

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第4956891号
(P4956891)

(45) 発行日 平成24年6月20日 (2012.6.20)

(24) 登録日 平成24年3月30日 (2012.3.30)

(51) Int.Cl.		F I			
G06F	9/46	(2006.01)	G06F	9/46	410
G06F	9/48	(2006.01)	G06F	9/46	455B
G06F	9/38	(2006.01)	G06F	9/38	350B

請求項の数 12 (全 21 頁)

(21) 出願番号	特願2004-217046 (P2004-217046)	(73) 特許権者	000005223
(22) 出願日	平成16年7月26日 (2004.7.26)		富士通株式会社
(65) 公開番号	特開2006-39815 (P2006-39815A)		神奈川県川崎市中原区上小田中4丁目1番1号
(43) 公開日	平成18年2月9日 (2006.2.9)	(74) 代理人	100092978
審査請求日	平成18年12月18日 (2006.12.18)		弁理士 真田 有
前置審査		(72) 発明者	吉田 利雄
			神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内
		審査官	井上 宏一

最終頁に続く

(54) 【発明の名称】 演算処理装置、情報処理装置および演算処理装置の制御方法

(57) 【特許請求の範囲】

【請求項1】

第1のスレッドに対する第1の処理又は第2のスレッドに対する第2の処理を実行する1以上の処理部と、

前記第1の処理に用いる第1のデータを格納する第1の記憶部と、

前記第2の処理に用いる第2のデータを格納する第2の記憶部と、

前記処理部と前記第1及び第2の記憶部に接続され、前記第1の記憶部からの第1のデータと、前記第2の記憶部からの第2のデータを保持する第1の保持部と、

前記第1のスレッド及び前記第2のスレッドのうち、実行状態のスレッドに関する情報を保持する第2の保持部と、

前記第1のスレッド及び前記第2のスレッドのうち、休止状態のスレッドに関する情報を保持する第3の保持部と、

前記第1の処理を実行する場合に、前記第1の保持部が保持する第1のデータを、前記第1の保持部から処理部に転送するとともに、前記第1のデータが転送された処理部が出力する前記第1の処理の結果である第1の結果データを、前記第1の保持部と前記第1のスレッドに対応する前記第1の記憶部に保持し、前記第1のスレッドから前記第2のスレッドに切替えることにより前記第2の処理を実行する場合に、スレッド切替え前の前記第3の保持部が保持する休止状態の前記第2のスレッドに関する情報を、前記第2の保持部に転送するとともに、スレッド切替え前の前記第2の保持部が保持する実行状態の前記第1のスレッドに関する情報を、前記第3の保持部に転送することにより前記第2のスレ

ドを休止状態から実行状態に切り替えて、前記第2のスレッドに対応する前記第2の記憶部から転送した第2のデータを前記第1の保持部に保持させ、前記第1の保持部が保持した第2のデータを、前記第1の保持部から処理部に転送するとともに、前記第2のデータが転送された処理部が出力する前記第2の処理の結果である第2の結果データを、前記第1の保持部と前記第2の記憶部に保持する制御部を有することを特徴とする演算処理装置。

【請求項2】

前記演算処理装置はさらに、

前記第1の処理に対応する第1の命令と、前記第2の処理に対応する第2の命令を解読するデコード部を有し、

10

前記制御部が、前記第1の処理の実行後に前記第1のスレッドから前記第2のスレッドに切替えることにより前記第2の処理を行なう場合に、前記第2のスレッドに対応する第2の記憶部から前記第1の保持部への第2のデータの転送が完了し、前記第1の保持部から前記第2のデータの読み出しが可能になるまで、前記デコード部による前記第2の命令の解読を停止させることを特徴とする請求項1記載の演算処理装置。

【請求項3】

前記演算処理装置はさらにキャッシュメモリ装置を有するとともに、主記憶装置に接続され、

前記制御部は、前記キャッシュメモリ装置に対するメモリアクセスがキャッシュミスが発生した場合に、前記第1のスレッドから前記第2のスレッドに切替えることを特徴とする請求項1又は2記載の演算処理装置。

20

【請求項4】

前記演算処理装置はさらに、

前記第3の保持部の複数の格納位置と、休止状態のスレッドとを対応付けるレジスタ識別情報を保持するレジスタ識別テーブルを有し、

前記制御部は、前記レジスタ識別情報に基づいて、前記第1のスレッドから前記第2のスレッドに切替えることにより前記第2の処理を実行する場合に、スレッド切替え前の前記第3の保持部が保持する休止状態の前記第2のスレッドに関する情報を、前記第2の保持部に転送するとともに、スレッド切替え前の前記第2の保持部が保持する実行状態の前記第1のスレッドに関する情報を、前記第3の保持部に転送することにより前記第2のスレッドを休止状態から実行状態に切り替えることを特徴とする請求項1～3のいずれか1項記載の演算処理装置。

30

【請求項5】

演算処理装置と前記演算処理装置に接続された記憶装置を有する情報処理装置において、

前記演算処理装置は、

第1のスレッドに対する第1の処理又は第2のスレッドに対する第2の処理を実行する1以上の処理部と、

前記第1の処理に用いる第1のデータを格納する第1の記憶部と、

前記第2の処理に用いる第2のデータを格納する第2の記憶部と、

40

前記処理部と前記第1及び第2の記憶部に接続され、前記第1の記憶部からの第1のデータと、前記第2の記憶部からの第2のデータを保持する第1の保持部と、

前記第1のスレッド及び前記第2のスレッドのうち、実行状態のスレッドに関する情報を保持する第2の保持部と、

前記第1のスレッド及び前記第2のスレッドのうち、休止状態のスレッドに関する情報を保持する第3の保持部と、

前記第1の処理を実行する場合に、前記第1の保持部が保持する第1のデータを、前記第1の保持部から処理部に転送するとともに、前記第1のデータが転送された処理部が出力する前記第1の処理の結果である第1の結果データを、前記第1の保持部と前記第1のスレッドに対応する前記第1の記憶部に保持し、前記第1のスレッドから前記第2のスレ

50

ッドに切替えることにより前記第2の処理を実行する場合に、スレッド切替え前の前記第3の保持部が保持する休止状態の前記第2のスレッドに関する情報を、前記第2の保持部に転送するとともに、スレッド切替え前の前記第2の保持部が保持する実行状態の前記第1のスレッドに関する情報を、前記第3の保持部に転送することにより前記第2のスレッドを休止状態から実行状態に切り替えて、前記第2のスレッドに対応する前記第2の記憶部から転送した第2のデータを前記第1の保持部に保持させ、前記第1の保持部が保持した第2のデータを、前記第1の保持部から処理部に転送するとともに、前記第2のデータが転送された処理部が出力する前記第2の処理の結果である第2の結果データを、前記第1の保持部と前記第2の記憶部に保持する制御部を有することを特徴とする情報処理装置。

10

【請求項6】

前記演算処理装置はさらに、

前記第1の処理に対応する第1の命令と、前記第2の処理に対応する第2の命令を解読するデコード部を有し、

前記制御部が、前記第1の処理の実行後に前記第1のスレッドから前記第2のスレッドに切替えることにより前記第2の処理を行なう場合に、前記第2のスレッドに対応する第2の記憶部から前記第1の保持部への第2のデータの転送が完了し、前記第1の保持部から前記第2のデータの読み出しが可能になるまで、前記デコード部による前記第2の命令の解読を停止させることを特徴とする請求項5記載の情報処理装置。

【請求項7】

20

前記演算処理装置はさらにキャッシュメモリ装置を有するとともに、主記憶装置に接続され、

前記制御部は、前記キャッシュメモリ装置に対するメモリアクセスがキャッシュミスが発生した場合に、前記第1のスレッドから前記第2のスレッドに切替えることを特徴とする請求項5又は6記載の情報処理装置。

【請求項8】

前記演算処理装置はさらに、

前記第3の保持部の複数の格納位置と、休止状態のスレッドとを対応付けるレジスタ識別情報を保持するレジスタ識別テーブルを有し、

前記制御部は、前記レジスタ識別情報に基づいて、前記第1のスレッドから前記第2のスレッドに切替えることにより前記第2の処理を実行する場合に、スレッド切替え前の前記第3の保持部が保持する休止状態の前記第2のスレッドに関する情報を、前記第2の保持部に転送するとともに、スレッド切替え前の前記第2の保持部が保持する実行状態の前記第1のスレッドに関する情報を、前記第3の保持部に転送することにより前記第2のスレッドを休止状態から実行状態に切り替えることを特徴とする請求項5～7のいずれか1項記載の情報処理装置。

30

【請求項9】

第1のスレッドに対する第1の処理又は第2のスレッドに対する第2の処理を実行する1以上の処理部と、前記第1の処理に用いる第1のデータを格納する第1の記憶部と、前記第2の処理に用いる第2のデータを格納する第2の記憶部と、前記第1のスレッド及び前記第2のスレッドのうち、実行状態のスレッドに関する情報を保持する第2の保持部と、前記第1のスレッド及び前記第2のスレッドのうち、休止状態のスレッドに関する情報を保持する第3の保持部とを有する演算処理装置の制御方法において、

40

前記演算処理装置が有する制御部が、前記第1の処理を実行する場合に、前記演算処理装置が有する第1の保持部が保持する第1のデータを、前記第1の保持部から処理部に転送し、

前記制御部が、前記第1のデータが転送された処理部が出力する前記第1の処理の結果である第1の結果データを、前記第1のスレッドに対応する前記第1の記憶部と前記第1の保持部に保持し、

前記制御部が、前記第1のスレッドから前記第2のスレッドに切替えることにより前記

50

第2の処理を実行する場合に、スレッド切替え前の前記第3の保持部が保持する休止状態の前記第2のスレッドに関する情報を、前記第2の保持部に転送し、

前記制御部が、スレッド切替え前の前記第2の保持部が保持する実行状態の前記第1のスレッドに関する情報を、前記第3の保持部に転送することにより前記第2のスレッドを休止状態から実行状態に切り替え、

前記制御部が、前記第2のスレッドに対応する前記第2の記憶部から転送した第2のデータを前記第1の保持部に保持させ、

前記第1の保持部が保持した第2のデータを、前記第1の保持部から処理部に転送し、前記制御部が、前記第2のデータが転送された処理部が出力する前記第2の処理の結果である第2の結果データを、前記第1の保持部と前記第2の記憶部に保持することを特徴とする演算処理装置の制御方法。

10

【請求項10】

前記演算処理装置の制御方法において、前記演算処理装置はさらに、前記第1の処理に対応する第1の命令と、前記第2の処理に対応する第2の命令を解読するデコード部を有し、

前記制御部が、前記第1の処理の実行後に前記第1のスレッドから前記第2のスレッドに切替えることにより前記第2の処理を行なう場合に、前記第2のスレッドに対応する第2の記憶部から前記第1の保持部への第2のデータの転送が完了し、前記第1の保持部から前記第2のデータの読み出しが可能になるまで、前記デコード部による前記第2の命令の解読を停止させることを特徴とする請求項9記載の演算処理装置の制御方法。

20

【請求項11】

前記演算処理装置の制御方法において、前記演算処理装置はさらにキャッシュメモリ装置を有するとともに、主記憶装置に接続され、

前記制御部は、前記キャッシュメモリ装置に対するメモリアクセスがキャッシュミスが発生した場合に、前記第1のスレッドから前記第2のスレッドに切替えることを特徴とする請求項9又は10記載の演算処理装置の制御方法。

【請求項12】

前記演算処理装置の制御方法において、前記演算処理装置はさらに、前記第3の保持部の複数の格納位置と、休止状態のスレッドとを対応付けるレジスタ識別情報を保持するレジスタ識別テーブルを有し、

30

前記制御部は、前記レジスタ識別情報に基づいて、前記第1のスレッドから前記第2のスレッドに切替えることにより前記第2の処理を実行する場合に、スレッド切替え前の前記第3の保持部が保持する休止状態の前記第2のスレッドに関する情報を、前記第2の保持部に転送するとともに、スレッド切替え前の前記第2の保持部が保持する実行状態の前記第1のスレッドに関する情報を、前記第3の保持部に転送することにより前記第2のスレッドを休止状態から実行状態に切り替えることを特徴とする請求項9～11のいずれか1項記載の演算処理装置の制御方法。

40

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、マルチスレッドプロセッサにおけるスレッド切り替えにかかるレジスタ制御に関する。

【背景技術】

【0002】

近年、コンピュータアーキテクチャの代表的なものとしては、複雑な処理を一命令で行なうCISC(Complex Instruction Set Computer)アーキテクチャの他、一命令中に実行する処理を簡素化したRISC(Reduced Instruction Set Computer)アーキテクチャ

50

、複数の同時処理可能な命令をソフトウェアで1つの長い命令にまとめたV L I W (Very Long Instruction Word) アーキテクチャ等が知られている。

【0003】

また、これらのアーキテクチャを実現するコンピュータの中央演算器装置 (CPU; Central Processing Unit) における処理手法は、インオーダー実行型とアウトオブオーダー実行型との2つに大別することができる。

図8はインオーダー実行型の処理方法を説明するための図、図9はアウトオブオーダー実行型の処理方法を説明するための図である。インオーダー実行型は、図8に示すように、プログラム順に沿って命令処理を行なう手法であり、アウトオブオーダー実行型は、図9に示すように、命令間の依存関係を見て、依存関係のない命令ならばプログラム順を追い越してでも処理を行なう手法である。

10

【0004】

そして、近年においては、一のプロセッサにおいて一のプログラム (スレッド) を実行するシングルスレッド処理の他に、物理的に一のプロセッサにおいて複数のスレッドを並列的に実行するマルチスレッドプロセッサ方式が注目されている。

図10(a), (b)はマルチスレッドプロセッサ方式を説明するための図であり、図10(a)はシングルスレッド処理を説明するための図、図10(b)はマルチスレッド処理を説明するための図である。なお、図10(b)においては、2つのCPUでA, Bの2つのプログラムを並列的に処理するマルチスレッド処理の例を示している。

20

【0005】

一般に、CPUは、ソフトウェアから可視なレジスタや状態レジスタ (CPU状態レジスタ) に加え、加減算や乗算、除算、またメモリデータをレジスタに読み出すロード処理や、レジスタのデータをメモリに書き込むストア処理を実行する資源を有する。マルチスレッドプロセッサとは、1つのCPU内にソフトウェアから可視なレジスタを多重化し、別々なプログラムを実行するようにしながら、加減算などの命令実行資源を複数のプログラムで共有するものである (例えば、特許文献1参照)。

【0006】

上述の如きマルチスレッド処理を実現するための方式としては、複数のスレッドを同時に実行する、fine grained multithreading方式やSimultaneous multithreading (SMT) 方式 (図11参照) の他、複数のスレッドを同時には実行せず、キャッシュミスのような事象が起きたときに別のスレッドに切り替えて実行する、coarse grained multithreading方式やVertical multithreading (VMT) 方式 (図12参照) がある。

30

【0007】

図11はSMT方式を説明するための図、図12はVMT方式を説明するための図である。

VMT方式は、処理に長時間かかるキャッシュミスした命令処理を隠蔽することを狙い、キャッシュミスを検出したときに、メモリからキャッシュまでデータを持ってくる処理をキャッシュ制御部 (図示省略) が実行している間に、メモリアクセス以外の処理について、実行部や制御部 (ともに図示省略) において他のスレッドに切り替え、実行するというものである。又、このVMT方式においては、キャッシュミスが発生しにくいスレッド

40

【特許文献1】特開2003-241961号公報

【発明の開示】

【発明が解決しようとする課題】

【0008】

しかしながら、マルチスレッド処理を実現するために、複数のスレッドでCPUの命令解読器や演算器等を共有する必要があり、又、ソフトウェアから可視のレジスタ類はスレッドごとに多重化する必要がある。従って、取り扱うレジスタ数が増えるため、レジスタ分の面積が増大し、レジスタ読み出し処理が遅くなるという課題がある。又、このように増加したレジスタを取り扱うための選択回路を追加する必要があり、これによっても、回

50

路が複雑化してレジスタ読み出し処理が遅くなる。

【0009】

本発明は、このような課題に鑑み創案されたもので、マルチスレッド処理を高速に処理することができるようにすることを目的とする。

【課題を解決するための手段】

【0010】

このため、本発明の演算処理装置は、第1のスレッドに対する第1の処理又は第2のスレッドに対する第2の処理を実行する1以上の処理部と、前記第1の処理に用いる第1のデータを格納する第1の記憶部と、前記第2の処理に用いる第2のデータを格納する第2の記憶部と、前記処理部と前記第1及び第2の記憶部に接続され、前記第1の記憶部からの第1のデータと、前記第2の記憶部からの第2のデータを保持する第1の保持部と、前記第1のスレッド及び前記第2のスレッドのうち、実行状態のスレッドに関する情報を保持する第2の保持部と、前記第1のスレッド及び前記第2のスレッドのうち、休止状態のスレッドに関する情報を保持する第3の保持部と、前記第1の処理を実行する場合に、前記第1の保持部が保持する第1のデータを、前記第1の保持部から処理部に転送するとともに、前記第1のデータが転送された処理部が出力する前記第1の処理の結果である第1の結果データを、前記第1の保持部と前記第1のスレッドに対応する前記第1の記憶部に保持し、前記第1のスレッドから前記第2のスレッドに切替えることにより前記第2の処理を実行する場合に、スレッド切替え前の前記第3の保持部が保持する休止状態の前記第2のスレッドに関する情報を、前記第2の保持部に転送するとともに、スレッド切替え前の前記第2の保持部が保持する実行状態の前記第1のスレッドに関する情報を、前記第3の保持部に転送することにより前記第2のスレッドを休止状態から実行状態に切り替えて、前記第2のスレッドに対応する前記第2の記憶部から転送した第2のデータを前記第1の保持部に保持させ、前記第1の保持部が保持した第2のデータを、前記第1の保持部から処理部に転送するとともに、前記第2のデータが転送された処理部が出力する前記第2の処理の結果である第2の結果データを、前記第1の保持部と前記第2の記憶部に保持する制御部を有することを特徴としている。

【0011】

なお、前記演算処理装置はさらに、前記第1の処理に対応する第1の命令と、前記第2の処理に対応する第2の命令を解釈するデコード部を有し、前記制御部が、前記第1の処理の実行後に前記第1のスレッドから前記第2のスレッドに切替えることにより前記第2の処理を行なう場合に、前記第2のスレッドに対応する第2の記憶部から前記第1の保持部への第2のデータの転送が完了し、前記第1の保持部から前記第2のデータの読み出しが可能になるまで、前記デコード部による前記第2の命令の解釈を停止させてもよい。

また、前記演算処理装置はさらにキャッシュメモリ装置を有するとともに、主記憶装置に接続され、前記制御部は、前記キャッシュメモリ装置に対するメモリアクセスがキャッシュミスが発生した場合に、前記第1のスレッドから前記第2のスレッドに切替えることとしてもよい。

【0012】

さらに、前記演算処理装置はさらに、前記第3の保持部の複数の格納位置と、休止状態のスレッドとを対応付けるレジスタ識別情報を保持するレジスタ識別テーブルを有し、前記制御部は、前記レジスタ識別情報に基づいて、前記第1のスレッドから前記第2のスレッドに切替えることにより前記第2の処理を実行する場合に、スレッド切替え前の前記第3の保持部が保持する休止状態の前記第2のスレッドに関する情報を、前記第2の保持部に転送するとともに、スレッド切替え前の前記第2の保持部が保持する実行状態の前記第1のスレッドに関する情報を、前記第3の保持部に転送することにより前記第2のスレッドを休止状態から実行状態に切り替えることとしてもよい。

【0013】

また、本発明の情報処理装置は、演算処理装置と前記演算処理装置に接続された記憶装置を有する情報処理装置において、前記演算処理装置は、第1のスレッドに対する第1の

10

20

30

40

50

処理又は第2のスレッドに対する第2の処理を実行する1以上の処理部と、前記第1の処理に用いる第1のデータを格納する第1の記憶部と、前記第2の処理に用いる第2のデータを格納する第2の記憶部と、前記処理部と前記第1及び第2の記憶部に接続され、前記第1の記憶部からの第1のデータと、前記第2の記憶部からの第2のデータを保持する第1の保持部と、前記第1のスレッド及び前記第2のスレッドのうち、実行状態のスレッドに関する情報を保持する第2の保持部と、前記第1のスレッド及び前記第2のスレッドのうち、休止状態のスレッドに関する情報を保持する第3の保持部と、前記第1の処理を実行する場合に、前記第1の保持部が保持する第1のデータを、前記第1の保持部から処理部に転送するとともに、前記第1のデータが転送された処理部が出力する前記第1の処理の結果である第1の結果データを、前記第1の保持部と前記第1のスレッドに対応する前記第1の記憶部に保持し、前記第1のスレッドから前記第2のスレッドに切替えることにより前記第2の処理を実行する場合に、スレッド切替え前の前記第3の保持部が保持する休止状態の前記第2のスレッドに関する情報を、前記第2の保持部に転送するとともに、スレッド切替え前の前記第2の保持部が保持する実行状態の前記第1のスレッドに関する情報を、前記第3の保持部に転送することにより前記第2のスレッドを休止状態から実行状態に切り替えて、前記第2のスレッドに対応する前記第2の記憶部から転送した第2のデータを前記第1の保持部に保持させ、前記第1の保持部が保持した第2のデータを、前記第1の保持部から処理部に転送するとともに、前記第2のデータが転送された処理部が出力する前記第2の処理の結果である第2の結果データを、前記第1の保持部と前記第2の記憶部に保持する制御部を有することを特徴としている。

10

20

【0014】

なお、前記演算処理装置はさらに、前記第1の処理に対応する第1の命令と、前記第2の処理に対応する第2の命令を解釈するデコード部を有し、前記制御部が、前記第1の処理の実行後に前記第1のスレッドから前記第2のスレッドに切替えることにより前記第2の処理を行なう場合に、前記第2のスレッドに対応する第2の記憶部から前記第1の保持部への第2のデータの転送が完了し、前記第1の保持部から前記第2のデータの読み出しが可能になるまで、前記デコード部による前記第2の命令の解釈を停止させてもよい。

また、前記演算処理装置はさらにキャッシュメモリ装置を有するとともに、主記憶装置に接続され、前記制御部は、前記キャッシュメモリ装置に対するメモリアクセスがキャッシュミスが発生した場合に、前記第1のスレッドから前記第2のスレッドに切替えること

30

【0015】

さらに、前記演算処理装置はさらに、前記第3の保持部の複数の格納位置と、休止状態のスレッドとを対応付けるレジスタ識別情報を保持するレジスタ識別テーブルを有し、前記制御部は、前記レジスタ識別情報に基づいて、前記第1のスレッドから前記第2のスレッドに切替えることにより前記第2の処理を実行する場合に、スレッド切替え前の前記第3の保持部が保持する休止状態の前記第2のスレッドに関する情報を、前記第2の保持部に転送するとともに、スレッド切替え前の前記第2の保持部が保持する実行状態の前記第1のスレッドに関する情報を、前記第3の保持部に転送することにより前記第2のスレッドを休止状態から実行状態に切り替えることとしてもよい。

40

【0016】

また、本発明の演算処理装置の制御方法は、第1のスレッドに対する第1の処理又は第2のスレッドに対する第2の処理を実行する1以上の処理部と、前記第1の処理に用いる第1のデータを格納する第1の記憶部と、前記第2の処理に用いる第2のデータを格納する第2の記憶部と、前記第1のスレッド及び前記第2のスレッドのうち、実行状態のスレッドに関する情報を保持する第2の保持部と、前記第1のスレッド及び前記第2のスレッドのうち、休止状態のスレッドに関する情報を保持する第3の保持部とを有する演算処理装置の制御方法において、前記演算処理装置が有する制御部が、前記第1の処理を実行する場合に、前記演算処理装置が有する第1の保持部が保持する第1のデータを、前記第1の保持部から処理部に転送し、前記制御部が、前記第1のデータが転送された処理部が出

50

力する前記第 1 の処理の結果である第 1 の結果データを、前記第 1 のスレッドに対応する前記第 1 の記憶部と前記第 1 の保持部に保持し、前記制御部が、前記第 1 のスレッドから前記第 2 のスレッドに切替えることにより前記第 2 の処理を実行する場合に、スレッド切替え前の前記第 3 の保持部が保持する休止状態の前記第 2 のスレッドに関する情報を、前記第 2 の保持部に転送し、前記制御部が、スレッド切替え前の前記第 2 の保持部が保持する実行状態の前記第 1 のスレッドに関する情報を、前記第 3 の保持部に転送することにより前記第 2 のスレッドを休止状態から実行状態に切り替え、前記制御部が、前記第 2 のスレッドに対応する前記第 2 の記憶部から転送した第 2 のデータを前記第 1 の保持部に保持させ、前記第 1 の保持部が保持した第 2 のデータを、前記第 1 の保持部から処理部に転送し、前記制御部が、前記第 2 のデータが転送された処理部が出力する前記第 2 の処理の結果である第 2 の結果データを、前記第 1 の保持部と前記第 2 の記憶部に保持することを特徴としている。

10

さらに、前記演算処理装置の制御方法において、前記演算処理装置はさらに、前記第 1 の処理に対応する第 1 の命令と、前記第 2 の処理に対応する第 2 の命令を解読するデコード部を有し、前記制御部が、前記第 1 の処理の実行後に前記第 1 のスレッドから前記第 2 のスレッドに切替えることにより前記第 2 の処理を行なう場合に、前記第 2 のスレッドに対応する第 2 の記憶部から前記第 1 の保持部への第 2 のデータの転送が完了し、前記第 1 の保持部から前記第 2 のデータの読み出しが可能になるまで、前記デコード部による前記第 2 の命令の解読を停止させてもよい。

また、前記演算処理装置の制御方法において、前記演算処理装置はさらにキャッシュメモリ装置を有するとともに、主記憶装置に接続され、前記制御部は、前記キャッシュメモリ装置に対するメモリアクセスがキャッシュミスが発生した場合に、前記第 1 のスレッドから前記第 2 のスレッドに切替えることとしてもよい。

20

【0017】

さらに、前記演算処理装置の制御方法において、前記演算処理装置はさらに、前記第 3 の保持部の複数の格納位置と、休止状態のスレッドとを対応付けるレジスタ識別情報を保持するレジスタ識別テーブルを有し、前記制御部は、前記レジスタ識別情報に基づいて、前記第 1 のスレッドから前記第 2 のスレッドに切替えることにより前記第 2 の処理を実行する場合に、スレッド切替え前の前記第 3 の保持部が保持する休止状態の前記第 2 のスレッドに関する情報を、前記第 2 の保持部に転送するとともに、スレッド切替え前の前記第 2 の保持部が保持する実行状態の前記第 1 のスレッドに関する情報を、前記第 3 の保持部に転送することにより前記第 2 のスレッドを休止状態から実行状態に切り替えることとしてもよい。

30

【発明の効果】

【0018】

本発明によれば、マルチスレッド処理を行なう場合においても、レジスタからの読み出し処理を高速に行なうことができる利点がある。

【発明を実施するための最良の形態】

【0019】

以下、図面を参照して本発明の実施の形態を説明する。

40

図 1 は本発明の一実施形態としてのマルチスレッドプロセッサの構成を示すブロック図、図 2 および図 3 はそれぞれ本マルチスレッドプロセッサにおけるジョイントワークレジスタの機能構成を説明するための図である。

本発明の一実施形態としてのマルチスレッドプロセッサである CPU (Central Processing Unit) 1 は、SPARC (Scalable Processor ARChitecture; Sun Microsystems 社) 等のレジスタウィンドウ方式 (詳細は後述) のアーキテクチャをそなえるプロセッサであり、複数のスレッド (プログラム) を 1 以上の演算器 15 (図 2 参照) により並列的に実行可能なマルチスレッドプロセッサとして構成されている。

【0020】

本 CPU 1 は、図 1 に示すように、マルチスレッド制御部 11, 状態レジスタ部 120

50

レジスタウィンドウ 13, ジョイントワークレジスタ (ワークレジスタ) 14, 実行部 150, 命令制御部 16, TLB (論理アドレス・物理アドレス変換部) 17, L1 命令キャッシュ 18, L1 データキャッシュ 19 および L2 キャッシュ 20 をそなえて構成され、又、バス 22 を介してメインメモリ 21 と通信可能に接続されている。

【0021】

L1 命令キャッシュ 18 は、実行部 150 が実行した命令を格納するための 1 次キャッシュであり、L1 データキャッシュ 19 は、実行部 150 が命令の実行に際して使用された各種データを格納するための 1 次キャッシュである。L2 キャッシュ 20 は、各種命令やデータを格納するための 2 次キャッシュである。

命令制御部 16 は、1 次命令キャッシュ 18 や L2 キャッシュ, メインメモリ 21 から命令を取得して実行部 150 により実行させるための制御を行なうものであり、TLB (Translation Look-aside Buffer; 論理アドレス・物理アドレス変換部) 17 は、論理アドレスから物理アドレスへの変換で発生するページテーブル参照のペナルティを小さくするために設けられた一種のキャッシュメモリであり、参照されたアドレスと変換情報の履歴を保存しておき、アドレス変換の高速化を実現するものである。

【0022】

実行部 (演算器) 150 は、レジスタウィンドウ 13 やジョイントワークレジスタ 14, 状態レジスタ 12 等を用いて、演算等の種々の処理 (命令処理) を行なうものであり、例えば種々の演算処理を実行する演算器 (実行演算器) 15 として機能するようになっており、レジスタ読み出しバス 23 を介してジョイントワークレジスタ 14 に接続されるとともに (図 2 参照)、レジスタ書き込みバス 25 を介してジョイントワークレジスタ 14 と、又、レジスタ書き込みバス 24 - 1 ~ 24 - 4 を介して各レジスタウィンドウ 13 - 1 ~ 13 - 4 とそれぞれ接続されている (図 3 参照)。

【0023】

具体的には、実行部 150 は、レジスタ書き込みバス 24 - 1 を介してレジスタウィンドウ 13 - 1 と、レジスタ書き込みバス 24 - 2 を介してレジスタウィンドウ 13 - 2 と、レジスタ書き込みバス 24 - 3 を介してレジスタウィンドウ 13 - 3 と、レジスタ書き込みバス 24 - 4 を介してレジスタウィンドウ 13 - 4 とそれぞれ接続されている。

メインメモリ 21 は、CPU 1 の外部にそなえられたメモリであって、種々の命令やデータを格納するものであり、CPU 1 とバス 22 を介して通信可能に接続され、実行部 150 による処理・実行に必要なとされる命令やデータが L1 命令キャッシュ 18, L1 データキャッシュ 19 および L2 キャッシュ 20 に格納されていない場合 (キャッシュミス時) には、CPU 1 はバス 22 等を介してこのメインメモリ 21 から命令やデータを読み出すようになっている。

【0024】

レジスタウィンドウ 13 は、実行部 150 による命令処理に使用されるデータを格納可能な記憶領域であり、図 3 に示すように、スレッド毎にそれぞれそなえられている。すなわち、本 CPU 1 は、マルチスレッドを実装するために、スレッドに対応させてレジスタ (レジスタウィンドウ 13) を多重化して構成されているのである。なお、このレジスタウィンドウ 13 に対するデータの書き込みや読み出しは、後述するマルチスレッド制御部 11 による制御に従って行なわれるようになっている。以下、本実施形態においては CPU 1 がスレッド 0 ~ スレッド 3 の 4 つのスレッドを並列的に実行 (処理) する例について説明するものであり、本 CPU 1 が 4 つのレジスタウィンドウ 13 - 1 ~ 13 - 4 をそなえて構成されている例について説明する。

【0025】

なお、以下、レジスタウィンドウを示す符号としては、複数のレジスタウィンドウのうち 1 つを特定する必要があるときには符号 13 - 1 ~ 13 - 4 を用いるが、任意のレジスタウィンドウを指すときには符号 13 を用いる。又、一般に、プロセッサ等にそなえられるレジスタは、実際には数ビットのデータを保持可能な複数のレジスタの集合 (レジスタ群) として構成されるものであるが、本実施形態においては、便宜上、単にレジスタと称

10

20

30

40

50

する。

【 0 0 2 6 】

図 2 に示す例においては、スレッド 0 がレジスタウィンドウ 1 3 - 1 を、スレッド 1 がレジスタウィンドウ 1 3 - 2 を、スレッド 2 がレジスタウィンドウ 1 3 - 3 を、スレッド 3 がレジスタウィンドウ 1 3 - 4 をそれぞれ用いるようになっており、マルチスレッド制御部 1 1 によって、これらのレジスタウィンドウ 1 3 - 1 ~ 1 3 - 4 がジョイントワークレジスタ 1 4 を共有するように制御されるようになっている。又、この図 2 中においては、演算処理毎に演算器 1 5 が示されている。

【 0 0 2 7 】

ジョイントワークレジスタ (ワークレジスタ) 1 4 は、上記複数 (4 つ) のレジスタウィンドウ 1 3 と実行部 1 5 0 との間にそなえられ、各レジスタウィンドウ 1 3 及び実行部 1 5 0 との間でデータを相互に転送可能に構成されている。すなわち、ジョイントワークレジスタ 1 4 は、各レジスタウィンドウ 1 3 から転送されるデータを格納したり、又、各レジスタウィンドウ 1 3 にデータを転送したりするように構成されている。このジョイントワークレジスタ 1 4 には、演算器 1 5 によって実行されるスレッドにかかるデータが格納され、演算器 1 5 は、このジョイントワークレジスタ 1 4 に格納されたデータを用いて種々の演算処理を行なうようになっている。

【 0 0 2 8 】

さて、本レジスタウィンドウ方式においては、例えば、1 つのウィンドウが 3 2 ワードで構成され、8 ワードのグローバルレジスタ、8 ワードのインレジスタ、8 ワードのアウトレジスタおよび 8 ワードのローカルレジスタによって 1 つのウィンドウが構成されている。なお、本実施形態においては、ウィンドウ数 = 8、すなわち、8 ウィンドウの例について説明する。

【 0 0 2 9 】

図 4 はレジスタウィンドウ方式を説明するための図である。本 CPU 1 において採用されているレジスタウィンドウ方式においては、この図 4 に示すように、複数のレジスタウィンドウがリング状に連結されている。

このレジスタウィンドウ方式においては、図 4 に示すように、CWP (Current Window Pointer) を用いて現在のウィンドウ (カレントウィンドウ) が示されるようになっており、この CWP によって指し示されるウィンドウ番号のレジスタを一時にプログラムで扱うことができるようになっている。なお、図 4 においては、CWP = 2 の例を示している。

【 0 0 3 0 】

そして、サブルーチンコールのときには、SAVE 命令を実行して CWP をプラス 1 して更新し、サブルーチンリターンのときには、RESTORE 命令や RETURN 命令を実行して CWP をマイナス 1 して更新するようになっている。ウィンドウポインタがプラス 1 したものは、プラス 1 する前のウィンドウのアウトレジスタ (outs) 8 ワードをインレジスタ (ins) として扱い、逆にマイナス 1 したものは、マイナス 1 する前のウィンドウのインレジスタをアウトレジスタとして扱うようになっている。このようにして、隣のウィンドウと 8 ワード分共有するようになっている。

【 0 0 3 1 】

なお、グローバルレジスタは、ウィンドウポインタを変更しても変わらないものであり、割り込み処理などが生じたとき等の特別な処理時にのみ入れ替えられるようになっている。

また、CWP を更新、つまりウィンドウを切り替えると、新しいレジスタを用いて処理が可能になり、レジスタのメモリ退避のペナルティ損失を軽減できるようになっている。

【 0 0 3 2 】

この図 4 に示すレジスタウィンドウにおいて、CWP によって特定されるカレントウィンドウと、その両隣のウィンドウとをジョイントワークレジスタ (JWR (Joint-Work-Register ; 結合作業レジスタ) 領域と呼ぶ。具体的には、この JWR 領域は、CWP, C

10

20

30

40

50

WP + 1およびCWP - 1の3ウィンドウ分のインレジスタ、アウトレジスタ、ローカルレジスタ、さらにCWPに関係ないグローバルレジスタを有する合計64ワードの領域として構成される。

【0033】

ウィンドウを切り替えるのに最も用いられるSAVE命令やRESTORE命令、RETURN命令の実行の際には、ウィンドウが切り替わった場合も、隣のウィンドウがすでにJWR領域内に格納されているため、命令処理をインターロックせずに、CWPを更新し、その後、隣のウィンドウ内のレジスタ読み出しを可能にしている。そして、上記命令の完了の際に、さらに一つ隣のウィンドウがバックグラウンドでJWR領域に転送されるようになっている。

10

【0034】

そして、このJWR領域のデータが、前述したジョイントワークレジスタ14内に格納されるようになっている。

また、演算器15の演算結果等のジョイントワークレジスタ14に対するデータの書き込みや、ジョイントワークレジスタ14からのデータの読み出しは、マルチスレッド制御部11による制御に従って行なわれるようになっている。

【0035】

マルチスレッド制御部11は、実行部150（演算器15）により処理を実行するスレッドを識別するための実行スレッド識別子に基づいて、複数のレジスタウィンドウ13とワークレジスタ14と演算器15との間におけるデータ転送を制御するものである。

20

このマルチスレッド制御部11は、図3に示すように、実行部150による命令完了時のレジスタ更新を、ジョイントワークレジスタ14と、その命令にかかるスレッドに対応するレジスタウィンドウ13との両方に対してそれぞれ行なわせるように制御するようになっている（レジスタ更新ステップ）。そして、演算器15によるスレッド実行中におけるレジスタ読み出しは、このジョイントワークレジスタ14から行なわれるようになっている。

【0036】

また、マルチスレッド制御部11は、演算器15による処理対象のスレッドの切替えを行なうに際して、切替え後に実行対象になるスレッドに対応するレジスタウィンドウ13からジョイントワークレジスタ14にデータの転送を行なうように（データ転送ステップ）、そのデータ転送を制御するようになっている。

30

さらに、マルチスレッド制御部11は、演算器15による処理対象のスレッドの切替えを行なうに際して、切替え後に実行対象になるスレッドに対応するレジスタウィンドウ13からジョイントワークレジスタ14へのデータの転送が完了し、ジョイントワークレジスタ14から演算器15による当該データの読み出しが可能になるまで、命令解読ステージで実行パイプラインをストールさせるようになっている。

【0037】

図5は本CPU1におけるスレッドの命令処理の開始タイミングの例を示す図である。例えば、キャッシュミスの発生時や最後にスレッド切り替えを行なった後の一定時間経過時等のタイミングでスレッドを切り替える際には、先ずCWPを新しいスレッドに置き換え、その後、LOAD-CWP信号やCWPをマルチスレッド制御部11から送信し、さらにこのマルチスレッド制御部11から新しいスレッド識別子を送信する。レジスタウィンドウ13からジョイントワークレジスタ14へは、LOAD-CWP信号により転送を開始するが、その時点のスレッド識別子を参照して、新しいスレッドのレジスタウィンドウを選択するようになっている。

40

【0038】

さて、図5に示す例において3ウィンドウ分のレジスタをレジスタウィンドウ13からジョイントワークレジスタ14に転送するためには9サイクルかかる。この間ジョイントワークレジスタ14から演算器15にデータを読み出すことができないため、マルチスレッド制御部11は、図5に示すように、インターロック（D_インターロック）を行なう

50

ことにより命令解釈ステージで実行パイプラインをストールさせるのである。

【0039】

図6は本発明の一実施形態としてのマルチスレッドプロセッサにおける状態レジスタ部120の構成を説明するための図であり、図7はそのスレッド切り替え時における状態レジスタ部120の処理を説明するための図である。

状態レジスタ部120は、例えば、実行部150が演算命令を実行したときの実行結果の状態等、プロセッサの状態を示す情報を保持するものであり、図6に示すように、休止用プロセッサ状態レジスタ(第2レジスタ)12-1, 12-2, 12-3, 実行用プロセッサ状態レジスタ(第1レジスタ; ハードウェア制御用プロセッサ状態レジスタ)121およびレジスタ識別テーブル122をそなえて構成されている。

10

【0040】

実行用プロセッサ状態レジスタ121は、複数のスレッドのうち演算器15により実行される実行対象スレッドに関するデータを格納可能なものであり、本実施形態においては1つそなえられている。

休止用プロセッサ状態レジスタ(第2レジスタ)12-1~12-3は、実行用プロセッサ状態レジスタ(第1レジスタ)121との間で該データを相互に転送可能に構成され、複数のスレッドのうち実行対象スレッド以外の休止スレッドに関するデータを格納可能であり、これらの休止スレッドと同数(本実施形態では3つ)そなえられている。

【0041】

なお、以下、休止用プロセッサ状態レジスタを示す符号としては、複数の休止用プロセッサ状態レジスタ12のうち1つを特定する必要があるときには符号12-1~12-3を用いるが、任意の休止用プロセッサ状態レジスタを指すときには符号12を用いる。

20

また、マルチスレッド制御部11(レジスタ制御部)は、休止用プロセッサ状態レジスタ12と実行用プロセッサ状態レジスタ121との間におけるデータ転送を制御するようになっており、休止スレッドとその休止スレッドに関するデータとを対応付けて成るレジスタ識別テーブル122をそなえ、このレジスタ識別テーブル122を参照してデータ転送を制御するようになっている。

【0042】

なお、本実施形態においては、レジスタ識別テーブル122は、休止スレッドのスレッド番号と、その休止スレッドにかかる休止用プロセッサ状態レジスタ12の格納位置(物理スタック; スタック#(番号))とを対応付けて管理している(スレッド対スタックのマップ)。

30

そして、マルチスレッド制御部11は、演算器15による処理対象のスレッドの切替えを行なうに際して、レジスタ識別テーブル122を参照して、切替え後に実行対象になるスレッド(実行対象スレッド)に対応する休止用プロセッサ状態レジスタ12を選択し(選択ステップ)、この休止用プロセッサ状態レジスタ12に格納されているデータと、実行用プロセッサ状態レジスタ121に格納されているデータとを置換し、更に、レジスタ識別テーブル122を置換後の状態に更新するようになっている。

【0043】

すなわち、スレッドの切り替えを行なう場合には、図7に示すように、マルチスレッド制御部11は、実行スレッド識別子に基づいて新たに実行対象スレッドを認識し、現在実行中のスレッド(図7に示す例においてはスレッド0)から、新たに実行状態になる実行対象スレッド(図7に示す例においてはスレッド1)への切り替え指示がスレッド切り替え制御部から行なわれると、マルチスレッド制御部11は、レジスタ識別テーブル122を参照して、実行対象スレッド(スレッド1)にかかる休止用プロセッサ状態レジスタ12のスタック番号(スタック0)を取得する。

40

【0044】

マルチスレッド制御部11は、この取得したスタック0から実行用プロセッサ状態レジスタ121にデータを転送するとともに、スタック0が空くので、実行用プロセッサ状態レジスタ121のデータをスタック0番に転送する。これにより、選択された休止用プロ

50

セッサ状態レジスタ12と実行用プロセッサ状態レジスタ121とのレジスタ内のデータの入れ替えを行なう(置換ステップ)。

【0045】

また、マルチスレッド制御部11は、レジスタ識別テーブル122の更新を行なう(更新ステップ)。すなわち、レジスタ識別テーブル122のスレッド0に対応するスタック番号に、次に実行状態になるスレッド1のスタック番号0をコピーし、つまりスタック0番を書き込む。

なお、レジスタ識別テーブル122においては、現在実行中のスレッドに対応するスタック#は、実行前にスタックされていた番号が入ったままとなっている(図7における更新後のレジスタ識別テーブル122のスレッド1参照)。

10

【0046】

また、プロセッサ状態レジスタの入れ替えは、スレッド切り替え後すぐに新しいスレッドの命令フェッチを開始するため、1サイクルで行なう。同時に行なうレジスタウィンドウからジョイントワークレジスタ14へのデータ転送に比べて処理サイクルが少ないので、プロセッサ状態レジスタ転送用には特別にインターロックをかけない。

このように、本発明の一実施形態としてのCPU1によれば、レジスタウィンドウ方式のアーキテクチャであり、カレントウィンドウを含んだジョイントワークレジスタ14を備えるプロセッサで、キャッシュミスや一定時間でスレッドを切り替えるマルチスレッド処理を行なう場合に、マルチスレッド用にウィンドウレジスタ13を多重化する際に、異なるスレッド(レジスタ)間でジョイントワークレジスタ14を共有することにより、演算器15のレジスタ読み出しをマルチスレッドをしない場合と同等に高速に行なうことが可能になる。

20

【0047】

また、スレッド数だけ多重化されたレジスタウィンドウ13とジョイントワークレジスタ14とを相互に通信可能に構成して、ジョイントワークレジスタ14を複数のスレッドで共有可能に構成することにより、マルチスレッドを実装し、スレッド実行中のレジスタ読み出しをマルチスレッド処理しない場合と同様に行なうことを可能にしている。

さらに、命令完了時のレジスタ書き込みは、ジョイントワークレジスタ14と同時にレジスタウィンドウにも行なうことにより、ジョイントワークレジスタ14からレジスタウィンドウに書き戻す処理が不要であり、処理を高速化することができる。すなわち、スレッド切り替えの際には、次に実行状態になるスレッドのレジスタウィンドウ13からジョイントワークレジスタ14に転送を行なうだけでよく、切り替わる前のスレッドにかかるデータをジョイントワークレジスタ14からそのスレッドにかかるレジスタウィンドウ13への転送をする必要がない。

30

【0048】

また、CPU1の状態を示すプロセッサ状態レジスタなどのように、常に参照する可能性があるためにワークレジスタ(ジョイントワークレジスタ)を持たないレジスタに関しては、実行用プロセッサ状態レジスタ121と、休止スレッド用の休止用プロセッサ状態レジスタ12とをそなえ、スレッドを切り替えるごとに実行用プロセッサ状態レジスタ121の内容を、対応する休止用プロセッサ状態レジスタ12と入れ替えて格納し、スレッドの実行時に、実行用プロセッサ状態レジスタ121のみを用いるようにすることにより、マルチスレッドをしない場合と同等にレジスタ参照を高速に行なうことが可能になる。

40

【0049】

本発明では、キャッシュミスや一定時間経過をトリガーにして、実行するスレッドを切り替えるマルチスレッド方式を採用するプロセッサにおいて、レジスタウィンドウ13を多重化し、これらのレジスタウィンドウ13によりレジスタ読み出し用のジョイントワークレジスタ(ワークレジスタ)14を共有することにより、マルチスレッドでレジスタが増大した場合も、スレッド実行状態の命令処理中のレジスタ読み出しにおいてマルチスレッド方式をしない場合と同じように高速に行なうことが可能になる。又、ワークレジスタをとらない状態レジスタにおいても、実行用プロセッサ状態レジスタ121と休止用プロ

50

セッサ状態レジスタ12とをそなえ、スレッドを切り替える時に実行用プロセッサ状態レジスタ121と、次に実行するスレッドを格納するスタックとを入れ替えることにより、マルチスレッドでレジスタが増大した場合も、スレッド実行状態の命令処理中のレジスタ読み出しにおいてマルチスレッド方式をしない場合と同じように高速に行なうことが可能になる。

【0050】

そして、本発明は上述した実施形態に限定されるものではなく、本発明の趣旨を逸脱しない範囲で種々変形して実施することができる。

例えば、上述した実施形態においては、マルチスレッドの例として4スレッドを実行する例について説明しているが、これに限定されるものではなく、3以下もしくは5以上のスレッドを実行してもよい。

10

【0051】

また、上述した実施形態においては、ワークレジスタを持たないレジスタの例として状態レジスタを用いて説明しているが、これに限定されるものではなく、状態レジスタ以外の他のレジスタに上記手法を適用してもよい。

なお、本発明の各実施形態が開示されていれば、本発明を当業者によって実施・製造することが可能である。

【0052】

そして、本発明は、以下に示すように要約することができる。

(付記1) 複数のスレッドを1以上の演算器により並列的に実行可能なマルチスレッドプロセッサであって、

20

該演算器による命令処理に使用されるデータを格納可能なレジスタウィンドウを該スレッド毎にそなえるとともに、

前記複数のレジスタウィンドウとの間及び該演算器との間で該データを相互に転送可能なワークレジスタと、

該演算器により処理を実行する該スレッドを識別するための実行スレッド識別子に基づいて、前記複数のレジスタウィンドウと該ワークレジスタと該演算器との間におけるデータ転送を制御するマルチスレッド制御部とをそなえることを特徴とする、マルチスレッドプロセッサ。

【0053】

30

(付記2) 該マルチスレッド制御部が、

該演算器による命令完了時のレジスタ更新を、該ワークレジスタとともに当該命令完了にかかるスレッドに対応するレジスタウィンドウに対しても行なうことを特徴とする、付記1記載のマルチスレッドプロセッサ。

(付記3) 該マルチスレッド制御部が、

該演算器による処理対象の該スレッドの切替えを行なうに際して、切替え後に実行対象になる該スレッドに対応する該レジスタウィンドウから該ワークレジスタに前記更新された該データの転送を行なうように、前記データ転送を制御することを特徴とする、付記2記載のマルチスレッドプロセッサ。

【0054】

40

(付記4) 該マルチスレッド制御部が、

該演算器による処理対象の該スレッドの切替えを行なうに際して、前記切替え後に実行対象になるスレッドに対応するレジスタウィンドウから該ワークレジスタへの該データの転送が完了し、該ワークレジスタから該演算器による当該データの読み出しが可能になるまで、命令解読ステージで実行パイプラインをストールさせることを特徴とする、付記3記載のマルチスレッドプロセッサ。

【0055】

(付記5) 複数のスレッドを1以上の演算器により並列的に実行可能なマルチスレッドプロセッサであって、

前記複数のスレッドのうち該演算器により実行される実行対象スレッドに関するデータ

50

を格納可能な第 1 レジスタと、

該第 1 レジスタとの間で該データを相互に転送可能に構成され、前記複数のスレッドのうち該実行対象スレッド以外の休止スレッドに関するデータを格納可能であり、該休止スレッドと同数そなえられた第 2 レジスタと、

該第 2 レジスタと該第 1 レジスタとの間におけるデータ転送を制御するレジスタ制御部とをそなえることを特徴とする、マルチスレッドプロセッサ。

【 0 0 5 6 】

(付記 6) 該休止スレッドと当該休止スレッドに関するデータを格納している該第 2 レジスタとを対応付けて成るレジスタ識別テーブルをそなえ、

該レジスタ制御部が、該レジスタ識別テーブルを参照して該データ転送を制御することを特徴とする、付記 5 記載のマルチスレッドプロセッサ。

10

(付記 7) 該レジスタ制御部が、

該演算器による処理対象の該スレッドの切替えを行なうに際して、該レジスタ識別テーブルを参照して、切替え後に実行対象になる該スレッドに対応する該第 2 レジスタを選択し、当該選択された該第 2 レジスタに格納されている該データと、該第 1 レジスタに格納されている該データとを置換するとともに、該データの置換結果に基づいて該レジスタ識別テーブルを更新することを特徴とする、付記 6 記載のマルチスレッドプロセッサ。

【 0 0 5 7 】

(付記 8) 前記複数のスレッドのうち該演算器により実行される実行対象スレッドに関するデータを格納可能な第 1 レジスタと、

20

該第 1 レジスタとの間で該データを相互に転送可能に構成され、前記複数のスレッドのうち該実行対象スレッド以外の休止スレッドに関するデータを格納可能であり、該休止スレッドと同数そなえられた第 2 レジスタと、

該第 2 レジスタと該第 1 レジスタとの間におけるデータ転送を制御するレジスタ制御部とをそなえることを特徴とする、付記 1 ~ 付記 4 のいずれか 1 項に記載のマルチスレッドプロセッサ。

【 0 0 5 8 】

(付記 9) 該休止スレッドと当該休止スレッドに関するデータを格納している該第 2 レジスタとを対応付けて成るレジスタ識別テーブルをそなえ、

該レジスタ制御部が、該レジスタ識別テーブルを参照して該データ転送を制御することを特徴とする、付記 8 記載のマルチスレッドプロセッサ。

30

(付記 1 0) 該レジスタ制御部が、

該演算器による処理対象の該スレッドの切替えを行なうに際して、該レジスタ識別テーブルを参照して、切替え後に実行対象になる該スレッドに対応する該第 2 レジスタを選択し、当該選択された該第 2 レジスタに格納されている該データと、該第 1 レジスタに格納されている該データとを置換するとともに、該データの置換結果に基づいて該レジスタ識別テーブルを更新することを特徴とする、付記 9 記載のマルチスレッドプロセッサ。

【 0 0 5 9 】

(付記 1 1) 複数のスレッドを 1 以上の演算器により並列的に実行可能であり、該演算器による命令処理に使用されるデータを格納可能なレジスタウィンドウを該スレッド毎にそなえるとともに、各レジスタウィンドウとの間で該データを相互に転送可能なワークレジスタとをそなえるマルチスレッドプロセッサにおけるレジスタ制御方法であって、

40

該演算器による命令完了時のレジスタ更新を、該ワークレジスタとともに当該命令完了にかかるスレッドに対応するレジスタウィンドウに対しても行なうレジスタ更新ステップと、

該演算器による処理対象の該スレッドの切替えを行なうに際して、切替え後に実行対象になる該スレッドに対応する該レジスタウィンドウから該ワークレジスタに、該レジスタ更新ステップにおいて更新された該データの転送を行なうデータ転送ステップとをそなえることを特徴とする、レジスタ制御方法。

【 0 0 6 0 】

50

(付記12) 該演算器による処理対象の該スレッドの切替えを行なうに際して、前記切替え後に実行対象になるスレッドに対応するレジスタウィンドウから該ワークレジスタへの該データの転送が完了し、該ワークレジスタから該演算器による当該データの読み出しが可能になるまで、命令解釈ステージで実行パイプラインをストールさせるストールステップをそなえることを特徴とする、付記11記載のレジスタ制御方法。

【0061】

(付記13) 複数のスレッドを1以上の演算器により並列的に実行可能であり、前記複数のスレッドのうち該演算器により実行される実行対象スレッドに関するデータを格納可能な第1レジスタと、該第1レジスタとの間で該データを相互に転送可能に構成され、前記複数のスレッドのうち該実行対象スレッド以外の休止スレッドに関するデータを格納可能であり、該休止スレッドと同数そなえられた第2レジスタとをそなえるマルチスレッドプロセッサにおけるレジスタ制御方法であって、

該演算器による処理対象の該スレッドの切替えを行なうに際して、該休止スレッドと当該休止スレッドに関するデータを格納している該第2レジスタとを対応付けて成るレジスタ識別テーブルを参照して、切替え後に実行対象になる該スレッドに対応する該第2レジスタを選択する選択ステップと、

該選択ステップにおいて選択された該第2レジスタに格納されている該データと、該第1レジスタに格納されている該データとを置換する置換ステップと、

該レジスタ識別テーブルを該置換ステップにおける置換結果に基づいて更新する更新ステップとをそなえることを特徴とする、レジスタ制御方法。

【産業上の利用可能性】

【0062】

S P A R C等の比較的大きなレジスタ領域を有するプロセッサでマルチスレッド処理を行なう際のレジスタ制御に適用できる。

【図面の簡単な説明】

【0063】

【図1】本発明の一実施形態としてのマルチスレッドプロセッサの構成を示すブロック図である。

【図2】本発明の一実施形態としてのマルチスレッドプロセッサにおけるジョイントワークレジスタの機能構成を説明するための図である。

【図3】本発明の一実施形態としてのマルチスレッドプロセッサにおけるジョイントワークレジスタの機能構成を説明するための図である。

【図4】レジスタウィンドウ方式を説明するための図である。

【図5】本発明の一実施形態としてのマルチスレッドプロセッサにおけるスレッドの命令処理の開始タイミングの例を示す図である。

【図6】本発明の一実施形態としてのマルチスレッドプロセッサにおける状態レジスタ部の構成を説明するための図

【図7】本発明の一実施形態としてのマルチスレッドプロセッサにおけるスレッド切り替え時における状態レジスタ部の処理を説明するための図である。

【図8】インオーダー実行型の処理方法を説明するための図である。

【図9】アウトオブオーダー実行型の処理方法を説明するための図である。

【図10】(a), (b)はマルチスレッドプロセッサ方式を説明するための図である。

【図11】SMT方式を説明するための図である。

【図12】VMT方式を説明するための図である。

【符号の説明】

【0064】

- 1 CPU (マルチスレッドプロセッサ)
- 11 マルチスレッド制御部 (レジスタ制御部)
- 12 - 1 ~ 12 - 3, 12 休止用プロセッサ状態レジスタ (第2レジスタ)
- 13, 13 - 1 ~ 13 - 4 レジスタウィンドウ

10

20

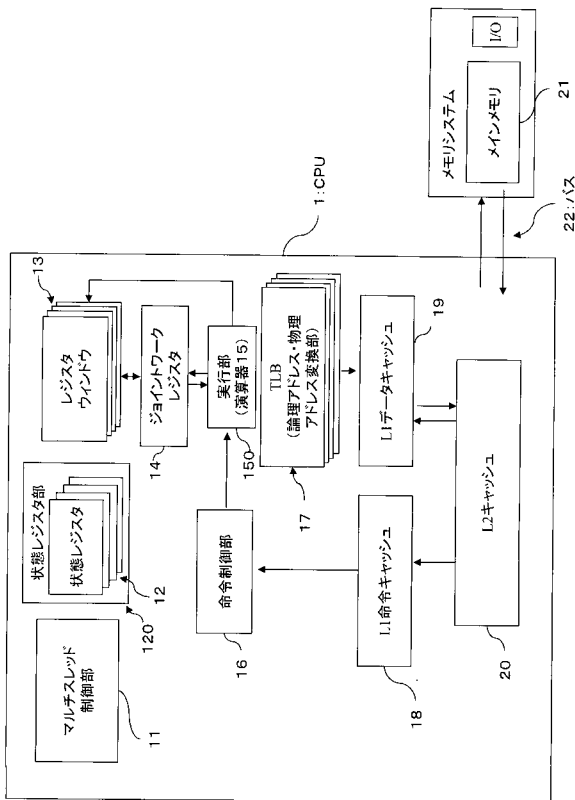
30

40

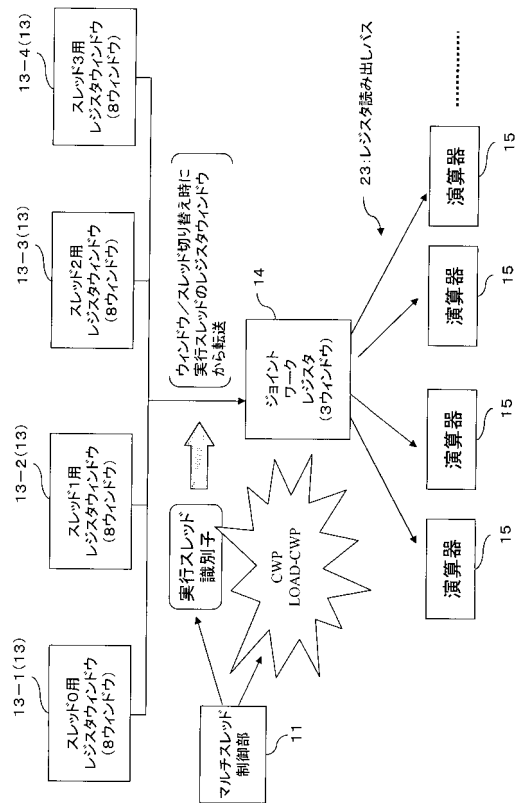
50

- 14 ジョイントワークレジスタ (ワークレジスタ)
- 15 演算器
- 16 命令制御部
- 17 TLB
- 18 L1命令キャッシュ
- 19 L1データキャッシュ
- 20 L2キャッシュ
- 21 メインメモリ
- 22 バス
- 23 レジスタ読み出しバス
- 24 - 1 ~ 24 - 4, 25 レジスタ書き込みバス
- 120 状態レジスタ部
- 121 実行用プロセッサ状態レジスタ (第1レジスタ)
- 122 レジスタ識別テーブル
- 150 実行部

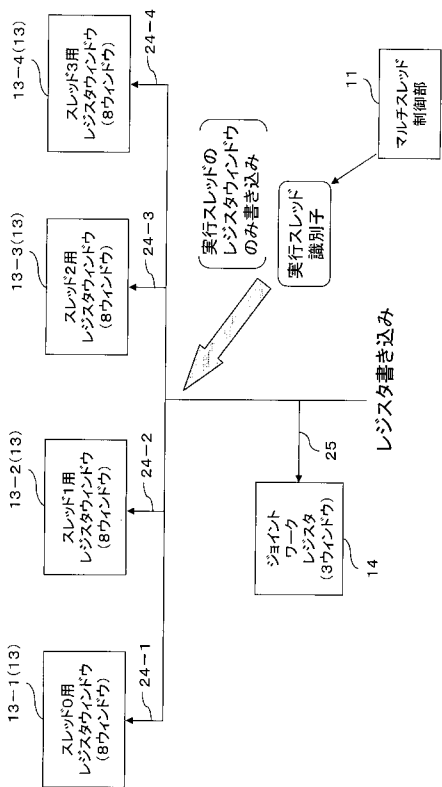
【図1】



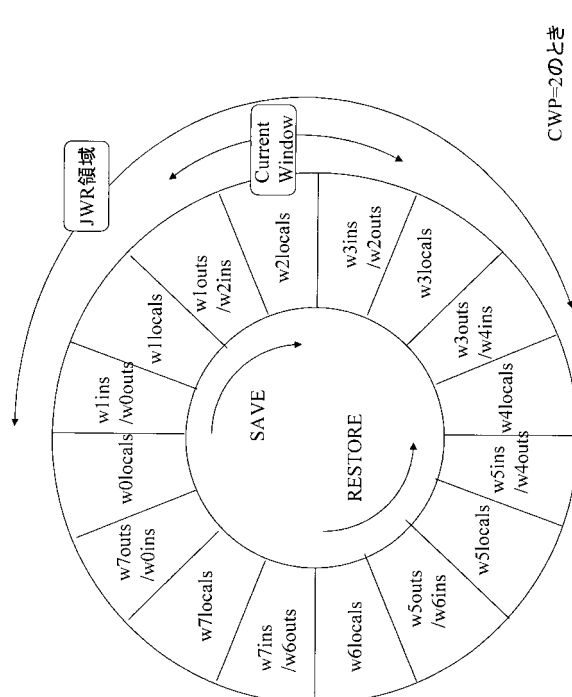
【図2】



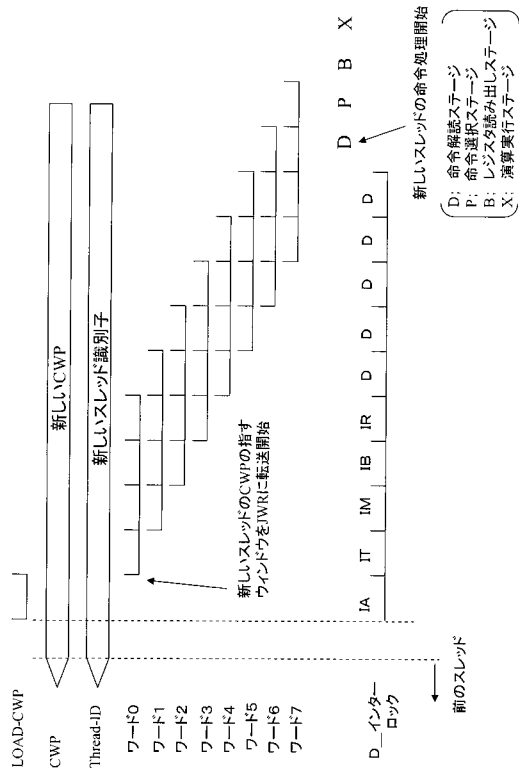
【 図 3 】



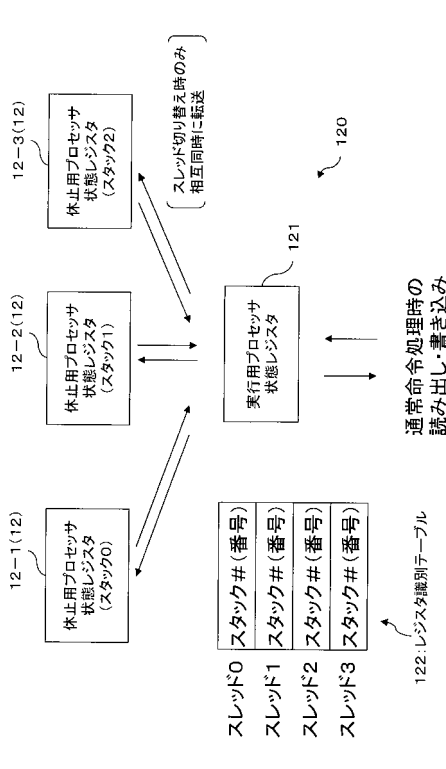
【 図 4 】



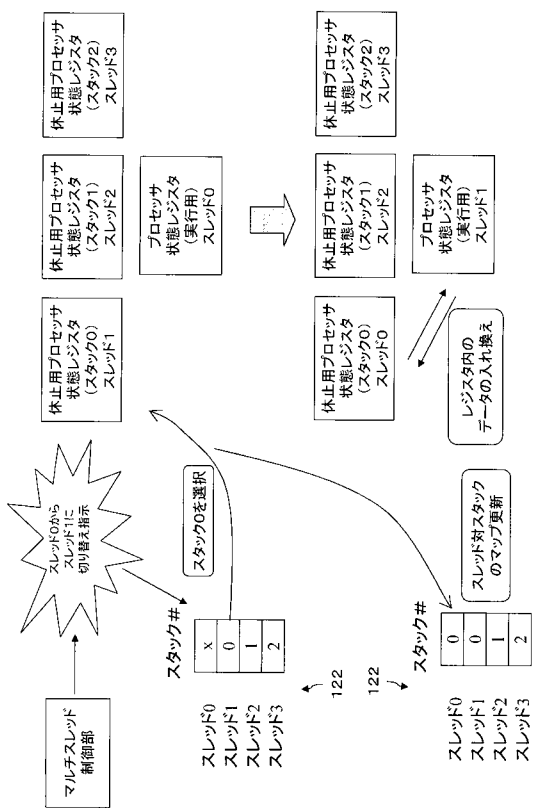
【 図 5 】



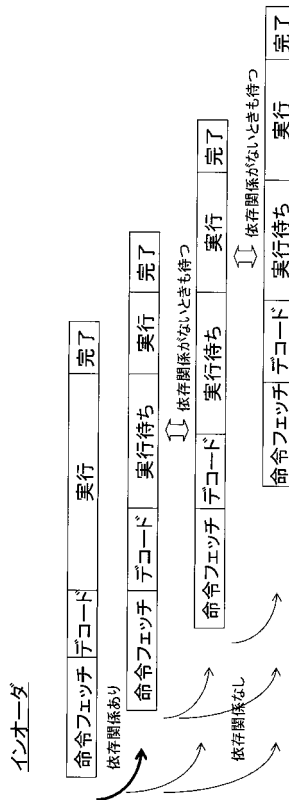
【 図 6 】



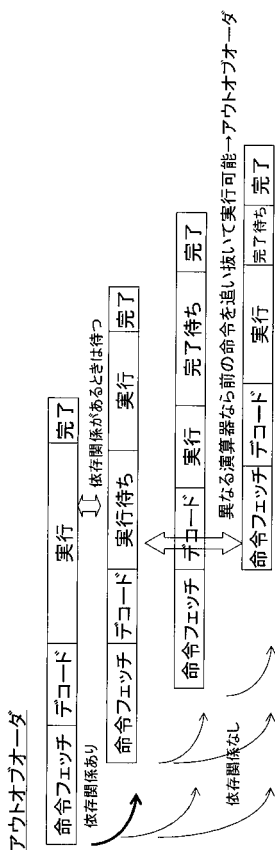
【図7】



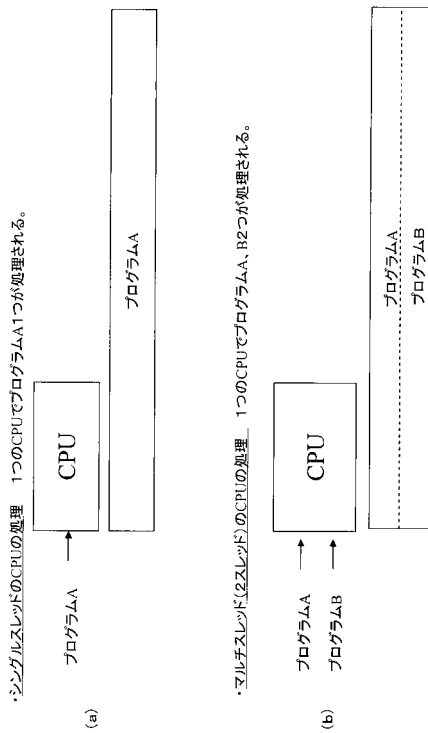
【図8】



【図9】



【図10】

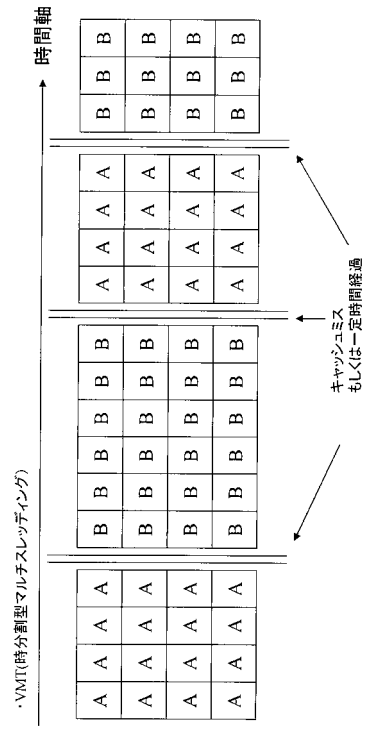


【 図 1 1 】

・SMT(同時マルチスレッディング)

A	A	A	A	B	A
A	B	A	B	B	A
B	B	A	B	A	A
B	A	B	B	A	B

【 図 1 2 】



フロントページの続き

- (56)参考文献 特開2003-196086(JP,A)
特開2003-241961(JP,A)
特開2002-287957(JP,A)
特開2002-163121(JP,A)

(58)調査した分野(Int.Cl., DB名)

G06F 9/38
G06F 9/46 - 9/54