

【公報種別】特許法第17条の2の規定による補正の掲載  
 【部門区分】第6部門第3区分  
 【発行日】平成26年8月14日(2014.8.14)

【公表番号】特表2012-507794(P2012-507794A)  
 【公表日】平成24年3月29日(2012.3.29)  
 【年通号数】公開・登録公報2012-013  
 【出願番号】特願2011-534680(P2011-534680)  
 【国際特許分類】

G 0 6 F 9/30 (2006.01)

G 0 6 F 9/318 (2006.01)

【 F I 】

G 0 6 F 9/30 3 3 0 A

G 0 6 F 9/30 3 1 0 B

G 0 6 F 9/30 3 2 0 C

【誤訳訂正書】

【提出日】平成26年6月3日(2014.6.3)

【誤訳訂正1】

【訂正対象書類名】特許請求の範囲

【訂正対象項目名】全文

【訂正方法】変更

【訂正の内容】

【特許請求の範囲】

【請求項1】

命令キューと、

前記命令キューに格納されている第1のフュージョン可能な命令の処理を最大で閾時間だけ遅延させて、前記第1のフュージョン可能な命令とフュージョン可能であるが前記命令キューに未だ格納されていない第2のフュージョン可能な命令が、該第2のフュージョン可能な命令が前記閾時間内に前記命令キューに格納される場合に、前記第1のフュージョン可能な命令とフュージョンされ得るようにするロジックと、

前記第2のフュージョン可能な命令が前記閾時間内に前記命令キューに格納された場合に、前記第1及び第2のフュージョン可能な命令をフュージョンする命令フュージョンロジックと、

を有する装置。

【請求項2】

前記第1のフュージョン可能な命令及び前記第2のフュージョン可能な命令は、前記命令キューに格納される前にフェッチ・バウンダリにわたって格納される、

請求項1に記載の装置。

【請求項3】

前記ロジックは、前記第1のフュージョン可能な命令が前記命令キューに格納されている最後の命令である場合にのみ、前記第1のフュージョン可能な命令の処理を遅延させる、

請求項1に記載の装置。

【請求項4】

前記ロジックは、前記閾時間に対応する閾数のサイクルに達するまで、前記第1のフュージョン可能な命令が前記命令キューに格納され且つ前記命令キューにおける最後の命令であるサイクルごとに1つインクリメントされるカウンタを有する、

請求項1に記載の装置。

【請求項5】

中間動作が前記命令キューに格納されている前記第 1 のフュージョン可能な命令と前記命令キューに格納される前記第 2 のフュージョン可能な命令との間で実行される場合に、P F R Q が前記第 1 のフュージョン可能な命令及び前記第 2 のフュージョン可能な命令に対応するエントリをロックすることを防止する状態機械を更に有し、該防止は、前記命令キュー内の命令が該エントリの充填領域を終わらせるが前記 P F R Q にセーブされた線形命令ポインタで開始しないときを検出して該エントリを割当解除することによる、

請求項 1 に記載の装置。

【請求項 6】

前記中間動作は、前記命令キューをクリアすることである、  
請求項 5 に記載の装置。

【請求項 7】

命令キュー内で目下アクセスされている命令が、前記命令キューに格納されるいずれかの後続の命令とフュージョン可能であるかどうかを決定するステップと、

前記目下アクセスされている命令が前記命令キューに格納される後続の命令とフュージョン可能でない場合に、前記命令キューの次の命令にアクセスし、遅延カウンタをリセットするステップと、

前記目下アクセスされている命令がフュージョン可能であり、且つ、前記命令キューにおける最後の命令である場合に、前記遅延カウンタをインクリメントするステップと、

前記目下アクセスされている命令及び前記後続の命令がフュージョン可能であり、且つ、前記遅延カウンタが閾値に達する前に前記後続の命令が前記命令キューに格納される場合に、前記目下アクセスされている命令を前記後続の命令とフュージョンするステップと

を有する方法。

【請求項 8】

前記目下アクセスされている命令及び前記後続の命令がフュージョン可能でない場合に、前記目下アクセスされている命令を前記後続の命令とは別々に処理するステップを更に有する、

請求項 7 に記載の方法。

【請求項 9】

前記遅延カウンタが前記閾値に達した場合に、前記目下アクセスされている命令を前記後続の命令とは別々に処理するステップを更に有する、

請求項 7 に記載の方法。

【請求項 10】

前記目下アクセスされている命令及び前記後続の命令がフュージョン可能であり、且つ、中間イベントが、前記後続の命令がキャッシュに格納される前であって、前記目下アクセスされている命令が前記キャッシュに格納された後に行われる場合に、P F R Q が前記目下アクセスされている命令及び前記後続の命令に対応するエントリをロックすることを防止するステップを更に有し、該防止は、前記命令キュー内の命令が該エントリの充填領域を終わらせるが前記 P F R Q にセーブされた線形命令ポインタで開始しないときを検出して該エントリを割当解除することによる、

請求項 7 に記載の方法。

【請求項 11】

第 1 のフュージョン可能な命令及び第 2 のフュージョン可能な命令を夫々第 1 のアクセス・バウンダリ及び第 2 のアクセス・バウンダリに格納する記憶部と、

前記第 1 のフュージョン可能な命令及び前記第 2 のフュージョン可能な命令を命令キューにフェッチするフェッチロジックを有するプロセッサと、

前記命令キューからの前記第 1 のフュージョン可能な命令の読出を閾数のサイクル分遅延させる遅延ロジックと

前記第 2 のフュージョン可能な命令が、前記第 1 のフュージョン可能な命令の後であって、前記閾数のサイクルに達する前に前記命令キューに格納される場合に、前記第 1 のフ

フュージョン可能な命令及び前記第2のフュージョン可能な命令をフュージョンする命令フュージョンロジックと

を有するシステム。

【請求項12】

前記第1のフュージョン可能な命令が前記命令キューにおける唯一の命令である場合にインクリメントし、前記閾数のサイクルに達したときにカウントを停止するカウンタを更に有する、

請求項11に記載のシステム。

【請求項13】

前記カウンタは、前記閾数のサイクルに達する前に前記第2のフュージョン可能な命令が前記命令キューに格納される場合にリセットされる、

請求項12に記載のシステム。

【請求項14】

前記記憶部は命令キャッシュを有し、前記第1のアクセス・バウンダリ及び前記第2のアクセス・バウンダリは夫々サイズが64バイトである、

請求項11に記載のシステム。

【請求項15】

前記記憶部はダイナミックランダムアクセスメモリを有し、前記第1のアクセス・バウンダリ及び前記第2のアクセス・バウンダリは夫々サイズが4096バイトである、

請求項11に記載のシステム。

【請求項16】

前記第1のフュージョン可能な命令はCMP/TEST命令であり、前記第2のフュージョン可能な命令はJCC命令である、

請求項11に記載のシステム。

【請求項17】

前記サイクルの閾数は2である、

請求項16に記載のシステム。

【請求項18】

中間イベントが、キャッシュに格納されている前記第1のフュージョン可能な命令と前記キャッシュに格納されている前記第2のフュージョン可能な命令との間で実行される場合に、PF RQが前記第1のフュージョン可能な命令及び前記第2のフュージョン可能な命令に対応するエントリをロックするのを防止する状態機械を更に有し、該防止は、前記命令キュー内の命令が該エントリの充填領域を終わらせるが前記PF RQにセーブされた線形命令ポインタで開始しないときを検出し、それに応答して該エントリを割当解除することによる、

請求項11に記載のシステム。

【誤訳訂正2】

【訂正対象書類名】明細書

【訂正対象項目名】全文

【訂正方法】変更

【訂正の内容】

【発明の詳細な説明】

【発明の名称】有効な命令フュージョンを進展させる技術

【技術分野】

【0001】

本発明の実施形態は、概して、情報処理の分野に関し、より具体的には、コンピュータシステム及びマイクロプロセッサにおける命令フュージョン (instruction fusion) の分野に関する。

【背景技術】

【0002】

命令フュージョンは、2つの命令を、プロセッサ内の1つの動作（又はマイクロオペレーション“uop”）シーケンスをもたらす単一の命令にまとめる処理である。プロセッサ命令キュー（IQ）（instruction queue）に格納されている命令は、IQから読み出された後、命令デコーダへ送信される前に、あるいは、命令デコーダによってデコードされた後に、フュージョンされてよい。通常、命令がデコードされる前に起こる命令フュージョンは「マクロフュージョン（macro-fusion）」と呼ばれ、他方、命令がデコードされた後に起こる命令フュージョンは「マイクロフュージョン（micro-fusion）」と呼ばれる。マクロフュージョンの例は、比較（“CMP”）命令又はテスト（“TEST”）命令（“CMP/TEST”）と条件付きジャンプ（“JCC”）命令との結合である。CMP/TEST及びJCCの命令の組は、プログラムにおいて、例えば、比較が行われ、比較の結果に基づいて、枝分かれが起こるか否かが決まるところのループの終わりに定期的に起こってよい。マクロフュージョンは命令スループットを有効に増大させるるので、可能な限り命令をフュージョンする多くの機会を見出すことが望ましい。

#### 【0003】

何らかの先行技術のプロセッサ・マイクロアーキテクチャで見出される命令フュージョンの機会に関し、CMP/TEST及びJCCの両命令は、同時にIQにある必要があり、それにより、それらの命令は、命令がIQから読み出されるときにフュージョンされ得る。しかし、IQにフュージョン可能なCMP/TEST命令はあるが、それ以外の命令はIQに書き込まれていない場合（すなわち、CMP/TEST命令がIQにおける最後の命令である場合）、CMP/TEST命令は、たとえプログラム順序における次の命令がJCC命令であるとしても、IQから読み出されて、フュージョンされることなくデコーダに送られうる。フュージョンの機会を逸する例は、CMP/TEST及びJCCが図らずも記憶バウンダリ（例えば、16バイトバウンダリ）にわたって起こり、CMP/TESTが1サイクルにおいてIQに書き込まれ、JCCが次のサイクルにおいてIQに書き込まれる場合である。この場合に、失速状態（stalling condition）が存在しないならば、JCCは、CMP/TESTがIQから読み出されるのと同時に又はその後IQに書き込まれるので、フュージョンの機会をのがし、IQの多数の不必要な読出が引き起こされ、命令スループットが低下するとともに、電力消費は増大する。

#### 【図面の簡単な説明】

#### 【0004】

【図1】本発明の少なくとも1つの実施形態が使用されるマイクロプロセッサのブロック図を表す。

【図2】本発明の少なくとも1つの実施形態が使用されるバス共有型コンピュータシステムのブロック図を表す。

【図3】本発明の少なくとも1つの実施形態が使用されるポイント・ツー・ポイント相互接続型コンピュータシステムのブロック図を表す。

【図4】本発明の少なくとも1つの実施形態を実施するために使用される状態機械のブロック図を表す。

【図5】本発明の少なくとも1つの実施形態を実行するために使用される動作のフロー図である。

【図6】少なくとも1つの実施形態で実行される動作のフロー図である。

#### 【発明を実施するための形態】

#### 【0005】

本発明の実施形態について、限定ではなく、一例として、添付の図面の各図を参照して説明する。図中、同じ参照符号は同じ要素を表している。

#### 【0006】

本発明の実施形態は、プロセッサにおける命令スループットを改善し、及び/又は、プロセッサの電力消費を削減するために、使用されてよい。一実施形態において、さもなければ命令フュージョンの機会を逸する可能性があるものが見つけられ、結果として、命令フュージョンが起こり得る。一実施形態において、逸する可能性がある命令フュージョン

の機会、命令キュー（IQ）からの最後の命令の読出又はIQから読み出される最後の命令の発行を閾数のサイクルだけデコード相まで遅らせることによって見つけられ、それにより、後続のフュージョン可能な命令は全てフェチされてIQに格納され（又は、必ずしもIQに格納される必要はなく、少なくとも識別され）、次いで、前記最後のフュージョン可能な命令とフュージョンされる。一実施形態において、閾数のサイクルだけ第1のフュージョン可能な命令の読出又は発行を遅延させることは、そのようにすることが、さもなければフュージョン可能な2つの命令が、単一の命令としてではなく、別々にデコードされて処理されてしまうことを回避するので、プロセッサ性能を改善することができる。

#### 【0007】

待機サイクルの閾数の選択は、特定の実施形態が使用されるマイクロアーキテクチャに依存してよい。例えば、一実施形態において、サイクルの閾数は2であってよく、一方、他の実施形態において、サイクルの閾数は2よりも多くても又は少なくともよい。一実施形態において、待機サイクルの閾数は、フュージョン可能な命令を別々の命令として処理することに対する、後続のフュージョン可能な命令を待つことの全体的なレイテンシ/性能上の利点を保ちながら、後続のフュージョン可能な命令がIQに格納されるのを待つための最大の時間量を提供する。他の実施形態において、電力がより重要である場合に、例えば、待機サイクルの閾数は、たとえ待機サイクルの数が（一時的ではあるが）命令スループットの低下を引き起こすとしても、余分の電力が2つのフュージョン可能な命令を別々に処理するために使用されないことを確かにするために、より大きくてよい。

#### 【0008】

図1は、本発明の少なくとも1つの実施形態が使用されてよいマイクロプロセッサを表す。具体的に、図1は、1又はそれ以上のプロセッサコア105及び110を有するマイクロプロセッサ100を表し、各プロセッサコアは、夫々自身に関連付けられたローカルキャッシュ107及び113を有する。また、図1には、ローカルキャッシュ107及び113の夫々に格納されている情報の少なくとも一部のバージョンを記憶することができる共有キャッシュメモリ115が表されている。幾つかの実施形態において、マイクロプロセッサ100は、図1に図示されていない他のロジック、例えば、集積メモリコントローラ、集積グラフィックコントローラ、及びI/O制御等のコンピュータ内の他の機能を実行する他のロジックを有してよい。一実施形態において、マルチプロセッサシステムにおける各マイクロプロセッサ又はマルチコアプロセッサにおける各プロセッサコアは、少なくとも1つの実施形態に従って、割込通信技術を可能にするロジック119を有してよく、又は別なふうに該ロジック119と関連付けられてよい。ロジックは、幾つかの先行技術実施よりも効率的な命令フュージョンを可能にする回路、ソフトウェア又はそれら両方を有してよい。

#### 【0009】

一実施形態において、ロジック119は、命令フュージョンの機会を逸する可能性を減らすロジックを有してよい。一実施形態において、ロジック119は、IQ又は他のフェッチ命令記憶構造に記憶されている後続の命令がない場合に、IQからの第1の命令（例えば、CMP）の読出を遅延させる。一実施形態において、ロジック119は、IQを読み出すこと、又は第1のフュージョン可能な命令をデコーダ若しくは他の処理ロジックへ発行する前の閾数のサイクル（例えば、2サイクル）の間に第1のフュージョン可能な命令の読出又は発行を引き起こし、それにより、（例えば、対象である2つの命令が、異なる記憶バウンダリにおけるメモリ又はキャッシュに格納されているために）IQに未だ記憶されていない、第1の命令とフュージョン可能な第2のフュージョン可能な命令が存在する場合に、これら2つのフュージョン可能な命令をフュージョンする機会をのがさない。幾つかの実施形態において、閾値は固定であってよく、一方、他の実施形態において、閾値は可変であってよく、ユーザによって又はユーザに依存しないアルゴリズムに従って変更可能である。一実施形態において、第1のフュージョン可能な命令はCMP命令であり、第2のフュージョン可能な命令はJCC命令である。他の実施形態において、第1の

命令及び第 2 の命令の一方又は両方は C M P 又は J C C 命令ではなく、如何なる フュージョン可能な命令であってもよい。更に、本発明の実施形態は、2 よりも多い命令を フュージョンすることに適用されてよい。

【 0 0 1 0 】

図 2 は、例えば、本発明の実施形態が使用されてよいフロントサイドバス ( F S B ) 型コンピュータシステムを表す。いずれのプロセッサ 2 0 1、2 0 5、2 1 0 又は 2 1 5 も、プロセッサコア 2 2 3、2 2 7、2 3 3、2 3 7、2 4 3、2 4 7、2 5 3、2 5 7 の 1 つの中の又は別なふうに該 1 つと関連付けられているいずれかの ローカルのレベル 1 ( L 1 ) キャッシュメモリ 2 2 0、2 2 5、2 3 0、2 3 5、2 4 0、2 4 5、2 5 0、2 5 5 の情報にアクセスしてよい。更に、いずれのプロセッサ 2 0 1、2 0 5、2 1 0 又は 2 1 5 も、共有レベル 2 ( L 2 ) キャッシュ 2 0 3、2 0 7、2 1 3、2 1 7 のいずれか 1 つの情報に、又はチップセット 2 6 5 を介してシステムメモリ 2 6 0 の情報にアクセスしてよい。図 2 におけるプロセッサの 1 つ又はそれ以上は、少なくとも 1 つの実施形態に従って、命令 フュージョン の効率改善を可能にするロジック 2 1 9 を有し、又は別なふうに該ロジック 2 1 9 と関連付けられてよい。

【 0 0 1 1 】

図 2 に表されている F S B 型コンピュータシステムに加えて、他のシステム構成が、ポイント・ツー・ポイント ( P 2 P ) 相互接続型システム及びリング相互接続型システムを含め、本発明の種々の実施形態とともに使用されてよい。図 3 の P 2 P システムは、例えば、複数のプロセッサを有してよい。それらの中の 2 つのプロセッサ 3 7 0 及び 3 8 0 のみが例として示されている。プロセッサ 3 7 0 及び 3 8 0 は、夫々、メモリ 3 2、3 4 と接続する ローカルのメモリコントローラハブ ( M C H ) 3 7 2、3 8 2 を有してよい。プロセッサ 3 7 0 及び 3 8 0 は、ポイント・ツー・ポイント ( P t P ) インターフェース回路 3 7 8、3 8 8 を用いて P t P インターフェース 3 5 0 を介してデータを交換してよい。プロセッサ 3 7 0 及び 3 8 0 は、夫々、P t P インターフェース回路 3 7 6、3 9 4、3 8 6、3 9 8 を用いて個別の P t P インターフェース 3 5 2、3 5 4 を介してチップセット 3 9 0 とデータを交換してよい。チップセット 3 9 0 は、更に、高性能グラフィックスインターフェース 3 3 9 を介して高性能グラフィックス回路 3 3 8 とデータを交換してよい。本発明の実施形態は、任意数のプロセッシングコアを有する何らかのプロセッサ内に、又は図 3 の P t P バスエージェントの夫々の中に、配置されてよい。一実施形態において、いずれのプロセッサコアも、ローカルキャッシュメモリ ( 図示せず。 ) を有し、又は別なふうに該 ローカルキャッシュメモリ と関連付けられてよい。更に、共有キャッシュ ( 図示せず。 ) が、いずれか一方のプロセッサ内に含まれ、P 2 P 相互接続を介してそれらのプロセッサと接続されてよく、それにより、一方又は両方のプロセッサの ローカルキャッシュ情報 は、プロセッサが低電力モードに入る場合に、共有キャッシュに格納されてよい。図 3 におけるプロセッサ又はコアの 1 つ又はそれ以上は、少なくとも 1 つの実施形態に従って、命令 フュージョン の効率改善を可能にするロジック 3 1 9 を有し、又は別なふうに該ロジック 3 1 9 と関連付けられてよい。

【 0 0 1 2 】

少なくとも 1 つの実施形態において、第 2 の フュージョン可能な命令 は、何らかの中間動作、例えば、( 第 1 及び第 2 の フュージョン可能な命令 の間に起こる ) I Q クリア動作が起こる前に、I Q に格納されないことがあり、このとき、さもなければフュージョン可能な 2 つの命令をフュージョンする機会をのがす。キャッシュ ( 又はバッファ ) が、( 命令が I Q から読み出されてデコードされた後に ) プロセッサによる実行のためにスケジューリングされるべき関連するデコードされた命令シーケンス又は u o p ( 例えば、デコードされたストリームバッファ ( “ D S B ” ) ( decoded stream buffer )、トレースキャッシュ ( “ T C ” ) ( trace cache ) ) を記憶するところの一実施形態において、第 1 の フュージョン可能な u o p ( 例えば、C M P ) は、同じアドレス可能な範囲 ( 例えば、同じキャッシュウェイ ) 内に フュージョン可能な第 2 の u o p ( 例えば、J C C ) が格納されることなく、キャッシュに格納されることがある。これは、例えば、J C C が ( キャ

ッシュミスのために) キャッシュラインを横断し、又は(トランスレーションルックアサイドバッファ(TLB)ミスのために) ページ・バウンダリを横断している場合に、起こりうる。その場合に、キャッシュは、JCCなしでCMPを格納し得る。その後、CMPが格納された後、しかし、JCCがキャッシュに格納される前に、プロセッサコア・パイプラインが(例えば、“クリア”信号がアサートされたために) クリアされる場合は、キャッシュは、JCCなしで、そのウェイのうちの1つにCMPのみを格納する。

【0013】

CMPを格納するキャッシュラインのその後の検索において、キャッシュは、欠けているJCCを、ミスしたアクセスとして解釈してよく、JCCは、次のキャッシュ充填(フィル)動作のために付加ポイントしてマークされてよい。しかし、この付加ポイントは、CMP+JCCがIQからフュージョンされたものとして読み出され得るので、見つけれないことがある。従って、要求されるJCCは、IQから到来する充填されるいずれのuopとも一致せず、従って、キャッシュは、欠けているJCCを充填することができず、フュージョンされたCMP+JCCが期待されるラインにおいて継続的に失敗する。更に、保留中のフィル要求キュー(PFRQ)(pending fill request queue)がuopキャッシュフィル要求を格納するために使用されるところの一実施形態において、特定のフュージョンされた命令の充填のためにリザーブされていたエントリは(期待されるフュージョンされた命令が決して起こらないために)解放せず、次のクリア動作まで無駄なままである。一実施形態において、PFRQエントリロックは、欠けているフュージョンされた命令のエントリがアクセスされるたびに起こってよく、従って、同じ場所へのその後のあらゆる充填を妨げうる。

【0014】

PFRQエントリの誤った又は好ましくないロックを防ぐために、一実施形態において、IQから読み出されるuopをモニタして、対応するPFRQエントリを有する領域(例えば、充填のためにマークされた領域)が、例えば、充填開始点が検出されることなくエントリの最後のuopに達したために完全にミスした場合を検出するよう、状態機械(state machine)が使用されてよい。一実施形態において、状態機械は、この条件が満たされる場合にPFRQエントリを解放してよい。他の実施形態において、好ましくないPFRQエントリのロックは、フュージョン可能な命令が両方とも存在しない場合には、IQから読み出されるフュージョン可能な命令をキャッシュ内に生成しないことによって、回避されてよい。例えば、CMPの後に非JCC命令が続く場合、フュージョンされた命令エントリがキャッシュ内に生成され得るが、CMPが(例えば、闕待機時間が満了した後に)単独でIQから読み出される場合に限り、それは、キャッシュに充填されないフュージョンされた命令エントリである。他の実施形態において、スキップされた充填領域を状態機械が検出した回数がカウントされ、充填領域がスキップされた闕カウント数の後に、キャッシュフラッシュ又は無効動作が実行されてよい。そして、その充填領域はキャッシュから除去されて、フュージョンされた命令が再充填されてよい。

【0015】

図4は、一実施形態に従う状態機械を表し、当該状態機械は、IQにおけるのがしたフュージョン可能な命令による好ましからざるPFRQエントリロック状態を回避するために使用されてよい。IQ内の命令が充填のためにマークされた領域にない状態401において、充填領域にマッピングされる命令(キャッシュハッシングに従う充填領域からの命令)をIQがまさに処理しようとしていることを示す「充填領域開始(fill region start; FRS)」信号は、しかし、PFRQにセーブされている線形命令ポインタ(“リップ”(lip))から始まらない(405)。これは、状態機械を状態410に移動させる。IQ内の(直ぐにデコードされる)次の命令が充填領域を終わらせる(例えば、キャッシュによってハッシュされるラインを終わらせる、又は成立分岐である)場合、状態機械は、対応するPFRQエントリの割当解除(deallocation)415を引き起こし、状態機械は状態401に戻る。しかし、充填ポインタ(fill pointer; FP)が、状態401又は状態410のいずれであろうと、充填領域リップ(fill region lip; FRL

)に等しい場合(430)、状態機械は、アクセスが充填領域内且つ充填開始点後にある状態420に入る。状態420から、充填領域インジケーションにおける最後のuopは、対応するPFRQエントリを解除することなく、状態機械を状態401に戻す(425)。図4の状態機械は、ハードウェアロジック、ソフトウェア、又はそれらの何らかの組合せにおいて実施されてよい。他の実施形態において、他の状態機械又はロジックが使用されてよい。

#### 【0016】

図5は、閾時間だけ第1のフュージョン可能な命令の処理を遅延させて、第2のフュージョン可能な命令が閾時間内にIQに格納された場合に第2のフュージョン可能な命令が第1のフュージョン可能な命令とフュージョンされ得るようにするために、本発明の少なくとも1つの実施形態とともに使用されてよい動作のフロー図を表す。動作501で、IQ内の目下アクセスされている命令がいずれかの後続の命令とフュージョン可能であるかどうか、故に、IQからの第1のフュージョン可能な命令の読み出しを閾数のサイクルだけ遅らせるべきかが決定される。一部の実施形態において、ロジックは、第1のフュージョン可能な命令がIQに格納されている最後の命令である場合にのみ、第1のフュージョン可能な命令の処理を遅延させる。目下アクセスされている命令が、IQに格納されるいずれの後続命令ともフュージョン可能でないと決定された場合は、目下アクセスされている命令を後続命令とは別々に処理するよう、動作505で、次の命令がIQからアクセスされ、遅延カウンタがリセットされる。そして、図示のように、この動作フローは動作501へと戻る。認識されるように、目下アクセスされている命令に対する図示の動作フローの完了を受け、このフローは動作501で再び開始する次のサイクルで始まって、再び、IQ内の目下アクセスされている命令が第1のフュージョン可能な命令であるかが決定される。他方、フュージョン可能である場合は、動作510で、遅延カウンタがインクリメントされて、IQからの第1のフュージョン可能な命令の読み出しが閾数のサイクルまで遅延され、動作515で、遅延カウンタ閾値に達したかどうか決定される。遅延カウンタ閾値に達していない場合は、動作520で、目下アクセスされている命令の命令フュージョンは実行されない。遅延カウンタ閾値に達した場合は、動作505で、次の命令がIQからアクセスされ、遅延カウンタがリセットされる。一部の実施形態において、動作505にて、命令フュージョンロジックは、第2のフュージョン可能な命令が、第1のフュージョン可能な命令の後であって、閾数のサイクルに達する前にIQに格納される場合に、第1及び第2のフュージョン可能な命令をフュージョンし得る。他の実施形態において、他の動作が、命令フュージョンの効率を改善するために使用されてよい。

#### 【0017】

図6は、閾数のサイクルだけIQから第1のフュージョン可能な命令を読み出すこと、又は第1のフュージョン可能な命令をデコーダ若しくは他の処理ロジックへ発行することを遅延させて、IQに未だ記憶されていない、第1の命令とフュージョン可能な第2のフュージョン可能な命令が存在する場合に、これら2つのフュージョン可能な命令をフュージョンする機会をのがさないようにするために、少なくとも1つの実施形態とともに実行されてよい動作のフロー図を表す。多数のデコーダ回路を有するプロセッサにおいて一実施形態を実行するために、第1のフュージョン可能な命令が、フュージョンされた命令をデコードすることができる特定のデコーダ回路においてデコードされるべきことを確かに行うことが有用である。図6において、動作601で、特定の命令がフュージョンされる命令の組の中の第1の命令であり得るかどうか決定される。第1の命令でない場合は、動作605で、次の命令がIQから処理され、フュージョンされた命令が発行される。他方、第1の命令である場合は、動作610で、IQにおいて第1のフュージョン可能な命令の後に有効な命令が続くかどうか決定される。有効な命令が後に続く場合は、動作605で、次の命令がIQから処理され、フュージョンされた命令が発行される。他方、有効な命令が後に続かない場合は、第1のフュージョン可能な命令はIQに格納されている最後の命令であり、動作615で、第1のフュージョン可能な命令は、フュージョンされた命令をサポート可能なデコーダに発せられるべきかどうか決定される。一実施形態に

において、デコーダ - 0 は、フュージョンされた命令をデコードすることができる。第 1 のフュージョン可能な命令がデコーダ - 0 に発せられなかった場合は、動作 6 2 0 で、第 1 のフュージョン可能な命令は、それがデコーダ - 0 に対応するまで、異なるデコーダへ移動又は削除される。動作 6 2 5 で、カウンタは初期値 N に設定され、動作 6 3 0 で、命令の後に有効な命令が続き、又はカウンタがゼロである場合は、次の命令が I Q から処理され、フュージョンされた命令は、動作 6 3 5 で発せられる。一部の実施形態において、動作 6 3 5 にて、命令フュージョンロジックは、第 2 のフュージョン可能な命令が、第 1 のフュージョン可能な命令の後であって、閾数のサイクルに達する前に I Q に格納される場合に、第 1 及び第 2 のフュージョン可能な命令をフュージョンし得る。さもなければ、動作 6 4 0 で、カウンタはデクリメントされ、無効な命令は削除される。他の実施形態において、カウンタは、最終値へとインクリメントしてよい。他の実施形態において、削除動作に加えて、他の動作が無効な命令をクリアしてよい。認識されるように、目下アクセスされている命令に対する図示の動作フローの完了を受け、このフローは動作 6 0 1 で再び開始する次のサイクルで始まって、再び、目下アクセスされている命令がフュージョンされる命令の組の中の第 1 の命令であり得るかどうかが決定される。

【 0 0 1 8 】

少なくとも 1 つの実施形態の 1 又はそれ以上の側面は、機械によって読み出される場合に該機械にここに記載される技術を実行するようロジックを組み立てさせる、プロセッサ内の種々のロジックを表す機械読出可能な媒体に記憶されている代表データによって実施されてよい。かかる表現は、「IP コア」として知られており、有形な機械読出可能な媒体（テープ）に記憶され、実際にロジック又はプロセッサを作る製造機械にロードするよう種々の顧客又は製造施設に供給されてよい。

【 0 0 1 9 】

このように、マイクロアーキテクチャにおけるメモリ領域アクセスを対象とする方法及び装置について記載してきた。当然、上記記載は例示であって限定ではない。多数の他の実施形態は、上記記載を読み理解することで当業者には明らかである。従って、本発明の適用範囲は、添付の特許請求の範囲が権利を有する同等な包括的範囲とともに、特許請求の範囲を参照して決定されるべきである。

【 誤訳訂正 3 】

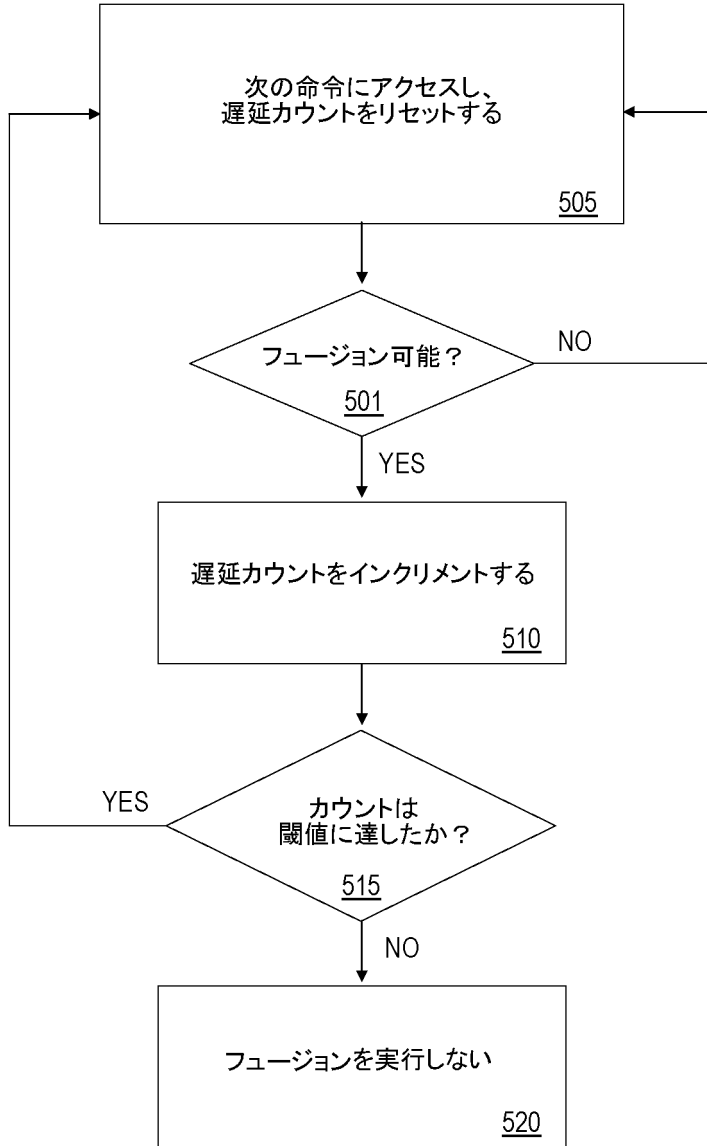
【訂正対象書類名】図面

【訂正対象項目名】図 5

【訂正方法】変更

【訂正の内容】

【 図 5 】



- 【 誤訳訂正 4 】
- 【 訂正対象書類名 】 図面
- 【 訂正対象項目名 】 図 6
- 【 訂正方法 】 変更
- 【 訂正の内容 】

【 図 6 】

