

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2004-326765

(P2004-326765A)

(43) 公開日 平成16年11月18日(2004. 11. 18)

(51) Int. Cl.⁷

G06F 9/46

G06F 9/38

F I

G06F 9/46 340B

G06F 9/38 310X

テーマコード(参考)

5B013

5B098

審査請求 有 請求項の数 20 O L (全 21 頁)

(21) 出願番号 特願2004-119571(P2004-119571)
 (22) 出願日 平成16年4月14日(2004. 4. 14)
 (31) 優先権主張番号 10/424530
 (32) 優先日 平成15年4月25日(2003. 4. 25)
 (33) 優先権主張国 米国(US)

(71) 出願人 390009531
 インターナショナル・ビジネス・マシー
 ズ・コーポレーション
 INTERNATIONAL BUSIN
 ESS MASCHINES CORPO
 RATION
 アメリカ合衆国10504 ニューヨーク
 州 アーモンク ニュー オーチャード
 ロード

(74) 代理人 100086243
 弁理士 坂口 博

(74) 代理人 100091568
 弁理士 市位 嘉宏

(74) 代理人 100108501
 弁理士 上野 剛史

最終頁に続く

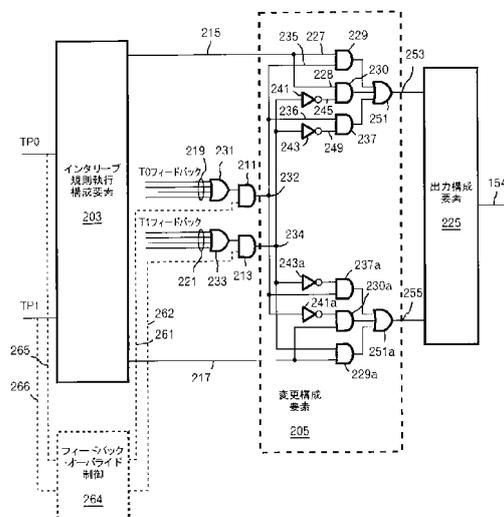
(54) 【発明の名称】 マルチスレッド・プロセッサにおいて処理する命令スレッドを選択するための方法および装置

(57) 【要約】

【課題】あるプロセッサ事象または条件に対処するために、異なる命令スレッドからの命令をインタリーブするためにSMTプロセッサ内の命令スレッドを選択する命令スレッド間の選択を変更する装置および方法を提供すること。

【解決手段】プロセッサ・クロック・サイクルごとに、インタリーブ規則執行構成要素が、インタリーブされた命令ストリームに命令を渡す特定の1つの命令スレッドを指示する少なくとも1つの基準命令スレッド選択信号を生成する。スレッド選択の変更は、基準スレッド選択信号と、さまざまなプロセッサ要素内の1つまたは複数の条件または事象に由来するフィードバック信号とに基づいて最終スレッド選択信号を生成するインタリーブ変更構成要素によって提供される。この最終スレッド選択信号は、インタリーブされた命令ストリームに命令を渡す命令スレッドとして、基準スレッド選択信号が指示した命令スレッドと同じ命令スレッドを指示し、または別の命令スレッドを指示することができる。

【選択図】 図2



【特許請求の範囲】**【請求項 1】**

同時マルチスレッディング・プロセッサにおけるいくつかの命令スレッド間の命令のサイクル単位のインタリーブを制御する方法であって、

(a) 前記いくつかの命令スレッドの中の処理する第 1 の命令スレッドを指示する基準スレッド選択信号を、命令インタリーブ実行規則に少なくとも部分的に基づいて生成する段階と、

(b) 前記いくつかの命令スレッドに含まれる特定の命令スレッドを指示する最終スレッド選択信号を、前記基準スレッド選択信号と前記いくつかの命令スレッドに含まれる 1 つの前記命令スレッドに関連付けられたフィードバック信号とに基づいて生成する段階と

(c) インタリーブされた命令ストリームに命令をインタリーブするために前記いくつかの命令スレッドのうちの 1 つの命令スレッドを選択する選択に、前記最終スレッド選択信号を適用する段階と

を含む方法。

【請求項 2】

(a) 前記いくつかの命令スレッドの中の処理する追加の命令スレッドを指示する追加の基準スレッド選択信号を、前記命令インタリーブ実行規則に少なくとも部分的に基づいて生成する段階と、

(b) 前記いくつかの命令スレッドに含まれるそれぞれの命令スレッドを指示する追加の最終スレッド選択信号を、前記追加の基準スレッド選択信号と前記いくつかの命令スレッドに含まれる 1 つの前記命令スレッドに関連付けられた追加のフィードバック信号とに基づいて生成する段階と

をさらに含む、請求項 1 に記載の方法。

【請求項 3】

前記最終スレッド選択信号と前記追加のスレッド選択信号を結合してスレッド選択 MUX 制御信号を生成する段階をさらに含む、請求項 2 に記載の方法。

【請求項 4】

前記追加の基準スレッド選択信号が前記基準スレッド選択信号の 2 進補数である、請求項 2 に記載の方法。

【請求項 5】

(a) 前記最終スレッド選択信号を、前記いくつかの命令スレッドに含まれる第 1 の命令スレッドに関連付けられた保留入力に適用する段階と、

(b) 前記追加の最終スレッド選択信号を、前記いくつかの命令スレッドに含まれる追加の命令スレッドに関連付けられた保留入力に適用する段階と

をさらに含む、請求項 2 に記載の方法。

【請求項 6】

前記フィードバック信号が前記第 1 の命令スレッドに関連付けられている、請求項 1 に記載の方法。

【請求項 7】

前記いくつかの命令スレッドが異なる処理優先度を有するときに前記フィードバック信号を無効にする段階をさらに含む、請求項 1 に記載の方法。

【請求項 8】

同時マルチスレッディング・プロセッサにおけるいくつかの命令スレッド間の命令のサイクル単位のインタリーブを制御する方法であって、

(a) 命令インタリーブ実行規則に少なくとも部分的に基づく基準スレッド選択信号を受け取る段階であって、前記基準スレッド選択信号が、前記いくつかの命令スレッドに含まれる処理のために選択された第 1 の命令スレッドを指示する段階と、

(b) 前記いくつかの命令スレッドに含まれる 1 つの前記命令スレッドに関連付けられたフィードバック信号を命令処理要素から受け取る段階と、

10

20

30

40

50

(c) 前記いくつかの命令スレッドに含まれる特定の命令スレッドを指示する最終スレッド選択信号を、前記基準スレッド選択信号および前記フィードバック信号に基づいて生成する段階と、

(d) インタリーブされた命令ストリームに命令をインタリーブするために1つの前記命令スレッドを選択する選択に、前記最終スレッド選択信号を適用する段階とを含む方法。

【請求項9】

(a) 前記命令インタリーブ実行規則に少なくとも部分的に基づく追加の基準スレッド選択信号を受け取る段階であって、前記追加の基準スレッド選択信号が、前記いくつかの命令スレッドに含まれる処理のために選択された追加の命令スレッドを指示する段階と、

(b) 前記いくつかの命令スレッドに含まれる1つの前記命令スレッドに関連付けられた追加のフィードバック信号を命令処理要素から受け取る段階と、

(c) 前記いくつかの命令スレッドに含まれるそれぞれの命令スレッドを指示する追加の最終スレッド選択信号を、前記基準スレッド選択信号および前記フィードバック信号に基づいて生成する段階と

をさらに含む、請求項8に記載の方法。

【請求項10】

前記最終スレッド選択信号と前記追加の最終スレッド選択信号を結合してスレッド選択MUX制御信号を生成する段階をさらに含む、請求項9に記載の方法。

【請求項11】

前記追加の基準スレッド選択信号が前記基準スレッド選択信号の2進補数である、請求項9に記載の方法。

【請求項12】

(a) 前記最終スレッド選択信号を、前記いくつかの命令スレッドに含まれる第1の命令スレッドに関連付けられた保留入力に適用する段階と、

(b) 前記追加の最終スレッド選択信号を、前記いくつかの命令スレッドに含まれる追加の命令スレッドに関連付けられた保留入力に適用する段階と

をさらに含む、請求項9に記載の方法。

【請求項13】

前記フィードバック信号が前記第1の命令スレッドに関連付けられている、請求項8に記載の方法。

【請求項14】

前記いくつかの命令スレッドが異なる処理優先度を有するときに前記フィードバック信号を無効にする段階をさらに含む、請求項8に記載の方法。

【請求項15】

同時マルチスレッディング・プロセッサにおけるいくつかの命令スレッド間の命令のサイクル単位のインタリーブを制御する回路であって、

(a) 前記いくつかの命令スレッドの中の第1の命令スレッドを指示する第1のスレッド選択信号を受け取るように接続された第1の選択入力と、

(b) 前記いくつかの命令スレッドに含まれる1つの前記命令スレッドに関連付けられたプロセッサ要素フィードバック信号を受け取るように接続されたフィードバック入力と、

(c) 前記基準スレッド選択信号および前記フィードバック信号に基づいて最終スレッド選択信号を生成する変更構成要素と、

(d) インタリーブされた命令ストリームに命令をインタリーブするために前記いくつかの命令スレッドのうちの1つ命令スレッドを選択する選択に前記最終スレッド選択信号を適用する出力構成要素と

を含む回路。

【請求項16】

前記いくつかの命令スレッドの中の第2の命令スレッドを指示する第2の基準スレッド

10

20

30

40

50

選択信号を受け取るように接続された第2の選択入力をさらに含む、請求項15に記載の回路。

【請求項17】

(a) 前記変更構成要素がさらに、前記第2のスレッド選択信号および第2のフィードバック信号に基づいて追加の最終スレッド選択信号を生成し、

(b) 前記出力構成要素がさらに、インタリーブされた命令ストリームに命令をインタリーブするために前記いくつかの命令スレッドのうちの1つの命令スレッドを選択する選択に前記追加の最終スレッド選択信号を適用する、

請求項16に記載の回路。

【請求項18】

(a) 前記最終スレッド選択信号の状態に基づいて第1の命令スレッドを選択的に保留する第1のスレッド保留要素と、

(b) 前記追加の最終スレッド選択信号の状態に基づいて第2の命令スレッドを選択的に保留する第2のスレッド保留要素と

をさらに含む、請求項17に記載の回路。

【請求項19】

前記第1のフィードバック信号と前記第2のフィードバック信号のそれぞれに対するフィードバック・オーバライド回路をさらに含み、前記フィードバック・オーバライド回路が、前記いくつかの命令スレッドが異なる処理優先度を有するときにそれぞれの前記フィードバック信号を無効にする、請求項17に記載の回路。

【請求項20】

前記いくつかの命令スレッドが異なる処理優先度を有するときに前記第1のフィードバック信号を無効にする第1のフィードバック・オーバライド回路をさらに含む、請求項15に記載の回路。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、インタリーブされた複数の命令スレッドを優先規則に従ってサイクル単位で処理するデータ・プロセッサに関する。詳細には本発明は、命令スレッドに関連した事象または条件に基づいて処理する、特定の命令スレッドを選択することに関する。

【背景技術】

【0002】

データ・プロセッサがソフトウェア・プログラムを実行する速度を向上させる目的にはいくつかの技法が使用される。これらの技法には、プロセッサのクロック速度を高めること、キャッシュ・メモリを使用すること、および分岐予測を使用することが含まれる。プロセッサのクロック速度を高めると、プロセッサは、与えられた時間内に相対的に多くの演算を実行できるようになる。キャッシュ・メモリは、プロセッサのすぐ近くに置かれ、メイン・メモリよりも速く動作する。これによってプロセッサがデータおよび命令にアクセスするのに要する時間が短縮される。分岐予測では、プロセッサが、以前の命令の結果に関する予測に基づいてある種の命令を実行することができ、そのため実結果を待つ必要がなくなり、これによって処理速度が向上する。

【0003】

いくつかのプロセッサはさらに、パイプライン方式の命令実行を使用してシステム性能を強化している。パイプライン命令実行では、処理タスクが、いくつかのパイプライン・ステップまたはステージに分割される。パイプライン処理では、以前に発行された命令が特定のプロセスを完了させる前に後続の命令が処理を開始できるようにすることによって、処理速度を高めることができる。プロセッサは、ある命令が完全に処理されるのを待つことなく次の命令の処理を開始することができる。

【0004】

パイプライン処理を使用するプロセッサは、プロセッサ内での異なるアクティビティに

10

20

30

40

50

充てられる異なるいくつかのパイプラインを含むことができる。例えばプロセッサは、一連の命令を、フェッチ・ステージ、解読/ディスパッチ・ステージ、発行ステージ、実行ステージ、終了ステージおよび完了ステージで処理することができる。これらの個々のステージはそれぞれ、自体のパイプライン・ステージ・セットを使用して、所望の処理タスクを実行することができる。

【0005】

マルチスレッド命令処理は、パイプライン処理とともに使用して処理速度を高めることができる追加の技法である。マルチスレッド命令処理は、1つのプログラム命令セットを、2つ以上の別個の命令群ないし命令スレッドに分割することを含む。このマルチスレッド技法では、何らかの理由で1つのスレッドが処理できない場合でも、その間に、別の1つのスレッドからの命令をパイプライン処理することができる。これによって、シングル・スレッド命令処理において遭遇する、例えば特定の命令を実行するのに必要なデータがすぐには使用できないキャッシュ・ミス状況などで特定の命令を実行できない間、全ての命令が停止される状況が回避される。複数の命令スレッドを処理する能力を有するデータ・プロセッサはしばしば、同時マルチスレッディング (simultaneous multithreading: SMT) プロセッサと呼ばれる。

10

【0006】

ソフトウェア分野での用語「マルチスレッディング」の使用方法和、コンピュータ・アーキテクチャ分野での用語「マルチスレッディング」の使用方法とは異なることに留意されたい。ソフトウェア分野では用語「マルチスレッディング」が、複数の関連スレッドに細分されたある1つのタスクについて使用される。コンピュータ・アーキテクチャ分野では、用語「マルチスレッディング」が、互いに独立したスレッドを含むスレッドに対して使用される。本明細書では用語「マルチスレッディング」を、コンピュータ・アーキテクチャ分野で使用されている意味と同じ意味で使用する。

20

【0007】

マルチスレッディングを容易にするため、異なるスレッドからの命令は、プロセッサ・パイプライン上のあるポイントで何らかの方法でインタリーブされる。SMTプロセッサで処理する命令をインタリーブする技法には一般に異なる2つの技法がある。1つの技法は、1つのスレッドの処理において遅延を引き起こすキャッシュ・ミスなどのある長い待ち時間事象 (long latency event) に基づいてスレッドをインタリーブすることを含む。この技法では、ある長い待ち時間事象によって1つのスレッドの処理に遅延が生じるまで、全てのプロセッサ・リソースがその1つのスレッドに向けられる。長い待ち時間事象が出現すると、プロセッサはすぐに別のスレッドに切り替わり、そのスレッドに対してある長い待ち時間事象が生じるか、または先のスレッドを停止させた状況が解決されるまで、そのスレッドの処理を進める。

30

【0008】

SMTプロセッサにおいて複数の命令スレッドからの命令をインタリーブするもう1つの一般的な技法は、命令を、あるインタリーブ規則に従ってサイクル単位でインタリーブすることを含む。単純なサイクル単位のインタリーブ技法は、異なるスレッドからの命令を1命令ずつインタリーブする。例えば2スレッドSMTプロセッサでは、第1のクロック・サイクルで第1のスレッドから1つの命令を取り出し、第2のクロック・サイクルで第2のスレッドから1つの命令を取り出し、第3のクロック・サイクルで第1のスレッドの別の命令を取り出し、...というように、2つの命令スレッド間で交互に命令を処理することができる。より複雑なサイクル単位のインタリーブ技法は、ソフトウェア命令を使用してそれぞれの命令スレッドに優先度を割り当て、次いで異なるスレッドからの命令をインタリーブして、相対的なスレッド優先度に基づくある規則を執行することを含む。例えば、2スレッドSMTプロセッサにおいて、一方のスレッドに他方のスレッドよりも高い優先度が割り当てられている場合、単純なインタリーブ規則は例えば、優先度が高いほうのスレッドからインタリーブされた命令ストリームに含める命令の数を、優先度が低いほうのスレッドから含める命令の数の2倍にすることを要求する。

40

50

【0009】

現在使用されているより複雑なサイクル単位インタリーブ規則では、それぞれのスレッドに「1」から「7」までの優先度を割り当て、優先度が低いほうのスレッドからの命令を、関数 $1 / (2^{|X - Y| + 1})$ に基づいてインタリーブされた命令ストリームに入れる。ただし、 X = ソフトウェアによって第1のスレッドに割り当てられた優先度、 Y = ソフトウェアによって第2のスレッドに割り当てられた優先度、である。2つのスレッドの優先度が等しい場合、例えば $X = 3$ 、 $Y = 3$ のときには、この関数から比 $1 / 2$ が得られ、2つのスレッドからそれぞれ2クロック・サイクルに1回、1つの命令が、インタリーブされた命令ストリームに入れられる。スレッド優先度の差が2の場合、例えば $X = 2$ 、 $Y = 4$ のときには、この関数から比 $1 / 8$ が得られ、優先度の低いほうのスレッドから、8クロック・サイクルに1回、1つの命令が、インタリーブされた命令ストリームに入れられる。

10

【発明の開示】

【発明が解決しようとする課題】

【0010】

特定のスレッドからの命令をインタリーブされた命令ストリームに含める頻度を優先規則を使用して選択するのは、一般に、それぞれのスレッドの優先度に基づいてプロセッサ・リソースが割り当てられることを保証するためである。マルチスレッディング・プロセッサ内の複数の命令スレッドの優先度が等しいときには、一般に、これらの命令スレッドはプロセッサ・リソースを同等に共用しなければならない。しかし、異なる全ての命令スレッドがソフトウェアによって割り当てられた同じ優先度を有するときでも、1つの命令スレッドがプロセッサ・リソースを過度に使用する状況がSMTプロセッサ内に存在する可能性がある。命令スレッドの優先度が等しいときにプロセッサ・リソースがこのように不均等に使用されると、軽視された命令スレッドの処理が思うように進まず、全体的な処理効率が低下する可能性がある。

20

【課題を解決するための手段】

【0011】

本発明は、異なる命令スレッドからの命令をインタリーブするためにSMTプロセッサ内の命令スレッドを選択する命令スレッド間の選択を変更する装置および方法を提供する。プロセッサ・クロック・サイクルごとに、インタリーブ規則執行構成要素または回路が、インタリーブされた命令ストリームに命令を渡す特定の1つの命令スレッドを指示する少なくとも1つの基準スレッド選択信号を生成する。本発明によれば、インタリーブ変更構成要素は、基準スレッド選択信号と、さまざまなプロセッサ要素内の1つまたは複数の条件または事象に由来するフィードバック信号とに基づいて最終スレッド選択信号を生成する。この最終スレッド選択信号は、インタリーブされた命令ストリームに命令を渡す命令スレッドとして、基準スレッド選択信号が指示した命令スレッドと同じ命令スレッドを指示し、または別の命令スレッドを指示することができる。

30

【0012】

本発明に基づく命令スレッド選択の調整または変更は特に、処理優先度が同じかまたは同様である異なる命令スレッド間でプロセッサ・リソースを適切に割り当てる際に適用することができる。本発明の一形態によれば、命令スレッドが、異なる処理優先度または実質的に異なる処理優先度を有するときには、異なる命令スレッドからの命令は、インタリーブ規則執行構成要素が執行するインタリーブ規則に従ってインタリーブされる。しかし、全ての命令スレッドの優先度が等しいか、またはほとんど等しいときには、インタリーブ変更構成要素が、インタリーブ規則によって決定された命令スレッドとは異なる命令スレッドを、インタリーブされたストリームに命令を追加する命令スレッドとして選択することができる。このスレッド選択の変更は、同様の優先度を有する命令スレッド間のプロセッサ・リソースの割当ての均等化を容易にする。

40

【0013】

本発明に基づく最終スレッド選択信号を直接に適用して、インタリーブされた命令スト

50

リームに命令をインタリーブする命令スレッドの選択を制御することができる。本発明の好ましい形態は、それぞれがそれぞれの基準スレッド選択信号およびプロセッサ内の命令スレッドに対応する2つの最終スレッド選択信号を生成する。この2つの最終スレッド選択信号を出力構成要素で結合して、インタリーブされた命令ストリームに命令を渡す命令スレッドを選択する2つの命令スレッド間の選択に使用することができる選択制御信号を生成する。

【0014】

プロセッサ内の別個のそれぞれの命令スレッドに対して最終スレッド選択信号を生成する本発明の形態では、ある状況下でそれぞれの最終選択信号が、インタリーブされたストリームに命令を渡す命令スレッドとしてそれぞれの命令スレッドを選択しないことを指示するように、変更構成要素を構成することができる。本発明のこれらの形態では、それぞれの最終スレッド選択信号を保留信号として使用し、プロセッサ内のそれぞれの保留要素に適用して、それぞれの命令スレッドからの有効な命令が、処理するインタリーブされたストリームに渡されることを防ぐ。

10

【0015】

1つまたは複数の基準スレッド選択信号が変更構成要素の影響を受けないようにして、それぞれの最終スレッド選択信号が、それぞれの基準スレッド選択信号が指示するスレッドと同じスレッドを指示するようにすることが望ましい状況が存在する可能性がある。これらの状況に対処するため、本発明の好ましい形態はフィードバック・オーバライド構成要素または回路を含む。フィードバック・オーバライド構成要素はそれぞれ、それぞれの基準スレッド選択信号からそれぞれの最終スレッド選択信号への変更を指示するそれぞれのフィードバック信号を選択的に無効にする（オーバライドする）。例えば、本発明に基づくフィードバック・オーバライド構成要素は、インタリーブする命令スレッドのスレッド優先度が等しくない場合に、それぞれのフィードバック信号を選択的に無効にすることができる。

20

【0016】

本発明のこれらの利点および特徴、ならびにその他の利点および特徴は、好ましい実施形態の以下の説明を添付の図面とともに検討することによって明白となる。

【発明を実施するための最良の形態】**【0017】**

図1に、本発明の原理を具体化したプロセッサ100を示す。プロセッサ100はマルチスレッディングに対応し、第1の命令待ち行列101（Queue₀）、第2の命令待ち行列102（Queue₁）、選択マルチプレクサ（MUX）105、および選択コントローラ106を含む。命令待ち行列101および102はそれぞれ、それぞれの命令スレッド中の命令の取扱いに携わる。具体的には、第1の命令待ち行列101は第1の命令スレッドT₀からの命令を待ち行列に入れ、第2の命令待ち行列102は第2の命令スレッドT₁からの命令を待ち行列に入れる。シーケンシャル・フェッチャ（sequential fetcher）120は、この2つの異なる命令スレッドを構成する命令群を、命令キャッシュ/メモリ管理ユニット（memory management unit: MMU）118から命令待ち行列101および102に転送する。命令待ち行列101および102は最終的に、それぞれの命令を選択MUX105の2つの入力に供給する。選択コントローラ106は、この2つの異なる命令スレッドからの命令を、インタリーブされた処理のための単一の命令ストリームにインタリーブする選択MUX105を制御する。

30

40

【0018】

図2および3を参照して後に詳細に説明するとおり、それぞれの命令スレッドには優先度が割り当てられ、選択コントローラ106に含まれるインタリーブ規則執行構成要素は、これらの優先度を使用して、それぞれの待ち行列（101または102）からの命令を、インタリーブされた処理ストリームに入れる頻度に関する規則を執行する。選択コントローラ106の出力は、スレッド選択制御経路154に適用される制御信号を含む。スレッド選択制御経路154上のこの制御信号は、MUX105に適用し、後段の処理のため

50

に特定の1クロック・サイクルの間に通過させるM U Xの入力を指定する。M U X 1 0 5のそれぞれの入力は、異なるスレッドからの命令を含む命令待ち行列に結合されているので、M U X 1 0 5の特定の入力を選択することは、処理をおこなう特定の命令スレッドを選択する効果を有する。

【0019】

図1に示した例示的な実施形態では、プロセッサ100が、単一の集積回路スーパーシステム・マイクロプロセッサを含む。したがってプロセッサ100は、さまざまな実行ユニット、レジスタ、バッファ、メモリ・デバイスおよび他の機能単位を含む。これらは全て集積回路設計によって形成される。本明細書では、マイクロプロセッサに適用するとして本発明を説明しているが、当然ながら、この選択コントローラ配置はマイクロプロセッサだけに限定されず、他のタイプのプロセッサに実装することもできる。さらに、図1に示した全体プロセッサ配置は、第1の命令待ち行列101、第2の命令待ち行列102および選択M U X 1 0 5に対する選択コントローラ106の動作を説明する際の例として示したに過ぎない。本明細書において説明し請求する選択コントローラは、複数の命令スレッドの同時処理をサポートする能力を有する実質的に任意のプロセッサ配置で利用することができることを当業者は理解されたい。

10

【0020】

図1に示すプロセッサ100は、バス・インタフェース・ユニット(B I U)114およびプロセッサ・バス115を介してシステム・バス113に結合されている。システム・バス113およびプロセッサ・バス115はともに、個別に示されていないが、アドレス・バス、データ・バスおよび制御バスを含む。B I U 1 1 4はバス裁定に関与して、プロセッサ100と、メイン・メモリ116、不揮発性大容量記憶装置117などのシステム・バス113に結合された他の装置との間の情報の転送を制御する。図1に示したデータ処理システムは、システム・バス113に結合した他の装置を含むことが好ましいが、本発明の理解にこれらの装置は不要であり、そのため無用な詳細によって本発明が不明瞭にならないようこれらは図面から除外した。

20

【0021】

B I U 1 1 4は、命令キャッシュ/MMU(メモリ管理ユニット)118およびデータ・キャッシュ/MMU119に接続されている。命令キャッシュ/MMU118およびデータ・キャッシュ/MMU119中のキャッシュなどの高速キャッシュは、メイン・メモリ116から以前にキャッシュに転送されたデータまたは命令のサブセットにプロセッサ100が相対的に高速にアクセスすることを可能にし、したがって全体的な処理速度を向上させる。それぞれデータ・キャッシュ119および命令キャッシュ118に記憶されたデータおよび命令は、メイン・メモリ116中のデータまたは命令の実アドレスに関係付けられた実効アドレスによって識別され、アクセスされる。

30

【0022】

命令キャッシュ/MMU118はさらに、シーケンシャル・フェッチャ120に結合されている。シーケンシャル・フェッチャ120は、実行する命令を、各プロセッサ・サイクルの間に命令キャッシュ/MMU118からフェッチする。シーケンシャル・フェッチャ120は、命令キャッシュ/MMU118からフェッチした分岐命令を、これを実行する分岐処理ユニット(B P U)121に伝送し、また、第1の命令待ち行列101および第2の命令待ち行列102中の一連の命令を一時的に記憶する。記憶された命令はいずれ、解読をする命令解読ユニット123、および実行ユニット130、140または150へのディスパッチする順序付けユニット124に転送される。

40

【0023】

シーケンシャル・フェッチャ120は、第1の命令待ち行列101と第2の命令待ち行列102の両方に命令を供給する。命令待ち行列101および102はともに、後段の処理のために選択M U X 1 0 5を通して解読ユニット123に命令が発行されるたびに、その中の命令を、1つの待ち行列位置から次の待ち行列位置へシフトさせるように構成されている。第1の命令待ち行列101の最後の記憶要素は第1のラッチ103(L C H _ 0

50

)を含み、第2の命令待ち行列102の最後の記憶要素は第2のラッチ104(LCH_1)を含む。ラッチ103および104は、ラッチされた命令を、選択MUX105のそれぞれの入力で使用可能なように維持し、そのため、MUXの1つの入力を選択されると、選択された入力に存在する命令が命令解読ユニット123に転送される。なお、ラッチ103および104を含む図示のさまざまな要素は、一度に1つの命令だけを取り扱うように構成し、または複数の命令を取り扱うように構成することができることに留意されたい。本発明の好ましい一形態では、プロセッサ100の図示のさまざまな要素が、複数の命令を同時に取り扱うことができる。例えば、命令待ち行列101および102中のそれぞれの待ち行列位置、ならびにラッチ103および104は、5つの命令を収容するのに十分な記憶要素を含むことができる。したがってこの例では、それぞれの命令スレッド(T0およびT1)、ならびに選択MUX105の出力のインタリーブされた命令ストリームが実際に5つの命令に相当する幅を有する。本発明は、スレッドT0およびT1を通過する任意の命令幅を包含する。

10

【0024】

プロセッサ100の実行回路は、一連の命令を実行する3つの別個の実行ユニット130、140および150を含む。個々の実行ユニット130、140および150はそれぞれ、特定の実行ユニットに固有の一連のパイプライン・ステージで命令を実行することが好ましい。この例示プロセッサの第1の実行ユニット130(EXU_1)および第2の実行ユニット140(EXU_2)はともに、固定小数点数値演算および論理演算、ならびにメモリからデータをロードするロード操作を実行するように適合させることができる。プロセッサ100の第3の実行ユニット150(EXU_3)は、複雑な固定小数点演算を実行するように適合させることができる。第3の実行ユニット150はさらに、メモリにデータを記憶する記憶操作を実行することができる。オペランドおよび演算結果を一時的に記憶するためにさまざまな汎用および浮動小数点レジスタが実行ユニット130、140および150に関連付けられていることを当業者は理解されたい。本発明の選択コントローラ配置を理解するのにこれらのレジスタの動作の理解は不要なので、図1にこれらのレジスタは示されていない。実行ユニット130、140および150には、命令のアウト・オブ・オーダー(out of order)実行をサポートするのに使用するさまざまな待ち行列、レジスタ、テーブルなど、他の数多くの要素を関連付けることができる。無用な詳細によって本発明が不明瞭にならないよう、これらの追加のプロセッサ要素も図1から

20

30

【0025】

プロセッサ100は一連のそれぞれの命令を、別個のパイプライン・ステージ、すなわちフェッチ、解読/ディスパッチ、発行/順序付け、実行、終了および完了ステージで処理する。それぞれ2つの命令待ち行列101および102に記憶された2つのスレッドT0およびT1からの命令は、解読/ディスパッチ・ステージの直前に、選択コントローラ106が執行する優先規則に従って単一の命令ストリームにインタリーブされる。2つのスレッドからの命令は、解読ユニット123によって実行される解読/ディスパッチ・ステージの前にサイクル単位でインタリーブされるので、解読/ディスパッチ以降の各ステージは、所与のクロック・サイクルに、いずれかのスレッドから命令を受け取ることができる。例えば、所与のクロック・サイクルに、プロセッサ100は、第1のスレッドからの命令を完了ユニット190で完了させ、第1または第2のスレッドからの命令を実行ユニット130、140および150で実行し、第2のスレッドからの命令を解読している。異なるスレッドからの命令を同時に処理することによって、一方の命令スレッドの中のある命令に関連した長い待ち時間事象によってその特定のスレッドが機能停止した場合でも、プロセッサ100は命令を処理し続けることができる。例えば、(第1の命令待ち行列101に入れられた)スレッドT0からの命令が、この命令がすぐに処理されるのを妨げるある長い待ち時間事象を経験しているとする。(第2の命令待ち行列102に入れられた)第2の命令スレッドT1からの命令はスレッドT0からの命令とインタリーブされているので、第2の命令スレッドからの命令は処理され続け、第2の命令待ち行列T1を

40

50

通過することができる。

【0026】

フェッチ・ステージの間に、シーケンシャル・フェッチャ120は、1つまたは複数のメモリ・アドレスに関連付けられた1つまたは複数の命令を命令キャッシュ/MMU118から取り出す。シーケンシャル・フェッチャ120は、命令キャッシュ/MMU118からフェッチした一連の命令を、第1の命令待ち行列101に記憶して第1の命令スレッドT0の一部とするか、または第2の命令待ち行列102に記憶して第2の命令スレッドT1の一部とする。両方のスレッドの分岐命令はシーケンシャル・フェッチャ120によって抜き取られ (removed or folded out)、実行のためBPU121に送られる。BPU121は、分岐予測機構 (別個には示されていない) を含む。一実施形態では分岐予測機構が、分岐履歴テーブルなどの動的予測機構 (図示せず) を含む。この分岐履歴テーブルによってBPU121は、分岐するか否かを予測することによって未解決の条件付き分岐命令を推測で実行することができる。

10

【0027】

第1および第2の命令待ち行列101および102を通過した命令は単一の命令ストリームにインタリーブされ、後段の処理のために命令解読ユニット123に送達される。命令解読ユニット123はこれらの命令を解読し、順序付けおよび実行ユニット130、140および150へのディスパッチをおこなう順序付けユニット124に渡す。実行ユニット130、140および150は、順序付けユニット124から発行された命令を実行する。発行された命令が完全に実行されると、実行ユニット130、140および150は、結果があればそれを関連汎用レジスタまたは他のレジスタ (図示せず) に記憶し、さらに、命令の実行が終了したことを完了ユニット190に通知する。

20

【0028】

図1に示した特定のプロセッサ構造は単に例示目的で示したに過ぎないこと、および本発明に基づく変更したスレッド選択はこの特定の構造とともに使用することだけに限定されないことを理解されたい。例えば、プロセッサ100には、シーケンシャル・フェッチャ120が命令ストリームから抜き取った分岐命令が示されているが、別のプロセッサ構造はこの分岐命令ユニットを、他の実行ユニット130、140および150と一緒に別の実行ユニットとして取り扱うことができる。いずれにせよ、本発明に基づく変更したスレッド選択は、実質的に任意の同時マルチスレッディング・プロセッサに組み込むことができる。

30

【0029】

さらに、図示のプロセッサ100は、2つの命令スレッドT0とT1をインタリーブして、フェッチ・ステージ後の各種プロセッサ・パイプライン・ステージで処理するインタリーブされた単一のストリームとるように適合されているが、他の実施形態では、2つ以上の命令スレッドをインタリーブして、インタリーブされた単一の命令ストリームとるように適合させることもできることに留意されたい。同時に処理している命令スレッドの数に関わらず、所与のクロック・サイクルにインタリーブされた命令ストリームに渡すことができる命令は一般に、単一のスレッドからの命令 (またはパイプラインが複数の命令に相当する幅を有する場合には命令セット) だけであることに留意されたい。したがって、MUX105などの選択装置を介して2つ以上のスレッドをインタリーブする場合には、装置は、MUXを通して転送する特定の1つのスレッドを選択するのに適したMUX制御信号を生成する論理を含んでいなければならない。図1に示した本発明の2スレッド形態では、信号経路154を通してMUX制御信号を適用することができる。しかし、2つ以上の入力を有する選択MUXに対する制御信号には、必要なMUX制御信号を収容する追加の信号経路が必要となる。

40

【0030】

本発明の好ましい実施形態は、一方または両方の命令スレッドT0、T1からの命令を保留し、それによって保留されたそれぞれのスレッドからの有効な命令が渡されて処理されないようにする能力を有する。この保留能力は、2つのラッチ103および104に適

50

用された保留信号によって実装することができる。図1には、選択コントローラ106から2つのラッチ103および104への破線152および153が示されている。選択コントローラ106は、それぞれのスレッドに対して必要な保留信号を生成し、それらを信号経路152または153に適用する。ラッチ103または104に適用されると、保留信号は、そのラッチに記憶された1つまたは複数の命令に実行無効の印を付ける。本発明に基づく保留信号については後に図3を参照して詳細に論じる。

【0031】

図2に、2つの命令スレッドからの命令のインタリーブを制御するように適合された好ましい選択コントローラ106を示す。選択コントローラは、インタリーブ規則執行構成要素203、変更構成要素205および出力構成要素225を含む。インタリーブ規則執行構成要素203は、別々の2つの基準スレッド選択信号を変更構成要素205に供給する。基準スレッド選択信号はそれぞれ、プロセッサ100内でインタリーブされる命令スレッドの1つに関連付けられている。変更構成要素205は、それぞれの基準スレッド選択信号について、基準スレッド選択信号の状態およびプロセッサ100の他の要素の事象または条件に由来するフィードバック信号に基づいて、別々の最終スレッド選択信号を生成する。この2つの最終スレッド選択信号は出力構成要素225によって結合されて、処理するインタリーブされた命令ストリームにインタリーブするためにプロセッサ100内の2つの命令スレッドの一方を選択する際に使用する単一のスレッド選択制御信号が生成される。出力構成要素225の好ましい一形態を図3を参照して後に詳細に説明する。それぞれの命令スレッドの保留信号としてこの最終スレッド選択信号を使用することについて図3を参照して論じる。

10

20

【0032】

図2を詳細に参照する。インタリーブ規則執行構成要素203は優先度信号TP0およびTP1を受け取り、別々の2つの出力信号をノード215および217に適用する。優先度信号TP0は命令スレッドT0に割り当てられた優先度を指示し、優先度信号TP1は命令スレッドT1に割り当てられた優先度を指示する。インタリーブ規則執行構成要素203によって生成され、ノード215および217に適用される2つの出力は、特定のスレッドにそれぞれ関連付けられた基準スレッド選択信号を含む。ノード215の基準スレッド選択信号はスレッドT0に関連付けられており、入力優先度TP0およびTP1に基づく2つのスレッド間のインタリーブ規則の執行に使用することができる2進信号を含む。ノード217の基準スレッド選択信号はスレッドT1に関連付けられており、ノード215の信号の補数である2進信号を含み、したがってやはり、2つのスレッド間のインタリーブ規則の執行に使用することができる。例えば、所与のクロック・サイクルにインタリーブ規則執行構成要素から出力されたノード215のレベル0の信号は、そのクロック・サイクルにスレッドT0を選択して、インタリーブされた命令ストリームへ命令を渡すことを指示する。ノード215のレベル1の論理信号は、そのクロック・サイクルにスレッドT1を選択して、インタリーブされた命令ストリームへ命令を渡すことを指示する。この例では、ノード217のレベル0の信号が、そのクロック・サイクルにスレッドT1を選択して、インタリーブされた命令ストリームへ命令を渡すことを指示し、ノード217のレベル1の論理信号が、そのクロック・サイクルにスレッドT0を選択することを指示する。

30

40

【0033】

インタリーブ規則執行構成要素203は、所望のインタリーブ規則を執行する特定の命令スレッドを指示する所望の出力信号を生成する適当な任意の構成要素または回路を含むことができる。無用な詳細によって本発明が不明瞭にならないよう、インタリーブ規則執行構成要素203のこれ以上の詳細は省く。

【0034】

ノード215および217の基準スレッド選択信号と同様に、図2の変更構成要素205によって生成されるノード253および255の最終スレッド選択信号もそれぞれ、インタリーブされた命令ストリーム処理に命令を渡すために選択する命令スレッドを指示す

50

る。しかし、最終スレッド選択信号が指示する命令スレッドと基準スレッド選択信号が指示する命令スレッドとは異なる場合がある。具体的には、変更構成要素205は、プロセッサに含まれるさまざまな要素内でのある事象または条件を考慮して、所与のクロック・サイクルにおける所与の最終スレッド選択信号の状態を、基準スレッド選択信号の状態とは逆の状態にすることができる。したがって本発明は、サイクル単位でインタリーブされるSMTプロセッサのインタリーブ規則執行構成要素の下流でスレッド選択を変更して、単に命令スレッド間の優先規則を執行することによって可能な競合命令スレッド間のプロセッサ・リソースの割当てよりも望ましい割当てを達成する能力を有する。

【0035】

図2の変更構成要素205は、2組のフィードバック入力経路219および221を通して、プロセッサ事象または条件フィードバックを受け取る。本発明の図示の形態では、フィードバック入力経路219がスレッドT0に関連付けられており、フィードバック経路221がスレッドT1に関連付けられている。これらの2つのフィードバック経路セット上の信号は、それぞれのORゲート231および233によって結合されて、変更構成要素205が使用するフィードバック信号が生成される。これらのフィードバック信号は、ノード215および217の基準スレッド選択信号とともに、ノード253および255の2つの最終スレッド選択信号を生成するのに使用される。本発明の好ましい形態では、後に詳細に論じるフィードバック・オーバーライド信号の制御下で、これらのフィードバック信号がそれぞれANDゲート211および213によってゲートされる。

【0036】

例示のため図2に示した変更構成要素205は、2つの命令スレッドT0およびT1ならびにノード215および217に適用された2つの基準スレッド選択信号に対応する2本のブランチを含む。この別個のブランチの配置は、後に論じるように、最終スレッド選択信号を保留信号として使用することを容易にする。図2に示した変更構成要素205の上側のブランチは、ノード215の信号を受け取り、さらにORゲート231および233によって生成され、最終的にノード232および234に適用された両方のフィードバック信号を受け取る。ノード215の信号は、T0基準スレッド選択信号と呼ぶことができ、ANDゲート229および230の入力227および228に適用される。ノード232のフィードバック信号は、ANDゲート229の入力235およびANDゲート237の入力236に適用される。ノード234のフィードバック信号はインバータ241および243によって反転され、ANDゲート230の入力ノード245およびANDゲート237の入力ノード249に適用される。ANDゲート229、230および237の出力はORゲート251の入力に適用されており、そのため、これらのANDゲートの1つが、アサートされた(論理レベル1の)出力信号を生成する場合には、ORゲート251の出力ノード253もアサートされる。

【0037】

変更構成要素205の上ブランチのこの論理装置配置は、ノード215、232、234および253の信号レベルを表す以下の真理値表を与える。

	ノード215	ノード232	ノード234	ノード253
(1)	0	0	0	0
(2)	0	0	1	0
(3)	0	1	0	1
(4)	0	1	1	0
(5)	1	0	0	1
(6)	1	0	1	0
(7)	1	1	0	1
(8)	1	1	1	1

表1

【0038】

上の真理値表から、本発明のこの形態では、ノード215のT0基準選択信号とノード

10

20

30

40

50

253の最終スレッド選択信号が異なるのは2例だけであることが分かる。表の第6行に示す最初の事例では、T0基準スレッド選択信号が1であり、この特定の論理においてこれは、当初、構成要素203によって執行されたインタリーブ規則に単純に基づいて、スレッドT0は処理するスレッドとして選択されていないことを指示している。しかし、変更構成要素205は、ノード215のこの状態の信号を変更し、そのクロック・サイクルにインタリーブされたストリームに命令を導入するスレッドとしてT0スレッドを選択することを指示するレベル0の最終スレッド選択信号をノード253に生成する。このことが起こるのは、ノード232の信号の論理レベルが0、ノード234の信号の論理レベルが1のときだけである。ノード234の信号は、任意のT1フィードバック入力経路がアサートされているとき、したがってその経路がスレッドT1に関連したある事象または条件を指示しているときに論理レベル1をとることができる。219のフィードバック入力はいずれもアサートされておらず、したがってスレッドT0が正常に処理されていることを指示しているので、変更構成要素205は、215のT0基準スレッド選択信号を変更して、T1命令スレッドの代わりにT0命令スレッドを選択することを指示するレベル0の最終スレッド選択信号をノード253に生成する。

【0039】

2番目の事例を表の(3)行に示す。ノード215の論理値0の基準スレッド選択信号が指示しているように、この例では当初、スレッドT0は処理するスレッドとして選択されている。しかし、ノード232のフィードバック信号は論理レベル1にアサートされており、したがってスレッドT0に関連したあるプロセッサ条件を指示しており、かつノード234のフィードバック信号の論理レベルは0であり、スレッドT1に関連した遅延プロセッサ条件がないことを指示しているので、変更構成要素205は、インタリーブされたストリームへ命令を渡すスレッドとしてT0スレッドを選択しないことを指示する、T0基準スレッド選択信号とは逆のT0スレッド最終スレッド選択信号をノード253に生成する。

【0040】

論理要素229a、230a、237a、251a、241aおよび243aを含む図2に示した変更構成要素205の下側のブランチは上ブランチの鏡像となっていることに留意されたい。したがって、変更構成要素205の下ブランチの論理に関連した真理値表は、上ブランチ論理に関して先に示した真理値表の鏡像であり、ノード253の最終スレッド選択信号とノード255の最終スレッド選択信号は全てのケースで互いの補数となる。

【0041】

本発明は、ノード253の最終スレッド選択信号とノード255の最終スレッド選択信号が常に互いの補数であるケースに限定されないことに留意されたい。変更構成要素205の代替論理配置では、ノード253の最終スレッド選択信号とノード255の最終スレッド選択信号がともに、インタリーブされた命令ストリームに命令を送るスレッドとしてそれぞれのスレッドを選択しないことを指示することができる。この状況については図3を参照して後に詳細に論じる。さらに、本発明は、セット219および221の特定のフィードバック入力に限定されないことに留意されたい。これらのフィードバック信号を、キャッシュ・ミスなどのある長い待ち時間事象、またはスレッド選択信号を変更することが好ましい他のプロセッサ事象に応答してアサートすることができる。

【0042】

本発明の好ましい一形態では、ORゲート231および233からのフィードバック信号出力がそれぞれANDゲート211および213によってゲートされる。ANDゲート211および213はそれぞれ、フィードバック信号の禁止ないしオーバライドを容易にする信号経路261および262上のフィードバック・オーバライド制御信号によって制御される。具体的には、経路261および262上の信号がアサートされている場合、すなわちその論理レベルが1である場合に、セット219および221の中の任意の入力のアサートされた信号が最終的にノード232および234に渡される。しかし、信号経路

261および262の低レベル論理信号は、高レベル信号がノード232および234に渡されることを事実上阻止する。変更構成要素の上ブランチに対する先の真理値表によれば、ノード232および234の高レベル論理信号が阻止されると、ノード253の最終スレッド選択信号の状態が、ノード215の基準スレッド選択信号の状態とは異なる状態をとることはない。すなわち、ゲート211および213によってフィードバックを無効にする（にオーバライドする）ことによって、変更構成要素205は基準スレッド選択信号を事実上変更なしに通過させる。

【0043】

図2に示した本発明の形態は、ORゲート231および233の出力に生成されるフィードバック信号をゲートするが、本発明の代替形態は、フィードバック入力経路セット219および221に含まれる個々のフィードバック入力経路をゲートする。本発明のこれらの代替形態では、セット219および221に含まれる個々のフィードバック入力がそれぞれ、ゲート211および213と同様の別個のゲートを含む。これらの個々のフィードバック入力経路ゲートはそれぞれ、他のフィードバック入力経路を使用可にしたままそれぞれのフィードバック入力経路を希望どおりに使用禁止にするそれぞれの制御信号によって制御される。ゲートまたは同様の装置を使用して1つまたは複数のフィードバック経路219および221を禁止にする本発明の任意の形態の制御信号は、ソフトウェアによって制御可能なレジスタによって供給することができる。

【0044】

本発明の一形態では、フィードバック・オーバライド制御構成要素264を使用して、オーバライド信号経路261および262上に信号を生成する。フィードバック信号を無効にするためにオーバライド信号を適用する具体的な1つの状況は、入力スレッド優先度が等しくない状況である。したがって、フィードバック・オーバライド制御264は、図2の破線265および266によって指示されているようにTP0およびTP1の値を受け取ることができ、TP0とTP1が等しいかまたはほぼ等しい場合にだけ高レベル論理信号をゲート211および213に供給する論理を含むことができる。TP0とTP1が等しくない場合には、フィードバック・オーバライド制御262の論理は、経路261および262上の低レベル阻止信号をそれぞれゲート211および213に生成して、ORゲート231および233の出力のフィードバック信号を事実上使用禁止にする。TP0の値とTP1の値が等しくない場合にフィードバック信号を使用禁止にするこのプロセスは、スレッド優先度の大きな違いによって決定されたスレッド選択頻度が変更構成要素205の影響を受けることを防ぐ。

【0045】

図1および2に示した選択コントローラ106の最終副構成要素である出力構成要素225は、ノード253および255からの最終スレッド選択信号を結合し、選択制御信号を生成する。この選択制御信号は信号経路154に適用される。図1に示したとおり、経路154上のこの信号は、2つのスレッドからの命令をインタリーブするための2つの命令スレッド間の選択を実施するMUX105を制御する。

【0046】

図3に、選択コントローラ106で使用される出力構成要素225の好ましい一形態を示す。この特定の出力構成要素225は、スレッドT0に関連したノード253の最終スレッド選択信号およびスレッドT1に関連したノード255の最終スレッド選択信号がともに、それぞれの関連スレッドを処理するスレッドとして選択しないことを指示している、すなわち両方の信号が論理1である状況に対処する。図示の選択結合構成要素225は、ANDゲート403、ANDゲート407およびORゲート411を含んでいる。ANDゲート403は、ノード253からT0最終スレッド選択信号を受け取り、さらに反転されたT1最終スレッド信号を受け取るように接続されている。反転されたT1最終スレッド信号は、インバータ417によって反転された後のノード255の信号を含む。ANDゲート407は、ノード253からT0最終スレッド選択信号を受け取り、ノード255からT1最終スレッド選択信号を受け取るように接続されている。ANDゲート407

の第3の入力は、1クロック・サイクルごとに両方の論理状態間を交互にトグルするトグル信号を含む。インタリーブ規則執行構成要素はトグル信号発生器を含むので、この信号は、図3に示すようにインタリーブ規則執行構成要素203から得ることができる。しかし、このトグル信号は適当な任意の方法で生成することができることを理解されたい。ANDゲート403および407の出力はともにORゲート411に入力される。ORゲート411は論理OR演算を実行して、MUX105（図1にも示されている）に適用される選択制御信号を信号経路154上に生成する。

【0047】

本発明の図示の形態に適用される論理では、レベル0の最終スレッド選択信号は、その信号に関連付けられたスレッドを、その特定のクロック・サイクルにインタリーブされたストリームに命令を含めるスレッドとして選択することを指示する。ノード253の信号とノード255の信号を互いの補数信号とし、または両方の信号がともに論理レベル1をとることができることに留意されたい。ただし、ノード253の信号とノード255の信号の論理レベルがともに0となることはない。図示の出力構成要素225の論理構成では、ノード253とノード255がともに論理1であるときを除いて、経路154上の信号はノード253の信号と同じになる。この場合、ゲート407のトグル信号は、ノード253および255の信号がともに論理レベル1である限り、経路154上の信号出力が論理0と論理1の間を交互に切り替わるように強制する。

【0048】

両方の最終スレッド選択信号がそのクロック・サイクルにそれぞれの関連スレッドを選択しないことを指示することができる本発明の実施態様では特に、最終スレッド選択信号（図3のノード253および255）を、図1の待ち行列101および102の命令に保留信号として適用することが有利であることがある。このような保留能力は、前の段落で論じたように出力構成要素225からの選択信号の制御下でMUX105が2つの命令スレッド間を交互にトグルしたときに、待ち行列101および102からの有効な命令がMUX105を介してインタリーブされた命令ストリームに入ることを妨げる。したがって、図3に示した本発明の形態は、ノード253のT0最終スレッド選択信号を保留信号経路153を通してラッチ103の保留入力に適用する。同様に、ノード255のT1最終スレッド選択信号は、保留信号経路152を通してラッチ104の保留入力に適用される。図1のラッチ103および104はそれぞれ、待ち行列101および102の最終位置を含むことに留意されたい。ラッチ103および104はそれぞれ、アサートされた（論理レベル1の）保留信号に 응답して、それぞれの1つまたは複数の記憶位置を無効状態にセットする。したがって、1つまたは複数の命令に対するデータはそれぞれのラッチに保持され、MUX105に駆動され、MUXがその特定のラッチの出力を選択したときにインタリーブされた命令ストリームに駆動されるが、無効と指示された命令は完全には処理されない。したがって、無効命令がさまざまなプロセッサ要素から「見える」場合であっても、ラッチ103および104は保留要素の働きをする。このようにインタリーブされたストリームに無効命令をインタリーブすることができることはいくつかの状況で有益であろう。

【0049】

以下の例は、スレッド選択MUX105の制御にも関与しながら保留信号としても機能するノード253および255の最終スレッド選択信号の働きを説明するのに役立つ。フィードバックが処理上の問題を指示しているため、T0命令スレッドとT1命令スレッドの両方を保留すると仮定する。本発明の図示の実施態様では両方の信号の論理レベルが1である。これらの高レベル論理信号はさらにラッチ103および104に適用され、これによって、ラッチされた命令が無効であることを指示するようにラッチがセットされる。ノード253および255のT0およびT1最終スレッド選択信号が、どちらのスレッドからの命令も処理にまわされないことを指示する高論理レベルにある場合でも、ANDゲート407のトグル入力が、選択MUX105の2つの入力間を交互に切り替わるよう経路154の出力を強制する。したがって、両方のスレッドが保留されている間も、それぞ

10

20

30

40

50

れのラッチ 103 および 104 に保持された命令がインタリーブされたストリームに渡される。しかし、これらの命令は無効と指示されているので、プロセッサを通過しても処理されない。

【0050】

本発明は、さまざまなスレッドに対する最終スレッド選択信号が相互排除であり、そのため、インタリーブされたストリームに命令をインタリーブするスレッドとして、1クロック・サイクルごとに1つのスレッドから命令が選択される実施形態を包含することに留意されたい。これらの実施形態では、最終スレッド選択信号を、命令待ち行列に関連付けられたラッチの保留信号として適用する必要はない。これらの実施形態では最終スレッド選択信号が、図1のMUX105などの選択MUXの選択制御信号を生成するためだけに使用される。命令スレッド間の排他性はさらに、本出願の図2および3、ならびに参照によって本明細書に組み込まれる関連出願に示した二重命令ブランチ配置を不要にすることを容易にする。

10

【0051】

以上に記載した好ましい実施形態は、本発明の原理を説明することを意図したものであり、本発明の範囲を限定しようとするものではない。当業者であれば、上記請求項の範囲から逸脱することなく、他のさまざまな実施形態およびこれらの好ましい実施形態に対するさまざまな変更が可能である。例えば、図示の論理および他の回路は、開示の発明を実現する方法の具体的な例を提供するが、開示の回路は正または負論理で実現することもできる。さらに、異なる組合せの論理ゲートを使用して同じ結果を得ることができ、このよう

20

【0052】

まとめとして、本発明の構成に関して以下の事項を開示する。

【0053】

(1) 同時マルチスレッディング・プロセッサにおけるいくつかの命令スレッド間の命令のサイクル単位のインタリーブを制御する方法であって、

30

(a) 前記いくつかの命令スレッドの中の処理する第1の命令スレッドを指示する基準スレッド選択信号を、命令インタリーブ実行規則に少なくとも部分的に基づいて生成する段階と、

(b) 前記いくつかの命令スレッドに含まれる特定の命令スレッドを指示する最終スレッド選択信号を、前記基準スレッド選択信号と前記いくつかの命令スレッドに含まれる1つの前記命令スレッドに関連付けられたフィードバック信号とに基づいて生成する段階と、

(c) インタリーブされた命令ストリームに命令をインタリーブするために前記いくつかの命令スレッドのうちの1つの命令スレッドを選択する選択に、前記最終スレッド選択信号を適用する段階と

40

を含む方法。

(2) (a) 前記いくつかの命令スレッドの中の処理する追加の命令スレッドを指示する追加の基準スレッド選択信号を、前記命令インタリーブ実行規則に少なくとも部分的に基づいて生成する段階と、

(b) 前記いくつかの命令スレッドに含まれるそれぞれの命令スレッドを指示する追加の最終スレッド選択信号を、前記追加の基準スレッド選択信号と前記いくつかの命令スレッドに含まれる1つの前記命令スレッドに関連付けられた追加のフィードバック信号とに基づいて生成する段階と

50

をさらに含む、上記(1)に記載の方法。

(3) 前記最終スレッド選択信号と前記追加のスレッド選択信号を結合してスレッド選択MUX制御信号を生成する段階をさらに含む、上記(2)に記載の方法。

(4) 前記追加の基準スレッド選択信号が前記基準スレッド選択信号の2進補数である、上記(2)に記載の方法。

(5) (a) 前記最終スレッド選択信号を、前記いくつかの命令スレッドに含まれる第1の命令スレッドに関連付けられた保留入力に適用する段階と、

(b) 前記追加の最終スレッド選択信号を、前記いくつかの命令スレッドに含まれる追加の命令スレッドに関連付けられた保留入力に適用する段階と

をさらに含む、上記(2)に記載の方法。

10

(6) 前記フィードバック信号が前記第1の命令スレッドに関連付けられている、上記(1)に記載の方法。

(7) 前記いくつかの命令スレッドが異なる処理優先度を有するときに前記フィードバック信号を無効にする段階をさらに含む、上記(1)に記載の方法。

(8) 同時マルチスレッディング・プロセッサにおけるいくつかの命令スレッド間の命令のサイクル単位のインタリーブを制御する方法であって、

(a) 命令インタリーブ実行規則に少なくとも部分的に基づく基準スレッド選択信号を受け取る段階であって、前記基準スレッド選択信号が、前記いくつかの命令スレッドに含まれる処理のために選択された第1の命令スレッドを指示する段階と、

(b) 前記いくつかの命令スレッドに含まれる1つの前記命令スレッドに関連付けられたフィードバック信号を命令処理要素から受け取る段階と、

20

(c) 前記いくつかの命令スレッドに含まれる特定の命令スレッドを指示する最終スレッド選択信号を、前記基準スレッド選択信号および前記フィードバック信号に基づいて生成する段階と、

(d) インタリーブされた命令ストリームに命令をインタリーブするために1つの前記命令スレッドを選択する選択に、前記最終スレッド選択信号を適用する段階と

を含む方法。

(9) (a) 前記命令インタリーブ実行規則に少なくとも部分的に基づく追加の基準スレッド選択信号を受け取る段階であって、前記追加の基準スレッド選択信号が、前記いくつかの命令スレッドに含まれる処理のために選択された追加の命令スレッドを指示する段階と、

30

(b) 前記いくつかの命令スレッドに含まれる1つの前記命令スレッドに関連付けられた追加のフィードバック信号を命令処理要素から受け取る段階と、

(c) 前記いくつかの命令スレッドに含まれるそれぞれの命令スレッドを指示する追加の最終スレッド選択信号を、前記基準スレッド選択信号および前記フィードバック信号に基づいて生成する段階と

をさらに含む、上記(8)に記載の方法。

(10) 前記最終スレッド選択信号と前記追加の最終スレッド選択信号を結合してスレッド選択MUX制御信号を生成する段階をさらに含む、上記(9)に記載の方法。

(11) 前記追加の基準スレッド選択信号が前記基準スレッド選択信号の2進補数である、上記(9)に記載の方法。

40

(12) (a) 前記最終スレッド選択信号を、前記いくつかの命令スレッドに含まれる第1の命令スレッドに関連付けられた保留入力に適用する段階と、

(b) 前記追加の最終スレッド選択信号を、前記いくつかの命令スレッドに含まれる追加の命令スレッドに関連付けられた保留入力に適用する段階と

をさらに含む、上記(9)に記載の方法。

(13) 前記フィードバック信号が前記第1の命令スレッドに関連付けられている、上記(8)に記載の方法。

(14) 前記いくつかの命令スレッドが異なる処理優先度を有するときに前記フィードバック信号を無効にする段階をさらに含む、上記(8)に記載の方法。

50

(15) 同時マルチスレッディング・プロセッサにおけるいくつかの命令スレッド間の命令のサイクル単位のインタリーブを制御する回路であって、

(a) 前記いくつかの命令スレッドの中の第1の命令スレッドを指示する第1のスレッド選択信号を受け取るように接続された第1の選択入力と、

(b) 前記いくつかの命令スレッドに含まれる1つの前記命令スレッドに関連付けられたプロセッサ要素フィードバック信号を受け取るように接続されたフィードバック入力と、

(c) 前記基準スレッド選択信号および前記フィードバック信号に基づいて最終スレッド選択信号を生成する変更構成要素と、

(d) インタリーブされた命令ストリームに命令をインタリーブするために前記いくつかの命令スレッドのうちの1つ命令スレッドを選択する選択に前記最終スレッド選択信号を適用する出力構成要素と

を含む回路。

(16) 前記いくつかの命令スレッドの中の第2の命令スレッドを指示する第2の基準スレッド選択信号を受け取るように接続された第2の選択入力をさらに含む、上記(15)に記載の回路。

(17) (a) 前記変更構成要素がさらに、前記第2のスレッド選択信号および第2のフィードバック信号に基づいて追加の最終スレッド選択信号を生成し、

(b) 前記出力構成要素がさらに、インタリーブされた命令ストリームに命令をインタリーブするために前記いくつかの命令スレッドのうちの1つの命令スレッドを選択する選択に前記追加の最終スレッド選択信号を適用する、

上記(16)に記載の回路。

(18) (a) 前記最終スレッド選択信号の状態に基づいて第1の命令スレッドを選択的に保留する第1のスレッド保留要素と、

(b) 前記追加の最終スレッド選択信号の状態に基づいて第2の命令スレッドを選択的に保留する第2のスレッド保留要素と

をさらに含む、上記(17)に記載の回路。

(19) 前記第1のフィードバック信号と前記第2のフィードバック信号のそれぞれに対するフィードバック・オーバライド回路をさらに含み、前記フィードバック・オーバライド回路が、前記いくつかの命令スレッドが異なる処理優先度を有するときにそれぞれの前記フィードバック信号を無効にする、上記(17)に記載の回路。

(20) 前記いくつかの命令スレッドが異なる処理優先度を有するときに前記第1のフィードバック信号を無効にする第1のフィードバック・オーバライド回路をさらに含む、上記(15)に記載の回路。

【図面の簡単な説明】

【0054】

【図1】本発明に基づくスレッド選択配置を含むプロセッサのブロック図である。

【図2】本発明の原理を具体化したスレッド選択変更構成要素を含む、図1のスレッド選択コントローラの好ましい一形態のブロック図/構成図である。

【図3】本発明に基づく選択制御回路を示すブロック図/構成図である。

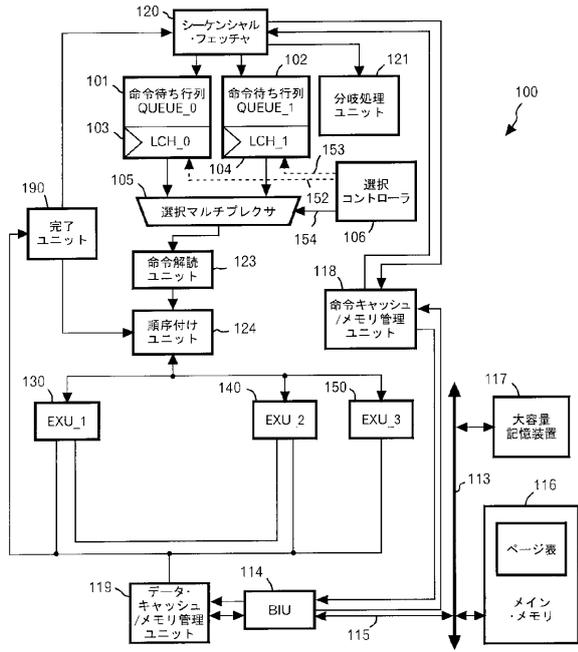
【符号の説明】

【0055】

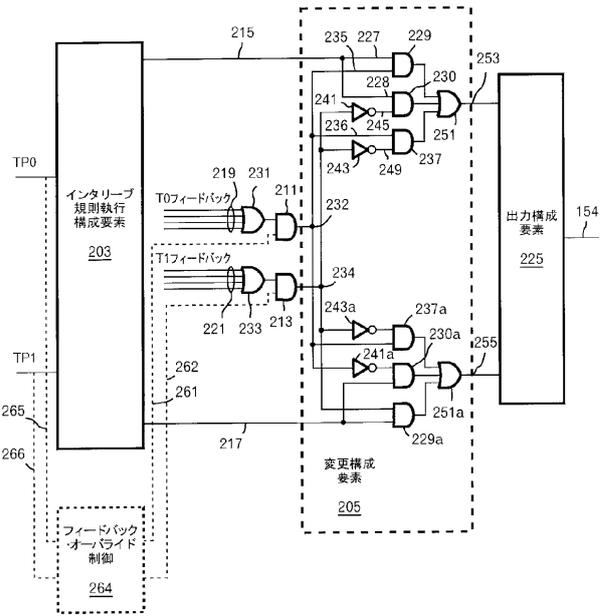
- 100 プロセッサ
- 101 第1の命令待ち行列
- 102 第2の命令待ち行列
- 103 第1のラッチ
- 104 第2のラッチ
- 105 選択マルチプレクサ
- 106 選択コントローラ
- 113 システム・バス

1 1 4	バス・インタフェース・ユニット	
1 1 5	プロセッサ・バス	
1 1 6	メイン・メモリ	
1 1 7	不揮発性大容量記憶装置	
1 1 8	命令キャッシュ/メモリ管理ユニット	
1 1 9	データ・キャッシュ/メモリ管理ユニット	
1 2 0	シーケンシャル・フェッチャ	
1 2 1	分岐処理ユニット	
1 2 3	命令解読ユニット	
1 2 4	順序付けユニット	10
1 3 0	実行ユニット	
1 4 0	実行ユニット	
1 5 0	実行ユニット	
1 5 2	保留信号経路	
1 5 3	保留信号経路	
1 5 4	スレッド選択制御経路	
1 9 0	完了ユニット	
2 0 3	インタリーブ規則執行構成要素	
2 0 5	変更構成要素	
2 1 1	A N Dゲート	20
2 1 3	A N Dゲート	
2 1 5	基準スレッド選択信号	
2 1 7	基準スレッド選択信号	
2 1 9	フィードバック入力経路	
2 2 1	フィードバック入力経路	
2 2 5	出力構成要素	
2 2 9	A N Dゲート	
2 2 9 a	A N Dゲート	
2 3 0	A N Dゲート	
2 3 0 a	A N Dゲート	30
2 3 1	O Rゲート	
2 3 3	O Rゲート	
2 3 7	A N Dゲート	
2 3 7 a	A N Dゲート	
2 4 1	インバータ	
2 4 1 a	インバータ	
2 4 3	インバータ	
2 4 3 a	インバータ	
2 5 1	O Rゲート	
2 5 1 a	O Rゲート	40
2 5 3	最終スレッド選択信号	
2 5 5	最終スレッド選択信号	
2 6 4	フィードバック・オーバライド制御構成要素	
4 0 3	A N Dゲート	
4 0 7	A N Dゲート	
4 1 1	O Rゲート	
4 1 7	インバータ	

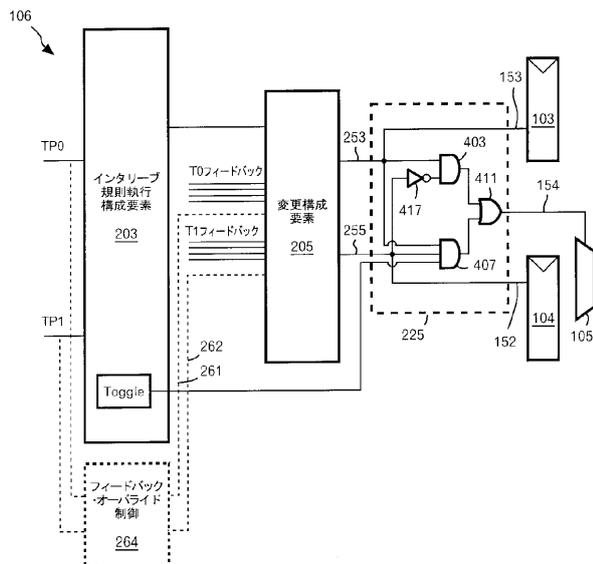
【図 1】



【図 2】



【図 3】



フロントページの続き

- (72)発明者 ロナルド・エヌ・カラ
アメリカ合衆国78664 テキサス州ラウンド・ロック ヒルトン・ヘッド・ドライブ 214
3
- (72)発明者 ミン・ミシェル・キュー・ファム
アメリカ合衆国78753 テキサス州オースチン ノース・ラマー 10401 ナンバー F
102
- (72)発明者 パララム・シンハロイ
アメリカ合衆国12601 ニューヨーク州ポキプシー ハドソン・ハーバー・ドライブ 15 E
- (72)発明者 ジョン・ダブリュー・ワード 3世
アメリカ合衆国78660 テキサス州プリューガーヴィル ガイザー・アベニュー 408
- Fターム(参考) 5B013 AA01
5B098 GC03 GC20

【要約の続き】