

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第5221546号
(P5221546)

(45) 発行日 平成25年6月26日 (2013. 6. 26)

(24) 登録日 平成25年3月15日 (2013. 3. 15)

(51) Int. Cl.

F I

G 0 6 F 9/38 (2006. 01)

G 0 6 F 9/38 3 3 O F

G 0 6 F 9/38 3 3 O A

請求項の数 31 (全 18 頁)

(21) 出願番号	特願2009-530609 (P2009-530609)	(73) 特許権者	595020643
(86) (22) 出願日	平成19年9月27日 (2007. 9. 27)		クアルコム・インコーポレイテッド
(65) 公表番号	特表2010-505210 (P2010-505210A)		Q U A L C O M M I N C O R P O R A T E D
(43) 公表日	平成22年2月18日 (2010. 2. 18)		アメリカ合衆国、カリフォルニア州 9 2
(86) 国際出願番号	PCT/US2007/079781		1 2 1 - 1 7 1 4、サン・ディエゴ、モア
(87) 国際公開番号	W02008/039947		ハウス・ドライブ 5 7 7 5
(87) 国際公開日	平成20年4月3日 (2008. 4. 3)	(74) 代理人	100108855
審査請求日	平成21年5月20日 (2009. 5. 20)		弁理士 蔵田 昌俊
(31) 優先権主張番号	11/535, 536	(74) 代理人	100109830
(32) 優先日	平成18年9月27日 (2006. 9. 27)		弁理士 福原 淑弘
(33) 優先権主張国	米国 (US)	(74) 代理人	100088683
前置審査			弁理士 中村 誠
		(74) 代理人	100103034
			弁理士 野河 信久

最終頁に続く

(54) 【発明の名称】 同時に予測された分岐命令を解決するための方法および装置

(57) 【特許請求の範囲】

【請求項 1】

予測された分岐命令の確認の前に同時に予測された分岐命令を解決する方法であって、
プロセッサにより、2つ以上の予測された分岐命令を処理することと、各予測された分岐命令はメモリに記憶された予測された状態および訂正された状態を有し、前記予測された分岐命令は、解決ステージに同時に入り、

前記プロセッサにより、前記メモリに記憶された前記予測された分岐命令の1つから前記訂正された状態の1つを選択することと、

前記プロセッサにより、前記予測された分岐命令の少なくとも1つが予測ミスされたことを決定することと、

前記プロセッサにより、前記分岐命令の少なくとも1つが予測ミスされたことを決定することに応答して、前記メモリに記憶された前記選択された訂正された状態に基づく先の命令フェッチを導くことと、

を含む方法。

【請求項 2】

前記予測された状態は、さらに分岐方向を含む請求項 1 に記載の方法。

【請求項 3】

前記予測された状態は、さらに前記プロセッサのモードを含む請求項 1 に記載の方法。

【請求項 4】

前記予測された状態は、さらにターゲットアドレスを含む請求項 1 に記載の方法。

【請求項 5】

前記選択された訂正された状態は、最も古い前記予測された分岐命令の 1 つに関連する請求項 1 に記載の方法。

【請求項 6】

前記選択された訂正された状態は、ランダムに選ばれる請求項 1 に記載の方法。

【請求項 7】

前記選択された訂正された状態は、最後に予測されたパイプラインに基づく請求項 1 に記載の方法。

【請求項 8】

前記選択された訂正された状態は、最後に予測ミスされたパイプラインに基づく請求項 1 に記載の方法。 10

【請求項 9】

前記選択された訂正された状態は、前記予測された分岐命令のタイプを基礎とする請求項 1 に記載の方法。

【請求項 10】

複数のパイプラインにおいて、予測された分岐命令の確認の前に、同時に予測された命令を解決する方法であって、

プロセッサにより、2 つ以上の予測された分岐命令を処理することと、各予測された分岐命令はメモリに記憶された予測された状態および訂正された状態を有し、前記予測された分岐命令は、別々のパイプラインの解決ステージに同時に入り、 20

前記プロセッサにより、前記メモリに記憶された前記予測された分岐命令の 1 つから前記訂正された状態の 1 つを選択することと、

前記プロセッサにより、前記予測された分岐命令の少なくとも 1 つが予測ミスされたことを決定することと、

前記プロセッサにより、前記分岐命令の少なくとも 1 つが予測ミスされたことを決定することに
応答して、前記メモリに記憶された前記選択された訂正された状態に基づいて先の命令フェッチを導くことと、

を含む方法。

【請求項 11】

前記予測された状態は、さらに分岐方向を含む請求項 10 に記載の方法。 30

【請求項 12】

前記予測された状態は、さらに前記プロセッサのモードを含む請求項 10 に記載の方法。
。

【請求項 13】

前記予測された状態は、さらにターゲットアドレスを含む請求項 10 に記載の方法。

【請求項 14】

前記選択された訂正された状態は、最も古い前記予測された分岐命令の 1 つに関連する請求項 10 に記載の方法。

【請求項 15】

前記選択された訂正された状態は、ランダムに選ばれる請求項 10 に記載の方法。 40

【請求項 16】

前記選択された訂正された状態は、最後に予測されたパイプラインに基づく請求項 10 に記載の方法。

【請求項 17】

前記選択された訂正された状態は、最後に予測ミスされたパイプラインに基づく請求項 10 に記載の方法。

【請求項 18】

前記選択された訂正された状態は、前記予測された分岐命令のタイプに基づく請求項 10 に記載の方法。

【請求項 19】

予測された分岐命令の確認の前に同時に予測された分岐命令を解決するシステムは、複数の分岐命令を予測するように構成された予測ロジックと、各予測された分岐命令は、メモリに記憶された予測された状態および訂正された状態を有しており、

2つ以上の予測された分岐命令が解決ステージにいつ同時に達したかを決定するように構成された解決ロジックであって、前記解決ロジックは、前記分岐命令の少なくとも1つが予測ミスされたとき、前記メモリに記憶された前記予測された分岐命令の1つから前記訂正された状態の1つを選び、前記分岐命令の少なくとも1つが予測ミスされたことを決定する、解決ロジックと、

前記分岐命令の少なくとも1つが予測ミスされたことを決定することに応答して、前記メモリに記憶された前記選択された訂正された状態に基づいて命令をフェッチするように構成されたフェッチロジックと、
を備えるシステム。

【請求項 20】

前記予測された状態は、さらに分岐方向を備える請求項 19 に記載のシステム。

【請求項 21】

前記システムは、プロセッサである請求項 19 に記載のシステム。

【請求項 22】

前記予測された状態は、さらに前記プロセッサのモードを備える請求項 20 に記載のシステム。

【請求項 23】

前記予測された状態は、さらにターゲットアドレスを備える請求項 19 に記載のシステム。

【請求項 24】

前記選択された訂正された状態は、最も古い前記予測された分岐命令の1つに関連する請求項 19 のシステム。

【請求項 25】

前記選択された訂正された状態は、ランダムに選ばれる請求項 19 に記載のシステム。

【請求項 26】

前記選択された訂正された状態は、最後に予測されたパイプラインに基づく請求項 19 に記載のシステム。

【請求項 27】

前記選択された訂正された状態は、最後に予測ミスされたパイプラインに基づく請求項 19 に記載のシステム。

【請求項 28】

前記選択された訂正された状態は、前記予測された分岐命令のタイプを基礎とする請求項 19 に記載のシステム。

【請求項 29】

前記先の命令フェッチを導くことは、

前記プロセッサにより、命令パイプラインの第1部分の第1のフラッシュを実行することと、

前記プロセッサにより、メモリに記憶された前記選択された訂正された状態に基づいて前記命令フェッチを開始することと、

を備え、前記方法はさらに、

前記第1のフラッシュを実行した後で、また、前記メモリに記憶された前記選択された訂正された状態に基づいて前記命令フェッチを開始した後で、予測ミスされた最も古い予測された分岐命令が前記選択された訂正された状態を有するかを決定することと、

予測ミスされた前記最も古い予測された分岐命令が前記メモリに記憶された前記選択された訂正された状態を有する場合、前記命令パイプラインの第2部分の第1のフラッシュを実行することと、ここにおいて、前記第2部分は前記第1部分とは異なる、

予測ミスされた前記最も古い予測された分岐命令が前記メモリに記憶された前記選択さ

10

20

30

40

50

れた訂正された状態を有さない場合には、

前記命令パイプラインの前記第 1 部分の第 2 のフラッシュを実行し、

予測ミスされた前記最も古い予測された分岐命令の前記メモリに記憶された前記選択された訂正された状態に基づいて、第 2 の命令フェッチを開始し、そして、

前記命令パイプラインの前記第 2 部分の第 2 のフラッシュを実行することと、

を備える、請求項 1 に記載の方法。

【請求項 3 0】

前記先の命令フェッチを導くことは、

前記プロセッサにより、命令パイプラインの第 1 部分の第 1 のフラッシュを実行することと、

前記プロセッサにより、前記メモリに記憶された前記選択された訂正された状態に基づいて前記命令フェッチを開始することと、

を備え、前記方法はさらに、

前記第 1 のフラッシュを実行した後で、また、前記メモリに記憶された前記選択された訂正された状態に基づいて前記命令フェッチを開始した後で、予測ミスされた最も古い予測された分岐命令が前記メモリに記憶された前記選択された訂正された状態を有するかを決定することと、

予測ミスされた前記最も古い予測された分岐命令が前記メモリに記憶された前記選択された訂正された状態を有する場合、前記命令パイプラインの第 2 部分の第 1 のフラッシュを実行することと、ここにおいて、前記第 2 部分は前記第 1 部分とは異なる、

予測ミスされた前記最も古い予測された分岐命令が前記メモリに記憶された前記選択された訂正された状態を有さない場合には、

前記命令パイプラインの前記第 1 部分の第 2 のフラッシュを実行し、

予測ミスされた前記最も古い予測された分岐命令の前記メモリに記憶された前記選択された訂正された状態に基づいて、第 2 の命令フェッチを開始し、そして

前記命令パイプラインの前記第 2 部分の第 2 のフラッシュを実行することと、

を備える、請求項 1 0 に記載の方法。

【請求項 3 1】

前記フェッチロジックは、

前記プロセッサにより、命令パイプラインの第 1 部分の第 1 のフラッシュを実行するように、

前記プロセッサにより、メモリに記憶された前記選択された訂正された状態に基づいて前記命令フェッチを開始するように、

構成され、前記システムはさらに、

前記フェッチロジックが前記第 1 のフラッシュを実行した後で、また、前記フェッチロジックが前記メモリに記憶された前記選択された訂正された状態に基づいて前記命令フェッチを開始した後で、予測ミスされた最も古い予測された分岐命令が前記メモリに記憶された前記選択された訂正された状態を有するかを決定するように、

予測ミスされた前記最も古い予測された分岐命令が前記メモリに記憶された前記選択された訂正された状態を有する場合、前記命令パイプラインの第 2 部分の第 1 のフラッシュを実行するように、ここにおいて、前記第 2 部分は前記第 1 部分とは異なる、

予測ミスされた前記最も古い予測された分岐命令が前記メモリに記憶された前記選択された訂正された状態を有さない場合には、

前記命令パイプラインの前記第 1 部分の第 2 のフラッシュを実行し、

予測ミスされた前記最も古い予測された分岐命令の前記メモリに記憶された前記選択された訂正された状態に基づいて、第 2 の命令フェッチを開始し、そして

前記命令パイプラインの前記第 2 部分の第 2 のフラッシュを実行するように、

構成された訂正ロジックを備える、請求項 1 9 に記載のシステム。

【発明の詳細な説明】

【技術分野】

10

20

30

40

50

【 0 0 0 1 】

本発明は、一般に、コンピュータシステムに関し、より具体的には、同時に予測された分岐命令を解決するための技術に関する。

【 背景技術 】

【 0 0 0 2 】

コンピュータプラットフォームの発展の基本は、プロセッサである。初期のプロセッサは、その時点で使用可能な技術に限定されていた。製造技術において新しい発展は、初期のプロセッサの 1 / 1 0 0 0 のサイズまで減少した、および、上回るトランジスタの設計を可能にする。これらのより小さいプロセッサの設計は、前の予想を上回り、さらに速く、さらに有効であり、処理パワーを出力している間に実質的により少ない電力を使用する。

10

【 0 0 0 3 】

プロセッサの物理的な設計が発展したため、情報を処理する、および、機能を実効する革新的な方法も変わった。例えば、命令の「パイプライン方式 (pipelining)」は、1960年代の初期からプロセッサ設計に実装されていた。パイプライン方式の一例は、命令が順次すばやく流れる間に実行パイプラインをユニットに切断する概念である。ユニットは、

いくつかのユニットがいくつかの命令の適切な部分を同時に処理することができるように設計されている。パイプライン方式の利点の 1 つは、命令が並列に評価されるため、命令の実行が重なっていることである。パイプライン方式は、命令レベル並列性 (instruction level parallelism) (I L P) とも呼ばれている。

20

【 0 0 0 4 】

プロセッサパイプラインは、多くのステージによって構成されている、ここで各ステージは命令を実行することと関連する機能を実行する。各ステージは、パイプステージ (pipe stage) またはパイプセグメント (pipe segment) と呼ばれている。このステージは、パイプラインを形成するために合わせて接続されている。命令は、パイプラインの一方の端から始まり、他方の端で終了する。

【 0 0 0 5 】

パイプライン処理は継続してプロセッサの設計に実装されているが、これは当初プロセッササイクル (processor cycle) ごとに 1 つの命令のみが実行するように制約されていた。プロセッサの処理する処理量を増加させるために、より最近のプロセッサの設計には、同時に複数の命令を処理することが可能な複数のパイプラインが実装されている。この複数のパイプラインを有するプロセッサのタイプは、スーパースカラプロセッサ (superscalar processor) として分類されることがある。

30

【 0 0 0 6 】

プロセッサ内において、条件付分岐命令などの特定の命令のタイプが予測されることがある。プロセッサ内の分岐予測ハードウェアは、条件付分岐命令の予測を提供するために設計されることがある。この予測に基づいて、プロセッサは、次の順次命令を実行することを継続するか、または、次の実行されるべき命令に導かれる。

【 0 0 0 7 】

分岐予測ハードウェアを使用するスーパースカラプロセッサは、同じ、または、別々のパイプラインにおいて、同じクロックサイクル内で 2 つ以上の予測された分岐命令を同時に生じさせ、解決することができる。一般に、このようなアプリケーションにおいて、プロセッサは、予測ミスの場合においていずれかの訂正のステップが行われる前に、最も古い予測ミスされている分岐を決定するために、両方の分岐予測の完全な解決を決定することができるまで待たなければならない。ここには、複数の分岐予測が生じる高速プロセッサのために、最も古い予測ミスされた分岐の決定から、訂正された状態の選択を分離する必要性がある。

40

【 発明の概要 】

【 0 0 0 8 】

50

本発明は、この必要性を認め、すべての同時分岐命令予測の完全な解決が可能な前に、適切な行動を予測し、適切な行動を実行することに向けてステップを踏むことによって、同時分岐命令予測の処理するプロセッサを開示する。

【0009】

予測された分岐命令の確認の前に同時に予測された分岐命令を解決する方法が開示されている。この方法は、最初に2つ以上の予測された分岐命令を処理することを備えている。各予測された分岐命令は、予測された状態と訂正された状態を有している。予測された分岐命令は、解決ステージに同時に入り、予測された分岐命令の1つから訂正された状態の1つが選択される。この方法は、予測された分岐命令の少なくとも1つが予測ミスされ、選択された訂正された状態が先の命令フェッチを導くために使用されることをさらに確認する。

10

【0010】

複数のパイプラインにおいて、予測された分岐命令の確認の前に同時に予測された分岐命令を解決する方法は、最初に2つ以上の予測された分岐命令を処理することを備えている。各予測された分岐命令は、予測された状態と訂正された状態を有する。予測された分岐命令は、別々のパイプラインの解決ステージに同時に入り、予測された分岐命令の1つから訂正された状態の1つが選択される。この方法は、予測された分岐命令の少なくとも1つが予測ミスされ、選択された訂正された状態が先の命令フェッチを導くために使用されることをさらに確認する。

【0011】

20

予測された分岐命令の確認の前に同時に予測された分岐命令を解決するシステムは、複数の分岐命令を予測するように構成された予測ロジックを備えている。各予測された分岐命令は、予測された状態と訂正された状態を有している。このシステムは、2つ以上の予測された分岐命令が解決ステージに同時に達したときに決定するように構成された解決ロジックをさらに備えている。この解決ロジックは、そして予測された分岐命令から1つから訂正された状態1つを選択する。このシステムは、選択された訂正された状態に基づいて先の命令をフェッチするように構成されたフェッチロジックを備えている。

【0012】

本発明のより完全な理解、ならびに発明のさらなる特徴および利点は、以下の詳細な説明および添付の図面から明らかになる。

30

【図面の簡単な説明】

【0013】

【図1】図1は、本発明の一実施形態を用いたプロセッサの高レベルロジックハードウェアブロック図を示す。

【図2】図2は、本発明の一実施形態を用いているスーパースカラプロセッサの低レベルロジックブロック図を示す。

【図3】図3は、図1のプロセッサのパイプラインにおいて解決ステージのフローチャートを示す。

【図4】図4は、図3の複数の同時分岐解決フローを示す。

【発明を実施するための形態】

40

【0014】

以下に説明される添付された図面に関連した詳細な説明は、本発明の種々の実施形態の説明を意図し、本発明で実施されることがある実施形態のみを表すことを意図していない。この詳細な説明は、本発明の全体を通した理解を提供する目的で特定の記述を含む。しかしながら、本発明がこれらの特定の記述なしで実施されることがあることは当業者にとって明らかである。ある例において、よく知られた構成および要素は、本発明の概念を不明瞭にすることを防ぐためにブロック図の形式で示されている。頭字語および他の記述的な専門語は、単に便宜および明瞭のために使用されることがあり、この発明の範囲の限定を意図していない。

【0015】

50

スーパースカラのプロセッサにおいて、プロセッサの内部のリソースは、並列な処理を容易にするように設計されている。設計のいくつかの様相は、事前フェッチ、分岐処理、レジスタ値を含むデータの依存性の解決、命令の開始などの命令を含む。プロセッサが大多数のメモリ装置より速く動作するため、プログラム命令は、プロセッサの可能性を最大限に適切に使用するのに、十分な速さでメモリから直接読み込むことができない。

【 0 0 1 6 】

命令キャッシュは、従来のメモリとプロセッサのスピードの差をうめるように設計された特別なメモリである。メモリからフェッチされた命令は、プロセッサクロックスピード (processor clock speed) で読み込むことが可能なより速い命令キャッシュに位置する。フェッチされた命令は、プログラムにおいて次の順次命令または予測が行われた分岐のターゲットであることがある。次の命令が予測された分岐のターゲットであるとき、プロセッサは、分岐がどこへ行くか予想し、前の適切な命令をフェッチするように試みる。分岐予測が誤っている場合、プロセッサは、その命令処理を、予測された分岐パスでフェッチした命令をパージすることによって訂正し、訂正された分岐パスで命令をフェッチすることを再び開始する。この処理は、さらに詳しく図 2、3 及び 4 の解説で説明される。

【 0 0 1 7 】

図 1 は、下に説明される 1 つの実施形態を使用するスーパースカラプロセッサ 1 0 0 の高レベルの図を示す。プロセッサ 1 0 0 は、確保された高速バス 1 0 4 を介して命令キャッシュ 1 0 6 に接続された中心処理ユニット (central processing unit) (CPU) 1 0 2 を備えている。この CPU は、データキャッシュ 1 0 8 に接続される他の分離された高速バス 1 1 0 を備えている。この命令キャッシュ 1 0 6 およびデータキャッシュ 1 0 8 は、汎用バス 1 1 6 を介して入力/出力ポート (I/O) 1 1 2 およびメモリ 1 1 4 とさらに接続される。

【 0 0 1 8 】

プロセッサ 1 0 0 内において、命令フェッチユニット (Instruction Fetch Unit) (IFU) 1 2 2 は、メモリ 1 1 4 から命令キャッシュ 1 0 6 への命令のロードを制御する。一度、命令キャッシュ 1 0 6 が命令によりロードされると、CPU 1 0 2 は、高速バス 1 0 4 を介して命令とアクセスすることが可能となる。命令キャッシュ 1 0 6 は、図 1 に示すように分離されたメモリ構造であることがあり、または、CPU 1 0 2 の内部の要素として内蔵されることもある。組み込みは、命令キャッシュ 1 0 6 のサイズならびに複雑性、および、CPU 1 0 2 の電力損失に基づいて決めることができる。

【 0 0 1 9 】

命令は、一度にいくつかの命令が命令キャッシュ 1 0 6 からフェッチされ、符号化されることがある。命令キャッシュ 1 0 6 内において、命令は、キャッシュラインとして知られているセクションにグループ分けされる。各キャッシュラインは、複数の命令を含んでいる。フェッチされた命令の数は、必要とされるフェッチの帯域幅ならびに各キャッシュラインの命令の数の数に依存することがある。1 つの実施形態において、CPU 1 0 2 は、各クロックサイクルの間に IFU 1 2 2 で命令キャッシュ 1 0 6 から上位のパイプライン 2 5 0 に 4 つの命令をロードする。上位のパイプライン 2 5 0 内において、命令は、演算タイプおよびデータ依存について解析される。命令を解析した後、プロセッサ 1 0 0 は、実行のために命令を上位のパイプ 2 5 0 から下位の機能的なユニットまたはパイプライン 2 1 0 および 2 2 0 に分散することができる。

【 0 0 2 0 】

命令は、命令機能、パイプラインの利用可能度、命令キャッシュ 1 0 6 からロードされた命令のグループ内の命令位置などに依存して下位のパイプライン 2 1 0 または 2 2 0 に送信されることがある。下位のパイプライン 2 1 0 および 2 2 0 内において、命令は、オリジナルプログラムシーケンス (original program sequence) ではなく利用可能なリソースに基づいて並列に処理される。この処理のタイプは、ダイナミック命令スケジューリング (dynamic instruction scheduling) とよく呼ばれている。

【 0 0 2 1 】

10

20

30

40

50

下位のパイプライン 210 および 220 は、算術ロジックユニット、フローティングポイントユニット、記憶ユニット、ロードユニットなどの種々の実行ユニット (Execution Units) (EU) 118 を含むことができる。例えば、算術ロジックユニットなどの EU 118 は、整数加算、減算、単純乗算、ビット単位ロジック演算 (例えば、AND、NOT、OR、XOR)、ビットシフティングなどの広範囲の算術機能実行することができる。命令が実行を終了した後、CPU 102 は、命令の結果が適正にプロセッサ 100 を更新するために使用できるように、命令の結果およびレコーダそれらを適正なシーケンスに加える。

【0022】

プロセッサ 100 によって実行された大多数のプログラムは、条件付分岐命令を含むことができる。条件付分岐命令の実際の分岐行動は、命令が下位のパイプライン 210 または 220 において深部で実行されるまで知られない。分岐命令の最後の実行を待つことによって生じるストールおよび次に分岐命令の結果に基づいて命令をフェッチすべきことを防ぐために、プロセッサ 100 は、分岐予測のいくつかの形式を使用することができる。分岐予測を使用することによって、プロセッサ 100 は、上位のパイプ 250 において条件付分岐命令の分岐行動を予測することができる。予測された分岐評価に基づいて、プロセッサ 100 は、推論的にフェッチし、予測されたアドレス - 分岐ターゲットアドレス (例えば、分岐の予測が行われている場合) または分岐命令の後の次のくるべきアドレス (例えば、分岐の予測が行われていない場合) のいずれかからの命令を実行するために準備する。

【0023】

条件付分岐命令の 1 つの例は、単純アセンブラ命令等しくなければジャンプ (jump not equal) (JNE) である。JNE 命令が実行されたとき、特定の値は、レジスタにロードされることがあり、値がゼロ (zero) に等しい場合、条件付命令は行われず、シーケンスにおいて次の命令がフェッチされ実行される。しかしながら、レジスタにおいて値がゼロと等しくない場合、条件付分岐は、考慮のうえで行われ、次のフェッチされた命令は、JNE 命令に関連したターゲットアドレスに位置する。ターゲットアドレスは、前の JNE 命令の実行で前に JNE 命令に関連されていてよい。

【0024】

命令を予測するとき、いくつかの条件または「状態」が予測されることがある。例えば、分岐方向、ターゲットアドレス、またはプロセッサモードなどのプロセッサ 100 の特定の状態が予測されることがある。プロセッサモードを予測することは、分岐命令の実行の後に、プロセッサ 100 がどのモードにいるか予測することを必要とする。例えば、進化した RISC プロセッサアーキテクチャにおいて、命令は、ARM モードまたはサム (Thumb) モードのいずれかで実行されることがある。

【0025】

条件付分岐の方向を予測する可能な 1 つの方法は、分岐活動記録テーブル (branch history table) を使用することである。分岐活動記録テーブルは、過去の分岐の数の活動記録を記憶する単純ルックアップテーブルであることがある。1 つの分岐活動記録は、過去の条件付分岐の 1024 方向を記憶することができる。複合演算は、予測技術の階層に基礎付けられた分岐予測を作成するために書き込まれることがある (複数のレベル分岐プレディクタ (predictors))。

【0026】

図 2 は、プロセッサ 100 内の上位のパイプ 250 および 2 つの下位のパイプライン 210 ならびに 220 の下位レベルの機能的なブロック図 200 を示し、本発明の 1 つの態様に従った命令の処理を示す。機能的なハードウェアブロック図 200 内の異なるロジックブロック (またはステージ) は、ハードウェア、ファームウェア、または両方の組み合わせを含むことができる。機能的なハードウェアブロック図 200 は、上位のパイプ 250 および 2 つの下位のパイプライン 210 ならびに 220 によって構成されている。以前に言及したように、上位のパイプ 250 のステージは、IFU 122 内に存在する。上位

10

20

30

40

50

のパイプ 250 内には、フェッチステージ 202、命令キャッシュステージ 204、命令符号化ステージ 206 がある。さらに上位のパイプ 250 と関連するのは、分岐予測ロジック 208 である。

【0027】

上位のパイプ 250 において第 1 ステージは、フェッチステージ 202 である。フェッチステージ 202 は、検索され取り出されるべき命令の次のグループの選択を制御する。プロセッサ 100 が起動すると、フェッチステージ 202 は、検索され取り出され、ロードされるべき開始命令を決定する。図 3 および図 4 の検討に関連して説明されているが、フェッチステージ 202 は、下位のパイプライン 210 および 220 からのフィードバックを受信することもできる。フィードバックは、先の命令の選択および命令がどのような順序で実行されるかに影響することがある。

10

【0028】

命令キャッシュステージ 204 において、フェッチステージ 202 の間で選択された命令アドレスは、そのアドレスに命令が存在しているか決定するように命令キャッシュ 106 をアクセスするために使用される。命令キャッシュヒットがある場合、CPU 102 は、命令を命令キャッシュ 106 から上位のパイプ 250 へと検索して取り出し、プロセッサ 100 がメモリ 114 に戻ることなく、プロセッサスピードでフェッチ命令することを可能にする。命令キャッシュにミスがある場合（例えば、命令キャッシュ 106 からフェッチされるべき命令が使用不可能な場合）、IFU 122 は、メモリ 114 から命令を検索して取り出し、命令キャッシュ 106 にそれらをロードし、CPU 102 にそれらを転送する。命令キャッシュステージ 204 の間に命令が検索して取り出された後、命令は、命令符号化ステージ 206 の間に解析される。

20

【0029】

命令符号化ステージ 206 の間に種々の命令に係する情報は、解析され、処理される。例えば、命令符号化ステージ 206 内で、プロセッサ 100 は、命令のタイプ（例えば、アクション、記憶、ロード、ジャンプなど）を決定する。命令が条件付分岐命令の場合、分岐予測ロジック 208 は、引き起こされる。命令符号化ステージ 206 は、分岐予測ロジック 208 が分岐命令を生じされたことを通知し、分岐予測ロジック 208 と通信する。

【0030】

30

分岐予測の部分として、分岐予測ロジック 208 は、予測された状態を提供する。予測された状態に記憶された情報は、予測された分岐方向、予測されたターゲットアドレス、またはプロセッサ 100 の予測された状態を含むことができる。この情報は、レジスタ、レジスタのグループ、分岐命令と関連するメモリ位置に記憶されることがある。本発明の 1 つの態様において、予測された状態は、予測された分岐方向のみ含むことができる。他の実施形態において、予測された状態は、予測された分岐方向および予測されたターゲットアドレスにのみ関連する情報を含むことができる。更なる実施形態において、予測された状態は、予測されたターゲットアドレスおよび予測されたプロセッサモードに関連する情報を含むことができる。なお、更なる実施形態において、予測された状態は、予測された分岐方向、予測されたターゲットアドレス、および予測されたプロセッサモードを含むことができる。

40

【0031】

分岐方向が予測されたとき、予測された状態は、分岐の予測が行われたか、または、行われていないかを予測する情報を含んでいる。1 つの実施形態において、予測された状態は、単一のビットであることがある。例えば、予測された状態に関連したレジスタまたはメモリ位置内で、ビット位置において記憶された「1」は、予測が行われたと分岐を示す。逆に、レジスタまたはメモリ位置内のビット位置に「0」が記憶されている場合、分岐が行われていないと予測されることがある。

【0032】

ターゲットアドレスが予測された場合、予測された状態は、フェッチされる次の命令の

50

位置を示すターゲットアドレスを含むことができる。ターゲットアドレスのサイズは、プロセッサ 100 のアーキテクチャに依存することができる。1つの実施形態において、ターゲットアドレスは、予測された状態に関連したレジスタにおいて記憶された 32 ビットアドレス識別子であることがある。

【0033】

プロセッサモードが予測されたとき、予測された状態は、一度条件付分岐が実行されると、プロセッサ 100 が所属される予測されたモードに関連する情報を含むことができる。

【0034】

例えば、プロセッサモードの予測に基づいて、プロセッサは、命令符号化を異なるように実行する行動をとることができる（つまり、命令の ARM 符号化対命令のサム符号化）。プロセッサモードの予測された状態は、レジスタおよびメモリ位置に記憶された単一のビットの値であることがある。

【0035】

予測された状態を補って完全にすると、分岐予測ロジック 208 は、さらに予測された分岐命令に関連する訂正された状態を計算し、記憶する。訂正された状態は、予測が誤った場合の情報を含む。訂正された状態の一部として記憶された情報は、回復アドレスおよびプロセッサの前の状態を含むことができる。訂正された状態は、分岐の予測ミスの場合の適切な命令の順序のシーケンスを回復するために、プロセッサ 100 によって使用されることがある。

【0036】

分岐予測ロジック 208 によってつくられた予測の結果として、情報は、次の命令フェッチを導くように、フェッチステージ 202 内のロジックをフェッチするために提供される。予測された状態は、予測に基づく適切な命令を検索し取り出すために、フェッチロジックによって使用される。例えば、予測された状態がターゲットアドレスを含んでいる場合、フェッチロジックは、ターゲットアドレスに位置する命令キャッシュ 106 から次の命令を検索取り出す。命令キャッシュ 106 において命令が利用可能でない場合、フェッチロジックは、メモリ 114 から命令キャッシュ 104 へ命令をロードし、そして命令を上位のパイプ 250 にロードする。

【0037】

初期の分岐予測が解決される前に、予測を要求する他の分岐命令に出くわすことは、まれである。この例において、プロセッサ 100 は、分岐予測ロジック 208 によって実行された予測ごとのトラックを維持する。このトラックリングは、最初に届いた予測を識別することを含む。予測の「時期」をトラックリングする 1つの方法は、各条件付分岐命令に関連する命令順序値を使用することである。各予測された状態が割り当てられると、命令順序値も割り当てられ、記憶され、または分岐命令とともに搬送される。一度、予測ロジックが予測を実行すると、または、命令符号化ステージ 206 において現在の命令が予測を要求していないと決定すると、命令は適切な下位のパイプライン 210 および 220 に伝わる。

【0038】

前に記載したように、下位のパイプライン 210 および 220 は、命令の特定のタイプと関連することがある。例えば、パイプラインは、算術本質の命令を実行するため、または、ロード / 記憶機能のすべてを対応するためのみに設計されることがある。パイプラインに予測された分岐命令を送信するために、パイプラインは、分岐命令に対応するように設計されていなければならない。図 2 に示すように、両方の下位のパイプライン 210 および 220 は、分岐命令に対応するように構成されている。下位のパイプライン 210 および 220 は、各プロセッササイクルの間に複数の命令を実行するように設計されていることもある。したがって、下位のパイプライン 210 および 220 内で、複数の分岐命令は、同じプロセッササイクルの間に実行されることがある。

【0039】

一度、命令が適切な下位のパイプライン 210 または 220 に入ると、分岐命令などの命令は、より効果的な実行を容易にするために並び換えられることがある。分岐命令が下位のパイプライン 210 または 220 に達し、実行を持続させるために更なる情報またはデータを要求する場合、プロセッサ 100 は、分岐命令を実行する前に、他の命令または命令のグループを実行することができる。この場合において、分岐命令は、実行を用意するために必要な情報が利用可能となる前に、リザーベーションステーション (reservation station) (図示しない) に保有されることがある。例えば、分岐命令は、分岐命令が特定のレジスタに記憶されたターゲットアドレスに分岐され、ターゲットアドレスがまだ利用可能でない場合、リザーベーション局に保有される。ターゲットアドレスの値は、他の次の実行された命令として決定されることがある。分岐命令は、次の命令が実行されるまで保有され、特定のレジスタを更新し、ターゲットアドレスが利用可能となる。ターゲットアドレスが利用可能となった後、分岐命令は、更なる実行のために開放される。このような方法で実行された命令は、従来のプログラムシーケンスではなく、利用可能なリソースに基づいて並列に実行される。下位のパイプライン 210 および 220 において命令が実行された後、結果は、プロセッサ 100 が正しくに更新されるように、適切なシーケンスに集められ、再び配列される。

10

【0040】

リザーベーション局内で、いくつかの命令は、同じ時間に保有され、各命令は、更なる情報、プロセッサリソースなどを待っている。一般的に、複数の命令は、プロセッサ 100 によって、同じプロセッササイクルの間にリザーベーション局から開放されることがある。したがって、複数の分岐命令が同じプロセッササイクルの間にリザーベーション局から開放されることは、可能である。

20

【0041】

プロセッサ 100 は、下位のパイプライン 210 および 220 においてそれらが実行されているため、命令を監視することを継続する。分岐命令がリザーベーション局から開放されるとき、または、最後の実行する準備ができたとき、分岐命令と関連する予測の処理は、各下位のパイプライン 210 および 220 の解決ステージ 215 内で、解決ロジック 225 によって実行される。解決ステージ 215 は、図 3 の記載と関連して説明される。

【0042】

解決ロジック 225 は、予測された状態の正しさを確認し、予測ミスの事象において訂正された状態を選択する。例えば、予測された状態がターゲットアドレスであり、ターゲットアドレスが決定された適切なターゲットアドレスと一致しない場合、予測ミスが生じる。予測ミスの場合において、解決ロジック 225 は、フェッチステージ 202 に、訂正された状態ならびにフラッシュされる必要がある命令を識別する情報を含むフィードバックを提供する。フラッシュされた命令は、誤った予測に基づいて前にフェッチされた命令である。適切な命令がフラッシュされた後、フェッチロジックは、訂正された状態に基づいた命令を再びフェッチすることを開始する。解決ロジック 225 が予測が正しいと決定する場合、何も行動が行われず、フェッチされた不確かな命令 (予測に基づいて) が次に実行される。

30

【0043】

図 3 は、下位のパイプライン 210 および 220 のいずれかの解決ステージ 215 に関連した処理のフロー 300 を説明するフローチャートである。処理のフロー 300 は、予測された分岐命令がすべてのそれらの依存性を解決したとき、開始ブロック 302 から始まる。分岐依存性は、予測を基礎とされたオペランドが利用可能なとき、解決する。この解決は、解決ステージ 215 において、下位のパイプライン 210 および 220 のいずれかで生じる。

40

【0044】

判断ブロック 312 で、分岐解決ステージ 215 に同時に入った複数の予測分岐があるかどうかを判断する。前に記載したように、複数の分岐命令は、同じ下位のパイプライン 210 または 220 の同じプロセッササイクルの間に解決ステージ 215 に入ることがで

50

きる。本発明の１つの態様は、同じ下位のパイプライン２１０または２２０において、複数の分岐命令を解決する。別の実施形態において、下位のパイプライン２１０において、解決ステージ２１５に入る分岐命令、下位のパイプライン２２０の解決ステージ２１５に入る分岐命令は、同時に解決することがある。プロセッサ１００がさらに下位のパイプラインを有している場合には、本発明の他の実施形態が下位のパイプラインの１つにおいて、複数の分岐予測を解決することができる。

【００４５】

プロセッサ１００は、このアセスメントを作るために、両方の下位のパイプライン２１０および２２０を監視する。プロセッサ１００が複数の分岐命令が解決ステージ２１５に同時に入ったと判断する場合、処理フロー３００は、複数の同時分岐解決フロー３２０に導かれる。複数の同時分岐解決フロー３２０は、分岐解決ステージ２１５に同時に入った２つ以上の予測された分岐命令をどのように解決するか決定し、図４に関連してさらに説明される。解決ステージ２１５に１つの分岐命令のみが入ってきた場合、処理フロー３００は、判断ブロック３１４に続く。

10

【００４６】

判断ブロック３１４で、分岐予測の結果は、分岐が正しく予測されているか決定するために解析される。判断ブロック３１４で、条件付分岐命令の予測が正しく予測されている場合（例えば、予測された条件付分岐命令のターゲットアドレスが解決されたターゲットアドレスと一致する）、上位のパイプ２５０ならびに下位のパイプライン２１０および２２０において、残っている命令は、正しく予測され、処理フロー３００は、そして終了ブロック３５０へと導かれる。

20

【００４７】

判断ブロック３１４で、分岐予測結果が予測ミスが生じたことを示す場合（例えば、予測ターゲットアドレスが解決ターゲットアドレスと一致しない）、予測ミスの命令より新しいすべての命令（分岐予測の命令順序値に基づく）は、ブロック３１６に示されるように、上位のパイプ２５０、下位のパイプライン２１０および２２０にフラッシュされる。処理フロー３００は、フェッチステージ２０２内で分岐の訂正された状態の情報がフェッチロジックに送られるブロック３１８に進む。フェッチロジックは、分岐の訂正された状態に基づいて命令をフェッチする。

【００４８】

図４は、複数の同時分岐解決処理フロー３２０をさらに詳しく示す。複数の同時分岐解決処理フロー３２０は、同じプロセッササイクルの間に２つ以上の予測された分岐が解決ステージ２１５に達したときに始まる。２つ以上の予測分岐が解決時期２１５に同時に入ったとき、プロセッサ１００は、同じプロセッササイクルの間に両方の分岐の解決をハンドルする。この解決は、分岐が予測ミスされたかどうかの決定を含み、パイプラインを再び導くなどの適切な行動を行う。

30

【００４９】

図４に示すように、複数の同時分岐解決処理フロー３２０は、最初に、ブロック４０２で解決されている分岐の訂正された状態の１つを選ぶ。予測ミスが生じたかどうかを複数の同時分岐解決処理フロー３２０が決定する前に、訂正された状態の１つの選択が生じる。複数の同時分岐解決処理フロー３２０の初期において、訂正された状態の１つ選択することによって、予測ミスの予期で更なる時間が保管されることがある。予測ミスが生じていない場合、この選択によって、更なる処理時間の損失はない。

40

【００５０】

特定の分岐の訂正された状態の選択は、いくつかの要因に基づくことが可能である。１つの実施形態において、訂正された状態の選択は、解決分岐の相対的な時期に基づき、最も古い分岐命令が選択される。別の実施形態において、訂正された状態の選択は、下位のパイプライン２１０および２２０の最近の予測ミスに基づくことができる。さらに他の実施形態において、選択は、最後の予測が届いた下位のパイプライン２１０または２２０に基づくことができる。さらなる実施形態において、命令のタイプは、訂正された状態を選

50

択する基礎として使用されることがある。一方、訂正された状態は、ランダムに選ばれることがある。選択の処理にかかわらず、選択された訂正された状態は、予測ミスの場合において、先の命令フェッチのフェッチロジックを操作するために使用されることがある。

【0051】

訂正された状態の選択は、プロセッサ100の実行スピードに影響があることがある。プロセッサの設計によるが、各前述の態様は、特定のスピードの利点がある。例えば、下位のパイプライン210または220に基づいて訂正された状態を選ぶことは、最後に予測ミスされた下位のパイプライン210または220の決定よりさらに速く実行することができる。訂正された状態の確実性は、複数の同時分岐解決処理フロー320において、後で確認される。

10

【0052】

前述した実施形態を使用するプロセッサ100の1つの利点は、プロセッサ100が最も古い分岐予測が予測ミスされたことを予測することである。したがって、プロセッサ100は、単一のサイクルにおいて、分岐が予測ミスされたことを厳密に決定するために必要とされる余分な時間を考慮するためのプロセッサの回数を減少する代わりに、上位のパイプ250および下位のパイプライン210ならびに220をフラッシュされるなどの適当な行動を実行する。プロセッサ100が訂正された予測ミス分岐を選ぶ場合、最も古い予測ミス分岐が対応する分岐予測を決定するために待つ前に、プロセッサ100が命令をフラッシュし、再びフェッチすることを開始するため、より高いクロック周波数を実現することができる。プロセッサ100にもたらす増加したクロックレートは、間違った訂正された状態を選んでいるため、どのクロックサイクルの損失より上回っている。

20

【0053】

ブロック402で訂正された状態の情報が選択された後、複数の同時分岐解決処理フロー320は、判断ブロック404に進む。判断ブロック404で、解決ロジック225は、任意の分岐命令の予測ミスが生じたかどうか決定する。判断ブロック404で、予測ミスが生じていない場合、複数の同時分岐解決処理フロー320は、ブロック450で終了する。この例において、両方の予測が訂正され、命令がロードされた上位のパイプ250および下位のパイプライン210および220は有効であり、訂正は必要ではない。

【0054】

予測ミスが生じた場合、複数の同時分岐解決処理フロー320は、上位のパイプ250においてすべての命令がフラッシュされるブロック406に進む。上位のパイプ250において、すべての命令がプログラムの順序にあるため、これらはいずれの解決分岐より新しい。上位のパイプ250の命令がいずれの解決分岐命令より新しいため、これらは予測ミスパスにフェッチされ、フラッシュされる。

30

【0055】

ブロック406で上位のパイプ250がフラッシュされた後、複数の同時分岐解決処理フロー320は、ブロック408に続く。ブロック408で、フェッチロジックは、上位のパイプ250においてフェッチすることを再び導くために、選択された分岐の訂正された状態を使用する。複数の同時分岐解決処理フロー320は、判断ブロック410に続く。判断ブロック410で、訂正された状態の情報の選択は、予測ミス分岐命令が訂正された状態が選択された分岐命令に関連するか調べることによって確認される。判断ブロック410で確認が成功した場合、ブロック418でプロセッサ100は、命令順序値に基づいて、下位のパイプライン210および220からのより最近の命令をフラッシュする。ブロック418の結果、複数の同時分岐解決処理フロー320は、終了し、ブロック450に進む。

40

【0056】

ブロック410で確認が失敗した場合（つまり、最も古い予測ミス分岐が選択されない）、すべての命令は、ブロック412で上位のパイプ250から再度フラッシュされる。ブロック412での命令のフラッシュは、ブロック408で上位のパイプ250からフェッチされた命令を効果的に移動する。第2の分岐の訂正された状態情報は、そしてブロッ

50

ク 4 1 4 でフェッチロジックに送られる。フェッチロジックは、第 2 分岐の訂正された状態に基づいて命令をフェッチする。第 2 の分岐命令（命令順序値に基づく）より新しいすべての命令は、ステップ 4 1 6 で下位のパイプライン 2 1 0 および 2 2 0 からフラッシュされる。下位のパイプライン 2 1 0 および 2 2 0 から命令がフラッシュされた後、ステップ 4 5 0 で複数の同時分岐解決処理フロー 3 2 0 は終了する。

【 0 0 5 7 】

本明細書に含まれた実施形態に関連して説明される例示的なロジックブロック、モジュール、回路、構成部品、および／または構成要素は、汎用プロセッサ、ディジタル信号プロセッサ（DSP）、特定用途向け集積回路（ASIC）、フィールドプログラマブルゲートアレイ（FPGA）もしくは他のプログラミング可能なロジックデバイス、ディスクリートのゲートもしくはトランジスタロジック、ディスクリートのハードウェア構成要素、あるいは本明細書で説明される機能を実行するように設計された任意の組み合わせを使用して、実施、または実行されることがある。汎用プロセッサは、マイクロプロセッサであることがあるが、代替として、プロセッサは、任意の従来のプロセッサ、コントローラ、マイクロコントローラ、または状態マシンであってもよい。プロセッサは、コンピューティングデバイスの組み合わせとして、例えば、DSP とマイクロプロセッサの組み合わせ、複数のマイクロプロセッサ、DSP コア と連携する 1 つ以上のマイクロプロセッサ、または他の任意のそのような構成として実施されることがある。

【 0 0 5 8 】

本明細書において特定の実施形態が図示され、記述されているが、当業者は、同じ効果を実現するために意図された任意の機構が、示された特定の実施形態の代わりに用いられること、および発明が他の環境において他の適用例があることが分かる。

【 0 0 5 9 】

この応用例は、本発明の任意の適用例または変形例に及ぶことを意図する。下記の特許請求の範囲は、発明の範囲を本明細書で説明された特定の実施例に限定することを全く意図していない。

以下に本件出願当初の特許請求の範囲に記載された発明を付記する。

[1] 予測された分岐命令の確認の前に同時に予測された分岐命令を解決する方法であって、

2 つ以上の予測された分岐命令を処理することと、各予測された分岐命令は予測された状態および訂正された状態を有し、前記予測された分岐命令は、解決ステージに同時に入り、

前記予測された分岐命令の 1 つから前記訂正された状態の 1 つを選択することと、
予測ミスされた前記予測された分岐命令の少なくとも 1 つを決定することと、
前記選択された訂正された状態に基づく先の命令フェッチを導くことと、
を含む方法。

[2] 前記予測された状態は、さらに分岐方向を含む [1] に記載の方法。

[3] 前記予測された状態は、さらにプロセッサのモードを含む [1] に記載の方法。

[4] 前記予測された状態は、さらにターゲットアドレスを含む [1] に記載の方法。

[5] 前記選択された訂正された状態は、最も古い前記予測された分岐命令の 1 つに関連する [1] に記載の方法。

[6] 前記選択された訂正された状態は、ランダムに選ばれる [1] に記載の方法。

[7] 前記選択された訂正された状態は、最後に予測されたパイプラインに基づく [1] に記載の方法。

[8] 前記選択された訂正された状態は、最後に予測ミスされたパイプラインに基づく [1] に記載の方法。

[9] 前記選択された訂正された状態は、前記予測された分岐命令のタイプを基礎とする [1] に記載の方法。

[1 0] 複数のパイプラインにおいて、予測された分岐命令の確認の前に、同時に予測された命令を解決する方法であって、

10

20

30

40

50

2つ以上の予測された分岐命令を処理することと、各予測された分岐命令は予測された状態および訂正された状態を有し、前記予測された分岐命令は、別々のパイプラインの解決ステージに同時に入り、

前記予測された分岐命令の1つから前記訂正された状態の1つを選択することと、
予測ミスされた前記予測された分岐命令の少なくとも1つを決定することと、
前記選択された訂正された状態に基づいて先の命令フェッチを導くことと、
を含む方法。

[1 1] 前記予測された状態は、さらに分岐方向を含む [1 0] に記載の方法。

[1 2] 前記予測された状態は、さらにプロセッサのモードを含む [1 0] に記載の方法。

[1 3] 前記予測された状態は、さらにターゲットアドレスを含む [1 0] に記載の方法。

[1 4] 前記選択された訂正された状態は、最も古い前記予測された分岐命令の1つに関連する [1 0] に記載の方法。

[1 5] 前記選択された訂正された状態は、ランダムに選ばれる [1 0] に記載の方法。

[1 6] 前記選択された訂正された状態は、最後に予測されたパイプラインに基づく [1 0] に記載の方法。

[1 7] 前記選択された訂正された状態は、最後に予測ミスされたパイプラインに基づく [1 0] に記載の方法。

[1 8] 前記選択された訂正された状態は、前記予測された分岐命令のタイプに基づく [1 0] に記載の方法。

[1 9] 予測された分岐命令の確認の前に同時に予測された分岐命令を解決するシステムは、

複数の分岐命令を予測するように構成された予測ロジックと、各予測された分岐命令は、予測された状態および訂正された状態を有しており、

2つ以上の予測された分岐命令が解決ステージに同時に達したときに決定するように構成された解決ロジックと、前記解決ロジックは、前記分岐命令の少なくとも1つが予測ミスされたとき、前記予測された分岐命令の1つから前記訂正された状態の1つを選ぶ、

前記選択された訂正された状態に基づいて命令をフェッチするように構成されたフェッチロジックと、

を備えるシステム。

[2 0] 前記予測された状態は、さらに分岐方向を備える [1 9] に記載のシステム。

[2 1] 前記システムは、プロセッサである [1 9] に記載の方法。

[2 2] 前記予測された状態は、さらに前記プロセッサのモードを備える [2 0] に記載のシステム。

[2 3] 前記予測された状態は、さらにターゲットアドレスを備える [1 9] に記載のシステム。

[2 4] 前記選択された訂正された状態は、最も古い前記予測された分岐命令の1つに関連する [1 9] のシステム。

[2 5] 前記選択された訂正された状態は、ランダムに選ばれる [1 9] に記載の方法。

[2 6] 前記選択された訂正された状態は、最後に予測されたパイプラインに基づく [1 9] に記載の方法。

[2 7] 前記選択された訂正された状態は、最後に予測ミスされたパイプラインに基づく [1 9] に記載の方法。

[2 8] 前記選択された訂正された状態は、前記予測された分岐命令のタイプを基礎とする [1 9] に記載の方法。

10

20

30

40

【図 1】

図 1

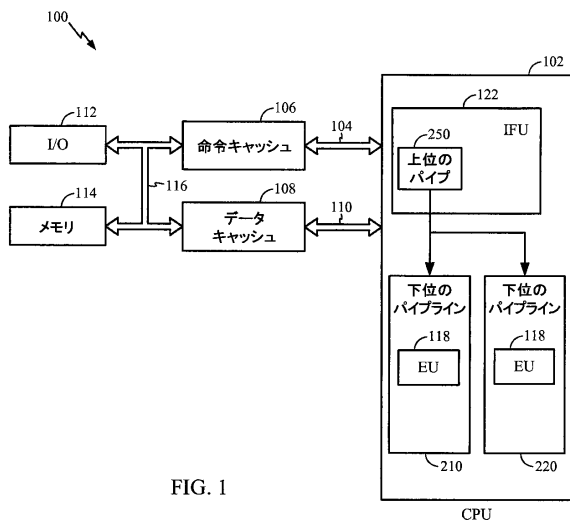


FIG. 1

【図 2】

図 2

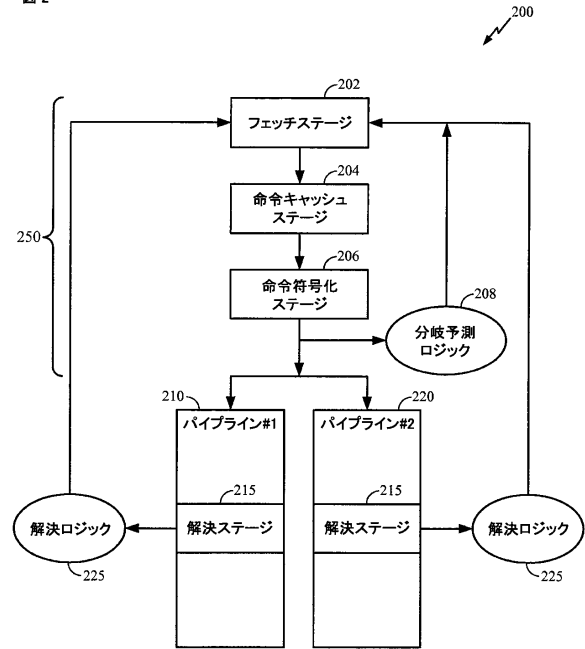


FIG. 2

【図 3】

図 3

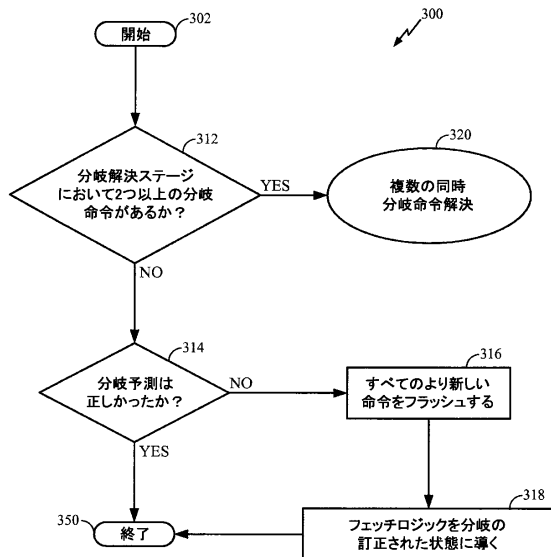


FIG. 3

【図 4】

図 4

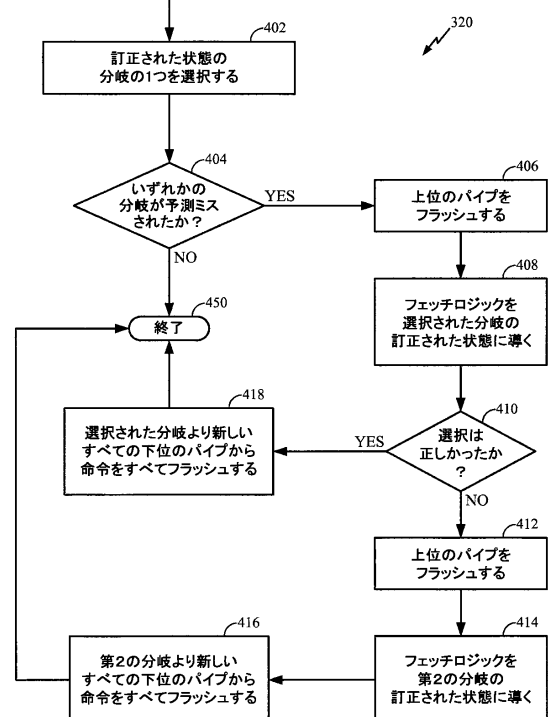


FIG. 4

 フロントページの続き

- (74)代理人 100095441
弁理士 白根 俊郎
- (74)代理人 100075672
弁理士 峰 隆司
- (74)代理人 100119976
弁理士 幸長 保次郎
- (74)代理人 100153051
弁理士 河野 直樹
- (74)代理人 100140176
弁理士 砂川 克
- (74)代理人 100158805
弁理士 井関 守三
- (74)代理人 100172580
弁理士 赤穂 隆雄
- (74)代理人 100179062
弁理士 井上 正
- (74)代理人 100124394
弁理士 佐藤 立志
- (74)代理人 100112807
弁理士 岡田 貴志
- (74)代理人 100111073
弁理士 堀内 美保子
- (74)代理人 100134290
弁理士 竹内 将訓
- (72)発明者 スミス、ロドニー・ウェイン
アメリカ合衆国、カリフォルニア州 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 ステムベル、ブライアン・マイケル
アメリカ合衆国、カリフォルニア州 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 ディーフェンダーファー、ジェームズ・ノリス
アメリカ合衆国、カリフォルニア州 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 サートリウス、トマス・アンドリュウ
アメリカ合衆国、カリフォルニア州 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 青木 重徳

- (56)参考文献 特開平 1 0 - 3 3 3 9 0 8 (J P , A)
特開平 1 0 - 1 3 3 8 7 4 (J P , A)
特開平 1 0 - 1 7 1 6 5 2 (J P , A)
特開平 0 9 - 0 0 6 6 1 2 (J P , A)
特開平 0 7 - 0 5 6 7 3 5 (J P , A)
中西知嘉子, 安藤秀樹, 町田浩久, 中屋雅夫, “ スーパスカラ・プロセッサ - S A R C H - のコ
ード・スケジューラ ” , 電子情報通信学会技術研究報告, 日本, 社団法人電子情報通信学会, 1
9 9 2 年 1 0 月 2 2 日, V o l . 9 2 , N o . 2 8 9 , p . 3 3 - 4 0
J. H. Pomerene and R. N. Rechtschaffen, “ REDUCTION OF BRANCH DELAY BY OUT-OF-SEQUENCE

DECODING UNDER CONTROL OF A DECODE HISTORY TABLE ” , IBM Technical Disclosure Bulletin
 , 米国 , IBM Corp. , 1 9 8 2 年 9 月 , vol.25, no.4 , p.2141

(58)調査した分野(Int.Cl. , D B 名)

G 0 6 F 9 / 3 8