

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第6257745号
(P6257745)

(45) 発行日 平成30年1月10日 (2018. 1. 10)

(24) 登録日 平成29年12月15日 (2017. 12. 15)

(51) Int. Cl.

F I

G 0 6 F 9/32 (2006.01)

G 0 6 F 9/42 3 3 0 A

G 0 6 F 9/42 3 2 0 A

請求項の数 5 (全 12 頁)

(21) 出願番号	特願2016-503221 (P2016-503221)	(73) 特許権者	507364838
(86) (22) 出願日	平成26年3月14日 (2014. 3. 14)		クアルコム, インコーポレイテッド
(65) 公表番号	特表2016-517092 (P2016-517092A)		アメリカ合衆国 カリフォルニア 9 2 1
(43) 公表日	平成28年6月9日 (2016. 6. 9)		2 1 サン ディエゴ モアハウス ドラ
(86) 国際出願番号	PCT/US2014/029778		イブ 5 7 7 5
(87) 国際公開番号	W02014/145101	(74) 代理人	100108453
(87) 国際公開日	平成26年9月18日 (2014. 9. 18)		弁理士 村山 靖彦
審査請求日	平成29年3月14日 (2017. 3. 14)	(74) 代理人	100163522
(31) 優先権主張番号	13/833, 844		弁理士 黒田 晋平
(32) 優先日	平成25年3月15日 (2013. 3. 15)	(72) 発明者	ロドニー・ウェイン・スミス
(33) 優先権主張国	米国 (US)		アメリカ合衆国・カリフォルニア・9 2 1
早期審査対象出願			2 1・サン・ディエゴ・モアハウス・ドラ
			イヴ・5 7 7 5
			最終頁に続く

(54) 【発明の名称】 プロセッサにおいてリターン分岐命令を実行する速度を向上させる方法

(57) 【特許請求の範囲】

【請求項 1】

プロセッサにおいて、コール分岐命令およびリターン分岐命令を実行するための方法であって、

前記プロセッサが前記コール分岐命令と異なるリンクレジスタ操作命令をデコードすることに応じて、カウンタを初期値に設定するステップと、

前記プロセッサがコール分岐命令をデコードすることに応じて、第1の定数だけ前記カウンタをインクリメントするステップと、

前記プロセッサが前記コール分岐命令をデコードすることに応じて、前記コール分岐命令を含むプログラムに記載された命令の順序で次の命令のアドレスをリンクレジスタスタックにプッシュするステップと、

前記プロセッサがリターン分岐命令をデコードすることに応じて、前記カウンタが前記初期値に等しくない値を有する場合に、第2の定数だけ前記カウンタをインクリメントするステップと、

前記プロセッサが前記リターン分岐命令をデコードすることに応じて、前記カウンタが前記初期値に等しくない値を有する場合に、前記リターン分岐命令のためのターゲットアドレスを前記リンクレジスタスタックからポップするステップであって、前記カウンタの前記値が前記初期値に等しくないとき、前記リンクレジスタスタックからポップされた前記ターゲットアドレスは前記リターン分岐命令の正確なターゲットアドレスである、ステップと、

10

20

前記カウンタが前記初期値に等しくない値を有する場合に、前記リターン分岐命令の実行の完了後に、前記リターン分岐命令を前記命令キャッシュから取り除くステップとを含む、方法。

【請求項 2】

前記初期値がゼロであり、前記第1の定数が1に等しく、前記第2の定数がマイナス1に等しい、請求項1に記載の方法。

【請求項 3】

リンクレジスタと、
リンクレジスタスタックと、
分岐カウンタと、
プロセッサであって、

10

前記リンクレジスタに書き込むコール分岐命令と異なる命令をデコードすることに応じて、前記分岐カウンタを初期値に設定し、

コール分岐命令をデコードすることに応じて、第1の定数だけ前記分岐カウンタをインクリメントし、

前記プロセッサが前記コール分岐命令をデコードすることに応じて、前記コール分岐命令を含むプログラムに記載された命令の順序で次の命令のアドレスを前記リンクレジスタスタックにプッシュし、

リターン分岐命令をデコードすることに応じて、前記分岐カウンタが前記初期値に等しくない値を有する場合に、第2の定数だけ前記分岐カウンタをインクリメントし、

20

前記プロセッサが前記リターン分岐命令をデコードすることに応じて、前記分岐カウンタが前記初期値に等しくない値を有する場合に、前記リターン分岐命令のためのターゲットアドレスを前記リンクレジスタスタックからポップし、

前記カウンタが前記初期値に等しくない値を有する場合に、前記リターン分岐命令の実行の完了後に、前記リターン分岐命令を前記命令キャッシュから取り除く

ように構成されたプロセッサであって、前記分岐カウンタの前記値が前記初期値に等しくないとき、前記リンクレジスタスタックからポップされた前記ターゲットアドレスは前記リターン分岐命令の正確なターゲットアドレスである、プロセッサと、
を備える、装置。

【請求項 4】

30

前記初期値がゼロであり、前記第1の定数が1に等しく、前記第2の定数がマイナス1に等しい、請求項3に記載の装置。

【請求項 5】

前記装置が、セルラー電話と、基地局とからなる集合から選択される、請求項3に記載の装置。

【発明の詳細な説明】

【技術分野】

【0001】

本発明の諸実施形態は、マイクロプロセッサを対象とする。

【背景技術】

40

【0002】

コール分岐命令により、プログラム制御が、サブルーチン(関数)を実行するコードのセクションに分岐できるようになる。たとえば、コール分岐命令が実行されるとき、プログラムカウンタ中のアドレスはインクリメントされ、後に再び呼び出されるようにリンクレジスタにロードされ、プログラム制御が、呼び出されるサブルーチンを表すコードのセクションに分岐するように、呼び出されるサブルーチンのコードの最初の行のアドレスは、プログラムカウンタにロードされる。コードのセクションの終わりのリターン分岐命令により、プログラム制御はリンクレジスタに記憶されたアドレスに戻り、それによってサブルーチンを呼び出したコール分岐命令後のプログラム順で次の命令に戻る。

【0003】

50

入れ子型または再帰的サブルーチン(関数)を実行するには、すなわち、サブルーチンが別のサブルーチン(それ自体であってもよい)を呼び出すには、ソフトウェアがメモリでスタックを利用することができる。たとえば、第1のサブルーチンコールによりアドレスがリンクレジスタにロードされていた場合、また第2のサブルーチンが、第1のサブルーチンを表すコード内で呼び出される場合、第1のサブルーチンが呼び出されたときリンクレジスタにロードされたアドレスは、ソフトウェアの規則によってメモリでスタックの最上部にプッシュされ、第2の分岐命令後のプログラム順で次の命令を表すアドレスが、リンクレジスタにロードされる。第2の呼び出されたサブルーチンに対するリターン分岐命令が実行されるとき、リンクレジスタに記憶されたアドレスは、プログラムカウンタにロードされ、スタックの最上部のアドレスは次にポップされ、リンクレジスタにロードされて、第1のサブルーチンが完了すると、プログラム制御が第1の分岐命令の(プログラム順で)直後の命令に戻ることができるようにする。

10

【発明の概要】

【課題を解決するための手段】

【0004】

本発明の諸実施形態は、リンクレジスタスタックを使用してリターン分岐命令を実行するためのプロセッササイクルの数を減少させるシステムおよび方法を対象とする。

【0005】

一実施形態では、方法は、プロセッサがコール分岐命令以外のリンクレジスタ操作命令をデコードすることに応じて、状態機械を初期状態に設定するステップと、コール分岐命令またはリターン分岐命令をデコードすることに応じて状態機械を、プロセッサがコール分岐命令以外のリンクレジスタ操作命令をデコードして以来、プロセッサによってデコードされたコール分岐命令の、それに対してプロセッサは対応するリターン分岐命令をデコードしていない、数を示す状態に変更するステップとを含む。この方法はさらに、状態機械の状態が、プロセッサがコール分岐命令以外のリンクレジスタ操作命令をデコードして以来、少なくとも1つのコール分岐命令がデコードされたことを示し、およびそれに対して対応するリターン分岐命令がデコードされていないならば、プロセッサがリターン分岐命令をフェッチし、デコードすることに応じて、リンクレジスタスタックからターゲットアドレスをポップし、このターゲットアドレスを次の命令をフェッチするアドレスとして使用するステップと、ターゲットアドレスの正当性をチェックすることなくリターン分岐命令の実行を完了するステップとを含む。

20

30

【0006】

別の態様では、方法は、プロセッサがコール分岐命令以外のリンクレジスタ操作命令をデコードすることに応じて、カウンタを初期値に設定するステップと、プロセッサがコール分岐命令をデコードすることに応じて、第1の定数だけカウンタをインクリメントするステップと、プロセッサがリターン分岐命令をデコードすることに応じて、第2の定数だけカウンタをインクリメントするステップとを含む。この方法はさらに、プロセッサがカウンタの値は初期値に等しくないと判定することに応じて、およびプロセッサがリターン分岐命令をデコードすることに応じて、リターン分岐命令のためのターゲットアドレスをリンクレジスタスタックからポップするステップと、ターゲットアドレスの正当性をチェックすることなくリターン分岐命令の実行を完了するステップとを含む。

40

【0007】

別の実施形態では、装置が、リンクレジスタと、分岐カウンタと、リンクレジスタに書き込むコール分岐命令以外の命令をデコードすることに応じて分岐カウンタを初期値に設定し、コール分岐命令をデコードすることに応じて第1の定数だけ分岐カウンタをインクリメントし、リターン分岐命令をデコードすることに応じて第2の定数だけ分岐カウンタをインクリメントするためのプロセッサとを含む。この装置はさらにリンクレジスタスタックを含み、プロセッサが、カウンタの値は初期値に等しくないと判定することに応じて、およびリターン分岐命令をデコードすることに応じて、リターン分岐命令のためのターゲットアドレスをリンクレジスタスタックからポップし、ターゲットアドレスの正当性を

50

チェックすることなく、実行を完了し、リターン分岐命令を回収する(retire)。

【0008】

別の実施形態では、プロセッサが、分岐カウンタと、リンクレジスタスタックと、カウンタを設定するための手段であって、プロセッサがコール分岐命令以外のリンクレジスタ操作命令をデコードすることに応じて、分岐カウンタを初期値に設定する、カウンタを設定するための手段と、カウンタをインクリメントするための手段であって、プロセッサがコール分岐命令をデコードすることに応じて第1の定数だけ分岐カウンタをインクリメントするように、およびプロセッサがリターン分岐命令をデコードすることに応じて第2の定数だけ分岐カウンタをインクリメントするように、カウンタをインクリメントするための手段とを含む。プロセッサはさらに、リンクレジスタスタックをポップするための手段であって、プロセッサが分岐カウンタの値は初期値に等しくないと判定することに応じて、およびプロセッサがリターン分岐命令をデコードすることに応じて、リターン分岐命令のためのターゲットアドレスをリンクレジスタスタックからポップする、リンクレジスタスタックをポップするための手段と、ターゲットアドレスの正当性をチェックすることなく、実行を完了し、リターン分岐命令を回収するプロセッサとを含む。

10

【0009】

別の実施形態では、記憶媒体が、プロセッサによって実行可能な命令を記憶し、プロセッサは、命令を実行するとき方法を実行し、方法は、プロセッサがコール分岐命令以外のリンクレジスタ操作命令をデコードすることに応じて、カウンタを初期値に設定するステップと、プロセッサがコール分岐命令をデコードすることに応じて、第1の定数だけカウンタをインクリメントするステップと、プロセッサがリターン分岐命令をデコードすることに応じて、第2の定数だけカウンタをインクリメントするステップとを含む。この方法はさらに、プロセッサがカウンタの値は初期値に等しくないと判定することに応じて、およびプロセッサがリターン分岐命令をデコードすることに応じて、リターン分岐命令のためのターゲットアドレスをリンクレジスタスタックからポップするステップと、ターゲットアドレスの正当性をチェックすることなくリターン分岐命令の実行を完了するステップとを含む。

20

【0010】

添付の図面は、本発明の諸実施形態の説明に役立つように提示し、単に諸実施形態の例示のために提供するものであり、これを限定するものではない。

30

【図面の簡単な説明】

【0011】

【図1】一実施形態によるマイクロプロセッサを抽象化した図である。

【図2】一実施形態による命令フローを示す図である。

【図3】一実施形態によるフロー図である。

【図4】一実施形態を含むワイヤレス通信システムの図である。

【発明を実施するための形態】

【0012】

本発明の特定の実施形態に向けた次の説明および関連図面において、本発明の諸態様を開示する。本発明の範囲を逸脱することなく、代替実施形態を考案することができる。さらに、本発明の関連する詳細をあいまいにしないために、本発明の周知の要素については詳しく説明しない、または省略する。

40

【0013】

「例示的」という用語は、本明細書では「一例、事例、または実例として役立つ」ことを意味するように使用される。「例示的」として本明細書で説明する実施形態が、必ずしも他の実施形態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。同様に、「本発明の諸実施形態」という用語は、本発明のすべての実施形態が、述べられる特徴、利点、または動作モードを含むことを求めない。

【0014】

本明細書で使用される用語は、単に特定の実施形態を説明するためであって、本発明の

50

実施形態を限定するものであることを意図されていない。本明細書で使用される単数形、「a」、「an」、および「the」は、文脈上明らかに他の意味を示す場合を除いて、複数形も含むものとする。「備える」、「備えている」、「含む」、および/または「含んでいる」という用語は、本明細書で使用されるとき、述べられる特徴、完全体、ステップ、動作、要素、および/または構成要素の存在を明記するが、1つまたは複数の他の特徴、完全体、ステップ、動作、要素、構成要素、および/またはその集まりの存在または追加を排除しないことはさらに理解されよう。

【0015】

さらに、多くの実施形態について、たとえばコンピュータデバイスの要素によって実行される一連の動作の観点から説明する。本明細書に記載する様々な動作は、特定の回路(たとえば、特定用途向け集積回路(ASIC))で、1つまたは複数のプロセッサによって実行されるプログラム命令で、または両方の組合せで行うことができることは認識されよう。さらに、本明細書に記載するこれらの一連の動作は、実行されると関連するプロセッサに本明細書に記載する機能を行わせる、対応するコンピュータ命令のセットを記憶したいかなる形態の非一時的コンピュータ可読記憶媒体内でも完全に具体化されるとみなすことができる。このように、本発明の様々な態様は、いくつかの異なる形態で具体化されることが可能であり、そのすべてが請求される対象の範囲内であると考えられている。また、本明細書に記載する実施形態のそれぞれについては、このようないかなる実施形態の対応する形態も、本明細書ではたとえば、記載する動作を行う「ように構成された論理」として記載されることがある。

【0016】

本明細書ではリンクレジスタスタックと呼ばれるハードウェア構造を使用することによって、ソフトウェアスタックを使用するとき、諸実施形態が改善されると期待される。リンクレジスタスタックは、適切な環境の下で、リターン分岐命令のためのターゲットアドレスが、ターゲットアドレスの正当性を確認することなく、リンクレジスタスタックからパイプラインに転送されることが可能であるように、リンクレジスタ値(アドレス)を記憶するために使用される。

【0017】

図1は、一実施形態に従ったマイクロプロセッサ100の簡略化された抽象概念である。パイプライン102が、データキャッシュ104からのデータ、および命令キャッシュ106からの命令を受信する。コントローラ108が、パイプライン102内の様々なステージおよび実行ユニット(フェッチステージ、デコードステージなど)を制御して、命令をデコードし、実行する。完了すると実行された命令からの結果を、レジスタファイル110に、データキャッシュ104に、およびメモリ112で表される他のバッファもしくはメモリユニットに、記憶することができる。メモリ112は、メモリ階層を表すことができる。背景技術において述べたように、ソフトウェアスタックは抽象的なメモリタイプであり、113と表示される。

【0018】

レジスタファイル110は、図1に示す3つ、すなわちプログラムカウンタレジスタ114と、リンクレジスタ116と、ソフトウェアスタック113の最上部へのポインタを記憶するためのスタックポインタレジスタ118とを有する、複数のレジスタを含む。

【0019】

リンクレジスタスタックは、120と表示されたハードウェアメモリ構造として、図1に示されている。リンクレジスタスタック120は、マイクロプロセッサ100と同じダイ(チップ)上に位置するレジスタファイルであってもよく、レジスタファイル110とは別個の構造として示されている。実際には、リンクレジスタスタック120は、レジスタファイル110の一部とすることができるが、マイクロプロセッサ100の設計済みレジスタ(architected register)の一部ではない可能性があることを示すために、別個の構造として示されている。リンクレジスタスタック120は、124と表示された分岐カウンタを含む。分岐カウンタ124の役割については、後で説明する。

【0020】

ー実施形態については、コントローラ108は、パイプライン102がコール分岐命令をデコードするとき、(プログラム順で)次の命令のアドレスがリンクレジスタスタック120上にプッシュされるように構成される。(背景技術において説明したように、このアドレスはまた、値としてリンクレジスタ116に記憶され、リンクレジスタ116に記憶された前の値は、ソフトウェアスタック113上にプッシュされ、スタックポインタレジスタ118に記憶されたアドレスによってポイントされる。)

【0021】

サブルーチン(関数)が呼び出されるソフトウェアプログラム中の各コール分岐命令については、サブルーチンの終わりにリターン分岐命令がある。このような各コール分岐命令およびリターン分岐命令を、分岐命令のペアとみなすことができる。すなわち、リターン分岐命令は、コール分岐命令とペアになっている。これを考慮して、分岐カウンタ124は、パイプライン102がデコードされたコール分岐命令とペアになったリターン分岐命令をデコードしていない、およびパイプライン102がリンクレジスタ操作命令に遭遇していない、デコードされたコール分岐命令の数の記録をとる構造として説明されてもよい。リンクレジスタ操作命令が、コール分岐命令以外の、リンクレジスタ116に書き込むいかなる命令も指す。

【0022】

分岐カウンタ124は、ゼロに初期化され、パイプライン102がリンクレジスタ操作命令をデコードするたびにゼロに設定される。コール分岐命令がデコードされ、命令アドレスがリンクレジスタスタック120上にプッシュされるたびに、分岐カウンタ124の値は1だけインクリメントされ、リターン分岐命令がデコードされ、命令アドレスがリンクレジスタスタック120からポップされるたびに、分岐カウンタ124の値は1だけデクリメントされる。

【0023】

リンクレジスタスタック120上に最初の命令アドレスがプッシュされて以来、リンクレジスタ操作命令がない限り、リンクレジスタスタック120に記憶された値(アドレス)は、リンクレジスタ116に書き込まれた値(アドレス)の正しい履歴を表す。したがって、分岐カウンタ124の値がゼロより大きい限り、リンクレジスタスタック120の状態は、リンクレジスタ116の正しい履歴を表す。

【0024】

ー実施形態については、コントローラ108は、パイプライン102がリターン分岐命令をデコードするとき、および分岐カウンタ124に記憶された値がゼロより大きいならば、リンクレジスタスタック120の最上部のアドレスが、デコードされたリターン分岐命令のためのターゲットアドレスとしてパイプライン102に転送されるように構成される。分岐カウンタ124に記憶された値はゼロより大きいので、リンクレジスタスタック120からポップされたアドレスは、正しいターゲットアドレスであり、ゆえにデコードされたリターン分岐命令が実行を完了し、回収するために正当性のチェックが行われる必要はない。

【0025】

しかしながら、パイプライン102がリターン分岐命令をデコードするとき、分岐カウンタ124の値がゼロである場合、リンクレジスタスタック120の状態が、リンクレジスタ116の正しい履歴を表しているという保証はなく、この場合コントローラ108は、パイプライン102を従来の方法で処理し、それによってソフトウェアスタック113は、背景技術で説明したように使用されて、予測ターゲットアドレスを提供し、正当性のチェックが行われた後に、デコードされたリターン分岐命令は実行を完了し、回収することができる。

【0026】

分岐カウンタ124を使用することによって、諸実施形態は、入れ子型サブルーチンの場合、ターゲットアドレスのためにリンクレジスタスタック120を使用することができ、リンクレジスタ116を操作する(リンクレジスタ116に書き込む)介入リンクレジスタ操作命令がない限り、呼び出されたサブルーチンが別のサブルーチンを呼び出す。

【0027】

このような実施形態によるプロセッサが、プロセッササイクルを節約すると期待され、

10

20

30

40

50

それによって電力および時間を節約する。

【 0 0 2 8 】

図2は、コードのセグメント中の命令フローを考慮することによって一実施形態を示す。文字Aから文字Lは、コードセグメント中の命令を表す。図2の実施形態を説明するために、変数xは、分岐カウンタ124の値を示す。コードセグメントを実行する前に、分岐カウンタ124の値は、表示200で示されるように、ゼロに設定される。

【 0 0 2 9 】

図2に示すコードセグメント中の第1の命令は、命令A、すなわち、図2の命令Dである、L1と表示された命令へのコール分岐命令である。コール分岐命令Aをデコードし、実行すると、命令Bのためのアドレスは、リンクレジスタ116に書き込まれ、またリンクレジスタスタック120上にプッシュされる。分岐カウンタ124は、図2の表示201で示すように、1だけインクリメントされる。分岐カウンタ124の値は、現在1である。

【 0 0 3 0 】

プログラムカウンタレジスタ114は、現在、命令Dのアドレスに設定され、パイプライン102の命令フローは、202と表示される矢印で示されるように、命令Aから命令Dに進む。命令フローは次いで、図2の204、206、および208と表示される矢印で示されるように、命令Dから命令Eに、命令Eから命令Fに、および命令Fから命令Gに進む。

【 0 0 3 1 】

命令Gは、図2の命令Jである、L2と表示される命令へのコール分岐命令である。命令Jのアドレスは、プログラムカウンタレジスタ114にロードされ、210と表示される矢印で示すように、制御は命令Jに導かれる。命令Hのためのアドレスは、リンクレジスタスタック120上にプッシュされ、リンクレジスタ116に書き込まれる。分岐カウンタ124は、表示212で示すように、1だけインクリメントされる。分岐カウンタ124の値は、現在2に等しい。

【 0 0 3 2 】

プログラムカウンタレジスタ114は命令Jのアドレスに設定されていたので、パイプライン102の命令フローは、214および216と表示される矢印で示されるように、命令Jから命令Kに、次いで命令Kから命令Lに進む。

【 0 0 3 3 】

命令Lは、リターン分岐命令であり、分岐カウンタ124の値(図2のx)がゼロより大きいので、リンクレジスタスタック120の状態は正しいことがわかり、したがってリンクレジスタスタック120の最上部からポップされるターゲットアドレスは正しい。ゆえに、命令Lが実行され、完了するとき、ポップされたアドレスの正当性をチェックする必要はない。このポップされたアドレスは、命令Hのためのアドレスである。分岐カウンタ124は、表示218で示すように、命令Lをデコードし、実行すると、1だけデクリメントされる。分岐カウンタ124の値は、現在1である。

【 0 0 3 4 】

図2に示す例を続けると、命令Lのターゲットは命令Hのアドレスであり、命令Hはプログラム順でコール分岐命令G直後の命令であるので、プログラム制御は、220と表示される矢印で示す命令Hに導かれる。

【 0 0 3 5 】

命令Hは、リターン分岐命令としてデコードされ、分岐カウンタ124の値がゼロより大きいので、リンクレジスタスタック120の状態が正しいことがわかり、ゆえにリンクレジスタスタック120からポップされたターゲットアドレスは、正しいターゲットアドレスである。このターゲットアドレスは、命令Bのための命令アドレスであり、命令Hが実行され、完了するとき、正当性をチェックされる必要はない。命令Hを実行すると、分岐カウンタ124は、表示222で示すように1だけデクリメントされ、制御は224と表示する矢印で示すように、命令Bに導かれる。分岐カウンタ124の値は、現在ゼロである。

【 0 0 3 6 】

前述のように、コントローラ108は、リンクレジスタ操作命令、すなわちリンクレジスタ116に書き込むコール分岐命令以外の命令をデコードし、実行すると、分岐カウンタ124

10

20

30

40

50

の値がゼロに設定されるように構成される。リターン分岐命令がデコードされ、その分岐カウンタ124の値がゼロである実行を開始する場合、リンクレジスタスタック120の状態が正しいという保証はない。したがって、予測ターゲットアドレスが、ソフトウェアスタック113からポップされる場合、リターン分岐命令が実行を完了して、回収する前に、予測ターゲットアドレスの正当性をチェックされるべきである。

【 0 0 3 7 】

図3は、一実施形態によるフロー図を示す。302と表示されるステップから始まって、命令が命令キャッシュ106からフェッチされ、デコードされるとき、図3の決定フローは、デコードされる命令がコール分岐命令、リターン分岐命令、またはリンクレジスタ操作命令であるかどうかに応じて決まる。一般性を失うことなく、これらの3つの命令のどれがデコードされる命令であるかを判定する順序は、表示304、306、および308で図3に示されている。しかしながら、他の実施形態については、これらの判定は、図3に示す順序ではない可能性があることは理解されよう。

【 0 0 3 8 】

ステップ304を参照すると、命令がコール分岐命令であるかどうかについて判定が行われる。そうである場合、(プログラム順で)次の命令のアドレスは、ステップ310に示すようにリンクレジスタスタック120上にプッシュされ、分岐カウンタ124の値は、ステップ312に示されるように1だけインクリメントされる($x \rightarrow x+1$)。図3は、ステップ312の動作を実行した後にステップ302に戻るフロー図を示している。ステップ312からステップ302へのデータフローは、ステップ312の後に実際の実施形態のパイプラインでとられる次の動作が必ず命令フェッチであることを暗示するものではないが、データフローは、一実施形態によってとられる動作を説明するために、単に便宜上そのようなものとして示されている。

【 0 0 3 9 】

ステップ306を参照すると、命令がリターン分岐命令であるかどうかについて判定が行われる。命令がリターン分岐命令であり、分岐カウンタ124の値がゼロより大きい場合(ステップ314)、ターゲットアドレスは、リンクレジスタスタック120からポップされて(ステップ316)、パイプライン102に転送され、プログラムカウンタレジスタ114に書き込まれ、分岐カウンタ124は1だけデクリメントされ(ステップ318)、命令が実行され、完了するとき、ターゲットアドレスの正当性をチェックする必要はない、ステップ320。

【 0 0 4 0 】

しかしながら、ステップ314において、分岐カウンタ124の値がゼロであると判定される場合、マイクロプロセッサ100は従来の方法で進行し、それによって予測ターゲットアドレスが使用される場合、その正当性をチェックすべきである(ステップ322)。

【 0 0 4 1 】

図3は、ステップ320またはステップ322の動作を実行した後にステップ302に戻るフロー図を示すが、上述のように、このようなプロセスフローは、単に便宜上紹介する。

【 0 0 4 2 】

ステップ308を参照すると、命令がリンクレジスタ116を操作する(たとえば、これに書き込む)(コール分岐命令以外の)いずれかの命令であるかどうかについて判定が行われる。そうである場合、分岐カウンタ124の値は、ステップ324に示すように、ゼロにリセットされる($x \rightarrow 0$)。そうではない場合、制御はステップ302に戻る。やはり、ステップ308およびステップ324の動作を実行した後にステップ302に戻るフロー図は、単に便宜上紹介する。

【 0 0 4 3 】

分岐カウンタ124はゼロを超えてデクリメントされないという規則を想定する。すなわち、パイプライン102がリターン分岐命令をデコードすることが起こり得るが、パイプライン102はコール分岐命令以外のリンクレジスタ操作命令をデコードしたので、分岐カウンタ124はすでにゼロに設定されている。この場合、分岐カウンタ124はゼロのままである。

10

20

30

40

50

【 0 0 4 4 】

一般に、状態機械は、分岐カウンタ124の動作を実行することができ、状態は、介入リンクレジスタ操作命令をデコードして以来、プロセッサ100によってデコードされたコール分岐命令の、それに対してデコードされたコール分岐命令とペアになったリターン分岐命令がまだデコードされていない、数の記録をとることは理解されよう。

【 0 0 4 5 】

たとえば、カウンタは、コール分岐命令がデコードされるたびに1だけインクリメントされる、リターン分岐命令がデコードされるたびに1だけデクリメントされる、またはリンクレジスタ操作命令がデコードされるたびにゼロにリセットされることは必要ではない。特定の例として、カウンタの方向は、逆にされてもよく、カウンタは、その最大値に初期化され、コール分岐命令がデコードされるたびにデクリメントされ、リターン分岐命令がデコードされるたびにインクリメントされる。図3によって表されるフロー図は、この特定の例に基づいて調整されることが可能である。

【 0 0 4 6 】

図3に関して説明した動作は、コントローラ108で実現されるハードウェアで実行されてもよい。いくつかの実施形態については、図3に示した動作の一部または全部が、ファームウェアまたはソフトウェアとみなされる命令など、コントローラ108で実行される命令によって実現されてもよく、このような命令がメモリ112に記憶されてもよい。

【 0 0 4 7 】

図4は、諸実施形態が適用されることが可能であるワイヤレス通信システムを示す。図4は、基地局404Aと、基地局404Bと、基地局404Cとを含んだワイヤレス通信ネットワーク402を示す。図4は、406と表示された通信デバイスを示し、これはいわゆるスマートフォン、タブレット、またはセルラー電話ネットワークに適した他の種類の通信デバイス、たとえばコンピュータなど、モバイルセルラー通信デバイスであってもよい。通信デバイス406は、モバイルである必要はない。図4の特定の例では、通信デバイス406は、基地局404Cと関連付けられるセル内に位置している。矢印408および矢印410は、それぞれアップリンクチャネルおよびダウンリンクチャネルを図示しており、これにより通信デバイス406は基地局404Cと通信する。

【 0 0 4 8 】

たとえば、通信デバイス406と関連付けられるデータ処理システム、または基地局404Cと関連付けられるデータ処理システム、または両方において、諸実施形態が使用され得る。図4は、本明細書に記載する諸実施形態が使用され得る多くの応用の中から単に1つの応用を示す。

【 0 0 4 9 】

情報および信号は、多種多様な技術および技法のいずれかをを用いて表され得ることを、当業者は理解するであろう。たとえば、上記の説明全体にわたって参照することができるデータ、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、磁界もしくは磁性粒子、光場もしくは光粒子、またはその組合せによって表すことができる。

【 0 0 5 0 】

さらに、本明細書で開示した実施形態と関連して説明した様々な説明的な論理ブロック、モジュール、回路、およびアルゴリズムのステップは、電子ハードウェア、コンピュータソフトウェア、または両方の組合せとして実装されることが可能であることを、当業者は理解するであろう。ハードウェアとソフトウェアのこのような互換性をわかりやすく説明するために、様々な例示的構成要素、ブロック、モジュール、回路、およびステップは、概してその機能性に関して上述した。このような機能性がハードウェアとして実装されるか、またはソフトウェアとして実装されるかは、特定の応用およびシステム全体に課される設計の制約によって決まる。当業者は、各特定の応用に対して様々な方法で記載した機能を実行することができるが、このような実行の決定は、本発明の範囲からの逸脱をもたらすと解釈されるべきではない。

【 0 0 5 1 】

本明細書に開示した実施形態と関連して記載した方法、シーケンス、および/またはアルゴリズムは、直接ハードウェアで、プロセッサによって実行されるソフトウェアモジュールで、またはこの2つの組合せで、具体化されることが可能である。ソフトウェアモジュールは、RAMメモリ、フラッシュメモリ、ROMメモリ、EPROMメモリ、EEPROMメモリ、レジスタ、ハードディスク、リムーバブルディスク、CD-ROM、または当技術分野で知られているその他の形態の記憶媒体に、常駐することができる。図1の特定の例では、1つまたは複数のソフトウェアモジュールは、メモリ112に記憶されてもよい。例示的な記憶媒体をプロセッサに結合し、プロセッサがこの記憶媒体から情報を読み取ること、およびこの記憶媒体に情報を書き込むことができるようにする。代替的には記憶媒体は、プロセッサと

10

【 0 0 5 2 】

したがって、本発明の一実施形態が、リターン分岐命令を実行する速度を向上させる方法を具体化する非一時的コンピュータ可読媒体を含むことができる。したがって、本発明は、図示した例に限定されず、本明細書に記載する機能を実行するためのいかなる手段も、本発明の諸実施形態に含まれる。

【 0 0 5 3 】

上述の開示は、本発明の例示的な実施形態を示すが、様々な変更形態および修正形態が、添付の特許請求の範囲によって定義される本発明の範囲から逸脱することなく本明細書において作成可能であることに注意すべきである。本明細書に記載する本発明の実施形態に従った方法の請求項の機能、ステップ、および/または動作は、特定の順序で行われる必要はない。さらに、本発明の要素は、単数で記載または請求される場合があるが、単数に限定することが明示的に表明されていない場合、複数が検討される。

20

【 符号の説明 】

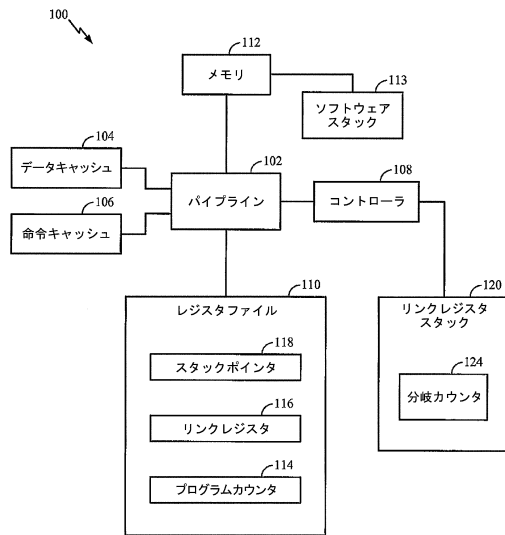
【 0 0 5 4 】

- 100 マイクロプロセッサ
- 102 パイプライン
- 104 データキャッシュ
- 106 命令キャッシュ
- 108 コントローラ
- 110 レジスタファイル
- 112 メモリ
- 113 ソフトウェアスタック
- 114 プログラムカウンタ
- 116 リンクレジスタ
- 118 スタックポインタ
- 120 リンクレジスタスタック
- 124 分岐カウンタ
- 404 基地局
- 406 通信デバイス

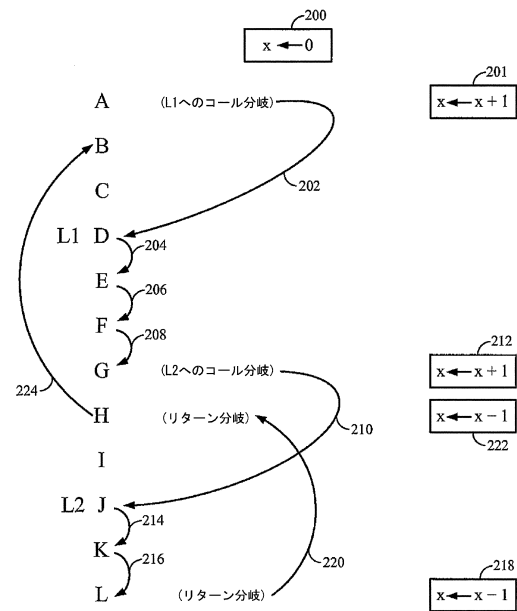
30

40

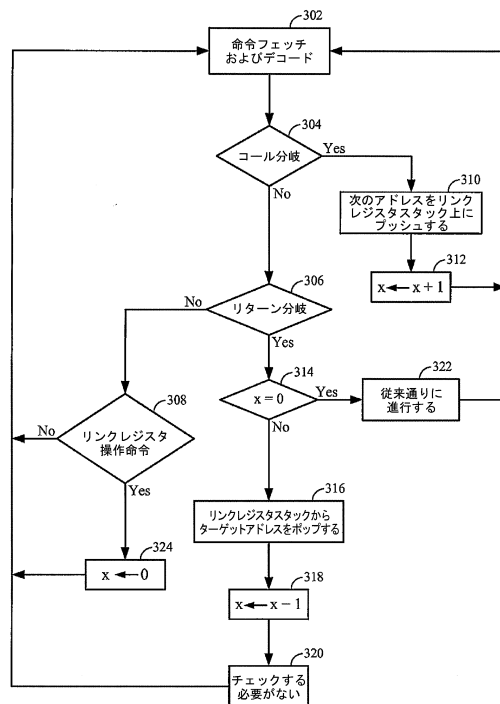
【図 1】



【図 2】



【図 3】



【図 4】

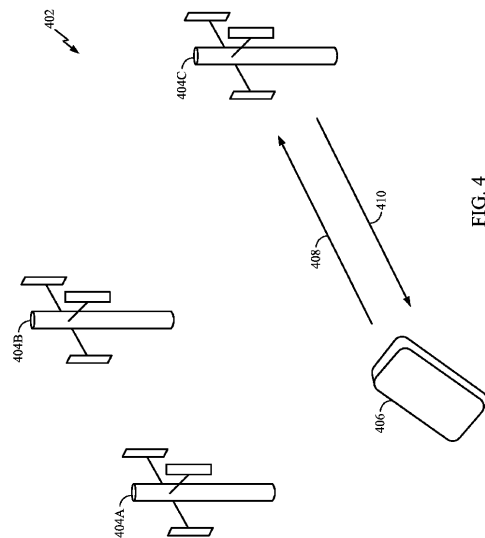


FIG. 4

フロントページの続き

- (72)発明者 ジェフェリー・エム・ショットミラー
アメリカ合衆国・カリフォルニア・9 2 1 2 1・サン・ディエゴ・モアハウス・ドライブ・5 7 7
5
- (72)発明者 マイケル・スコット・マッキルヴェイン
アメリカ合衆国・カリフォルニア・9 2 1 2 1・サン・ディエゴ・モアハウス・ドライブ・5 7 7
5
- (72)発明者 ブライアン・マイケル・ステンペル
アメリカ合衆国・カリフォルニア・9 2 1 2 1・サン・ディエゴ・モアハウス・ドライブ・5 7 7
5
- (72)発明者 メリンダ・ジェイ・ブラウン
アメリカ合衆国・カリフォルニア・9 2 1 2 1・サン・ディエゴ・モアハウス・ドライブ・5 7 7
5
- (72)発明者 ダレン・ユージン・ストリート
アメリカ合衆国・カリフォルニア・9 2 1 2 1・サン・ディエゴ・モアハウス・ドライブ・5 7 7
5

審査官 田中 幸雄

- (56)参考文献 特表2 0 1 0 - 5 4 1 1 0 6 (J P , A)
米国特許出願公開第2 0 0 2 / 1 2 9 2 2 6 (U S , A 1)
米国特許第6 5 5 0 0 0 3 (U S , B 1)

- (58)調査した分野(Int.Cl. , D B 名)
G 0 6 F 9 / 4 2