

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第3715057号
(P3715057)

(45) 発行日 平成17年11月9日(2005.11.9)

(24) 登録日 平成17年9月2日(2005.9.2)

(51) Int.Cl.⁷

G06F 9/38

F I

G06F 9/38 380X

G06F 9/38 310F

請求項の数 3 (全 22 頁)

(21) 出願番号	特願平9-27423	(73) 特許権者	398038580
(22) 出願日	平成9年2月12日(1997.2.12)		ヒューレット・パッカード・カンパニー
(65) 公開番号	特開平9-244895		HEWLETT-PACKARD COMPANY
(43) 公開日	平成9年9月19日(1997.9.19)		アメリカ合衆国カリフォルニア州パロアルト
審査請求日	平成16年1月26日(2004.1.26)		ハノーバー・ストリート 3000
(31) 優先権主張番号	609,807	(74) 代理人	100081721
(32) 優先日	平成8年3月1日(1996.3.1)		弁理士 岡田 次生
(33) 優先権主張国	米国 (US)	(72) 発明者	グレッグ・レサルター
			アメリカ合衆国80525コロラド州フォート・コリンズ、エス・ギャラクシー 524
		審査官	後藤 彰
			最終頁に続く

(54) 【発明の名称】 パニック・トラップ・システム

(57) 【特許請求の範囲】

【請求項1】

プロセッサ内ソフトウェアの順序外命令の実行で生じた不正確な結果から回復するためのシステムであって、

前記プロセッサに接続され、命令を読み出すフェッチ・メカニズムと、

前記プロセッサに接続され、前記命令を前記フェッチ・メカニズムから受け取り、前記命令を順序外で実行する再処理メカニズムと、

前記再処理メカニズムに接続され、前記再処理メカニズムの命令の構造的に不正確な実行を認識し、前記実行の後、前記命令をパニック・トラップ・インジケータと関連付けるパニック・トラップ論理手段と、

前記再処理メカニズムと前記フェッチ・メカニズムに接続され、前記再処理メカニズム内の命令を消去し、前記フェッチ・メカニズムに再読み出しをさせ、前記パニック・トラップ・インジケータに基づいて前記命令を開始するトラップ・ベクトル・ジェネレータと、
を備え、前記パニック・トラップ・インジケータは、前記構造的に不正確な実行を生じさせた命令を前記再処理メカニズムから引退させる引退信号による制御によって、前記トラップ・ベクトル・ジェネレータに送られるようにした、前記システム。

【請求項2】

プロセッサ内ソフトウェアの順序外命令の実行で生じた不正確な結果から回復するためのシステムであって、

前記順序外命令を集め、実行する第1の手段と、

前記第 1 の手段に接続され、命令の構造的に不正確な実行を認識し、前記不正確な実行を生じた命令を示す、前記ソフトウェアに不可視の非構造的ハードウェア信号である、パニック・トラップ・インジケータを、該命令を引退させる引退信号によってアサートする第 2 の手段と、

前記アサートされたパニック・トラップ・インジケータを認識し、前記不正確な実行を生じた命令から順に命令の再読出しを行う第 3 の手段と、
を有する前記システム。

【請求項 3】

プロセッサ内ソフトウェアの順序外命令の実行で生じた不正確な結果から回復するための方法であって、

再処理メカニズムに命令を集めるステップと、

前記命令を順序外で実行するステップと、

命令の構造的に不正確な実行を認識するステップと、

命令の構造的に不正確な命令を認識し、該不正確な実行を生じた命令を示す、前記ソフトウェアに不可視の非構造的ハードウェア信号であるパニック・トラップ・インジケータを、該命令を引退させる引退信号によってアサートするステップと、

前記アサートされたパニック・トラップ・インジケータを認識し、前記再処理メカニズム内の命令を消去し、前記不正確な実行を生じた命令から順に命令の再読出しを行うステップと、

を有する前記方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、一般にコンピュータ・プロセッサ・アーキテクチャに関連する。特に、プロセッサの命令の実行順序が狂っているときに、不正確な処理からリカバリするためのパニック・トラップ・システムと方法に関する。また、パニック・トラップ・システムと方法は、タイミングと同期を図るためにも使用される。

【0002】

【従来の技術】

コンピュータ・プロセッサ（演算処理装置）は、一般に、システムのオペレーションを実行する制御装置と、計算型のオペレーションを実行する算術論理演算装置（ALU）を含む。プロセッサのデザインは、レジスタ・セット、これらのレジスタ間の通信経路、及びこれらが稼働する方法を指示し制御する手段の選択によって決定される。通常、プロセッサは、メイン・メモリの中に保持されている一連の命令から成るプログラムによって稼働される。各々の命令は、ビット群であり、通常、プロセッサによって実行されるオペレーションを指定する 1 又は複数のワードの長さである。

【0003】

一般に、プロセッサの基本サイクルは、以下のステップから成る。

（a）命令をメイン・メモリから命令レジスタに取り出す。

（b）命令を復号する（すなわち、命令が示す実行すべきことを決定する。各命令が、実行されるオペレーションとそのオペレーションが使用するデータを示す）。

（c）命令によって指定されたオペレーションを実行する。

（d）次の命令の位置を決定する。通常、次の命令が、現在の命令の直後に続いている。

【0004】

しかしながら、複数のスカラ・オペレーションが平行して実行されるスーパースカラ・プロセッサのような高性能なプロセッサにおいて、プロセッサが、混乱した順序またはプロセッサを操作しているソフトウェアによって定義された順序と一致していない順序で命令を実行するように設計される場合がある。これらのシステムでは、プログラムによって定義された順序で命令が発生したときではなく、命令が実行されることができるときに、命令は実行される。更に、順序の混乱した命令の実行の後、プログラムにその結果を渡す前

10

20

30

40

50

に、命令順序に対応させて、最終的に再処理される。

【0005】

キャッシュ・メモリは、しばしばパフォーマンスを最適化するために、コンピュータのプロセッサに関連して、使われている。キャッシュ・メモリは、コンピュータのプロセッサとメイン・メモリの間に置かれた高速バッファである。プロセッサで使用される現在のデータと命令が、キャッシュ・メモリに移され、それによって、2つの利益を生じる。第1に、プロセッサのメモリ使用要求のための平均のアクセス時間が減少し、プロセッサのスループットが増加する。第2に、それによって、利用可能なメモリ・バンド幅によるプロセッサの使用が減り、システム・バス上の他の装置が、プロセッサのじゃまをすることなくメモリを使用できるようになる。キャッシュ・メモリは、メイン・メモリからプロセッサへの命令とデータの転送スピードをアップさせるために使用される。メイン・メモリ・サイクル・タイムが一般的にプロセッサ・クロッキング率より遅いので、このキャッシュ機能が重要となる。

10

【0006】

プロセッサがデータ・キャッシュをデータ・ラインのためにアクセスするとき、プロセッサはアドレスをキャッシュに転送する。キャッシュが、アドレスからキャッシュ・インデックスを解析して、要求されたデータ・ラインを含む記憶領域を選択するためにそれを使用する。キャッシュが、実ページ番号(RPN)であり、領域と状態インジケータに対応し、タグに対応するデータ・ラインが有効であるか否かを示す、タグを出力する。

【0007】

一般的に、キャッシュと関連するサポート回路は、状態インジケータとタグを受け取る。状態インジケータが無効データを示すとき、サポート回路がプロセッサに「ミス」表示を転送し、その場合には、プロセッサがデータ・ラインのためにメイン・メモリにアクセスしなければならない。状態インジケータが有効なデータを示すとき、キャッシュが要求されたデータ・ラインを現在記憶しているか否かを決定するために、サポート回路はタグとアドレスの残余とを比較する。キャッシュがタグの比較によって決められるような要求されたデータ・ラインを持たないとき、サポート回路が、プロセッサに「ミス」表示を転送する。その場合、プロセッサが、メイン・メモリをデータ・ラインのためにアクセスしなければならない。キャッシュがタグの比較によって決められるような要求されたデータ・ラインを持つとき、サポート回路が、プロセッサに「ヒット」表示を転送して、要求されたデータ・ラインを讀出するためにプロセッサにプロンプトを出す。

20

30

【0008】

命令の順序外実行を行うプロセッサにおいて、命令の不正確な実行処理のために不正確な結果が生じる場合がある。また、命令がそれが依存する命令より先に実行されるとき、不正確な結果が生成し、例えば、記憶(ストア)命令に依存するロード命令によって、ストア命令が転送される場合を考える。一般に、「ロード」命令はストレージ(キャッシュ又はメイン・メモリ)からレジスタへのデータ転送要求であるのに、「ストア」命令はレジスタからストレージへのデータ転送要求である。ロード命令がストア命令の前に処理される場合、ロード命令は無効データを送信する。従って、ストア命令は、正しい結果を得るためにロード命令の前に実行される。順序外命令を実行するスーパースカラ・プロセッサにおいて、不正確な結果が得られた場合、不正確な結果を認識して、そこから回復するためのメカニズムが必要である。

40

【0009】

【発明が解決しようとする課題】

従って、本発明の目的は、順序外命令を実行するスーパースカラ・プロセッサにおいて、不正確な結果が得られた場合、不正確な結果を認識して、そこから回復するためのメカニズムを提供することである。

【0010】

また、本発明の他の目的は、スーパースカラ・プロセッサ及び順序外命令を実行するプロセッサのパフォーマンスを改善することである。

50

【 0 0 1 1 】

更に、本発明の他の目的は、命令の不正確な順序の実行からプロセッサを回復するためのシステムと方法を提供することである。

【 0 0 1 2 】

即ち、本発明の目的は、命令の順序外実行から生成された不正確な結果からプロセッサを回復するのに使われ、更には、タイミングと同期を図るパニック・トラップ・システム及び方法を提供することである。

【 0 0 1 3 】

【課題を解決するための手段】

パニック・トラップ (panic trap) ・システムは、以下の構造から成る。

10

【 0 0 1 4 】

命令フェッチ・メカニズム (I F E T C H) は、命令を命令キャッシュ (I C A C H E) から取り出す。 I F E T C H は、タイミングや同期を取るためにパニック・トラップ・インジケータと命令を関連付ける。パニック・トラップ・インジケータとは、プロセッサを操作しているソフトウェアには発生しない非構造的な信号である。命令が終了したならば、このパニック・トラップ・インジケータは、最終的に、パニック・トラップ・インジケータを持っている命令から順に取出しを再開する。

【 0 0 1 5 】

2つの待ち行列は、命令を I F E T C H から受け取る。 A L U (算術論理演算装置) 命令待ち行列 (A Q U E U E) は、 A L U で示される命令を受け取って、順序外命令を実行する。各々の命令は、それぞれの命令レジスタの中で A Q U E U E に置かれる。メモリ命令待ち行列 (M Q U E U E) は、データ・キャッシュ (D C A C H E) がメイン・メモリで示される命令を受け取って、順序外命令を実行する。 M Q U E U E は、命令レジスタを含み、メモリ命令と命令実行の結果に応じたデータ・アドレスを受け取るアドレス・リオーダー・バッファ・スロット (A R B S L O T S) と関連付けられる。

20

【 0 0 1 6 】

トラップ・インジケータ・ロジックは、メモリ命令の構造的に不正確な実行を認識し、パニック・トラップ・インジケータを実行後の命令と関連付けるための各 A R B S L O T と関連する。実行されたロード命令が実行されていないストア命令に依存する場合は、命令の構造的に不正確な実行の一例を示す場面である。パニック・トラップ・インジケータ・ロジックは、この場面を決定し、発見した場合にパニック・トラップ・インジケータを発生する依存ロジックを持つ。

30

【 0 0 1 7 】

A R B S L O T は、さらに、命令が可干渉性チェック信号を基に転送されたデータに影響を与える場合を決定するキャッシュ可干渉性チェック (C C C) ハザード分解システムを含む。 C C C ハザード分解システムは、パニック・トラップ・インジケータをセットすることができる。

【 0 0 1 8 】

システム・インタフェースへのアクセスを制御するシステム・インタフェース制御は、同期信号や遠隔装置からの外部割込の受信に基づいて、パニック・トラップ・インジケータを起動することがある。この場合、システム・インタフェース制御が、実行後の引退し (R E T I R E) た命令を担当して、パニック・トラップ・インジケータと命令とを関係付ける引退メカニズムにパニック・トラップ・インジケータを転送する。

40

【 0 0 1 9 】

命令に応じてキャッシュ・アクセスに関係付けられたパリティ・エラーが生じた場合、 D C A C H E はパニック・トラップを起動することがある。 D C A C H E は、パニック・トラップ・インジケータを引退メカニズムに提供することによってパニック・トラップを発生する。

【 0 0 2 0 】

命令が A Q U E U E か M Q U E U E の中で実行された後、命令は引退メカニズムによって

50

引退させられる。引退処理の間、パニック・トラップ・インジケータを認識しながら、引退メカニズムに関連するトラップ・ベクトル・ゼネレータが、A Q U E U E や M Q U E U E からの命令を消去し、I F E T C H に再び読出しをさせ、パニック・トラップを引き起こした命令から順に実行する。

【 0 0 2 1 】

また、本発明はいくつかの新しい方法を提供している。最初の方法は、プロセッサを命令の順序外実行で処理された不正確な結果から回復することを許可し、特に、実行段階の間のパニック・トラップの関係を管理する。

【 0 0 2 2 】

第1の方法の概要を次に示す。再処理メカニズムの中で命令を集め、順序外命令を実行する。命令の構造的に不正確な実行を認識し、命令と関連するパニック・トラップ・インジケータを発生する。発生したパニック・トラップ・インジケータは不正確な実行を示している。実行後の命令を引退させるとき、発生したパニック・トラップ・インジケータを認識し、再処理メカニズムを除去して、無効な命令から順に読出しを再び始める。

10

【 0 0 2 3 】

第2の方法は、読出段階の間、順序外命令を実行し、特に、命令と関連付けられたパニック・トラップを示すプロセッサで、不正確な実行結果を生じた命令から回復を可能とする。第2の方法の概要を次に示す。命令を読出し、命令が読出しの間、無効であることを決める。パニック・トラップ・インジケータを無効な命令と関連付け、他の命令によって無効な命令を再処理メカニズムに置く。再処理メカニズムの中で順序外命令を実行する。そして、実行後の命令を引退させるとき、パニック・トラップ・インジケータを認識し、再処理メカニズムを除去して、無効な命令から順に再び読出し始める。

20

【 0 0 2 4 】

3番目の方法は、順序外命令を実行するプロセッサと関連するデータ・キャッシュによって発見されたパリティ・エラーからの回復を可能とする。第3の方法の概要を次に示す。再処理メカニズムの中に命令を集め、順序外命令を実行する。データ・キャッシュに対するアクセスを含んでいる命令を実行し、パリティ・エラーがアクセス中に起こったかを決める。プロセッサでパニック・トラップを生成し、データ・キャッシュからトラップ・ベクトル・ゼネレータへ転送する。再処理メカニズムを除去して、その命令で命令の実行を再び始める。

30

【 0 0 2 5 】

前述した目的を達成することに加えて、本発明は、次の他の特徴を持つ。

【 0 0 2 6 】

本発明の特徴は、I F E T C H がタイミングと同期目的のためにパニック・トラップ・システムを使用することができるということである。

【 0 0 2 7 】

また、本発明の他の特徴は、D C A C H E がキャッシュ・アクセスに関係するパリティ・エラーを示し、回復するためにパニック・トラップ・システムを使用することができるということである。

【 0 0 2 8 】

更に、本発明の他の特徴は、システム・インタフェース制御がタイミングと同期を図るためにパニック・トラップ・システムを使用することができるということである。

40

【 0 0 2 9 】

また更に、本発明の他の特徴は、パニック・トラップ・システムのデザインが単純でその処理が信頼でき、大量生産でコンピュータが安価にできるということである。

【 0 0 3 0 】

【 発明の実施の形態 】

図1には、コンピュータ11内で、特に、コンピュータ11のプロセッサ14内の命令読出/実行システム12内で、実行するパニック・トラップ・システム100(図3~5参照)が表示されている。コンピュータ11は、一般に、プロセッサ14と、プロセッサ14を操作するソフ

50

トウェア (S / W) 18を持つメイン・メモリ16と、図中の矢印23で示すようにプロセッサ14と接続するデータ・キャッシュ24 (D C A C H E) と、プロセッサ14とメイン・メモリ16を相互に連結させる 1 又は複数のバスのようなシステム・インタフェース22とから成る。

【 0 0 3 1 】

オペレーションにおいて、プロセッサ14の中の命令読出 / 実行システム12がソフトウェア18を実行できるように、プロセッサ14の中で現在使用されているデータが、D C A C H E 24へ移される。それによって、プロセッサのメモリ要求の平均アクセス時間が減少し、システム・インタフェース22上の通信量が最少となる。ここで、新しいパニック・トラップ・システム100を除く前述のコンピュータ装置とそれらの構成の相互作用は従来技術として知られている。

10

【 0 0 3 2 】

D C A C H E 24の一般的なキャッシュ・ラインは、タグ、状態インジケータ、及びデータを含む。キャッシュ・インデックスは、D C A C H E 24に転送されて、要求されたデータ・ラインを含む記憶域を選択するために、D C A C H E 24によって使用される。キャッシュ・インデックスの受信に応じて、D C A C H E 24が、タグを出力する。タグは、記憶域に対応している実ページ番号 (R P N) であり、状態インジケータは、タグに対応しているデータ・ラインが有効であるか無効であることを示し、データは、有効又は無効のどちらかである。一般的に、状態インジケータは次の状態を示す。「無効 (invalid) 」は、データが供給されないことを意味する。「有効共有 (valid shared) 」は、データが供給されるが、どこか他の場所にあることを意味する。「有効プライベート・クリーン (valid private clean) 」は、ラインがソール (sole) ・コピーを持ち、D C A C H E 24がラインにまだ書込んでいないことを意味する。「有効プライベート・ダーティ (valid private dirty) 」は、ラインがソール・コピーを持ち、D C A C H E 24がラインに書込まれる (即ち、メイン・メモリ16にラインをコピーする必要がある) 。

20

【 0 0 3 3 】

D C A C H E 24と接続しているタグ比較メカニズム228 (図 6) は、状態 (ステータス) インジケータとタグを受け取る。状態インジケータが無効データを示すとき、タグ比較メカニズム228はプロセッサ14に「ミス」表示を転送する。その場合、プロセッサ14はメイン・メモリ16をデータ・ラインのためにアクセスする。状態インジケータが有効データを示すとき、D C A C H E 24が要求されたデータ・ラインを現在記憶しているか否かを決定するために、タグ比較メカニズム228は、タグとアドレスの残余とを比較する。タグの比較によって決定され要求されたデータ・ラインをD C A C H E 24が持たないとき、タグ比較メカニズム228はプロセッサ14に「ミス」表示を転送する。その場合、プロセッサ14はメイン・メモリ16をデータ・ラインのためにアクセスする。タグ比較によって決定され要求されたデータ・ラインをD C A C H E 24が持つとき、タグ比較メカニズム228はプロセッサ14に「ヒット」表示を転送し、要求されたデータ・ラインを読取るためにプロセッサ14にプロンプトを出す。

30

【 0 0 3 4 】

命令読出 / 実行システム12の実行を図 2 のブロック図で示す。図 2 の中で示されるように、命令読出 / 実行システム12が、命令をソフトウェア18 (図 1) から記憶するための命令キャッシュ (I C A C H E) 26を持つ。命令フェッチ・メカニズム (I F E T C H) 28が命令キャッシュ26と接続され、最後に、実行のためにキャッシュ26から命令を検索する。ここで、命令 I F E T C H 28は、一度に 4 つの命令 (各 3 2 ビット) を読取り、ソート・メカニズム32へ命令を転送する。

40

【 0 0 3 5 】

命令は、I F E T C H 28から、待ち行列か保留場所のような適当な再処理メカニズムに送り出される。ここで、命令は、ソートされ、算術論理ユニット (A L U) とメモリ待ち行列に (M Q U E U E) に分配される。

【 0 0 3 6 】

50

特に、ソート・メカニズム32は、I F E T C H 28から命令を受取り、各命令が算術論理ユニット (A L U) がメモリかに前もって分別されているかどうかを決定する。それに応じて、ソート・メカニズム32が、図 2 の矢印36aと36bに示すように、それぞれ命令を A Q U E U E 38aと M Q U E U E 38bに分配する。

【 0 0 3 7 】

A Q U E U E 38aは、矢印43で示されるように、算術論理演算装置 (A L U) 42に転送されるそれぞれの命令を記憶するレジスタ41aを持つ A L U 命令処理メカニズム39a (総計 28 個) を含む。 A Q U E U E 38aの命令は、順序付けられて (即ち、データフロー型式で) 実行される。命令が完了すると、 A Q U E U E リネーム・レジスタ44aによって結果が獲得され、命令に A Q U E U E 38aの中で完了したことを示す印が付けられる。 A Q U E U E 38aは、ソート・メカニズム32からサイクル毎に 4 つの命令 (それぞれ 3 2 ビット) を受信し、矢印51aによって示されるように引退 (RETIRE) メカニズム52へ、サイクル毎に 2 つの命令 (それぞれ 3 2 ビット) を送信する。

10

【 0 0 3 8 】

インタフェース45で示すように、 A Q U E U E 38aの制御の下に、 A L U 42がリネーム・レジスタ44a、44bと汎用レジスタ46からオペランドを検索することができる。 A L U 42がオペランドを実行した後、矢印49で示されるように、実行結果は、 A Q U E U E リネーム・レジスタ44aに記憶される。 M Q U E U E 38bは、命令処理メカニズム39bを含む。各々の命令処理メカニズム39bは、それぞれのメモリ命令を記憶するレジスタ41bと、それぞれのアドレスを記憶するアドレス・リオーダ・バッファ・スロット (A R B S L O T : 総計 28 個) 48を含む。 M Q U E U E 38b中のメモリ命令は、メモリに対して「ロード (load) 」及び「記憶 (store) 」として類別することができる。「ロード」は、メモリ (キャッシュやメイン・メモリ) からレジスタへデータを伝送する要求であるのに対して、「記憶」はレジスタからメモリへデータを伝送する要求である。

20

【 0 0 3 9 】

命令の実行の間、最初の行程は、アドレスを計算するために定められた数学的オペレーションをオペランドの上で実行することを意味し、2番目の行程は、計算されたアドレスに基づくデータのためにメモリやキャッシュをアクセスすることを意味する。 M Q U E U E 38bは、各々の命令と、2つの行程と、順次 (即ち、データフロー型式) メモリ / キャッシュ・アクセスとを実行する。命令が完了すると、結果が M Q U E U E リネーム・レジスタ44bによって獲得され、命令に M Q U E U E 38bの中で完了したことを示す印が付けられる。 M Q U E U E 38bは、ソート・メカニズム32からサイクル毎に 4 つの命令 (それぞれ 3 2 ビット) を受信し、矢印51bによって示されるように引退 (RETIRE) メカニズム52へ、サイクル毎に 2 つの命令 (それぞれ 3 2 ビット) を送信する。

30

【 0 0 4 0 】

特に、命令実行の最初行程の間、アドレスはアドレス計算機58で生成される。アドレス計算機58が、 M Q U E U E リネーム・レジスタ44bから検索されたオペランドに基づいてアドレスを計算し、矢印62で示されるように、 M Q U E U E 38bの中の命令に対応する A R B S L O T 48に (実または仮想) アドレスを渡す。

【 0 0 4 1 】

命令による計算の制御は、図 2 の矢印64で示される。メモリ命令実行の2番目の行程が遂行されるとき、 D C A C H E 24でロード又は記憶を達成するために、矢印54で示されるように、計算されたアドレス (キャッシュ・インデックスを含む) が D C A C H E 24へ転送される。ここで、可能であれば、2つアドレスが、各々の各サイクルで M Q U E U E 38bから D C A C H E 24へ転送される。このマルチ・アドレス機能を適応させるため、例えば、 D C A C H E 24は、デュアル・ポートで実行される。これに対して他の例として、奇数と偶数アドレスが、単一移行されたランダム・アクセス・メモリ (R A M) 中のそれぞれの奇数と偶数キャッシュ・バンクに同時に送られるものもある。この詳細は、同じ発明者によって同日、同時に出版された "Address Aggregation System And Method For Increasing Throughput Of Addresses To A Data Cache From A Processor" に記述されている。

40

50

一度、D C A C H E 24がアドレスを処理すると、矢印56に示すように、データ結果はM Q U E U E リネーム・レジスタ44bに転送される。

【 0 0 4 2 】

引退メカニズム52は、実行された命令（即ち、サイクル毎に2つの32ビット・ワード）を待ち行列38a、38bの各々から受け取る。引退メカニズム52は、構造的状態で命令結果を入れる。ソフトウェア18（図1）は、結果が引退メカニズム52によって構造的状態に変換されることに関与しない。引退メカニズム52が、それぞれ矢印73、74によって示されるように、命令の属性に基づいて命令結果を汎用レジスタ46や制御レジスタ72に移動することにより、ソフトウェア18によって定義されたプログラム命令の中の待ち行列38a、38bの命令を引退させ、矢印76a、76bによって示すように命令結果をリネーム・レジスタ44a、44bから汎用レジスタ46に渡す。

10

【 0 0 4 3 】

引退メカニズム52が、キャッシュ24のデータ・ラインに結果的に記憶された命令を、引退させるとき、ラインが変わり、メイン・メモリ16でラインを更新し、メイン・メモリ16に最終的に転送することを示すために、引退メカニズム52が、ラインに応じて状態インジケータを、「ダーティ」とマークする。

【 0 0 4 4 】

また、引退メカニズム52は、命令に関係する「例外」があるか否かを決定するロジックを持つ。「例外」は、1又は複数の以前の命令に対応する特別な状況を示すフラグである。引退メカニズム52は、「例外」を示した命令に先行する待ち行列38a、38b内の全ての命令を捨てて、命令フェッチメカニズム28に、もう一度再処理するために問題となっている命令を検索させるか、または、特別な状況を扱う特別なソフトウェアを検索させるかする。

20

【 0 0 4 5 】

パニック・トラップ・システム及び方法

ここで、パニック・トラップ・システム100とそれに関連する方法を図3～5を参照して詳述する。一般に、パニック・トラップ・システム100及び方法が、エラーを認識し、エラーの生じた命令を再実行することによって、命令の不正確な順序外実行からの回復を可能とする。それによって、2回目の実行で、正しい結果となる。このことは、ソフトウェア18（図1）では見られない非構造的なハードウェア信号であるパニック・トラップの概念を利用することによって達成される。

30

【 0 0 4 6 】

パニック・トラップは、以下の構成によって発生する。

（a）命令に関するタイミング又は同期要求時、I F E T C H 28。

（b）メモリ命令の不正確な順序外実行時、M Q U E U E 38bのメモリ命令処理メカニズム39bの中のA R B S L O T 48。

（c）メモリ命令に従うアクセスに関する発見されたパリティ・エラー時、D C A C H E 24。

（d）同期命令（例えば、マルチ・プロセッサ（MP）システムの中での）、外部割込み、その他の事象での、システム・インタフェース22へ（又は、から）のアクセスを制御するシステム・インタフェース制御22。

40

【 0 0 4 7 】

特に、I F E T C H 28は、タイミングや同期を図るためにパニック・トラップを始める。例えば、I F E T C H 28は、命令と関連してパニック・トラップをセットすることによって、待ち時間サイクルを挿入することができる。ここで、パニック・トラップ・インジケータ102（1又は複数のビット）は、命令処理メカニズム39a、39bに渡される各命令104と関連する。I F E T C H 28は、インジケータ102が潜在的に不正確な動作を引き起こす命令104と関係するパニック・トラップを発生する。この発生又は非発生に基づき、パニック・トラップ・インジケータ102は、特別な命令104が引退メカニズム52によって最終的に引退するときパニック・トラップを始めるか否かを示めす。パニック・トラップ・インジケータ102として1又は複数のビットを実行する代わりに、パニック・トラップ・ロード

50

命令が、引退メカニズム52でパニック・トラップを始める I F E T C H 28によって、A Q U E U E 38aに置かれる。

【 0 0 4 8 】

命令の実行が不正確な結果を引き起こすとき、A R B S L O T 48自身によるパニック・トラップの生成に関して、メモリ命令104と関連するパニック・トラップ・インジケータ102が、対応するA R B S L O T 48によって、発生される。発生又は非発生に基づき、パニック・トラップ・インジケータ102は、特別な命令104が引退メカニズム52によって最終的に引退するときパニック・トラップを始めるか否かを示す。パニック・インジケータ102が命令104と関連するA R B S L O T 48によって発生される時の例としては、命令104がストア命令に従うロード命令であって、ストア命令がロード命令に先立って実行されなかったという状況がある。この場合、ロード命令は、無視される無効データを生成する結果となる。

10

【 0 0 4 9 】

また、矢印106で示されるように、D C A C H E 24はパニック・トラップ・インジケータ102を引退メカニズム52に供給する。この事によって、D C A C H E 24は、D C A C H E 24への記憶及びD C A C H E 24からのロードが有効であることを確保するために、パリティ・チェック・ロジックを持つ。特に、ロードの間、D C A C H E 24はパリティ・エラーのためにタグとデータをチェックし、記憶の間、D C A C H E 24はパリティ・エラーのためにタグをチェックする。パリティ・エラーが発見されたならば、D C A C H E 24は、パニック・トラップ・インジケータを引退メカニズム52に発生する。幾つかのケースでは、パリティ・エラーから回復でき、他のケース（例えば、ダーティ・データ、タグ・エラー等）では、回復できない。回復ができる場合において、引退メカニズム52によって、パリティ・エラーを発生した命令の再実行を行い、新しいデータ・ラインがメイン・メモリ16から検索されるようになる。回復が可能でない場合において、最終的にコンピュータ11のオペレーションを停止するトラップ・コードに、引退メカニズム52が、向くことになる。

20

【 0 0 5 0 】

また、矢印112で示すように、プロセッサ14に接続されたシステム・インタフェース制御108は、引退メカニズム52にパニック・トラップ・インジケータを提供する。システム・インタフェース制御108がシステム・インタフェース22へ（又はから）のアクセスを制御する。例えば、マルチ・プロセッサ（M P）システムで、プロセッサ14がM Pシステムの他のプロセッサと同期するように、システム・インタフェース制御108が、同期命令をシステム・インタフェース22から受け取ることができる。同期命令がシステム・インタフェース制御108によって受け取られるとき、システム・インタフェース制御108は、パニック・トラップ・インジケータを引退メカニズム52に発生する。待ち行列38a、38bがフラッシュされたあと、パニック・トラップを得る命令がリフェッチされ、フラッシュとリスタート・オペレーションの結果、プログラムの実行の同期がそのまま続くことになる。

30

【 0 0 5 1 】

引退メカニズム52が発生されたパニック・トラップ・インジケータ102を持つ命令104を引退させるとき、あるいは、引退メカニズム52が発生されたパニック・トラップ・インジケータ106、112をD C A C H E 24又はシステム・インタフェース制御108から受け取るとき、引退メカニズム52は、トラップ表示をトラップ・ベクトル・ゼネレータ114へ知らせる（矢印116）。引退メカニズム52は、パニック・トラップが発生されたトラップ・ベクトル・ゼネレータ114に対して指示を出し、パニック・トラップに対応する命令アドレスを転送する。次に、矢印118、118a、118bによって示されるように、トラップ・ベクトル・ゼネレータ114は、A Q U E U E 38aとM Q U E U E 38bの内容を削除する。さらに、矢印121で示すように、トラップ・ベクトル・ゼネレータ114は、命令アドレスをI F E T C H 28へ送る。I F E T C H 28は、矢印27で示されるように、新しい命令アドレスから始まる命令アドレスをI C A C H E 26から検索し、順序外実行のためにアドレスをA Q U E U E 38aとM Q U E U E 38bに置く。従って、パニック・トラップにかけられた命令の命令アドレスで、命令の実行がもう一度始まる。

40

50

【 0 0 5 2 】

また、引退メカニズム52は、構造的トラップ（非構造的パニック・トラップでなく）を受け取る場合がある。例えば、システム・インタフェース制御108は、外部割込みをシステム・インタフェース22から受け取る場合がある。この場合、システム・インタフェース制御108は、外部割込みを扱うためにコードと対応している命令アドレスとともに、引退メカニズム52へトラップ信号を発生する。さらに、引退メカニズム52は、トラップ・ベクトル・ゼネレータ114へ、トラップが発生したことを示し、トラップと対応している命令アドレスを転送する。上述のように、トラップ・ベクトル・ゼネレータ114は、順次に応答する。

【 0 0 5 3 】

10

従来の構造的トラップと同様に非構造的パニック・トラップを生成するための各 A R B S L O T 48の内のトラップ・インジケータ・ロジックのオペレーションを図4に示す。図4において、トラップ・インジケータ・ロジック122は、構造的トラップ・インジケータ（T）122からの構造的トラップ信号126と同様にパニック・トラップ・インジケータ（P）102からのパニック・トラップ信号121を発生できる入力を、受け取る。ここで、トラップ・インジケータ102、122は、ラッチ又は他の適当なロジック記憶装置である。

【 0 0 5 4 】

ここで、トラップ・インジケータ・ロジック122が、記憶・ロード（S T / L D）ハザード信号124（図7のST/LD_HAZARD）とキャッシュ可干渉性チェック（C C C）ハザード信号125（図6のCCC_HAZARD）を受け取る。ハザード信号124、125は、パニック・トラップが発生される時を示すためにA N D 論理ゲート135に渡される信号132を生成するO R 論理ゲート128に伝えられる。また、A N D 論理ゲート135は、UPDATE信号131を受け取る否定回路133から、-UPDATE信号134を受け取る。トラップ・インジケータ・ロジック122と対応する命令がそのデータ・アドレスを対応するA R B S L O T 48に示すとき、UPDATE信号134が、発生する。UPDATE信号134の発生が、P信号121を消滅するか、クリアする。UPDATE信号134の発生は、TLB_TRAP信号141が消滅されている場合、トラップ信号126をクリアし、TLB_TRAP141がトランジスタ145を通じて発生する場合、トラップ信号126をセットする。さらに、A N D 論理ゲート135に与えられる-UPDATE信号134が、クリア処理の後に発生し、ハザード信号124、125が、命令が有効な間に獲得され、A N D 論理ゲート135からの信号136が発生する。

20

30

【 0 0 5 5 】

A N D 論理ゲート135からの信号136は、トランジスタ137、138に作用する。トランジスタ137が、最終的にパニック・トラップを生成し、トランジスタ138が、最終的に構造的トラップを生成する。パニック・トラップに関連して、UPDATE信号123が発生していないとき、トランジスタ137は、パニック・トラップ信号121を発生することができる。構造的トラップに関連して、UPDATE信号123が発生せず、且つ変換ルックアサイド・バッファ（T L B）214（図6）からTLB_TRAP信号141が発生するとき、トランジスタ138は、構造的トラップ信号126を発生する。

【 0 0 5 6 】

A R B S L O T 48と関連する命令が、引退メカニズム52（図3）によって引退されるとき、矢印146で示すように、引退信号R E T [i]（ここで、i はトラップ・インジケータ・ロジック122を含む特別のA R B S L O T 48に対応する）によって制御されるようなドライバ142、144は、パニック・トラップ信号121と構造的トラップ信号126のそれぞれを引退メカニズム52へ送り出す。

40

【 0 0 5 7 】

ここで、28個の引退ポインタRET[27:0]は、A Q U E U E 38aとM Q U E U E 38bの各々のために生成される。これらの引退ポインタRET[27:0]は、次の2つの命令が各待ち行列38a、38bの中で引退する場所を示す。各待ち行列38a、38bに与えられた時間で、前述の2つの引退ポインタが各々2つの連続的の命令処理メカニズム39a、39bに対して発生し、それによって、まだ引退していない最も古い命令を含む2つの命令処理メカニズム39a、39bを

50

示す。命令が引退するとき、その対応するポインタは消滅し、次の最も古いものが発生する。実行の例として、引退ポインタRET[27:0]は、循環桁移動チェーンから28個のマスタ・スレーブ・ラッチで生成される。

【0058】

図5は、本発明を実行するために、ロジックの実行が引退メカニズム52と関連することを示す。引退メカニズム52が、AQUEUE 38aとMQUEUE 38b中に引退命令と関連データで構成され、構造的状態になる。図5において、引退メカニズム52は、第1の行程52aと第2の行程52bを含む2つの行程パイプを含む。

【0059】

第1行程52aにおいて、AQUEUE 38aとMQUEUE 38bからの命令データは、構造的
10
トラップと非構造的パニック・トラップの表示のために、それぞれのAQUEUEとMQUEUEの一部52a'、52a''によって、別々に分析される。構造的トラップの概念は、よく知られている技術である。構造的トラップの例は、アドレス・カルキュレータ58(図2)の変換ルックアサイド・バッファ(TLB)214(図6)によって生成される(TLBミス、書込違反、調整違反、参照違反など)である。(構造的又はパニック)トラップが発生するならば、関連命令は機能停止し、トラップ表示がトラップ・ベクトル・ゼネレータ114に転送される。更に、第1行程52aは、AQUEUE 38aとMQUEUE 38bからの命令を再処理して、再結合するために、命令と関連データを分析する。第1行程52aが第2
20
行程52bに情報を伝送するように、再結合処理が発生する。最終的に、命令結果は、順に、汎用レジスタ46(図2)や制御レジスタ72(図2)に転送される。

【0060】

特に、図5の中で示されるように、第1行程52aが4つのレジスタ148(簡潔にするため148a、148bの2つのみ示す)を含む。レジスタ148aは命令データをAQUEUE 38aから受け取るのに対して、レジスタ148bは命令データをMQUEUE 38bから受け取る。

【0061】

レジスタ148aが、ALUオペレーション・コード(OPCODE)154と、命令アドレス(INSTR_ADDR)155と、トラップ・タイプ・インジケータ(T_TYPE: 1又は複数のビット)156と、
30
構造的トラップ・インジケータ(T: 1又は複数のビット)157と、パニック・トラップ・インジケータ(P: 1又は複数のビット)102とを受け取る。T_TYPE156は、構造的トラップのタイプを示す。RET_INSERT信号152によって記録されるそれぞれのトランジスタ159に接続されるそれぞれの入力158を介して、前述の情報がレジスタ148に置かれる。

【0062】

第1行程52aのMQUEUE部分において、レジスタ148bは、メモリ・オペレーション・コード(INSTR_OPCODE)164と、命令アドレス(INSTR_ADDR)165と、トラップ・タイプ・インジケータ(T_TYPE: 1又は複数のビット)166と、構造的トラップ・インジケータ(T: 1又は複数のビット)167と、パニック・トラップ・インジケータ(P)102と、データ・アドレス(DATA_ADDR)62とを受け取る。前述の情報は、RET_INSERT152信号によって記録されるそれぞれの入力168とそれぞれのトランジスタ169を介して、レジスタ148bに移
40
される。

【0063】

OR論理ゲート172は、パニック・トラップ・インジケータ106、112と同様に、構造的トラップ入力をDCACHE 24とシステム・インタフェース制御108から受け取るために配置される。このゲート172の結果は、この引退した命令がトラップに必要なか否かを示す。

【0064】

図5において、OR論理ゲート173が、MQUEUE 38bと、DCACHE 24と、システム・インタフェース制御108とから、パニック・トラップ・インジケータを受け取るために配置されている。このゲート173の結果は、パニック・トラップ・インジケータ102がこの引退した命令と関連しているか否かを示す。DCACHE 24及びシステム・インタフェース制御108からのパニック・トラップ・インジケータ102は、外部システムによって生成されるので、命令と関係付けることができる。トラップを引退する次のMQUEUE命令に
50

割り当てる。構造的トラップ (T_TYPE166で示される) とパニック・トラップとを選択できるならば、パニック・トラップが準備されていることがソフトウェア18に対して意味があるので、構造的トラップが機能する。

【 0 0 6 5 】

他の実施の形態として、INSTR_ADDR155、165が、実際の命令アドレスを記憶するルックアップ・テーブルに、ポインタで置換される場合がある。この場合は、引退メカニズム52の第1行程52a内で、記憶域を節約する。

【 0 0 6 6 】

第2行程52bにおいて、ADVANCE信号174が、レジスタ148a、148bの内容をそれぞれのレジスタ178a-178dに伝送するために、それぞれのトランジスタ176a-176dに作用する。上述では、レジスタ178a-178dの中で位置決めされた命令は、プログラム順となっている。トラップされた命令があるならば、最初のトラップされた命令だけが、レジスタ178a、178dの中で許可される。各々の命令は、命令の順序を示す一組のb4ビット182と関係付けられる。b4ビット182が引退メカニズム52からの出力であるとき、b4ビット182は命令を処理するために利用される。

【 0 0 6 7 】

構造的トラップ・インジケータT 157、167が、(図4の構造的又はパニック)トラップが生成されたか否かを示すトラップ信号187を生成するために、矢印185で示されるように、OR論理ゲート184に転送される。トラップ信号187は、トラップ・ベクトル・ゼネレータ114へ渡される。

【 0 0 6 8 】

トラップ・タイプ・インジケータ156、166とパニック・トラップ・インジケータ102が結合され、トラップ・タイプ信号196を生じるために、矢印194で示されるように、マルチプレクサ(MUX) 192に伝えられる。トラップされた命令が、レジスタ178a-178dの1つの中にあるとき、他の有効なトラップされた命令は、他のレジスタ178a-178dのいずれにも存在しない。更に、トラップされた命令を持っているレジスタ178a-178dが、T_TYPEとPを出力するとき、矢印196で示すように、トラップ・ゼネレータ114にこれらのビットを渡すために、レジスタ178a-178dは、MUX 192を制御する。トラップ・タイプ・インジケータ156、166は、構造的トラップのタイプを識別するトラップ・インジケータT 157、167に対応している。

【 0 0 6 9 】

INSTR_ADDR155、165は、トラップ・ベクトル・ゼネレータ114のための命令アドレス201を生成するために、矢印199で示すように、MUX 198に伝えられる。MUX 198は、MUX 192と同時に制御され、従って、トラップされた命令に関連するINSTR_ADDR201がトラップ・ベクトル・ゼネレータ114に転送される。

【 0 0 7 0 】

発生されたトラップ信号187、トラップ・タイプ信号196、パニック・トラップ・インジケータ102、及び命令アドレス201の受取りに基づいて、トラップ・ベクトル・ゼネレータ114は、新しい命令アドレス121を生成して、それをIFETCH 28(図3)に渡す。トラップが構造的トラップTであるとき、新しい命令アドレス121がトラップ・ハンドラ、例えば、TLBミス・ハンドラ、外部割込ハンドラなど、のアドレスに対応する。トラップが、IFETCH 28又は命令処理メカニズム39a、39bからの非構造的パニック・トラップPであるとき、新しい命令アドレス121は、パニック・トラップされた命令に対応する。

【 0 0 7 1 】

キャッシュ可干渉性チェック(CCC)ハザード分解システム

キャッシュ可干渉性チェック(CCC)ハザード125(図4)の生成を、図6を基に説明する。各々ARBSLOT 48は、DCA CHE 24から出力されプロセッサ14で利用可能でないすでに使用されたCCCデータを、ロード命令が持つかどうかを決定するCCCハザード分解システム202を含む。

【 0 0 7 2 】

10

20

30

40

50

キャッシュ可干渉性チェックの間、システム・インタフェース22に接続される、遠隔処理装置や遠隔入出力（I/O）装置のような他の装置が、データ・ラインの独占排他的なコピーを行う。従って、遠隔装置は、D C A C H E 24がデータ・ラインのコピーを持つか否かを決定するために、要求をD C A C H E 24へ送り出す。D C A C H E 24がデータ・ラインを持つならば、データ・ラインの所有権が、D C A C H E 24からシステム・インタフェース22に接続される遠隔装置に移される。

【0073】

本発明のM Q U E U E 38bにおいて、個別のA R B S L O T 48は、C C Cデータ・ラインと対応しているアドレス（キャッシュ・インデックスとタグを含む）を受け取るか否か及びデータ・ラインが存在するか否かを決定するために、D C A C H E 24をアクセスすること
10
に使用される。C C C照会に対するレスポンスが、システム・インタフェース制御108（最終的には、システム・インタフェース22）に送り出される。レスポンスは、その状態インジケータと同様に、（存在するならば）データ・ラインを含む。

【0074】

C C Cデータ・ラインのアドレスは、いずれかのA R B S L O T 48の命令がC C Cデータ・ラインを使用するか否かを決定するために、M Q U E U E 38b内の全てのA R B S L O T 48へ移される。C C Cデータ・ラインのロードを意味する命令がすでに実行されるならば、パニック・トラップは、その対応するパニック・トラップ・インジケータ102（図4）
20
を介してその命令に関連付けられる。

【0075】

図6において、C C Cハザード分解システム202は、ヒット（H I T）インジケータ206と、キャッシュ・インデックス207と、タグ208とを有するレジスタ204を含む。キャッシュ・インデックス207とタグ208は、アドレス・カルキュレータ58から受け取られる。アドレス・カルキュレータ58は、リネーム・レジスタ44a、44bや汎用レジスタ46（図2）からオペランド（O P）212を受け取って、実行する加算機211を含む。加算機211が、矢印216で示すように、A R B S L O T 48及び変換ルックアサイド・バッファ（T L B）214に渡される仮想アドレスを生成する。T L B 214は仮想アドレス216に基づいて実アドレス・タグを生成し、矢印218で示すように、レジスタ204にそれを渡す。
30

【0076】

レジスタ204のキャッシュ・インデックス207は、矢印222で示すように、D C A C H E 24
30
へアクセスするのに使用される。キャッシュ・インデックス222を受信して、D C A C H E 24が、タグ224と、状態インジケータ226と、データとを選別して出力するために、着信キャッシュ・インデックスを使用する。タグ224と状態インジケータ226は、ヒットかミスかを判断するために、タグ比較メカニズム228に渡される。データは、M Q U E U E リネーム・レジスタ44bに移される。また、タグ比較メカニズム228が、A R B S L O T レジスタ204から受け取ったタグ208とD C A C H E 24からのタグ224を比較する。比較の結果が一致したならば、矢印227で示すように、タグ比較メカニズム228が、ヒット・インジケータ206を通してレジスタ204の中にヒットを発生する。

【0077】

更に、C C Cハザード分解システム202が、矢印232で示すように、遠隔C C C A R B S L O T 48から（キャッシュ・インデックスを持つ）アドレスMISS_ADDR231と、ローカルA R B S L O T レジスタ204からキャッシュ・インデックス207とを受け取るように構成された、キャッシュ・インデックス比較メカニズム228を含む。キャッシュ・インデックスの比較に基づいて、比較メカニズム228は、ローカルA R B S L O T 48の中の命令が、C C Cデータ・ラインを使用したどうかを示すマッチ信号（MATCH）234を生成する。MATCH信号234は、C C Cハザード信号125を発生するA N D 論理ゲート236と、C C Cハザード信号125を消滅するA N D 論理ゲート238とに転送される。
40

【0078】

A N D 論理ゲート236は、MATCH信号234と、ローカルA R B S L O T 48がキャッシュ・ヒットを持ち、既にデータを受信しているかどうかを示すDONE信号239と、C C C A R B
50

S L O T 48が比較のためにそのアドレスMISS_ADDR231を配信したことを示す、C C C A R B S L O T 48からのC C C制御信号(C C C -CNTL) 241と、を受け取る。前述の信号の状態に基づいて、A N D論理ゲート236は、信号237を生成して、C C Cハザード信号125を生成するA N D論理ゲート247にそれを渡す。

【 0 0 7 9 】

DONE信号239は、A N D論理ゲート243によって生成される。このDONE信号239を生成するために、A N D論理ゲート243は、以下のものを受け取る。

(a) A R B S L O T 48が他のA R B S L O T 48の中のストア命令に依存するロード命令を持つときを示すストア非依存信号(-DS: 図7) 242。

(b) A R B S L O T 48の中に有効アドレスがあるときを示すキャッシュ・アクセス有効信号(CAV) 245。 10

(c) D C A C H E 24がA R B S L O T 48の中の命令によってアクセスしたときを示す非キャッシュ保留信号(-CP: 図7) 246'。

(d) ヒット又はミスがD C A C H E 24で起こったことを示すヒット信号(HIT) 247。

(e) ローカルA R B S L O T 48の中の命令がD C A C H E 24には無い現在発生しているミス・データに依存しないときを示す非依存ノーマス(onmiss) 信号(-DM) 248。

【 0 0 8 0 】

A N D論理ゲート238が、キャッシュ・インデックス比較メカニズム228からのMATCH信号234と、A N D論理ゲート243からのDONE信号239と、最も古い命令ポインタOLDEST_INSTR237と、を受け取る。ポインタOLDEST_INSTR237が、ローカルA R B S L O T 48が最も古い命令を持って、既にその命令を引退させるために準備している時を示す。ここで、現在のA R B S L O T 48と以前のA R B S L O T 48とに対応する引退ポインタRETを考慮することによって、OLDEST_INSTR信号237が生成される。もし、現在のA R B S L O T 48に対応する引退ポインタが存在し、以前のA R B S L O T 48のそれが消滅しているならば、現在のA R B S L O T 48が最も古く、OLDEST_INSTR信号237が発生する。しかしながら、もし、現在のA R B S L O T 48に対応する引退ポインタが存在し、また、以前のA R B S L O T 48のそれも存在するならば、現在のA R B S L O T 48が一番古いものではなく、OLDEST_INSTR信号237が消滅する。 20

【 0 0 8 1 】

ローカルA R B S L O T 48の中の命令が、引退メカニズム52(図3)によって引退するとき、A N D論理ゲート238が、否定回路245によって反転され、C C Cハザード信号125を消滅するA N D論理ゲート247に送り出す、待ち時間信号(WAIT) 244を生成する。従って、もし、A R B S L O T 48が、引退メカニズム52(図3)にその命令を始めるために準備するならば、C C Cハザード分解システム202がパニック・トラップを防ぎ、A R B S L O T 48を完了させる。この機能は、不必要で役に立たないパニック・トラップを防いで、パフォーマンスを最適化する。 30

【 0 0 8 2 】

記憶・ロード(ST/LD) ハザード分解システム

各々A R B S L O T 48は、実行されるロード命令がまだ実行されていないストア命令に依存したデータを利用する場合、パニック・トラップを生成する記憶・ロード(ST/LD) ハザード分解システム252を有し、それを利用する。ロード命令が、実行されていないストア命令に依存するデータを使用したとき、データは無効であり、よって、ロード命令の結果も無効となり捨てられ、ロード命令が再実行される。 40

【 0 0 8 3 】

図7において、S T / L Dハザード分解システム252は、キャッシュ・インデックスの一致があるか否かを示す比較信号258を得るために、遠隔A R B S L O T 48からの遠隔キャッシュ・インデックス253とローカル・レジスタ204の中のローカル・キャッシュ・インデックス207とを比較するキャッシュ・インデックス比較メカニズム228を持つ。一致する場合、それらのそれぞれの命令は、同じデータを必要とするオペレーションを指示する場合がある。後述するように、それらのキャッシュ・インデックスが一致するならば、命令が 50

同じデータを必要とするか否かを確定するために、更に、それらの相対データ・サイズに関係して、チェックが実行される。

【 0 0 8 4 】

スロット比較メカニズム259は、遠隔 A R B S L O T 48からの遠隔スロット番号 SLOT261受け取って、それを、2つの A R B S L O T 48の2つの命令がより新しい命令であることを示す YOUNGER信号264を得るために、ローカル・レジスタ204の中のローカル・スロット番号 SLOT263と比較する。命令が、順番に A R B S L O T 48に置かれ、したがって、スロット番号が、より新しい命令を含む A R B S L O T 48の番号を直接意味する。

【 0 0 8 5 】

A N D 論理ゲート266は、キャッシュ・インデックス比較メカニズム254からの比較信号258と、MATCH信号269を得るために命令に応じたデータ・サイズが一致するか否かを示す SIZE_MATCH信号267とを受け取る。ここで、このデータ・サイズは、1バイト(8ビット)、ハーフ・ワード(hw:16ビット)、1ワード(32ビット)、及びダブル・ワード(dw:64ビット)を含む。データが一致する場合、それらが重なればよく、データ・サイズが同一である必要はない。図8に示されるサイズ・マッチ決定メカニズム326によって、SIZE_MATCH信号267が得られる。MATCH信号269が発生するとき、キャッシュ・インデックス207、255が一致し、そして、それらに対応するデータ・サイズが一致する。

【 0 0 8 6 】

A N D 論理ゲート272は、YOUNGER信号264と、遠隔 A R B S L O T 48によって D C A C H E 24に送られるストア命令があるか否かを示す STORE_LAUNCH信号274と、ローカル A R B S L O T 48の中の命令がロード命令であるか否かを示す LOAD信号275と、キャッシュ・インデックスとそれらのデータ・サイズが一致するか否かを示す A N D 論理ゲート266からの MATCH信号269と、を受け取る。前述の信号に基づいて、A N D 論理ゲート272は、A N D 論理ゲート278、279に送る信号277を生成する。信号277によって、ラッチあるいは他の適当なロジック・メカニズムのような依存ストア(DS)インジケータ281でDS信号315が発生し、若しくは、A R B S L O T 48がすでに実行してデータ・アドレスを受け取ったか否かに応じて、S T / L D ハザード信号124(図4)が発生する。一般に、他の A R B S L O T 48が D C A C H E 24にアドレスを渡すことを実行するストア命令を持つとき、A N D 論理ゲート272は、ロード命令を持つ A R B S L O T 48の中で稼働するようになる。

【 0 0 8 7 】

A N D 論理ゲート278が A N D 論理ゲート272からの信号277と否定信号-DONE282を受け取り、DSインジケータ281でDS信号315を発生するトランジスタ286に対するアクチュエータ信号284を得る。

【 0 0 8 8 】

A N D 論理ゲート279が A N D 論理ゲート272からの信号277と肯定信号DONE288を受け取り、最終的にパニック・トラップを発生させる ST/LD_HAZARD信号124(図4)を得る。

【 0 0 8 9 】

A N D 論理ゲート292は、ローカル A R B S L O T 48の命令が遠隔 A R B S L O T 48の命令より古いかどうかを示すスロット比較メカニズム259からの YOUNGER信号264と、遠隔 A R B S L O T 48で実行されている命令がロード命令であるかどうかを示す LOAD_LAUNCH信号294と、ローカル A R B S L O T 48がストア命令を含むかどうかを示す STORE信号295と、対応するデータ・サイズと同様にキャッシュ・インデックス207、255が一致するかどうかを示す A N D 論理ゲートロジック266からのマッチ信号269と、を受け取る。前述の信号に応じて、A N D 論理ゲート292は、一致がトリガーとなって、同じ遠隔 A R B S L O T 48の中の D S インジケータ281を介して最終的にDS信号315を発生する ASSERT_DS信号を生成する。HIT_DS信号302を他の遠隔 A R B S L O T 48に送り出すドライバ299へ ASSERT_DS信号298が送られる。A N D 論理ゲート303と駆動信号305に基づく GRANTED_+_2信号307が真であるならば、HIT_DS信号302が、D S インジケータ281にDS信号315を発生するトランジスタ304を駆動する。D C A C H E 24へのアクセスが A R B S L O T 48に与えられた後に、GRANTED_+_2信号が2サイクルで発生するという問題がある。従って、ロード命令が D

10

20

30

40

50

CACHE 24で開始され、ローカルARBSLOT 48がロード命令に依存するデータに関連するストア命令を含むとき、AND論理ゲート292は、上述のように稼働する。この場合、ロード命令が始められていて、データがまだ受け取られていない場合には、パニック・トラップは必要ない。

【0090】

AND論理ゲート306が、ストア命令が遠隔ARBSLOT 48で引退するときを示す遠隔ARBSLOT 48からのSTORE_RETIRE信号308と、(アドレスMISS_ADDRのキャッシュ・インデックスがローカル・キャッシュ・インデックスと一致するかどうかを示す図6のAND論理ゲート228からの)MATCH信号234と、を受け取る。前述の信号に基づいて、AND論理ゲート306は、DSインジケータ281によって維持されたDS信号315を最終的に消滅するDEASSERT_DS信号309を生成する。図7で示すように、DEASSERT_DS信号309が、DSインジケータ281によって生じたDS信号315を消滅するトランジスタ312に作用する。

10

【0091】

DSインジケータ281によるDS信号315の発生は、いつARBSLOT 48がその命令を実行しその結果に対応するデータ・アドレスを受け取るかを示すDONE信号239の生成に影響する。これについて、AND論理ゲート243(図6)は、否定回路318でDS信号315の反転によって生成される-DS信号242、CAV信号245、-CP信号2461、-DM信号248、及びHITインジケータ206からHIT信号247によって、DONE信号239を生成する。DS信号315が発生されるとき、この信号は、DCACHE 24に対するアクセス要求からの依存するロード命令を防止する。

20

【0092】

更に、ストア命令が引退メカニズム52のパイプを通じて、スムーズに実行されなかったため、ARBSLOT 48がデータを得るためにDCACHE 24をもう一度アクセスするとき、ARBSLOT 48が、引退メカニズム52からHIT_RETRY信号322を受け取る。このHIT_RETRY信号322は、否定回路323を通じて、AND論理ゲート324へ渡される。AND論理ゲート324も、GRANT_+2信号327を受け取る。DCACHE アクセスが与えられた後、2サイクルでHIT_RETRY信号322が受け取られるとき、AND論理ゲート324は信号325を発生する。信号325は、順番に、CPインジケータ327から作用するトランジスタ326に作用する。PCインジケータ327は、それぞれARBSLOT 48がDCACHE 24をアクセスしたのか否かを示すCP信号246を消滅するか発生するために構成されるラッチか他の適当なロジック装置である。CPインジケータ327は、論理ハイ・ボルテージ(high voltage)に接続され、UPDATE信号123によって作用するトランジスタ328によって、CP信号246を発生するためにセットされる。さらに、CP信号246は、DONE信号239を生成するAND論理ゲート243に送出される-CP信号246'を生成するために、否定回路329に送出される。従って、HIT_RETRY信号322が発生するとき、-CP信号246'がAND論理ゲート243に発生せず、DONE信号239が消滅する。

30

【0093】

図8は、SIZE_MATCH信号267(図7)を生成する各ARBSLOT 48のサイズ一致決定ロジック326を示す。一般に、図8のロジック326は、ロード命令が、それらのそれぞれのデータ・サイズに基づくストア命令に依存するか否かを決定する。各命令と関連するデータは、次のサイズのいずれかである。1バイト(8ビット)、ハーフ・ワード(16ビット)、1ワード(32ビット)、又はダブル・ワード(64ビット)。データ・アドレスの重複がある限り、データが異なるサイズのとき、ロード・データは記憶データに依存することができる。例えば、ロード・バイトが記憶ワードと重複するならば、ロード・バイトは、記憶ワードに依存する場合がある。また例えば、記憶バイトがロード・ハーフ・ワードの範囲内であるならば、ロード・ハーフ・ワードは、記憶バイトに依存する場合がある。

40

【0094】

ARBSLOT 48の各々に対応するアドレスは、そのサイズを示すSIZE信号(例えば、2ビット)を含む。サイズの符号化を次に示す。「00」は1バイトを意味し、「01」は

50

ハーフ・ワードを意味し、「10」は1ワードを意味し、「11」はダブル・ワードを意味する。

【0095】

図8において、SIZE_MATCHメカニズム326は、開始命令に対応するREMOTE_ADDR[62]信号331と、ローカル命令に対応するLOCAL_ADDR[62]信号256a、334とを受け取るXOR論理ゲート328を含む。これらの信号に基づいて、XOR論理ゲート328が、これらのアドレス・ビットが一致するか否かを示す信号332を生成する。

【0096】

NAND論理ゲート334が、遠隔発生信号と対応しているREMOTE_SIZE[0]信号333と、ローカル命令と対応しているLOCAL_SIZE[0]信号335とを受け取り、これらの1つ又は両方のアドレスがハーフ・ワード・サイズより大きいサイズで発生したことを示す信号336を生成する。よって、オーバーラップ（重複）オペレーションは、これらの命令のアドレス・ビット[63,62]の一致を必要としない。

10

【0097】

XOR論理ゲート338が、遠隔発生信号と対応しているREMOTE_ADDR[63]信号339と、ローカル命令と対応しているLOCAL_ADDR[63]信号342とを受け取り、これらのアドレス・ビットが一致するか否かを示す信号346を生成する。

【0098】

NAND論理ゲート348が、遠隔発生信号と対応しているREMOTE_SIZE[1]信号352と、ローカル命令と対応しているLOCAL_SIZE[1]信号354とを受け取り、これらの1つ又は両方のアドレスが1バイト・サイズより大きいサイズで発生したことを示す信号356を生成する。よって、オーバーラップ（重複）オペレーションは、この命令のアドレス・ビット[63]の一致を必要としない。

20

【0099】

AND論理ゲート358は、信号332、336を受け取って、それらの信号から、アドレス・ビット[62]が一致しないことを示す信号362を生成する。

【0100】

AND論理ゲート364が、信号346、356を受け取って、それらの信号から、アドレス・ビット[63]が一致するか否かを示す信号366を生成する。

【0101】

NOR論理ゲート368が、信号362、366を受け取って、SIZE_MATCH信号267を生成する。この信号により、最適な比較ができるようにデータ・サイズが一致する。

30

【0102】

サイズ一致決定メカニズム326（図8）と同様に、ST/LDハザード分解システム252（図7）の全体的なオペレーションを以下に説明する。

【0103】

ストア命令が開始されローカルARBSLOT48内でロード命令と一致するとき、ストア命令よりロード命令が新しいかどうかを確認される。ロード命令がストア命令より新しい場合、なにも起きない。ロード命令がストア命令より新しい場合、ロード命令が実行されていれば、ST/LD_HAZARD信号124が発生し、ロード命令が実行されていないならば、DSインジケータが発生する。

40

【0104】

ロード命令が開始され他のARBSLOT48内でストア命令と一致するとき、ストア命令を持っているARBSLOT48は、ロード命令を持っているARBSLOT48の中で、DS信号315を発生させる。

【0105】

他のARBSLOT48の中で一致するストア命令が引退するとき、DSインジケータ281によって発生し消滅するDS信号315は、ARBSLOT48の中のロード命令によってクリアされる。DSインジケータがARBSLOT48で消滅するとき、ARBSLOT48は、DCACHE 24をアクセスできるようになる。

50

【0106】

本願発明は、例として次の実施態様を含む。

(1) プロセッサの順序外命令の実行で生じた不正確な結果から回復するためのシステムであって、前記プロセッサに接続され、命令を読み出すフェッチ・メカニズムと、前記プロセッサに接続され、前記命令を前記フェッチ・メカニズムから受け取り、前記命令を順序外で実行する再処理メカニズムと、前記再処理メカニズムに接続され、前記再処理メカニズムの命令の構造的に不正確な実行を認識し、前記実行の後、前記命令をパニック・トラップ・インジケータと関連付けるパニック・トラップ論理手段と、前記再処理メカニズムと前記フェッチ・メカニズムに接続され、前記再処理メカニズムを消去し、前記フェッチ・メカニズムに再読出しをさせ、前記パニック・トラップ・インジケータに基づいて前記命令を開始するトラップ・ベクトル・ゼネレータと、を有する前記システム。

10

(2) 前記パニック・トラップ論理手段が、ロード命令がストア命令より先に実行される場合と前記ロード命令が前記ストア命令によって供給されるデータに依存する場合とを決定し、前記決定に基づいて前記パニック・トラップインジケータを発生する依存論理手段を有する、(1)記載のシステム。

(3) 更に、命令が、可干渉性チェック信号に基づいて転送されるデータを必要とする場合を決定するハザード論理手段を有する(1)記載のシステム。

(4) 前記フェッチ・メカニズムが前記パニック・トラップ・インジケータと前記命令の1つとを関連付ける手段を有する(1)記載のシステム。

(5) 更に、前記命令の1つがデータ・キャッシュへのアクセスの結果パリティ・エラーを生じたことを検出し、前記パニック・トラップ・インジケータと前記命令を関連付ける手段を有する(1)記載のシステム。

20

【0107】

(6) 順序外命令を実行するプロセッサにおいて、不正確な実行結果を生じる命令から回復するシステムであって、前記プロセッサに接続され、命令を読み出し、パニック・トラップ・インジケータと再実行される命令とを関連付けるフェッチ・メカニズムと、前記プロセッサに接続され、前記命令を前記フェッチ・メカニズムから受け取り、前記命令を順序外で実行する再処理メカニズムと、前記再処理メカニズムと前記フェッチ・メカニズムに接続され、命令の実行後に該命令を受け取り、前記再処理メカニズムを消去し、前記フェッチ・メカニズムに前記パニック・トラップ・インジケータに基づいて前記命令を順番に再読出しをさせるトラップ・ベクトル・ゼネレータと、を有する前記システム。

30

【0108】

(7) 順序外命令を実行するプロセッサに接続されたデータ・キャッシュによって発生したパリティ・エラーから回復するためのシステムであって、命令を読み出すフェッチ・メカニズムと、前記命令を前記フェッチ・メカニズムから受け取り、前記命令を順序外で実行し、前記データ・キャッシュへアクセスする再処理メカニズムと、アクセス中にパリティ・エラーが発生したことを検知し、アクセスによってパリティ・エラーが生じたときパニック・トラップ・インジケータを生成するデータ・キャッシュと、実行後に前記命令を引退させるタイミングを操作し、前記パニック・トラップ・インジケータを認識し、前記再処理メカニズムを消去し、前記フェッチ・メカニズムに再読出しをさせるパニック・トラップゼネレータと、を有する前記システム。

40

【0109】

(8) プロセッサにおいて、順序外命令の実行で生じた不正確な結果から回復するための方法であって、再処理メカニズムに命令を集めるステップと、前記命令を順序外で実行するステップと、命令の構造的に不正確な実行を認識するステップと、前記命令に応じて前記不正確な命令を示すパニック・トラップ・インジケータを発生するステップと、実行後前記命令を引退させる場合、前記パニック・トラップ・インジケータ認識し、前記再処理メカニズムを消去し、無効命令を順に再読出しするステップと、を有する前記方法。

【0110】

(9) 順序外命令を実行するプロセッサにおいて、不正確な実行結果を生じる命令から回

50

復する方法であって、命令を読出すステップと、前記読出し中に、命令が無効であることを決定するステップと、パニック・トラップ・インジケータと前記無効命令とを関連付けるステップと、前記無効命令を他の命令と共に再処理メカニズムへ配置するステップと、前記命令を前記再処理メカニズムの中で順序外に実行するステップと、実行後に前記命令を引退させる場合、パニック・トラップ・インジケータを認識し、前記再処理メカニズムを消去し、前記無効命令を順番に再読出しするステップと、を有する前記方法。

(10) 順序外命令を実行するプロセッサに接続されたデータ・キャッシュによって発生したパリティ・エラーから回復するための方法であって、再処理メカニズムに命令を集めるステップと、前記命令を順序外で実行するステップと、前記キャッシュにアクセスする命令を実行するステップと、前記アクセス中にパリティ・エラーが発生したことを検知するステップと、前記プロセッサにおいて、前記データ・キャッシュからトラップ・ベクトル・ゼネレータへパニック・トラップ・インジケータを生成して転送するステップと、を有する前記方法。

10

【0111】

【発明の効果】

本発明によると、命令の順序外実行から生成された不正確な結果からプロセッサを回復するのに使われ、更には、タイミングと同期を図るパニック・トラップ・システム及び方法を提供することができる。

【図面の簡単な説明】

【図1】本発明の新しいパニック・トラップ・システム及び方法を実行するコンピュータを示すブロック図である。

20

【図2】図1のプロセッサの命令読出/実行システムの実行処理を示すブロック図である。

【図3】図1の新しいパニック・トラップ・システムの実行処理を示すブロック図である。

【図4】図3の各アドレス・リオーダ・バッファ・スロット(ARBSLOT)中のパニック・インジケータを発生するトラップ・インジケータ・ロジックの実行処理を示すブロック図である。

【図5】図3の引退メカニズム(RETIRE)の実行処理を示すブロック図である。

【図6】図4のハザード信号CCC_HAZARD(最終的にパニック・トラップ)を生成するためのキャッシュ可干渉性チェック(CCC)システムを示すブロック図である。

30

【図7】図4のハザード信号ST/LD_HAZARD(最終的にパニック・トラップ)を生成するための記憶/ロード(ST/LD)ハザード分解システムの実行処理を示すブロック図である。

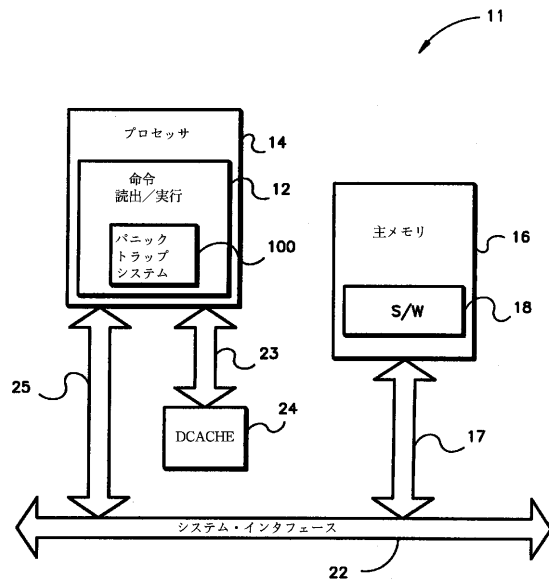
【図8】図7の信号SIZE_MATCHを生成するサイズ・マッチ決定メカニズムの実行処理を示すブロック図である。

【符号の説明】

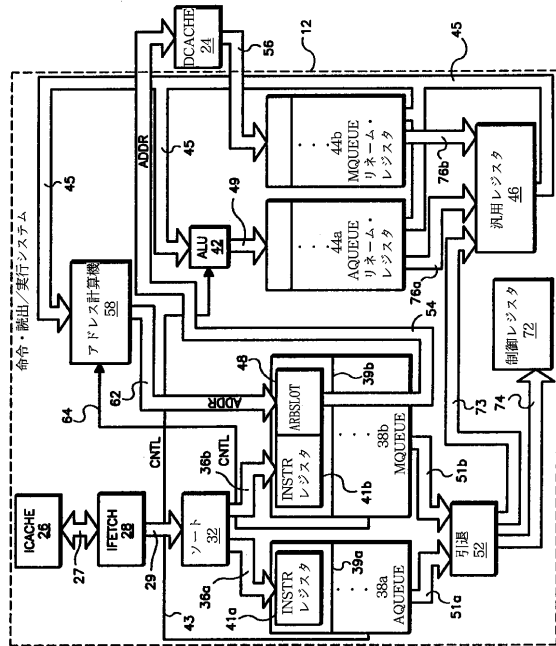
- 12 命令読出/実行システム
- 24 データ・キャッシュ
- 26 命令キャッシュ
- 28 命令フェッチ・メカニズム
- 32 ソート・メカニズム
- 38a ALU待ち行列
- 38b メモリ待ち行列
- 42 算術演算論理装置
- 48 アドレス・リオーダ・バッファ・スロット
- 52 引退メカニズム

40

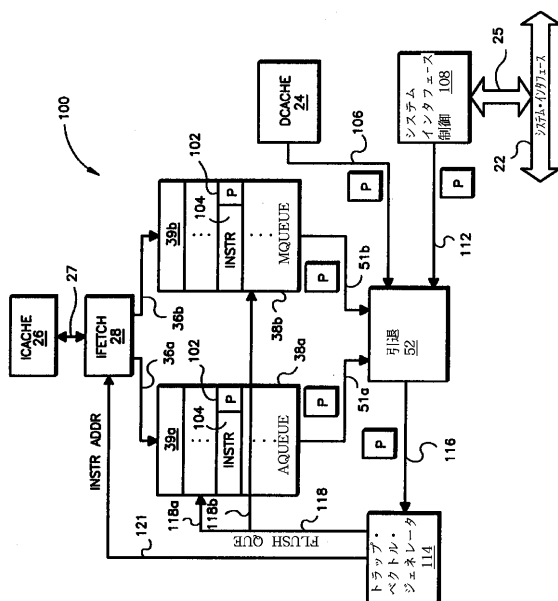
【図 1】



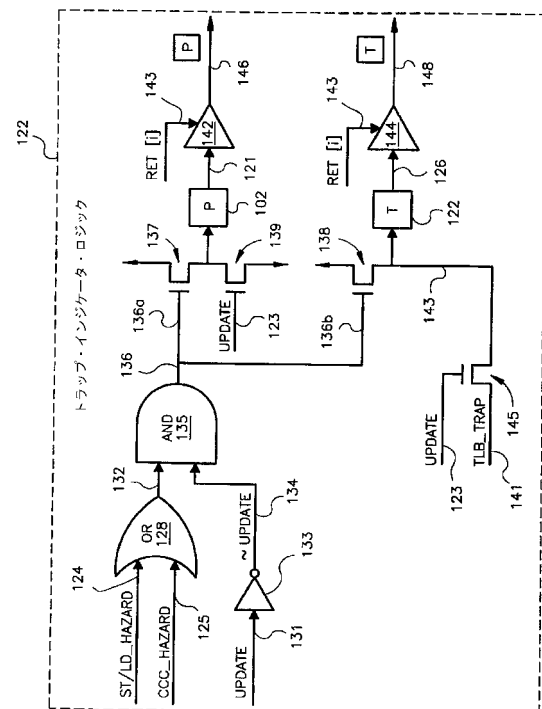
【図 2】



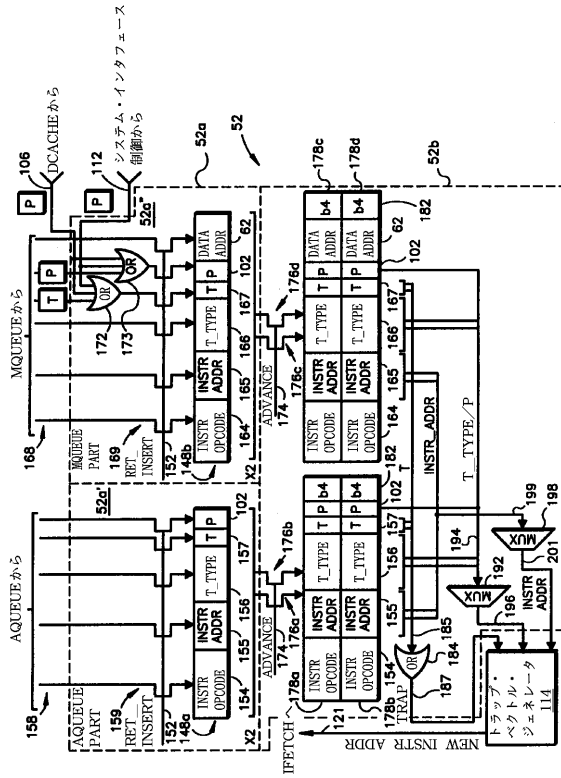
【図 3】



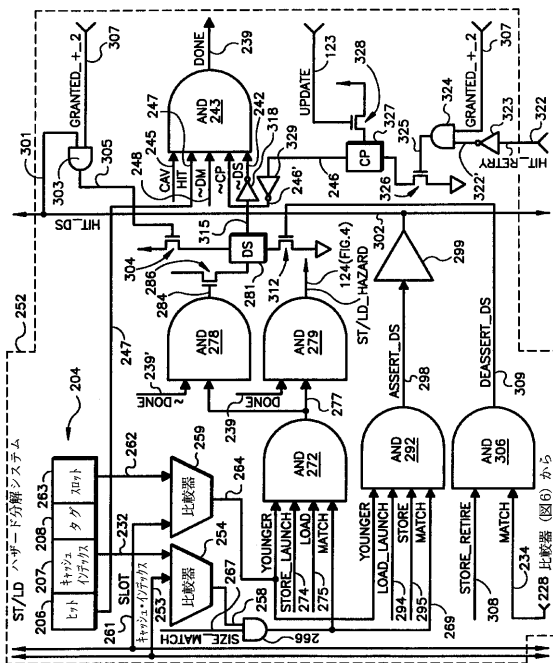
【図 4】



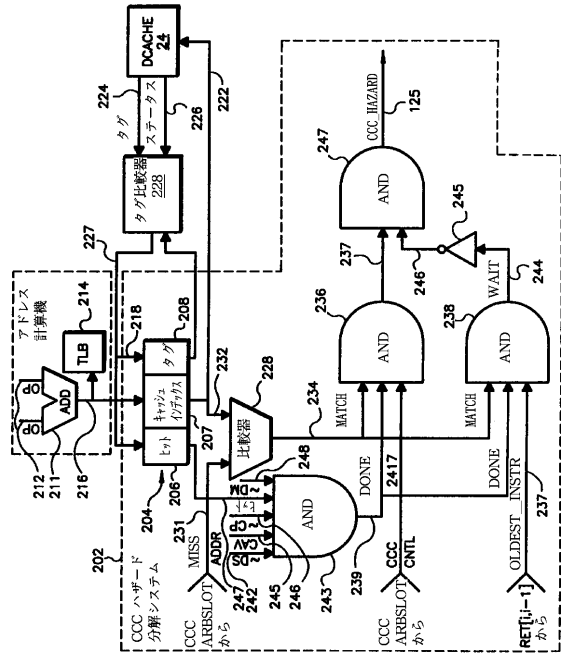
【図 5】



【図 7】



【図 6】



フロントページの続き

(56)参考文献 特開平7 - 3 2 5 7 1 6 (J P , A)
特開平7 - 1 6 0 5 0 1 (J P , A)

(58)調査した分野(Int.Cl.⁷, D B名)
G06F 9/38