

(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2004-295195

(P2004-295195A)

(43) 公開日 平成16年10月21日(2004.10.21)

(51) Int. Cl.⁷

G06F 9/38
G06F 9/46

F I

G06F 9/38 310F
G06F 9/38 370X
G06F 9/46 313C
G06F 9/46 340B

テーマコード(参考)

5B013
5B098

審査請求有 請求項の数 13 O L (全 18 頁)

(21) 出願番号 特願2003-83001 (P2003-83001)
(22) 出願日 平成15年3月25日(2003.3.25)

(71) 出願人 503360115
独立行政法人 科学技術振興機構
埼玉県川口市本町4丁目1番8号
(71) 出願人 899000079
学校法人慶應義塾
東京都港区三田2丁目15番45号
(74) 代理人 100107010
弁理士 橋爪 健
(72) 発明者 山崎 信行
神奈川県横浜市神奈川区六角橋2-21-4
Fターム(参考) 5B013 AA05 AA12 AA13 DD02
5B098 AA02 DD03 GA05 GC03 GC07

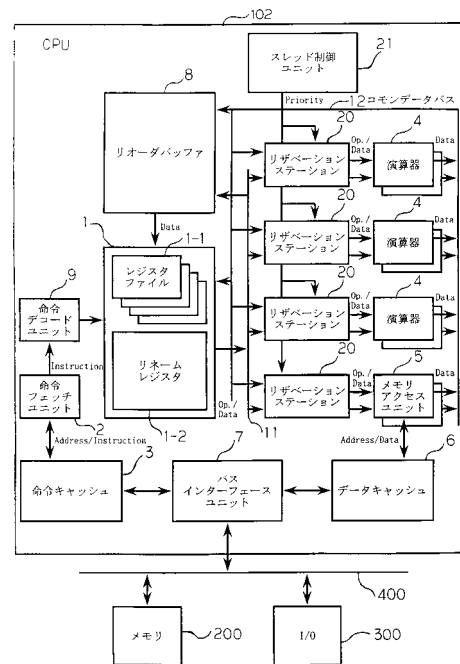
(54) 【発明の名称】 命令発行方法及び装置、中央演算装置、命令発行プログラム及びそれを記憶したコンピュータ読み取り可能な記憶媒体

(57) 【要約】

【課題】 マルチスレッドプロセッサにおいて実時間処理を、命令の優先度による追い越しにより実現する。

【解決手段】 命令フェッチユニット2は、命令キャッシュ3から命令をフェッチし、命令デコードユニット9に送る。命令デコードユニット9は、命令をデコードする。リザベーションステーション20は、レジスタファイル1-1から、命令と命令実行に必要なオペランドを得る。スレッド制御ユニット21は、各スレッドに対する優先度を設定する。リザベーションステーション20は、命令実行に必要な全てのオペランドを得ていない場合、コモンデータバス12を監視し、必要なオペランドが来ればそれを取り込む。リザベーションステーション20は、命令実行に必要な全てのオペランドがそろったエンタリの中から最も高い優先度を持つエンタリを選択し、その命令及びオペランドを演算器4に送る。演算器4はその命令及びオペランドに従い命令を実行する。

【選択図】 図5



【特許請求の範囲】

【請求項 1】

中央演算装置が複数のスレッドの命令を実行するマルチスレッドプロセッサにおける命令発行装置であって、
各スレッドに対応して優先度を設定するためのスレッド制御ユニットと、
演算処理及び/又はメモリアクセス処理において実行すべきスレッドの命令及びオペランドが記憶されたレジスタファイルと、
スレッド識別子(スレッドID)に対応して、優先度情報と、命令と、該命令の実行に用いる各オペランドを示すデータとを含むエントリを記憶し、前記レジスタファイルからデータを受け取り、命令及びオペランドを発行するリザベーションステーションと、
前記リザベーションステーションから発行された命令及びオペランドを受け取り、該命令に従い各種演算を行い、演算結果を前記レジスタファイルに書き戻す演算器と、
を備え、
前記リザベーションステーションは、
前記演算器からの出力を監視し、前記レジスタファイルで未解決のオペランドである必要なデータが前記演算器から演算結果として出力された場合、その演算結果をオペランドとして取り込む手段と、
スレッドIDに対応した優先度情報を前記スレッド制御ユニットから求める手段と、
命令実行に必要な全てのオペランドがそろった命令であって、且つ、優先度情報に従い優先度の高いスレッドの命令を前記演算器へ送る手段と
を有することにより、前記リザベーションステーションから前記演算器への命令発行を高い優先度のスレッドの命令が低い優先度の命令を追い越すようにした前記命令発行装置。

10

20

【請求項 2】

前記リザベーションステーションは、デコードされた命令を基に前記レジスタファイルから読み出されたオペランドを格納し、各オペランドに対してデータが有効であることを示す有効ビットをさらに含む、エントリを作成する手段と、
前記リザベーションステーションは、該リザベーションステーションに記憶された各エントリに対して、命令実行に必要なオペランドが全てそろうまで前記演算器の演算結果から必要なデータをオペランドとして取り込み、そのデータに対応する有効ビットをセットし、各エントリに対するスレッドIDに従い前記スレッド制御ユニットを参照して優先度を得てセットすることにより、各エントリを更新する手段と、
前記リザベーションステーションは、全ての有効ビットがセットされているエントリをオペランドがそろった演算可能なエントリを判断する手段と、
前記リザベーションステーションは、オペランドがそろったエントリが複数の演算可能である場合、各エントリの優先度を比較して、より高い優先度を持つエントリを選択して、前記演算器に該エントリの命令及びオペランド送る手段と
を含む請求項 1 に記載の命令発行装置。

30

【請求項 3】

前記リザベーションステーションは、スレッドの命令の優先度を、エントリ更新又は命令発行の処理クロック毎に更新制御することを特徴とする請求項 1 に記載の命令発行装置。

40

【請求項 4】

前記リザベーションステーションは、命令がエントリに入力又は作成されたときにカウンタをリセットし、以後エントリを更新する処理クロック毎に又は命令を発行する処理クロック毎に該カウンタをカウントアップする手段と、
前記リザベーションステーションは、優先度の高いエントリが複数ある場合に、カウンタに従いその値の大きいエントリを選択して、その命令及びオペランドを前記演算器に発行する手段と、
をさらに含むことを特徴とする請求項 1 に記載の命令発行装置。

【請求項 5】

前記リザベーションステーションからの命令及びオペランドを受け取り、前記レジスタフ

50

ファイル又はデータキャッシュへのロード又はストアを実行するメモリアクセスユニットをさらに備えた請求項 1 に記載の命令発行装置。

【請求項 6】

前記リザベーションステーション及び前記演算器の組は複数設けられ、各々の前記演算器によりアウトオブオーダーに演算された命令の順番を元の順番に戻し、元の順番にもどした演算結果を前記レジスタファイルへ書き込むリオーダーバッファと、各々の前記演算器の出力を、前記リザベーションステーション及び前記レジスタファイル及び前記リオーダーバッファに供給するためのコモンデータバスとをさらに備えた請求項 1 に記載の命令発行装置。

【請求項 7】

前記リザベーションステーション及び / 又は前記スレッド制御ユニットは、スレッド ID に対応して優先度情報を記憶したメモリを含む請求項 1 に記載の命令発行装置。

10

【請求項 8】

前記請求項 1 乃至 7 のいずれかに記載の命令発行装置と、命令及びデータをそれぞれキャッシュする命令キャッシュ及びデータキャッシュと、前記命令キャッシュから命令をフェッチ及びデコードし、デコードされた命令を基に前記レジスタファイルにアクセスする命令フェッチユニットと、外部の記憶装置及び / 又は入出力装置とデータの入出力を行うためのインターフェースユニットと、を備えた中央演算装置。

20

【請求項 9】

スレッド識別子 (スレッド ID) に対応して、優先度情報と、命令と、該命令の実行に用いる各オペランドを示すデータとを含むエントリを記憶し、実行すべきスレッドのオペランドが記憶されたレジスタファイルからデータを受け取り、命令及びオペランドを発行するリザベーションステーションと；前記リザベーションステーションから発行された命令及びオペランドを受け取り、該命令に従い各種演算を行い、演算結果を前記レジスタファイルに書き戻す演算器と；を備え、マルチスレッドプロセッサにおける命令発行装置を用いた命令発行方法であって、

前記リザベーションステーションは、前記演算器からの出力を監視し、前記レジスタファイルで未解決のオペランドである必要なデータが前記演算器から演算結果として出力された場合、その演算結果をオペランドとして取り込むステップと、

30

前記リザベーションステーションは、スレッド ID に対応した優先度情報を、各スレッドに対応して優先度を設定するためのスレッド制御ユニットから求めるステップと、

前記リザベーションステーションは、命令実行に必要な全てのオペランドがそろった命令であって、且つ、優先度情報に従い優先度の高いスレッドの命令を前記演算器へ送るステップと

を含むことにより、前記リザベーションステーションから前記演算器への命令発行を高い優先度のスレッドの命令が低い優先度の命令を追い越すようにした前記命令発行方法。

【請求項 10】

前記リザベーションステーションは、デコードされた命令を基に前記レジスタファイルから読み出されたデータを格納し、各オペランドに対してデータが有効であることを示す有効ビットをさらに含む、エントリを作成するステップと、

40

前記リザベーションステーションは、該リザベーションステーションに記憶された各エントリに対して、命令実行に必要なオペランドが全てそろうまで前記演算器の演算結果をオペランドとして取り込み、そのデータに対応する有効ビットをセットし、各エントリに対するスレッド ID に従い前記スレッド制御ユニットを参照して優先度を得てセットすることにより、各エントリを更新するステップと、

前記リザベーションステーションは、全ての有効ビットがセットされているエントリをオペランドがそろった演算可能なエントリと判断するステップと、

前記リザベーションステーションは、オペランドがそろったエントリが複数の演算可能で

50

ある場合、各エントリの優先度を比較して、より高い優先度を持つエントリを選択して、前記演算器に該エントリの命令及びオペランド送るステップとを含む請求項 9 に記載の命令発行方法。

【請求項 1 1】

前記リザベーションステーションは、命令がエントリに入力又は作成されたときにカウンタをリセットし、以後エントリを更新する処理クロック毎に又は命令を発行する処理クロック毎に該カウンタをカウントアップするステップと、

前記リザベーションステーションは、優先度の高いエントリが複数ある場合に、カウンタに従いその値の大きいエントリを選択して、その命令及びオペランドを前記演算器に発行するステップと、

をさらに含むことを特徴とする請求項 9 に記載の命令発行方法。

【請求項 1 2】

スレッド識別子（スレッド ID）に対応して、優先度情報と、命令と、該命令の実行に用いる各オペランドを示すデータとを含むエントリを記憶し、実行すべきスレッドの命令及びオペランドが記憶されたレジスタファイルからデータを受け取り、命令及びオペランドを発行するリザベーションステーションと；前記リザベーションステーションから発行された命令及びオペランドを受け取り、該命令に従い各種演算を行い、演算結果を前記レジスタファイルに書き戻す演算器と；を備え、マルチスレッドプロセッサにおける命令発行装置を用い、コンピュータに次の各ステップを実行させるための命令発行プログラムであって、

前記リザベーションステーションは、前記演算器からの出力を監視し、前記レジスタファイルで未解決のオペランドである必要なデータが前記演算器から演算結果として出力された場合、その演算結果をオペランドとして取り込むステップと、

前記リザベーションステーションは、スレッド ID に対応した優先度情報を、各スレッドに対応して優先度を設定するためのスレッド制御ユニットから求めるステップと、

前記リザベーションステーションは、命令実行に必要な全てのオペランドがそろった命令であって、且つ、優先度情報に従い優先度の高いスレッドの命令を前記演算器へ送るステップと

を、コンピュータに実行させるための命令発行プログラム。

【請求項 1 3】

スレッド識別子（スレッド ID）に対応して、優先度情報と、命令と、該命令の実行に用いる各オペランドを示すデータとを含むエントリを記憶し、実行すべきスレッドのオペランドが記憶されたレジスタファイルからデータを受け取り、命令及びオペランドを発行するリザベーションステーションと；前記リザベーションステーションから発行された命令及びオペランドを受け取り、該命令に従い各種演算を行い、演算結果を前記レジスタファイルに書き戻す演算器と；を備え、マルチスレッドプロセッサにおける命令発行装置を用い、コンピュータに次の各ステップを実行させるための命令発行プログラムを記憶したコンピュータ読み取り可能な記憶媒体であって、

前記リザベーションステーションは、前記演算器からの出力を監視し、前記レジスタファイルで未解決のオペランドである必要なデータが前記演算器から演算結果として出力された場合、その演算結果をオペランドとして取り込むステップと、

前記リザベーションステーションは、スレッド ID に対応した優先度情報を、各スレッドに対応して優先度を設定するためのスレッド制御ユニットから求めるステップと、

前記リザベーションステーションは、命令実行に必要な全てのオペランドがそろった命令であって、且つ、優先度情報に従い優先度の高いスレッドの命令を前記演算器へ送るステップと

を、コンピュータに実行させるための命令発行プログラムを記憶したコンピュータ読み取り可能な記憶媒体。

【発明の詳細な説明】

【0001】

10

20

30

40

50

【発明の属する技術分野】

本発明は、命令発行方法及び装置、中央演算装置、命令発行プログラム及びそれを記憶したコンピュータ読み取り可能な記憶媒体に係り、特に、各種ロボット、自動車、プラント、ホームオートメーション等の種々の分散実時間制御を実現するために必要な実時間処理をハードウェアで支援する命令発行方法及び装置、中央演算装置、命令発行プログラム及びそれを記憶したコンピュータ読み取り可能な記憶媒体に関する。

【0002】**【従来の技術】**

従来、マルチスレッドプロセッサを用いた技術として、特許文献1には、複数の命令フローを独立に処理し、命令フロー単位で処理性能を柔軟に制御するものが記載されている。このマルチスレッドプロセッサでは、命令フロー中の命令によって命令フロー毎の優先順位を制御し、一つの機能ユニットに対して複数同時に命令発行要求が出力された場合、機能ユニットに発行すべき命令を優先度によって決定することで命令フロー単位に必要なとされる処理性能を動的に実現する。なお、スレッドとは、一般に、OSがあるプロセス又はタスクを並列処理するため、プロセス又はタスクを分割した処理単位又は最小単位のことをいう。この際、プロセス又はタスクが分割されることなく、1プロセス又は1タスクが1スレッドとなる場合もある。

10

【特許文献1】

特開平10-124316号公報

【0003】**【発明が解決しようとする課題】**

従来のマルチスレッドプロセッサのリザーベーションステーションでは、命令がどのスレッドのものかということに関係なく処理が行われる。そのため、複数のスレッドが同時に実行されていると、あるスレッドの実行時間が他のスレッドの実行に影響を受け、実行時間を予測することが難しくなる。そのためリアルタイム処理のように時間制約に厳しいシステムにおいては、この点が課題となる。

20

【0004】

本発明は、以上の点に鑑み、マルチスレッドプロセッサにおいて実時間処理を、

(1) 命令の優先度による追い越し、及び/又は、

(2) クロック毎の優先度の付け替え、

という手法を用いて実現することにより、他のスレッドに影響されることなく、スレッドの実時間処理の実行を可能とすることを目的とする。

30

【0005】**【課題を解決するための手段】**

本発明では、特に、

1. 実行する命令に優先度を付け、リザーベーションステーションにおいて高い優先度の命令が低い優先度の命令を追い越すこと、及び/又は、

2. 命令の優先度をクロック毎に付け替えること、

により、他のスレッドの影響を受けずにスレッドがリアルタイム処理を行うことを実現する。

40

【0006】

本発明の第1の解決手段によると、

中央演算装置が複数のスレッドの命令を実行するマルチスレッドプロセッサにおける命令発行装置であって、

各スレッドに対応して優先度を設定するためのスレッド制御ユニットと、

演算処理及び/又はメモリアクセス処理において実行すべきスレッドの命令及びオペランドが記憶されたレジスタファイルと、

スレッド識別子(スレッドID)に対応して、優先度情報と、命令と、該命令の実行に用いる各オペランドを示すデータとを含むエントリを記憶し、前記レジスタファイルからデータを受け取り、命令及びオペランドを発行するリザーベーションステーションと、

50

前記リザベーションステーションから発行された命令及びオペランドを受け取り、該命令に従い各種演算を行い、演算結果を前記レジスタファイルに書き戻す演算器と、
を備え、

前記リザベーションステーションは、

前記演算器からの出力を監視し、前記レジスタファイルで未解決のオペランドである必要なデータが前記演算器から演算結果として出力された場合、その演算結果をオペランドとして取り込む手段と、

スレッドIDに対応した優先度情報を前記スレッド制御ユニットから求める手段と、

命令実行に必要な全てのオペランドがそろった命令であって、且つ、優先度情報に従い優先度の高いスレッドの命令を前記演算器へ送る手段と

を有することにより、前記リザベーションステーションから前記演算器への命令発行を高い優先度のスレッドの命令が低い優先度の命令を追い越すようにした前記命令発行装置が提供される。

10

【0007】

本発明の第2の解決手段によると、

上述のような命令発行装置と、

命令及びデータをそれぞれキャッシュする命令キャッシュ及びデータキャッシュと、

前記命令キャッシュから命令をフェッチ及びデコードし、デコードされた命令を基に前記レジスタファイルにアクセスする命令フェッチユニットと、

外部の記憶装置及び/又は入出力装置とデータの入出力を行うためのインターフェースユニットと、

を備えた中央演算装置が提供される。

20

【0008】

本発明の第3の解決手段によると、

スレッド識別子(スレッドID)に対応して、優先度情報と、命令と、該命令の実行に用いる各オペランドを示すデータとを含むエントリを記憶し、実行すべきスレッドのオペランドが記憶されたレジスタファイルからデータを受け取り、命令及びオペランドを発行するリザベーションステーションと；前記リザベーションステーションから発行された命令及びオペランドを受け取り、該命令に従い各種演算を行い、演算結果を前記レジスタファイルに書き戻す演算器と；を備え、マルチスレッドプロセッサにおける命令発行装置を用いた命令発行方法であって、

前記リザベーションステーションは、前記演算器からの出力を監視し、前記レジスタファイルで未解決のオペランドである必要なデータが前記演算器から演算結果として出力された場合、その演算結果をオペランドとして取り込むステップと、

前記リザベーションステーションは、スレッドIDに対応した優先度情報を、各スレッドに対応して優先度を設定するためのスレッド制御ユニットから求めるステップと、

前記リザベーションステーションは、命令実行に必要な全てのオペランドがそろった命令であって、且つ、優先度情報に従い優先度の高いスレッドの命令を前記演算器へ送るステップと

を含むことにより、前記リザベーションステーションから前記演算器への命令発行を高い優先度のスレッドの命令が低い優先度の命令を追い越すようにした前記命令発行方法が提供される。さらに、第3の解決手段に記載された各ステップを、コンピュータに実行させるための命令発行プログラム及びその命令発行プログラムを記憶したコンピュータ読み取り可能な記憶媒体が提供される。

30

40

【0009】

【発明の実施の形態】

1. 関連技術

本実施の形態は、一例として、レスポンス・マルチスレッドプロセッサ(Responsive Multi-Threaded(RMT)Processor)等のマルチスレッドプロセッサの中央演算装置(CPU)上で利用されている。そこで、まず、本実施

50

の形態に関連するCPUを有するマルチスレッドプロセッサについて説明する。

【0010】

図1に、本実施の形態に関連するアウトオブオーダー実行を行うマルチスレッドプロセッサの構成図を示す。ここで、アウトオブオーダー実行とは、プログラムで記述されている順番と異なる順番でも命令の演算を行うことをいう。そのために、アウトオブオーダー実行では、以下に説明するように、リオーダーバッファとリザベーションステーションを用いて、実行可能な命令から順に演算器に送り、リオーダーバッファで順番を元に戻すことが行われる。

【0011】

このプロセッサは、CPU101、メモリ200、I/O300、バス400を有する。CPU101は、バス400を介して、メモリ200、I/O300等と接続されている。CPU101は、レジスタ群1、命令フェッチユニット2、命令キャッシュ3、演算器4、メモリアクセスユニット5、データキャッシュ6、バスインターフェースユニット7、リオーダーバッファ8、命令デコードユニット9、リザベーションステーション10、演算バス11、コモンデータバス12を備える。レジスタ群1は、レジスタファイル1-1、リネームレジスタ1-2を有する。

10

【0012】

命令キャッシュ3及びデータキャッシュ6は、例えば、SRAM、フリップフロップ(F F)等の素子が用いられ、アクセス、読み出し、書き込み等の処理速度が速いものの記憶容量が小さい。一方、CPU101外部のメモリ200は、SDRAM、DRAM等の素子が用いられ、アクセス、読み出し、書き込み等の処理速度がキャッシュより遅いものの記憶容量が大きい。

20

【0013】

命令フェッチユニット2は、命令キャッシュ3へアドレス(Address)を出力し、命令キャッシュ3から命令(instruction)をフェッチし、フェッチした命令を命令デコードユニット9に送る。命令デコードユニット9は、命令フェッチユニット2でフェッチされた命令をデコードし、それを基にレジスタ群1のレジスタファイル1-1から演算に必要なデータを読み出す。

【0014】

レジスタ群1のレジスタファイル1-1は、汎用レジスタ(GPR)、浮動小数点レジスタ(FPR)、プログラムカウンタ(PC)、ステータスレジスタ(SR)等の各種レジスタを含む。レジスタファイル1-1は、現在実行中の命令に必要なオペランド又はデータ等を記憶する。マルチスレッドプロセッサの場合は、レジスタファイル1-1が並列に複数存在することになる。レジスタ群1のリネームレジスタ1-2は、オペランドの依存関係を解決するためにオペランド名を変更する。リネームレジスタ1-2は、必要なオペランドがまだ演算中で結果が出ていない場合、データの代わりにリネームされたレジスタIDを出力する。レジスタファイル1-1又はリネームレジスタ1-2から出力された、オペランド又はデータは、デコードされた命令と共に演算バス11を介してリザベーションステーション10へ送られる。

30

【0015】

演算バス11は、レジスタ群1のレジスタファイル1-1又はリネームレジスタ1-2、リオーダーバッファ8とリザベーションステーション10を並列に接続する。

40

【0016】

リザベーションステーション10は、レジスタファイル1-1及び/又はリネームレジスタ1-2から、デコードされた命令に従ってレジスタ群1から読み出された、演算に必要なオペランド又はデータを格納する。また、リザベーションステーション10は、各演算器4及びメモリアクセスユニット5からの出力をコモンデータバス12を介して監視し、レジスタファイル1-1で未解決のオペランドについては必要なデータが演算器4から出力されると、その結果をオペランドとして取り込む。リザベーションステーション10は、全てのオペランドがそろった命令から順に接続されている演算器4又はメモリアクセス

50

ユニット5へ送る。なお、ロード(Load)又はストア(Store)等のメモリアクセス命令の場合、メモリアクセスユニット5に接続されたリザベーションステーション10にその命令及びオペランドが格納される。

【0017】

演算器4は、局所性原理に従い、命令キャッシュ3、データキャッシュ6又はメモリ200から必要なデータ及び命令をリザベーションステーション10又はレジスタ群1を介して利用する。演算器4は、リザベーションステーション10から受け取った命令に従い演算を実行し、計算した結果をコモンデータバス12に出力する。

【0018】

また、メモリアクセスユニット5は、データキャッシュ6にアクセスしロード又はストアを実行し、コモンデータバス12に出力する。ストア命令の場合、メモリアクセスユニット5は、アドレスとデータをデータキャッシュ6に送り、データをデータキャッシュ6に格納する。ロード命令の場合、メモリアクセスユニット5は、アドレスをデータキャッシュ6に送りデータキャッシュ6からデータを読み出す。読み出したデータは、コモンデータバス12を介してレジスタファイル1-1に書き戻される。このとき、データキャッシュ6に求めるデータがなければ、バスインターフェースユニット7を介してメモリ200からそれを読み出す。バスインターフェースユニット7は、命令キャッシュ3、データキャッシュ6とCPU外部のメモリ200、I/O300等をバス400を介して接続し、CPU内部と外部の間でデータの入出力を行うユニットである。

【0019】

コモンデータバス12は、演算器4の演算結果、メモリアクセスユニット5のロード又はストアの実行結果をレジスタファイル1-1又はリネームレジスタ1-2、リオーダバッファ8、リザベーションステーション10に受け渡す。

【0020】

リオーダバッファ8は、各演算器4によりアウトオブオーダで実行された命令の順番を元の順番にもどしてから、演算結果を実際のレジスタファイル1-1へ書き込む。

【0021】

図2に、リザベーションステーション10のエントリのフォーマットを示す。リザベーションステーション10のエントリは、ビジービット(busy bit)、命令(operation)、ひとつ又は複数の有効ビット及びデータの組、(valid0及びdata0、valid1及びdata1、・・・)を含む。

【0022】

「ビジービット」は、エントリが有効であるか無効であるか、即ちエントリに命令があるかないかを示し、「命令」は、演算器4で演算するための命令又はメモリアクセスユニット5で実行するための命令を示す。また、「データ」は、命令実行に必要なオペランドを示し、「有効ビット」は、演算に用いる対応するオペランドに対して値が有効であるか否かを示す。リザベーションステーション10は、命令に応じて有効ビットとデータの組の数を設定することができる。リザベーションステーション10は、有効ビットを用いて、全てのオペランドがそろったことを判断することができる。

【0023】

図3及び図4に、リザベーションステーション10における演算器4への命令発行処理のフローチャート(1)及び(2)を示す。図3は、ステップS101からステップS109までのステップを、図4は、ステップS111からステップS117までのステップを示す。命令発行処理は、ステップS101からステップS109までのステップを処理した後、ステップS111からステップS117までのステップを処理する。

【0024】

リザベーションステーション10は、レジスタファイル1-1から読み出されたデータを受け取り、その命令実行に必要なデータを記憶するためのエントリを作成して記憶している。リザベーションステーション10は、このようなエントリをひとつ又は複数内部に記憶する。

10

20

30

40

50

【0025】

命令発行処理が開始されると、リザベーションステーション10の全てのエントリに対して、ステップS101からステップS109の繰り返しループ処理を行う。繰り返しループ処理では、まず、リザベーションステーション10は、各エントリについて、そのエントリが示す命令を実行するために必要なオペランドが全てそろっているか否か判断する(S103)。リザベーションステーション10は、必要なオペランドの数を命令に従って判別してもよいし、予め命令毎に必要な数又は領域をエントリに定めてもよい。リザベーションステーション10は、例えば、各エントリの有効ビットが全て有効であるか否かを判断する。ステップS103で、命令の実行に必要な全てのデータがオペランドとして得られた場合、例えば、該当する命令実行に必要な全てのオペランドに対する全ての有効ビットがセットされている場合、ステップS109に移る。一方、必要なオペランドがそろっていない場合、例えば、全ての有効ビットがセットされていない場合(S103)、リザベーションステーション10は、必要なデータをまだ得ていないことになり、ステップS105に移る。この場合、リザベーションステーション10は、コモンデータバス12から流れてくる演算結果を監視し、必要なデータが来れば(S105)、それをエントリのデータにオペランドとして取り込み、該当する有効ビットをセットする(S107)。一方、必要なデータが来なければ(S105)、ステップS109に移る。

10

【0026】

リザベーションステーション10の全てのエントリに対して、ステップS101からステップS109の処理を繰り返し行った後、ステップS111に進んで処理を継続する。

20

【0027】

つぎに、図4を用いて、ステップS111からステップS117の処理について説明する。

ステップS111では、リザベーションステーション10は、命令実行に必要な全てのオペランドがそろったエントリがあるか否か判断する(S111)。例えば、上述のように、リザベーションステーション10は全ての有効ビットが有効かどうかでこれを判断することができる。全てのオペランドがそろったエントリが無い場合は、命令発行を終了する。一方、全てのオペランドがそろったエントリがある場合は、リザベーションステーション10は、そのようなエントリがひとつか複数か判断する(S113)。例えば、全ての有効ビットがセットされると、命令実行に必要な全てのオペランドがそろったことになる。

30

【0028】

リザベーションステーション10は、全てのオペランドがそろったエントリが1つの場合、該当エントリの命令及びオペランドを演算器4に送る(S115)。一方、複数のエントリが全てのエントリがそろい演算可能である場合、リザベーションステーション10は、より古いエントリから演算器4に送る(S117)。選択されなかったエントリは次の命令発行の処理クロック以降まで発行を待たされる。

【0029】

2. 優先度による複数スレッドの命令発行装置を備えたCPU

上述の「1. 関連技術」では、リザベーションステーション10の動作は、主に、オペランドがそろい、演算が可能になった命令から演算器4に送り、複数の命令が同時に演算可能になっている場合は最も古い命令から実行することについて説明した。以下では、複数の命令が同時に演算可能になった場合、優先度の最も高いスレッドの命令から先に実行するための命令発行について説明する。

40

【0030】

図5に、優先度を用いたアウトオブオーダー実行を行うマルチスレッドプロセッサの構成図を示す。

このプロセッサは、CPU102、メモリ200、I/O300、バス400を有する。CPU102は、バス400を介して、メモリ200、I/O300等と接続されている。CPU102は、レジスタ群1、命令フェッチユニット2、命令キャッシュ3、演算器

50

4、メモリアクセスユニット5、データキャッシュ6、バスインターフェースユニット7、リオーダバッファ8、命令デコードユニット9、演算バス11、コモンデータバス12、リザベーションステーション20、スレッド制御ユニット21を備える。レジスタ群1は、レジスタファイル1-1、リネームレジスタ1-2を含む。

【0031】

命令フェッチユニット2は、命令キャッシュ3へアドレス(Address)を出力し、命令キャッシュ3から命令(instruction)をフェッチし、命令デコードユニット9で命令をデコードする。

【0032】

スレッド制御ユニット21は、各スレッドの優先度を設定する。また、スレッド制御ユニット21は、設定した優先度を全てのリザベーションステーション20に送る。スレッド制御ユニット21内部には、各スレッドの優先度を識別するためのデータを所定数保持する。スレッド制御ユニット21は、そのために、例えば、各スレッドに対応した優先度の情報を記憶したメモリを備えることができる。また、スレッド制御ユニット21は、他に、スレッドの実行、停止、コンテキストをキャッシュするコンテキストキャッシュとの入れ替え制御等を行うことができる。ここで、コンテキストとは、例えば、汎用レジスタ、浮動小数点レジスタ、プログラムカウンタ、ステータスレジスタ等、記憶部(例、レジスタファイル1-1)に記憶されている各スレッドの実行のための情報又は現在実行中の状態のことをいう。

10

【0033】

リザベーションステーション20は、上述したように、デコードされた命令に従ってレジスタファイル1-1及び/又はリネームレジスタ1-2から読み出された、演算に必要なオペランド又はデータを格納する。

20

【0034】

また、リザベーションステーション20は、各演算器4及びメモリアクセスユニット5からの出力をコモンデータバス12を介して監視し、レジスタファイル1-1で未解決のオペランドについては必要なデータが演算器4から出力されると、その結果をオペランドとして取り込む。さらに、リザベーションステーション20は、スレッド制御ユニット21から設定された優先度情報を受け取る、又は、スレッド制御ユニット21をアクセスし、所定のスレッドの優先度情報を得る。また、リザベーションステーション20は、スレッド毎に優先度情報を記憶したテーブルを内部のレジスタに記憶してもよい。リザベーションステーション20は、全てのオペランドがそろった命令のうち優先度が高い命令から順に接続されている演算器4又はメモリアクセスユニット5へ送る。なお、ロード又はストア等のメモリアクセス命令の場合、メモリアクセスユニット5に接続されたリザベーションステーション20にその命令及びオペランドが送られる。

30

【0035】

他の各部の構成及び動作は、上述したように、図1のCPU101の同符号で示される各部の構成及び動作と同様である。

【0036】

図6に、リザベーションステーション20のエントリのフォーマットを示す。このエントリは、リザベーションステーション10のエントリに、命令の優先度情報(priority)が付加されている。

40

【0037】

リザベーションステーション20のエントリは、ビジービット(busy bit)、スレッドID(thread ID)、優先度情報(priority)、命令(operation)、ひとつ又は複数の有効ビット及びデータの組、(valid0及びdata0、valid1及びdata1、・・・)を含む。

【0038】

ビジービット、命令、有効ビット、データは、図2で説明したものと同様である。「スレッドID」は、エントリの命令がどのスレッドのものかを示す。なお、スレッドIDを省

50

略し、スレッドID毎に各エントリを識別する情報を記憶したテーブルを用いること等により、リザベーションステーション20は、各スレッドIDを適宜把握することもできる。「優先度情報」は、命令の優先度を示し、リザベーションステーション20に命令が入られるとき、オペランドを得たとき、処理クロック毎、及び/又は全てのオペランドがそろったとき等にスレッド制御ユニット21からセットされる。リザベーションステーション20は、スレッド毎に別途指定されたレジスタやスレッド制御ユニット21の設定に基づき、スレッド毎に優先度の値を調べ、優先度情報を変更する。本実施の形態ではスレッド制御ユニット21は、例えば、スレッド毎に8bitのレジスタを用意し、優先度情報を256レベルの値として指定する。この値は、例えば、高い(大きい)ほど優先度が高く、低い(小さい)ほど優先度が低いとすることができる。

10

【0039】

図7及び図8に、優先度を用いた、リザベーションステーション20における演算器4への命令発行処理のフローチャート(1)及び(2)を示す。図7は、ステップS201からステップS209までのステップを、図8は、ステップS211からステップS217までのステップを示す。命令発行処理は、ステップS201からステップS209までのステップを処理した後、ステップS211からステップS217までのステップを処理する。

【0040】

リザベーションステーション20は、レジスタファイル1-1から送られてくるデータを及び命令を格納し、その命令実行に必要なデータを記憶するためのエントリを作成して記憶している。リザベーションステーション20は、このようなエントリをひとつ又は複数内部に記憶する。

20

【0041】

命令発行処理が開始されると、リザベーションステーション20の全てのエントリに対して、ステップS201からステップS209の繰り返しループ処理を行う。繰り返しループ処理では、まず、リザベーションステーション20は、各エントリについて、そのエントリが示す命令を実行するために必要なオペランドが全てそろっているか否か判断する(S203)。リザベーションステーション20は、必要なオペランドの数を命令に従って判別してもよいし、予め命令毎に必要な数又は領域をエントリに定めてもよい。リザベーションステーション20は、例えば、各エントリの有効ビットが全て有効であるか否かを判断する。ステップS203で命令の実行に必要な全てのオペランドがそろっている場合、例えば、該当する命令実行に必要な全てのオペランドに対する全ての有効ビットがセットされている場合、ステップS208に移る。一方、必要なオペランドがそろっていない場合、例えば、全ての有効ビットがセットされていない場合(S203)、リザベーションステーション20は、当該エントリの命令の実行のために必要なデータをまだ得ていないことになり、ステップS205に移る。ここで、リザベーションステーション20は、コモンデータバス12を監視し必要なデータが来れば(S205)、それをエントリにオペランドとして取り込み該当する有効ビットをセットする(S207)。一方、必要なデータが来なければ(S205)、ステップS208に移る。

30

【0042】

つぎに、リザベーションステーション20は、各エントリのスレッドIDから優先度情報を更新する(S208)。例えば、リザベーションステーション20は、各エントリにはどのスレッドのものかを示すフィールド、即ちスレッドIDがあるため、それを参照することによりそのエントリがどのスレッドのものかを判断することができる。また、スレッド制御ユニット21では、メモリ等によりスレッドID毎に優先度情報が指定されているため、リザベーションステーション20は、スレッド制御ユニット21のメモリをアクセスする等により設定された優先度を得て、該当するエントリの優先度情報を更新することができる。逆に、スレッド制御ユニット21が、各リザベーションステーション20のエントリのスレッドIDを得て、そのスレッドIDに該当する優先度情報を与えることで、リザベーションステーション20が優先度情報を得るようにしてもよい。また、リザベ

40

50

ションステーション 20 は、この優先度のデータを適当なタイミングでスレッド制御ユニット 21 から得て、スレッド ID に対応して優先度情報を内部メモリに記憶しておくようにして、それを参照することでスレッド ID に従い優先度情報を得てもよい。

【0043】

リザベーションステーション 20 は、全てのエントリに対して、ステップ S 201 からステップ S 209 の処理を繰り返し行った後、ステップ S 211 に進む。

【0044】

つぎに、図 8 を用いて、ステップ S 211 からステップ S 217 の処理について説明する。

ステップ S 211 では、リザベーションステーション 20 は、命令実行に必要な全てのオペランドがそろったエントリがあるか否か判断する (S 211)。例えば、上述のように、リザベーションステーション 20 は全ての有効ビットが有効かどうかでこれを判断することができる。全てのオペランドがそろったエントリが無い場合は、命令発行を終了する。一方、全てのオペランドがそろったエントリがある場合は、リザベーションステーション 20 は、そのようなエントリがひとつか複数か判断する (S 213)。例えば、上述のように、全ての有効ビットがセットされると、命令実行に必要なオペランドがそろったことになる。

【0045】

リザベーションステーション 20 は、全てのオペランドがそろったエントリが 1 つの場合、該当エントリの命令及びオペランドを演算器 4 に送る (S 215)。一方、複数のエントリが全てのオペランドがそろい演算可能である場合、リザベーションステーション 20 は、命令の実行可能な各エントリの優先度情報を比較して、より高い優先度を持つエントリを選択する (S 217)。なお、ここで、リザベーションステーション 20 は、優先度が等しい又は略等しいエントリが複数あれば、その中でより古いエントリを選択して、又は、予め定められた順序によりエントリを選択して演算器 4 に送る。なお、選択されなかったエントリは次の命令発行の処理クロック以降まで発行を待たされる。

【0046】

このように、リザベーションステーション 20 は、優先度の値が高いエントリを優先的に演算器 4 に送ることができ、これにより優先度の高い命令が先に演算される。なお、ステップ S 208 の処理をステップ S 203 の前に実行してもよい。

【0047】

3. 優先度とカウンタを用いた複数スレッドの命令発行装置及び CPU
この実施の形態では、優先度とカウンタに基づき、先に実行すべき命令を発行するための命令発行について説明する。この実施の形態の命令発行装置及び CPU の構成及び動作は、上述の図 5 及びその説明箇所の通りであり、リザベーションステーション 20 で用いるエントリが異なる。

【0048】

図 9 に、リザベーションステーション 20 のカウンタ (counter) を付加したエントリのフォーマットを示す。

【0049】

このエントリは、ビジービット (busy bit)、スレッド ID (thread ID)、優先度情報 (priority)、命令 (operation)、ひとつ又は複数の有効ビット及びデータの組、(valid0 及び data0、valid1 及び data1、・・・)、カウンタ (counter) を含む。ビジービット、スレッド ID、優先度情報、命令、有効ビット、データは、図 2 又は図 5 で説明したものと同様である。カウンタは、例えば、リザベーションステーション 20 により命令がエントリに入れられた時にリセットされ、命令発行の処理クロック毎にカウントアップされるカウンタ値である。

【0050】

図 10 及び図 11 に、カウンタを用いた、リザベーションステーション 20 における演算

器 4 への命令発行のフローチャート (1) 及び (2) を示す。図 1 0 は、ステップ S 2 0 1 からステップ S 2 0 9 までのステップを、図 1 1 は、ステップ S 2 1 1 からステップ S 3 0 7 までのステップを示す。命令発行処理は、ステップ S 2 0 1 からステップ S 2 0 9 までのステップを処理した後、ステップ S 2 1 1 からステップ S 3 0 7 までのステップを処理する。

【 0 0 5 1 】

図 1 0 及び図 1 1 のフローチャートは、上述した図 7 及び図 8 のフローチャートと同じステップ番号の各ステップにおける処理は、上述した通りである。以下の説明では、主に、図 1 0 及び図 1 1 のステップ S 3 0 1 からステップ S 3 0 7 の処理、図 1 0 及び図 1 1 の処理の順序と図 7 及び図 8 の順序で異なる部分について説明する。

10

【 0 0 5 2 】

リザベーションステーション 2 0 は、上述のように、レジスタファイル 1 - 1 から送られてくるデータ及び命令を格納し、その命令実行に必要なデータを記憶するためのエントリを作成して記憶している。リザベーションステーション 2 0 は、このようなエントリをひとつ又は複数内部に記憶する。

【 0 0 5 3 】

命令発行処理が開始されると、リザベーションステーション 2 0 の全てのエントリに対して、ステップ S 2 0 1 からステップ S 2 0 9 の繰り返しループ処理を行う。繰り返しループ処理では、まず、リザベーションステーション 2 0 は、各エントリについて、エントリが有効か無効か、(例えば、命令が記憶されているか否か) を判断する (S 3 0 1) 。リザベーションステーション 2 0 は、例えば、各エントリのビジービットにより、エントリが有効か無効かを判断する。ステップ S 3 0 1 でエントリが無効の場合、ステップ S 2 0 9 に移る。一方、エントリが有効である場合、リザベーションステーション 2 0 は、そのエントリのカウンタ値を 1 つ増やす (S 3 0 3) 。カウンタは、例えば、命令がエントリに入れられた時にリセットされ、以後命令発行の処理クロック毎にカウントアップされる。

20

【 0 0 5 4 】

つぎに、上述のように、リザベーションステーション 2 0 は、各エントリのスレッド ID から優先度情報を更新する (S 2 0 8) 。そして、上述のように、リザベーションステーション 2 0 は、各エントリが示す命令を実行するために必要なオペランドが全てそろっているか否か判断し (S 2 0 3) 、命令の実行に必要な全てのオペランドがそろっている場合、ステップ S 2 0 9 に移り、ステップ S 2 0 1 から繰り返しループ処理を実行する。一方、必要なオペランドがそろっていない場合 (S 2 0 3) 、リザベーションステーション 2 0 は、コモンデータバス 1 2 を監視して必要なデータが来れば (S 2 0 5) 、それをエントリのデータにオペランドとして取り込み該当する有効ビットをセットする (S 2 0 7) 。一方、必要なデータが来なければ (S 2 0 5) 、ステップ S 2 0 9 に移り、ステップ S 2 0 1 から繰り返しループ処理を実行する。

30

【 0 0 5 5 】

つぎに、図 1 1 を用いて、ステップ S 2 1 1 からステップ S 3 0 7 の処理について説明する。

40

ステップ S 2 1 1 では、上述のように、リザベーションステーション 2 0 は、命令実行に必要な全てのオペランドがそろったエントリがあるか否か判断する (S 2 1 1) 。全てのオペランドがそろったエントリが無い場合は、命令発行を終了する。一方、全てのオペランドがそろったエントリがある場合は、リザベーションステーション 2 0 は、そのようなエントリがひとつか複数か判断する (S 2 1 3) 。

【 0 0 5 6 】

リザベーションステーション 2 0 は、全てのオペランドがそろったエントリが 1 つの場合、該当エントリの命令及びオペランドを演算器 4 に送る (S 2 1 5) 。一方、複数のエントリが全てのオペランドがそろい演算可能である場合、リザベーションステーション 2 0 は、命令の実行可能な各エントリの優先度情報を比較して、より高い優先度を持つエント

50

リを選択する (S 3 0 5)。さらに、リザベーションステーション 2 0 は、最も高い優先度のエントリが複数ある場合は、その中で最もカウンタ値の大きいエントリの命令を演算器 4 に送る (S 3 0 5)。なお、複数のエントリが演算可能だった場合、リザベーションステーション 2 0 は、例えば、上位 b i t に優先度情報、下位 b i t にカウンタの連結値を用いて比較し、より数値の大きい命令を選択するようにしてもよい。選択されなかったエントリは、次の命令発行の処理クロック以降まで発行を待たされる。ステップ S 2 1 5 又は S 3 0 5 の処理を終了すると、リザベーションステーション 2 0 は、演算器 4 に命令を送ったエントリのカウンタ値をクリアし、ビジービットを変更してエントリを無効とする (S 3 0 7)。なお、ステップ S 2 0 8 の処理をステップ S 2 0 7 の後に実行してもよい。

10

これにより優先度の高い命令が選択され、優先度が等しい場合はより古い命令が選択される。

【 0 0 5 7 】

4. その他

本発明の命令発行方法又は命令発行装置・システムは、その各手順をコンピュータに実行させるための命令発行プログラム、命令発行プログラムを記録したコンピュータ読み取り可能な記録媒体、命令発行プログラムを含みコンピュータの内部メモリにロード可能なプログラム製品、そのプログラムを含むサーバ等のコンピュータ、等により提供されることもできる。

【 0 0 5 8 】

20

【発明の効果】

本発明によれば、マルチスレッドプロセッサにおいて実時間処理を、

(1) 命令の優先度による追い越し、及び / 又は、

(2) クロック毎の優先度の付け替え、

という手法を用いて実現することにより、他のスレッドに影響されることなく、スレッドの実時間処理の実行を可能とすることができる。

【図面の簡単な説明】

【図 1】本実施の形態に関連するアウトオブオーダー実行を行うマルチスレッドプロセッサの構成図。

【図 2】リザベーションステーション 1 0 のエントリのフォーマット。

30

【図 3】リザベーションステーション 1 0 における演算器 4 への命令発行のフローチャート (1)。

【図 4】リザベーションステーション 1 0 における演算器 4 への命令発行のフローチャート (2)。

【図 5】優先度を用いたアウトオブオーダー実行を行うマルチスレッドプロセッサの構成図。

【図 6】リザベーションステーション 2 0 のエントリのフォーマット。

【図 7】優先度を用いた、リザベーションステーション 2 0 における演算器 4 への命令発行のフローチャート (1)。

【図 8】優先度を用いた、リザベーションステーション 2 0 における演算器 4 への命令発行のフローチャート (2)。

40

【図 9】リザベーションステーション 2 0 のカウンタ (c o u n t e r) を付加したエントリのフォーマット。

【図 1 0】カウンタを用いた、リザベーションステーション 2 0 における演算器 4 への命令発行のフローチャート (1)。

【図 1 1】カウンタを用いた、リザベーションステーション 2 0 における演算器 4 への命令発行のフローチャート (2)。

【符号の説明】

1 レジスタ群

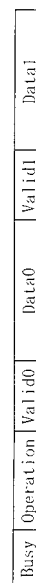
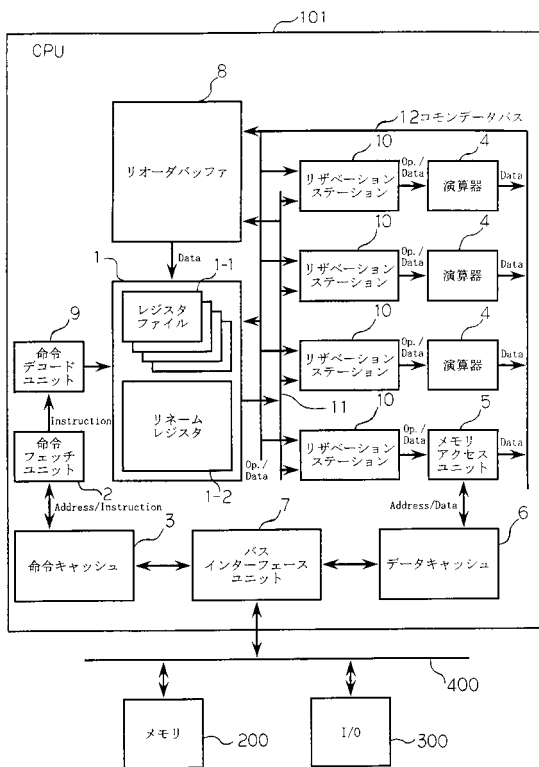
1 - 1 レジスタファイル

50

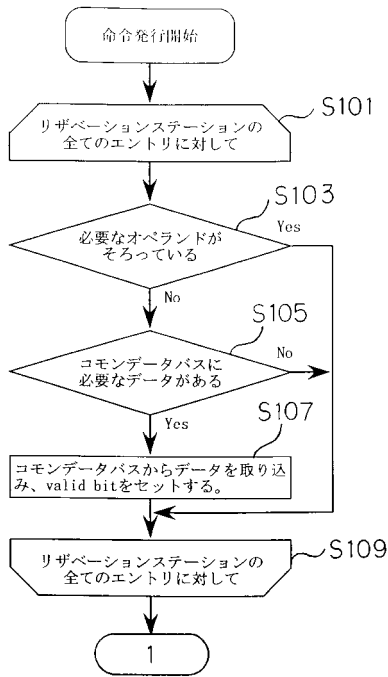
- 1 - 2 リネームレジスタ
- 2 命令フェッチユニット
- 3 命令キャッシュ
- 4 演算器
- 5 メモリアクセスユニット
- 6 データキャッシュ
- 7 バスインターフェースユニット
- 8 リオーダーバッファ
- 9 命令デコードユニット
- 10 リザベーションステーション
- 11 演算バス
- 12 コモンデータバス
- 20 リザベーションステーション
- 21 スレッド制御ユニット
- 101、102 CPU
- 200 メモリ
- 300 I/O
- 400 バス

【図1】

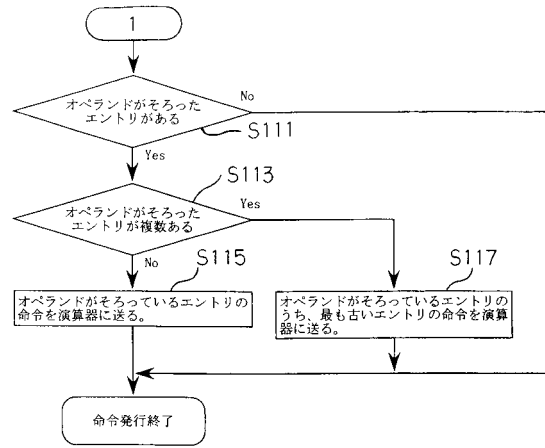
【図2】



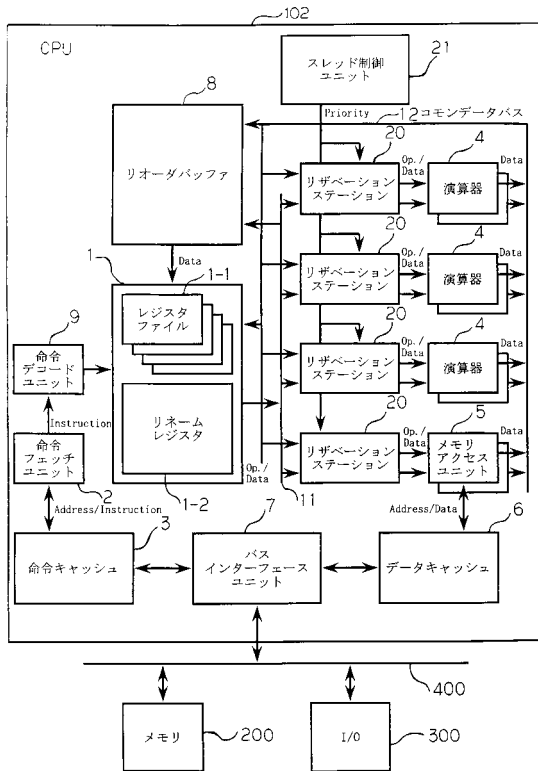
【 図 3 】



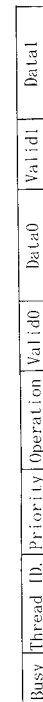
【 図 4 】



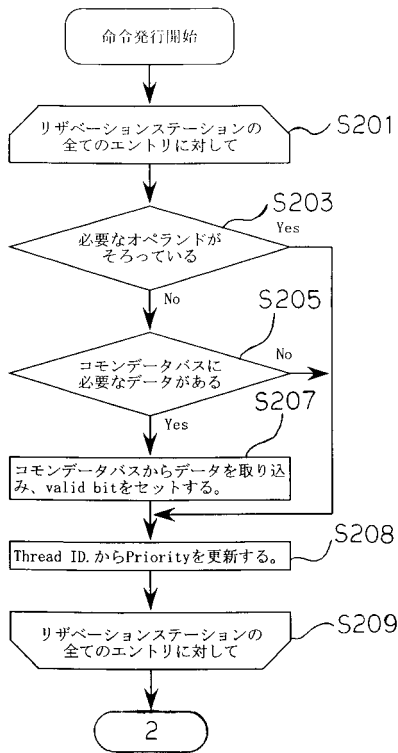
【 図 5 】



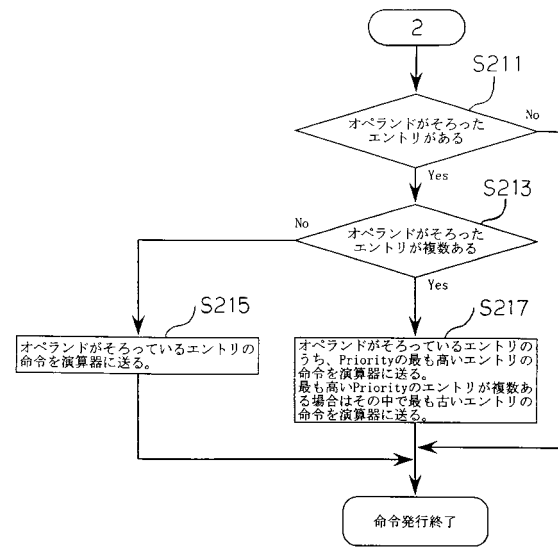
【 図 6 】



【 図 7 】



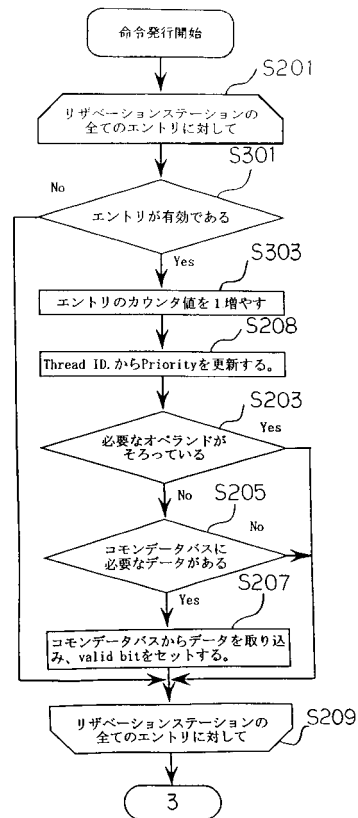
【 図 8 】



【 図 9 】



【 図 10 】



【 図 1 1 】

