占有するとき、分岐命令自体、および分岐命令スロットと割り込みスロットの間のパイプラインスロットにおける任意の命令は無効化または「中止」されてもよい。図 5 におけるように分岐命令が検出されなければ、命令は無効化なしで処理され、割り込みペナルティはない。

【 0 0 4 7 】

図 3 において、プロセッサ（たとえば、図 1 A および 1 B のプロセッサ 1）は、命令 A + 1 をデコードし（行 3、列 7）、この命令が分岐命令でないと判断する。したがって、命令 A + 1 は無効化されずに最後（たとえば、段 E 4 において：行 3、列 1 1 - 図示せず）まで中断することなく処理される。プロセッサは、命令 A + 2 をデコードし（行 4、列 8）、命令が分岐命令であると判断する。分岐命令 A + 2 は、典型的に、適切に処理される複数の（たとえば、三つの）後続の遅延スロットを必要とする。割り込みは、分岐命令 A + 2（たとえば、三つの遅延スロットの 2 番目）を適切に処理するために必要な遅延スロットの少なくとも一つを占有する。したがって、分岐命令 A + 2 は適切に処理されることができず、たとえば、分岐を解析するには不十分な単一遅延スロットを占有している分岐命令 A + 2 および A + 3 は、無効化または「中止」される（それぞれ、行 4 および 5、列 8）。

【 0 0 4 8 】

リターンレジスタ（たとえば、図 1 B のレジスタファイル 9 における）は、一連の無効化された命令の第 1 の命令、たとえば、命令 A + 2 のアドレスをロードしてもよい。第 1 の無効化された命令は、命令 A + 1 として再ロード・レジスタ・アドレス 1 0 1 に書き込まれてもよい。割り込みサブルーチンから戻った後、または割り込みサブルーチンを完了した後、プロセッサは、プロセッサパイプラインにおける次の有効なスロットで無効化された命令の処理および実行を再開してもよい。

【 0 0 4 9 】

10

20

30

40

50

この場合、割込みルーチンから戻った後に命令に対して再び割り込む（列 5 および 6 において）前に無効化された命令 A + 2 および A + 3 で実行された動作の繰り返しに対して割込みペナルティが生じる。本発明の実施形態を用いてこの場合に生じる割込みペナルティは、二計算サイクルであり、図 2 に関連して記載される従来の機構を用いる同じ割込みに対して生じる三計算サイクルよりも減少している。

【 0 0 5 0 】

図 4 は、命令 A + 2 ではなく、命令 A + 3 が分岐命令であることを除いて、図 3 に関連して記載される割込みと同じ割込みを示す。

プロセッサ（たとえば、図 1 A および 1 B のプロセッサ 1）は命令 A + 1 および A + 2（それぞれ、行 3、列 7 および行 4、列 8）をデコードしてもよく、命令 A + 1 および A + 2 が分岐命令でないと判断する。したがって、命令 A + 1 および A + 2 は、命令が完全に処理されるまで中断することなく進行が許可される（たとえば、段 E 4 において：行 3、列 1 1、および行 4、列 1 2、いずれも図示せず）。プロセッサは、この後、命令 A + 3 をデコードし（行 5、列 9）、命令が分岐命令であると判断する。割込みは、たとえば、分岐命令 A + 3 を適切に処理するために必要な命令 A + 3 の後に続く遅延スロットを占有するので、命令 A + 3 は無効化または「中止」される（行 5、列 9）。

【 0 0 5 1 】

リターンレジスタ（たとえば、図 1 B のレジスタファイル 9 における）は、一連の無効化された命令における第 1 の命令、たとえば、命令 A + 3 のアドレスをロードしてもよい。第 1 の無効化された命令は、命令 A + 1 として再ロード・レジスタ・アドレス 1 0 1 に書き込まれてもよい。割込みサブルーチンから戻った後、または割込みサブルーチンを完了した後、プロセッサは、プロセッサパイプラインにおける次の有効なスロットで無効化された命令の処理および実行を再開してもよい。

【 0 0 5 2 】

この場合、命令に対して再び割り込む（行 6 において）前に無効化された命令 A + 3 で既に実行された動作の繰り返しに対して割込みペナルティが生じる。本発明の実施形態を用いたこの場合に生じる割込みペナルティは、一計算サイクルであり、たとえば、図 3 に関連して記載されるように、命令 A + 3 ではなく命令 A + 2 が分岐命令であるときの同じ割込みに対して生じた二計算サイクルよりも減少している。本発明の実施形態によると、命令が分岐命令であると判断されるとき、分岐命令自体、および分岐命令スロットと割込みスロットの間のパイプラインスロットにおける任意の命令は無効化または「中止」されてもよい。図 3 において、分岐命令スロット（行 4）と割込みスロット（行 6）の間には一つの命令 A + 3（行 5）がある。したがって、したがって、分岐命令自体 A + 2 および介在命令 A + 3 は、いずれも無効化されて二命令または二計算サイクルの総ペナルティをもたらす。図 4 において、分岐命令 A + 3（行 5）と割込み（行 6）の間には介在スロットがなく、したがって、分岐命令 A + 3 のみが無効化されて一命令または一計算サイクルの総ペナルティをもたらす。

【 0 0 5 3 】

図 5 は、命令 A、A + 1、A + 2、および A + 3 のどれも分岐命令でないことを除いて、図 3 および 4 に関連して記載される割込みと同じ割込みを示す。

プロセッサ（たとえば、図 1 A および 1 B のプロセッサ 1）は、命令 A + 1、A + 2、および A + 3（それぞれ、行 3、列 7。行 4、列 8。行 5、列 9。）をデコードし、これら命令のいずれも分岐命令でないと判断してもよい。したがって、命令 A + 1、A + 2、および A + 3 が完全に処理されるまで（たとえば、段 E 4 において、それぞれ、行 3、列 1 1。行 4、列 1 2。行 5、列 1 3。図示せず）、これらの命令は中断することなく進行が許可される。割込みに先立つ命令 A + 1、A + 2、および A + 3 のいずれも分岐命令ではないので、命令は無効化も「中止」もされない。

【 0 0 5 4 】

リターンレジスタ（たとえば、図 1 B のレジスタファイル 9 における）は、一連の命令における次の命令、たとえば、命令 A + 4 をロードしてもよく、命令 A + 4 は命令 A + 1

10

20

30

40

50

として再ロード・レジスタ・アドレス 101 に書き込まれる可能性がある。割込みサブルーチンから戻った後、または割込みサブルーチンを完了した後、プロセッサは、プロセッサパイプラインにおける次の有効なスロットで無効化された命令の処理および実行を再開してもよい。

【0055】

この場合、命令は無効化または「中止」されないので、割込みペナルティは生じない。この場合の割込みペナルティはゼロ計算サイクルであり、たとえば、図4に関連して記載されるように、命令 A + 3 が分岐命令であるとき同じ割込みに対して生じた一計算サイクルよりも減少している。

【0056】

割込みが処理されるとき、従来の機構は、命令が実際に分岐命令であるか否かに関わらず、たとえば、まだデコードされていない分岐命令であるかどうか分からない命令を常に無効化する。図2に関連して記載される例では、三つの上記のような命令、A + 1、A + 2、および A + 3 があり、三計算サイクルの割込みペナルティをもたらす。対照的に、本発明の実施形態に従って記載される機構は、分岐命令であると実際に判断される命令、および分岐命令スロットと割込みスロットの間のパイプラインスロットにおける任意の命令を無効化する。図3において、関連遅延スロットにおける一つの分岐命令 A + 2 および一つの命令 A + 3 は、中断されて、二計算サイクルの割込みペナルティをもたらす。図4において、一つの分岐命令 A + 3 は中断されて、一計算サイクルの割込みペナルティが生じる。図5において、分岐命令は中断されることがなく、したがって、割込みペナルティが生じない。したがって、本発明の実施形態では、コンピュータシステムにおいて割込み処理に関連する割込みペナルティが減少する。

【0057】

本発明の実施形態による方法のフローチャートである図6を参照する。

操作 600 において、プロセッサ（たとえば、図1Aおよび1Bのプロセッサ1）は、たとえば、図3～5に関連して記載されるように、さらに、それぞれのプロセッサ・パイプライン・スロットで一連の命令処理を開始してもよい。

【0058】

操作 610 において、プロセッサは、操作 600 における命令を処理するために使用される一連のスロットに続くプロセッサ・パイプライン・スロットで割込み処理を開始してもよい。

【0059】

操作 620 において、割込み処理またはサブルーチンが開始される時点で、操作 600 の各連続命令は命令の一つが分岐命令であると判断されるまで処理される。操作 600 の命令の一つが分岐命令であれば、処理は操作 630 に進んでもよい。そうでなければ、処理は操作 670 に進んでもよい。プロセッサは、たとえば、命令がデコードされるとき（たとえば、プロセッサパイプラインの D2 段において）、命令が分岐命令であるかどうかを判断してもよい。

【0060】

操作 630 において、プロセッサは、検出された分岐命令に関連する遅延スロットで割込みが処理されるかどうかを判断してもよい。典型的に、遅延スロットは、分岐命令に続き、たとえば、どの命令が分岐命令に続くかを判断するために分岐を適切に解析する必要があるかもしれない。割込みが上記の遅延スロットの一つを占有するとき、プロセッサは分岐命令に続く命令を判断しないかもしれない、分岐は未解析のままとなるかもしれない。操作 610 の割込みが操作 620 で検出された分岐命令に関連する遅延スロットで処理されるものとプロセッサが判断すれば、処理は命令 640 に進んでもよい。そうでなければ、処理は操作 670 に進んでもよい。

【0061】

操作 640 において、プロセッサは、操作 620 で検出された分岐命令を無効化または「中止」してもよく、分岐命令および割込みが処理されるパイプラインスロットの間にパ

10

20

30

40

50

イブラインスロット内で処理された命令があればこれらの命令も無効化または「中止」してよい。

【 0 0 6 2 】

操作 6 5 0 において、プロセッサは、たとえば、少なくとも第 1 の無効化された命令のアドレスをリターンレジスタ（たとえば、図 1 B のレジスタファイル 9 における）に書き込んでよい。

【 0 0 6 3 】

操作 6 6 0 において、分岐命令が検出されなくても検出されても、割込みがこれに関連する遅延スロットを占有しないとき、プロセッサは、無効化なしで、したがって、割込みペナルティなしで命令処理を完了してよい。

10

【 0 0 6 4 】

操作 6 7 0 において、割込みサブルーチンから戻った後、プロセッサは、操作 6 1 0 の割込みプロセッサ・パイプライン・スロットに続くプロセッサ・パイプライン・スロットでリターンレジスタからの命令を実行してもよい。

【 0 0 6 5 】

他の操作または一連の操作が採用されてもよい。

本発明の実施形態は、コンピュータまたはプロセッサ可読媒体、あるいは、プロセッサまたはコントローラ（たとえば、図 1 A および 1 B においてデバイス 1 0 0 のプロセッサ 1 ）によって実行されるときに本明細書で開示された方法を実行する命令をエンコードし、含み、または記憶する、たとえば、メモリ、ディスクドライブ、または U S B フラッシュメモリーなどのコンピュータまたはプロセッサ記憶媒体などの品目を含んでいてもよい。

20

【 0 0 6 6 】

先に図示され説明された具体的な実施形態は本発明に関わる多くの分散システムにとって有用であることが分るであろうが、本発明のさらなる修正形態が当業者には思い浮かぶであろう。このようなすべての修正形態は、添付の特許請求の範囲によって規定される本発明の範囲および趣旨の範囲内にあるものと見なされる。



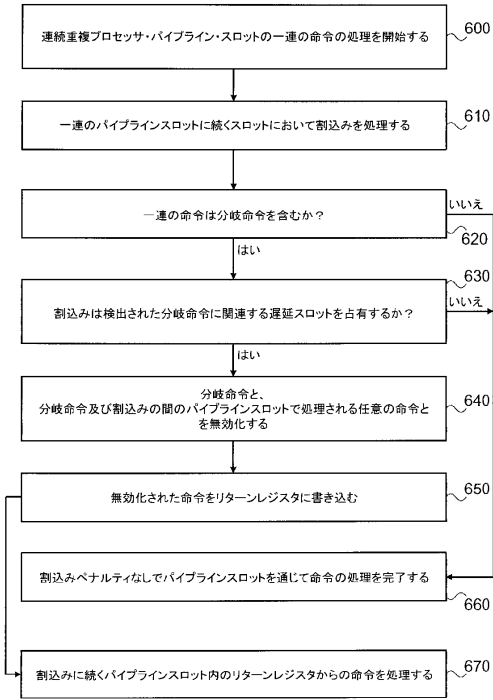
【図 4】

	1	2	3	4	5	6	7	8	9	10	11
	命令	プログラム アドレス									
1	A	100	IF1	IF2	D1	D2	D3	E1	E2	E3	E4
2	IRQ	*****	*****	*****	*****	*****	*****	*****	*****	*****	*****
3	A+1	101	IF1	IF2	D1	D2	D3	E1	E2	E3	E4
4	A+2	102	IF1	IF2	D1	D2	D3	E1	E2	E3	E4
5	A+3= 分岐	103	IF1	IF2	D1	D2	D3	E1	E2	E3	E4
6	割込みアドレス	200									
7	IA+1	201									
8	IA+2	202									
9	IA+3	203									
10	IA+4	204									

【図 5】

	1	2	3	4	5	6	7	8	9	10	11
	命令	プログラム アドレス									
1	A	100	IF1	IF2	D1	D2	D3	E1	E2	E3	E4
2	IRQ	*****	*****	*****	*****	*****	*****	*****	*****	*****	*****
3	A+1	101	IF1	IF2	D1	D2	D3	E1	E2	E3	E4
4	A+2	102	IF1	IF2	D1	D2	D3	E1	E2	E3	E4
5	A+3	103	IF1	IF2	D1	D2	D3	E1	E2	E3	E4
6	割込みアドレス	200									
7	IA+1	201									
8	IA+2	202									
9	IA+3	203									
10	IA+4	204									

【図 6】



## フロントページの続き

(72)発明者 エイタン ハイ

イスラエル国 52601 ラマツ ガン クリニツィ ストリート 106

審査官 清木 泰

(56)参考文献 特開2001-84149(JP,A)

特開平10-222367(JP,A)

特開平10-21074(JP,A)

特開平7-200294(JP,A)

特開平7-152564(JP,A)

特開平7-6037(JP,A)

特開平5-143363(JP,A)

特開平5-143362(JP,A)

特開平4-133132(JP,A)

特開平1-123325(JP,A)

特開昭62-54342(JP,A)

特開昭62-3342(JP,A)

米国特許第5774709(US,A)

Gerry Kane著,前川守監訳,mips RISC アーキテクチャ -R2000/R3000-,日本,共立出版株式会社,1992年10月1日,初版,Pages:67-70,130

Lisa Guerra et al., Cycle and Phase Accurate DSP Modeling and Integration for HW/SW Co-Verification, Proceedings of 36th Design Automation Conference, IEEE, 1999年, Pages:964-969

(58)調査した分野(Int.Cl., DB名)

G06F 9/38

G06F 9/46 - 9/54