

(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2006-146921

(P2006-146921A)

(43) 公開日 平成18年6月8日(2006.6.8)

(51) Int. Cl.

G06F 11/28 (2006.01)

F I

G06F 11/28 310B

テーマコード(参考)

5B042

審査請求 有 請求項の数 30 O L (全 24 頁)

(21) 出願番号 特願2005-332188 (P2005-332188)
 (22) 出願日 平成17年11月16日(2005.11.16)
 (31) 優先権主張番号 10/989,750
 (32) 優先日 平成16年11月16日(2004.11.16)
 (33) 優先権主張国 米国 (US)

(特許庁注: 以下のものは登録商標)

1. フロッピー

(71) 出願人 395015319
 株式会社ソニー・コンピュータエンタテインメント
 東京都港区南青山二丁目6番21号
 (74) 代理人 100099324
 弁理士 鈴木 正剛
 (74) 代理人 100108604
 弁理士 村松 義人
 (74) 代理人 100111615
 弁理士 佐野 良太
 (72) 発明者 平岡 大輔
 東京都港区南青山二丁目6番21号 株式会社ソニー・コンピュータエンタテインメント内

最終頁に続く

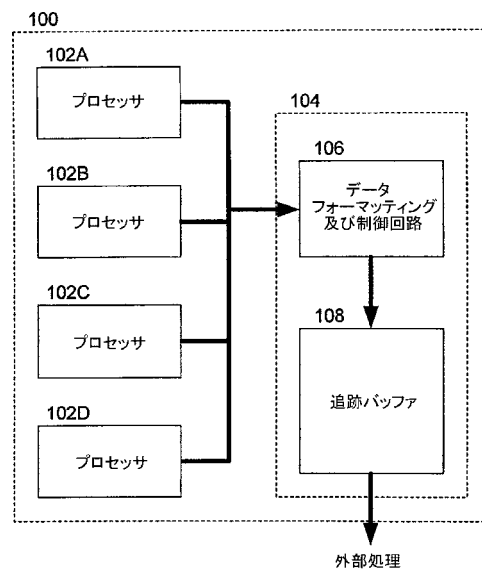
(54) 【発明の名称】 追跡バッファでブックマークを使用するための方法及びシステム

(57) 【要約】

【課題】 マルチプロセッサシステムなどの高速プロセッサのプロセッサとソフトウェアのパフォーマンス情報を取得するための新たな方法及び装置を提供する。

【解決手段】 少なくとも1つのプロセッサから受信したプログラムステータスデータから追跡データを生成する方法及び装置100が提供され、この少なくとも1つのプロセッサは、プログラムカウンタから取得したアドレスによるアドレッシング命令によりソフトウェアを実行するように動作できる。また、追跡バッファに追跡データを記録する方法及び装置100が提供され、この追跡データは、(i)各々のプログラム実行イベントの総カウント数、(ii)任意の時点でプログラムカウンタから取得したアドレス、及び(iii)プログラムのパフォーマンス情報を含むブックマークデータ、が含まれる。

【選択図】 図1



【特許請求の範囲】

【請求項 1】

各々がプログラムカウンタから取得したアドレスによるアドレッシング命令によってソフトウェアを実行するように動作可能な複数のプロセッサと、

前記プロセッサから受信したプログラムステータスデータから追跡データを生成し、かつ記録するように動作可能なパフォーマンスモニタ回路とを含み、前記追跡データは、(i) 各々のプログラム実行イベントの総カウント数、(i i) 任意の時点で前記プログラムカウンタから取得したアドレス、及び(i i i) プログラムのパフォーマンス情報を含むブックマークデータ、とを含む、システム。

【請求項 2】

前記パフォーマンスモニタ回路は前記追跡データを記録する追跡バッファを備えている、請求項 1 記載のシステム。

【請求項 3】

前記プロセッサ及び前記パフォーマンスモニタ回路は共通の集積回路内に配置されている、請求項 1 又は 2 記載のシステム。

【請求項 4】

前記集積回路はデータポートを備えており、前記追跡データは前記データポートを通じて前記追跡バッファから抽出されうる、請求項 3 記載のシステム。

【請求項 5】

前記追跡データは一定時間毎に生成され、かつ記録される、請求項 1 ~ 4 のいずれか 1 項に記載のシステム。

【請求項 6】

各々のプログラム実行イベントの前記総カウント数及び前記アドレスの少なくとも一方は、一定時間毎に生成され、かつ記録される、請求項 5 記載のシステム。

【請求項 7】

前記プログラム実行イベントは、キャッシュミスイベント、特定のソフトウェア命令の実行、及びプログラムのストール、のうちの少なくとも 1 つを含む、

前記特定のソフトウェア命令はロード命令である、

前記プログラムのストールは分岐ミスのストール及びダイレクトメモリアクセスのストールのうちの少なくとも 1 つを含む、あるいは、

前記ブックマークデータは少なくとも 1 つのプログラムスレッド情報を含む、の条件のうち少なくともいずれか一つが満たされる、請求項 1 ~ 6 のいずれか 1 項に記載のシステム。

【請求項 8】

前記パフォーマンスモニタ回路は、

前記プログラム実行イベントの発生を表示する前記プロセッサから各々の信号を受信するように動作可能な少なくとも 1 つのマルチプレクサと、

前記マルチプレクサから出力を受信し、かつ前記各々のプログラム実行イベントの総カウント数を生成するように動作可能な少なくとも 1 つのデジタルカウンタ、を含む、請求項 1 ~ 7 のいずれか 1 項に記載のシステム。

【請求項 9】

前記マルチプレクサと前記デジタルカウンタとのうちの少なくとも一方は、各プロセッサに対応付けられている、請求項 8 記載のシステム。

【請求項 10】

前記パフォーマンスモニタ回路は前記総カウント数、前記アドレス、及び前記ブックマークデータを、各種類の追跡データに割当てられた順序に従い各々のストリングにパッキングするように動作でき、前記ブックマークデータの前記順序はその他の種類の追跡データよりも優先される、請求項 1 記載のシステム。

【請求項 11】

プログラムカウンタから取得したアドレスによるアドレッシング命令によりソフトウェ

10

20

30

40

50

アを実行するように動作可能な少なくとも1つのプロセッサと、

前記少なくとも1つのプロセッサから受信したプログラムステータスデータから追跡データを生成し、かつ記録するように動作可能なパフォーマンスモニタ回路を含み、前記追跡データは、(i)各々のプログラム実行イベントの総カウント数、(ii)任意の時点で前記プログラムカウンタから取得したアドレス、及び(iii)プログラムのパフォーマンス情報を含むブックマークデータを含む、システム。

【請求項12】

各々が第1プログラムカウンタから取得したアドレスによるアドレッシング命令によりソフトウェアを実行するように動作可能な複数のプロセッサと、

第2プログラムカウンタから取得したアドレスによるアドレッシング命令によりソフトウェアを実行するように動作可能なメインプロセッシングユニットと、

前記プロセッサから受信したプログラムステータスデータから追跡データを生成かつ記録するように動作可能なパフォーマンスモニタ回路とを含み、前記追跡データは、(i)各々のプログラム実行イベントの総カウント数、(ii)任意の時点で前記第1及び第2プログラムカウンタの少なくとも1つから取得したアドレス、及び(iii)プログラムのパフォーマンス情報を含むブックマークデータを含み、

前記メインプロセッシングユニットは更に管理ソフトウェアプログラムを実行し、1つ以上の条件が合う場合に、前記総カウント数、前記アドレス、及び前記ブックマークデータのうちの少なくとも1つを前記追跡データに導入させるように動作することができる、システム。

【請求項13】

前記1つ以上の条件には、(i) `b c l r (l) w / t a k e n`、(ii) `b c c t r (l) w / t a k e n`、(iii) `r f i d`、のうちの少なくとも1つのアセンブリ言語命令の発生が含まれる、請求項1～9、11、12のいずれか1項に記載のシステム。

【請求項14】

前記パフォーマンスモニタ回路は前記総カウント数、前記アドレス、及び前記ブックマークデータを、長さがNビットの各々のストリングにパッキングし、それらを追跡バッファに記録するように動作可能な、請求項1～9、11、12のいずれか1項に記載のシステム。

【請求項15】

前記パフォーマンスモニタ回路は前記総カウント数の少なくとも一部と前記アドレスの少なくとも一部とを前記同じストリングにパッキングするように動作可能な請求項1～9、11、12のいずれか1項に記載のシステム。

【請求項16】

前記パフォーマンスモニタ回路は前記ブックマークデータを前記総カウント数と前記アドレスを含まないストリングにパッキングするよう動作可能な、請求項1～9、11、12のいずれか1項に記載のシステム。

【請求項17】

前記パフォーマンスモニタ回路は1つ以上の前記プロセッサからの前記アドレスを前記同じストリングにパッキングするように動作可能な、請求項1～9、11、12のいずれか1項に記載のシステム。

【請求項18】

前記パフォーマンスモニタ回路は前記メインプロセッシングユニットからの前記アドレスを、前記総カウント数、1つ以上のプロセッサからの前記アドレス、前記メインプロセッシングユニットからの前記アドレス、及び前記ブックマークデータから分離されたストリングにパッキングするように動作可能な、請求項12記載のシステム。

【請求項19】

プロセッサのパフォーマンスを監視する装置であって、

少なくとも1つのプロセッサから受信したプログラムステータスデータから追跡データを生成するように動作可能なフォーマッティング回路を含み、前記少なくとも1つのプロ

10

20

30

40

50

セッサはプログラムカウンタから取得したアドレスによるアドレッシング命令によりソフトウェアを実行するように動作可能なものであって、

前記追跡データを記録するように動作可能な追跡バッファと、

前記追跡データの追跡バッファへの書込みと、前記追跡データの追跡バッファからの読出しを管理するように動作可能なコントローラとを含み、

前記追跡データは、(i) 各々のプログラム実行イベントの総カウント数、(i i) 任意の時点で前記プログラムカウンタから取得したアドレス、(i i i) プログラムのパフォーマンス情報を含むブックマークデータ、を含む、装置。

【請求項 20】

前記コントローラは、

前記追跡データが一定時間毎に生成され、かつ記録される、

各々のプログラム実行イベントの前記総カウント数が一定時間毎に生成され記録される

10

、
前記プログラムカウンタから取得した前記アドレスが一定時間毎に記録される、又は、
前記ブックマークデータが一定時間毎には生成及び記録されない、の条件のうち少なくとも一つが満たされるように動作可能である、請求項 19 記載の装置。

【請求項 21】

前記コントローラは、タイミング信号に応じて前記追跡データを生成し、記録するように動作可能な書込み制御回路を含む、請求項 19 又は 20 記載の装置。

【請求項 22】

前記タイミング信号は、前記追跡データの少なくとも一部が生成され、かつ記録されるタイムインターバルを制御するようにプログラムできる、請求項 21 記載の装置。

20

【請求項 23】

前記コントローラはタイミング信号にตอบสนองして前記追跡バッファから前記追跡データを抽出させ、前記プロセッサのパフォーマンスを監視する装置から出力させるように動作可能な読出し制御回路を含む、請求項 19 又は 20 記載の装置。

【請求項 24】

前記タイミング信号は前記追跡データが前記追跡バッファから抽出されるタイムインターバルを制御するようにプログラムできる、請求項 19 ~ 21 のいずれか 1 項に記載の装置。

30

【請求項 25】

前記コントローラは、

前記追跡データを書込みタイミング信号にตอบสนองして生成させ、かつ記録させるように動作可能な書込み制御回路と、

前記追跡データを読出しタイミング信号にตอบสนองして前記追跡バッファから抽出させ、前記パフォーマンスモニタから出力させるように動作可能な読出し制御回路とを含み、

前記コントローラは前記追跡バッファ内の追跡データの総数をモニタリングし、前記書込み制御回路を操作して追跡データが上書きされないようにするよう動作可能な、請求項 19 記載の装置。

【請求項 26】

少なくとも 1 つのプロセッサから受信したプログラムステータスデータから追跡データを生成するステップを含み、前記少なくとも 1 つのプロセッサはプログラムカウンタから取得したアドレスによるアドレッシング命令によりソフトウェアを実行するように動作可能な方法であって、かつ、

追跡バッファに前記追跡データを記録するステップを含み、

前記追跡データは、(i) 各々のプログラム実行イベントの総カウント数、(i i) 任意の時点で前記プログラムカウンタから取得した前記アドレス、(i i i) プログラムのパフォーマンス情報を含むブックマークデータ、を含む、方法。

【請求項 27】

前記プログラム実行イベントは、キャッシュミスイベント、特定のソフトウェア命令の

40

50

実行、及びプログラムのストール、のうちの少なくとも1つを含む、

前記特定のソフトウェア命令はロード命令である、

前記プログラムのストールの少なくとも1つは分岐ミスのストール及びダイレクトメモリアクセスのストールのうちの少なくとも1つを含む、又は、

前記ブックマークデータはプログラムスレッド情報のうちの少なくとも1つを含む、の条件のうち少なくとも一つを満たす、請求項26記載の方法。

【請求項28】

前記プログラム実行イベントの前記発生を表示する前記プロセッサから各々の信号を受信するステップと、

前記総カウント数を生成するために前記各々のプログラム実行イベントをカウントするステップ、とを更に含む、請求項26又は27記載の方法。 10

【請求項29】

コンピュータを、

プログラムカウンタから取得したアドレスによるアドレッシング命令によりソフトウェアを実行できるように動作可能な少なくとも1つのプロセッサから受信したプログラムステータスデータから追跡データを生成する手段、及び、

追跡バッファに前記追跡データを記録する手段、として機能させるためのプログラムが記録されたコンピュータ読み取り可能な記録媒体であって、

前記追跡データは、(i)各々のプログラム実行イベントの総カウント数、(ii)任意の時点で前記プログラムカウンタから取得したアドレス、及び(iii)プログラムのパフォーマンス情報を含むブックマークデータ、を含むものである、 20

コンピュータ読み取り可能な記録媒体。

【請求項30】

コンピュータを、

プログラムカウンタから取得したアドレスによるアドレッシング命令によりソフトウェアを実行できるように動作可能な少なくとも1つのプロセッサから受信したプログラムステータスデータから追跡データを生成する手段、及び、

追跡バッファに前記追跡データを記録する手段、として機能させるためのプログラムであって、

前記追跡データは、(i)各々のプログラム実行イベントの総カウント数、(ii)任意の時点で前記プログラムカウンタから取得したアドレス、及び(iii)プログラムのパフォーマンス情報を含むブックマークデータ、を含むものである、 30

プログラム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、トレースバッファとも称される、追跡バッファに含まれるどの種類のプロセッサ情報も挿入することができる技術に関する。

【背景技術】

【0002】

最先端のコンピュータアプリケーションは、リアルタイムのマルチメディア機能を伴っているために、近年はデータスループットが高くより高速なコンピュータが常に望まれている。グラフィックアプリケーションは処理システムへの要求が大きいアプリケーションの1つであり、その理由は、グラフィックアプリケーションが所望の映像的な処理結果を実現するために、比較的短時間で非常に多くのデータアクセス、データの演算処理、及びデータ操作を要求するからである。これらのアプリケーションは、1秒間に数千メガビットのデータ処理等の非常に高速な処理速度を要求する。シングルプロセッサを採用し、高速の処理速度を実現している処理システムもある一方で、マルチプロセッサアーキテクチャを利用して実装されている処理システムもある。マルチプロセッサシステムでは、複数のプロセッサが並列に(あるいは少なくとも協調して)動作し、所望の処理結果を実現す 40 50

ることができる。

【発明の開示】

【発明が解決しようとする課題】

【0003】

しかし、最新の処理システムから、具体的にはマルチプロセッサシステムからプロセッサのパフォーマンス及び/又はソフトウェアのパフォーマンスデータを取得するのは困難である。処理システムからパフォーマンスデータを取得するための手法の幾つかの詳細は米国特許出願第10/881,971(2004年6月30日出願)において知ることができ、該出願は本明細書中に参考のために組み込まれている。

【0004】

プロセッサパフォーマンスとソフトウェアパフォーマンスのデータを取得する従来技術の手法では、プロセッサが配置される半導体パッケージ上の非常に多くの信号ピン上の信号伝達をモニタリングすることが必要とされる。一般に、そのようなピンは約10から20あり、キャッシュミス、データのロード、分岐の発生、アドレス情報、(アドレス追跡)などの信号を運んでいる。この手法は多くのプロセッサチップにとっては実用的であるが、超高速のインターフェース、超高速の処理スピードなどを有す最新のプロセッサには実用的でない。実際に、データが非常に高速に出現し、また、データ量が非常に多く、専用のピンにデータを展開することは困難である。高速のデータ転送率とデータ量に対応するようにピンの数を増やすことも可能であるが、評価目的のためだけに使用される外部ピンを多数設けるには多くの費用がかかる。

10

20

【0005】

従って、マルチプロセッサシステムなどの高速プロセッサのプロセッサとソフトウェアのパフォーマンス情報を取得するための新たな方法と装置が求められている。評価プロセスを向上するために用いられうる、その他の種類のパフォーマンス情報(従来技術のキャッシュミス、データのロード、分岐の発生、及びアドレス情報以外に)を取得することもまた望ましい。

【課題を解決するための手段】

【0006】

本発明の一形態によれば、プロセッサに対してオンチップに、又はチップ外に配置された追跡バッファに含まれる(プロセッサ及び/又はソフトウェアのパフォーマンス情報など)、どの種類の情報も挿入できる方法及び装置が提供される。

30

【0007】

本発明の一形態では、キャッシュミス、データのロード、分岐の発生、分岐ミスのストール、DMAストールなどの信号の発生を蓄積する多くのオンチップカウンタを採用している。その蓄積した総数はオンチップの又はチップ外の追跡バッファに記録され、後でダウンロードと評価が行われる。相異なる信号毎に単一のカウンタが用いられ得るように、カウンタと信号の間に多くのマルチプレクサを配置し得る。更に、ブックマークレジスタは実際にプロセッサから(スレッドナンバーなどの)どのデータでも受信することができ、追跡バッファへ送信することができる。スレッドナンバーは追跡バッファに記録されたデータの一部が取得され記録された場合に、“プログラムがどこにあったか(where the program was)”を表示するために用いられることもできる。その後、ブックマークデータを含むプロセッサ情報が追跡バッファから読み出され解析されることもできる。

40

【0008】

本発明の1つ以上の形態によれば、少なくとも1つのプロセッサから受信したプログラムステータスデータから追跡データを生成する方法とシステムが提供される。この少なくとも1つのプロセッサは、プログラムカウンタから取得したアドレスによるアドレッシング命令によってソフトウェアを実行するように動作できる。また本発明の1つ以上の態様によれば、追跡データを追跡バッファに記録する方法とシステムも提供され、追跡データには、(i)各々のプログラム実行イベントの総カウント数、(ii)任意の時点でプログラムカウンタから取得したアドレス(あるいは、プログラムカウンタから少なくとも2

50

回以上の時点で取得したアドレス)、及び(iii)プログラムのパフォーマンス情報を含むブックマークデータ、が含まれる。

【0009】

本発明の1つ以上の態様によれば、複数のプロセッサ(プロセッサの各々は第1プログラムカウンタから取得したアドレスによるアドレッシング命令によってソフトウェアを実行するように動作できる)、メインプロセッシングユニット(第2プログラムカウンタから取得したアドレスによるアドレッシング命令によってソフトウェアを実行するように動作できる)、及びパフォーマンスモニタ回路(プロセッサから受信したプログラムステータスデータから追跡データを生成し記録するように動作できる)、を含むシステムが提供される。このシステムにおいて、追跡データに、(i)各々のプログラム実行イベントの総カウント数、(ii)第1及び第2プログラムカウンタのうち少なくとも一つから任意の時点で取得したアドレス、(iii)プログラムのパフォーマンス情報を含むブックマークデータ、が含まれるシステムが提供される。

10

メインプロセッシングユニットは更に管理ソフトウェアプログラムを実行し、これにより、1つ以上の条件が合う場合に、総カウント数、アドレス、ブックマークデータのうちの少なくとも一つを追跡データに導入させるように動作することができる。

【0010】

本発明の1つ以上の態様によれば、フォーマッティング回路と、追跡バッファとコントローラとを備えた、プロセッサのパフォーマンスを監視する(モニタする)装置が提供される。フォーマッティング回路は、少なくとも一つのプロセッサから受信したプログラムステータスデータから追跡データを生成するように動作でき、この少なくとも一つのプロセッサは、プログラムカウンタから取得したアドレスによるアドレッシング命令によりソフトウェアを実行するように動作できる。追跡バッファは追跡データを記録するように動作できる。コントローラは追跡データの追跡バッファへの書込みと、追跡データの追跡バッファからの読出しを管理するように動作できる。追跡データには、(i)各々のプログラム実行イベントの総カウント数、(ii)任意の時点でプログラムカウンタから取得したアドレス、及び(iii)プログラムのパフォーマンス情報を含むブックマークデータ、が含まれる。

20

【0011】

添付の図面を参照しつつ、ここに記載する本発明の説明を読めば、他の態様、機能および利点などは当業者に自明となるであろう。

30

【0012】

本発明の様々な態様を例示するために、図面には現在のところ好まれる形状を示しているが、本発明、以下に開示される配置そのものや手段に限定されるものではないことを理解されたい。

【発明を実施するための最良の形態】

【0013】

図面において、同一の要素には同じ参照符号を付して参照する。図1に、本発明の1つ以上の特徴を実行するために適する処理システム100を例示した。簡潔を期すと共に明確になるように、図1のブロック図を装置100を例示するものとして本明細書において参照かつ記載する。しかし、この記載は同等の効力を有する方法のさまざまな態様に容易に適用できることを理解されたい。処理システム100は好ましくは複数のプロセッサ102-Dと、バス配線、専用の信号線及び/又は双方の組合せであってよい、1つ以上の配線により相互接続されたパフォーマンスモニタ104を備えている。

40

【0014】

プロセッサ102-Dは好ましくは並列に(あるいは少なくとも協調して)動作し、所望の処理結果を実現することができる。一例では4台のプロセッサ102が例示されているが、本発明の趣旨及び範囲から逸脱せずに、いずれの数のプロセッサでも利用し得る。プロセッサ102A-Dは同様の構造又は別の構造によって実装されることができる。例えば、プロセッサ102は標準マイクロプロセッサ、分散型マイクロプロセッサなどを含

50

む、ソフトウェア及び/又はファームウェアを実行できる、どの周知のマイクロプロセッサを用いて実装されることができ。一例としては、プロセッサ102は、データ(ピクセルデータなどであり、グレースケール情報、色情報、テキスチャデータ、ポリゴナル情報、ビデオフレーム情報などを含む)の要求と操作が可能、グラフィックスプロセッサであることができる。

【0015】

本発明の1つ以上の実施形態によれば、プロセッサ102Aなどのプロセッサ102の1つは、好ましくはメインプロセッサの役割を果たす。例えば、メインプロセッサ102Aは、その他のプロセッサ102上で実行されているアプリケーションプログラムやデータをモニタリングし、メインプロセッサ102Aが処理の変更を呼び出しできるようにすることもできる。このような処理の変更には、プロセッサ102などの間にデータを割り当てるために、1つのプロセッサ102から別のプロセッサ102へタスクを移動することが含まれる。

10

【0016】

好ましくは、メインプロセッサ102Aは、追跡データを生成して記録するように、パフォーマンスモニタ104と合わせて動作する管理プログラムを実行するように動作することもできる。本発明の別の実施形態では、アプリケーションプログラムはパフォーマンスデータを追跡するために、パフォーマンスモニタ回路104とともに動作することもできる。図示されているように、パフォーマンスモニタ回路104には、データフォーマッティング及び制御回路106と追跡バッファ108とが含まれる。データフォーマッティング及び制御回路106は好ましくは、プロセッサ102から受信したプログラムステータスデータから追跡データを生成し、該データを追跡バッファ108へ記録するように動作できる。追跡データには、キャッシュミスイベント、特定のソフトウェア命令の実行(ロード命令など)、及びプログラムのストール(分岐ミスのストール、ダイレクトメモリアクセスのストールなど)などの、各々のプログラム実行イベントの総カウント数を含めることもできる。追跡データにはまた、プログラム実行中の様々な時点で、プロセッサのプログラムカウンタから取得したプログラムアドレスも含めることもできる。更にまた、追跡データには、プログラムスレッドID、フレームID(画像処理用などの)などのプログラムパフォーマンス情報を有するブックマークデータ、又は、総カウント数やプログラムアドレス以外のその他のいずれのプログラム情報も有するブックマークデータを含め

20

30

【0017】

以下に議論しているように、ブックマークデータはブックマークデータを生成させるトリガイベントに応じて、不定期に生成され記録されるが、好ましくは、データフォーマッティング及び制御回路106は追跡データを生成し、その追跡データを一定時間毎に追跡バッファ108へ記録させる。十分な量の追跡データが追跡バッファ108に記録されると、そのようなデータは(例えばデータポートなどを通して)追跡バッファ108から読み出され、外部処理を行うこともできる。そのような外部処理としては、追跡データをソートする、つまり分類することが挙げられ、1つ以上のソフトウェアプログラムの実行時に、処理システム100がいかに上手く又は下手にプログラムを実行したかが判断される。そのような解析はプログラム実行イベント、プログラムアドレス、及び、ある種のイベントがプログラム実行中に発生した場合に、“プログラムがどこにあったか”を判断することができるブックマークデータ、に基づいて行われるのが便利である。これにより解析処理を大幅に改善することができ、また、処理システム100の処理パフォーマンスをより一層理解することができる。

40

【0018】

図2に、図1のマルチプロセッシングシステム100の更に詳しいブロック図が示されている。好ましくは、データフォーマッティング及び制御回路106は1つ以上のマルチプロセッサ110A-Dと、1つ以上のデジタルカウンタ112A-Dを備えている。マルチプロセッサ110A-Dは各々のプロセッサ102A-Dから信号を受信するように動作

50

できる。このような信号はプログラム実行イベントの発生を表示している。例えば、マルチプレクサ110Aに入る信号は、キャッシュミス、ロード命令の実行、分岐ミスのストール、分岐アドレス発生イベント(branch address taken events)、DMAのストールなどの発生を各々表わし得る。同様に、その他のマルチプレクサ110B-Dはそのような信号をその他のプロセッサ102B-Dから受信する。上記のイベントは論議を目的として提供されているが、システム100の1つのプロセッサ102、バスシステム、I/Oインターフェース、メモリインターフェース、メモリアクセスコントローラ(ダイレクトメモリアクセスコントローラなど)などのいずれのイベント/条件がデータフォーマッティング及び制御回路106へ入力されうことは理解されよう。信号多重化技術で周知のように、マルチプレクサ110A-Dは、特定の時間における入力信号の各々を表わす信号を出力するように制御されることもできる。

【0019】

マルチプレクサ110A-Dからの各々の出力は各々のデジタルカウンタ112A-Dへの入力であり、特定のプログラム実行イベントが発生するたびにインクリメントされてその値が増加する。従って、各々のカウンタ112A-Dからの値は、特定のタイムインターバルの終了時に追跡バッファ108に記録され、後で解析が行われう。一例として、各々のカウンタ112A-Dは最大255カウントを与えるように動作できる8ビットカウンタであることができる。従って、プログラム実行イベントをカウントする間の適切なタイムインターバルは約200プロセッササイクルであり、各サイクルは4GHz速度で起こりう。しかし、カウンタ112は特定のアプリケーションによって決定されうように、16ビット以上のキャパシティを利用して実装されることもできる。

【0020】

なお、ここでは、説明のために、各プロセッサ102に対し、1つのマルチプレクサ110と1つのカウンタ112が例示している。しかし、実際には、いずれの数のマルチプレクサ110及び/又はカウンタ112が本発明の趣旨と範囲から逸脱することなく採用されることもできる。好ましい実施形態では、2つの64ビットの信号グループが各アイランドから取得されうるように、各カウンタ112に対し、64対1マルチプレクサ110が存在する。いずれのサイズのいずれの数のカウンタを採用することもできる。

【0021】

追跡バッファ108に記録される追跡データはグループ又はブロック114にまとめられることもでき、各ブロック114は特定のタイムインターバル中に取得したデータを表す。他の実施形態では、データのブロック114は、キャッシュミス、ロード命令の実行などの、各々の種類の追跡データを表わすこともでき、追跡バッファ108内のデータの特定のまとめりは、特定のアプリケーションに基づいて容易に調整されう。

【0022】

図3はデータフォーマッティング及び制御回路106の他の構成のブロック図であり、ブックマークデータは追跡データに導入され、追跡バッファ108に記録されることもできる。理論上は、ブックマークデータは特殊用途のレジスタ116Aにより導入されることもできる。該レジスタ116Aによって、そのようなデータの導入を正当化する特定のイベント発生時にはいつでもブックマークデータが受信される。例えば、スレッドIDは特殊用途のレジスタ116AにスレッドIDを書込み、そのようなデータと総カウント数とを更なるマルチプレクサ118Aによって多重化することにより、追跡データに導入することもできる。特定用途のレジスタ116Aはパフォーマンスモニタ104内に配置することもでき、又は1つ以上のプロセッサ102A-D内に配置することもできる。

【0023】

好ましい実施形態では、プロセッサ102Aはメインプロセッサの役割を果たし、その中に特殊用途のレジスタ116Aを備えている。好ましくは、1つ以上の条件が合う場合には、メインプロセッサ102Aで実行中の管理プログラムは、ブックマークデータを特殊用途のレジスタ116Aへ書込みさせることができる。管理プログラムが特殊用途のレジスタ116Aへブックマークデータを開始できる特定の条件は非常に多数存在するが、

そのような条件の例としては、各プログラムスレッドの開始（又は終了）、画像処理のにおける所定のフレームの開始又は終了、などが挙げられる。

【0024】

総カウント数やブックマークデータに加え、プロセッサ102のプログラムカウンタから取得したプログラムアドレスも追跡データ間に含まれ、かつ、追跡バッファ108に記録されるのが好ましい。好ましくは、プロセッサ102のプログラムアドレスがキャプチャされ、特定の条件に応じて追跡データに導入される。例えば、プロセッサ102がメインプロセッサとして機能している場合、プログラムアドレスを捕獲させる条件は、特定の種類のプログラム命令（分岐命令など）の実行としてもよい。ある種のアセンブリ言語プログラム命令セットでは、このような分岐命令として、`bclr(l)w/taken`、`bcctr(l)w/taken`、及び/又は`rfid`イベントを含むことができる。

【0025】

メインプロセッサ102Aに関連するアドレスキャプチャイベントとは対照的に、その他のプロセッサ102Bに関連するアドレスキャプチャイベントは、特定の時間間隔で行われるものであり得る。例えば、アドレスキャプチャイベントは16プロセッシングサイクルの倍数でありうる。プログラム実行イベントの総カウント数に関連するインターバルが約200プロセッシングサイクルで発生すると仮定すると、その他のプロセッサ102B-Dのプログラムアドレスにおける速度が極めて高速であることは明らかである。総カウント数とプロセッサ102のプログラムカウンタから取得したアドレスが追跡データに導入される周波数を比較すると、ブックマークデータは具体的にはブックマークデータがスレッドIDである場合に、大幅に低い周波数で導入されよう。

【0026】

本発明の具体的な実施形態によれば、プロセッサ102B-Dに関連するアドレスの長さは、メインプロセッサ102Aに関連するアドレスの長さよりも短い。このことはプロセッサ102B-Dによってアクセス可能なアドレススペースが、メインプロセッサ102Aによってアクセス可能なアドレススペースよりも小さい場合に起こりうる。

【0027】

図4を参照すると、パフォーマンスモニタ回路104、具体的にはデータフォーマッティング及び制御回路106は、好ましくは長さがNビットの各々のストリング、例えばビットやビット列の中に追跡データをパッキングし、該データを追跡バッファ108に記録するように動作できる。好ましい実施形態では、ストリングは長さが128ビットである。しかし、当業者であればいずれの長さのストリングが本発明の精神と範囲から逸脱することなく採用されうることは理解されよう。Oとラベル付けされている、第1ビットのストリングはオーバーフロー条件を示す。Rとラベル付けされている、次の2ビットのストリングはレコードタイプを示す。thとラベル付けられている次のビットのストリングはプログラム可能ビットであり、このビット値に基づいて捕獲イベントを開始又は中止する。cntとラベル付けされている次の3ビットのストリングは取得イベントを表わすように用いられうる。resとラベル付けされている次の57ビットのストリングはリザーブされる。メインプロセッサ102Aのプログラムカウンタから取得されるプログラムアドレスは、残りの64ビットのストリングに記録される。

【0028】

図5を参照すると、その他のプロセッサ102B-Dのプログラムカウンタから捕獲された16ビットのプログラムアドレスは、`add`とラベル付けされた16ビットロケーションに記録されうる。図5には、`ct`とラベル付けされた多数の8ビットのロケーションも示されており、各々のカウンタ112のサイズが8ビットであると仮定すると、このような8ビットの場所は、プログラム実行イベントの総カウント数を記録するために用いられうる。図6を参照すると、16ビットのプログラムアドレスとその他の16ビットの総カウント数は各々のストリングにパッキングされ、追跡バッファ108に記録されうる。更にまた、図7に例示しているように、特定のストリングは長さが8ビット又は16ビットなどの、総カウント数だけを含むこともできる。図8を参照すると、ブックマークデー

タは62ビットまでの長さであり、Tとラベル付けされた2ビットのセグメントがブックマークデータに対する開始/終了のレコードトリガーを示すために用いられる。

【0029】

上述のように、データフォーマッティング及び制御回路106は、好ましくは各々のNビットのストリングに追跡データをパッキングし、該データを追跡バッファ108に、所定のタイムインターバルで、又は特定の条件にตอบสนองして、記録するように動作することができる。これに関連して、図9-10を参照する。図9は、メインプロセッサ102A上に行っている管理プログラムにตอบสนองして、追跡バッファ108に記録される追跡データの一例を例示した概念図である。図10はパフォーマンスモニタ104の具体的な実装品を例示したブロック図である。

10

【0030】

パフォーマンスモニタ104、具体的にはデータフォーマッティング及び制御回路106は、好ましくはタイマー120、書込み制御回路122、タイマー124、読出し制御回路126、及びマルチプレクサ/ラッチ回路128を備えている。好ましくは、書込み制御回路122は、タイマー120からのタイミング信号にตอบสนองして、また、プロセッサ102A-Dのうちの一つにより及び/又はブックマークデータが存在することにより、分岐命令が実行されるといった、特定のイベントの発生にตอบสนองして、追跡データが生成され、記録されるように動作できる。タイマー120は好ましくは、上述しているような、あるプログラムアドレスキャプチャに対しては16プロセッシングサイクル、及びプログラム実行イベントの総カウント数に対しては200プロセッシングサイクルなど、特定のタイムインターバルで追跡データが生成され記録される必要があることを示す信号を生成するように動作できる。好ましくは、ブックマークデータが最優先され、また、追跡データが生成され、追跡バッファ108に書込まれる必要があることを示す、どのその他の信号伝達イベントよりも早くサービスされるように、書込み制御回路122への信号伝達が優先されるのが望ましい。好ましくは、読出し制御回路は、タイマー124からの読出しタイミング信号にตอบสนองして、追跡バッファ108から追跡データが抽出され、パフォーマンスモニタ104から出力されるように動作可能である。

20

【0031】

図9に示すように、書込み制御回路122によって、タイムインターバル t_0 中にブックマークデータが追跡バッファ108の中に書込まれ、タイムインターバル t_1 中に16ビットのプログラムアドレスと総カウント数が追跡バッファ108に記録され、タイムインターバル t_2 中に64ビットのプログラムアドレスが追跡バッファ108に記録される。好ましくは、追跡データが追跡バッファ108に記録される速度と、追跡データが追跡バッファ108から抽出される速度が制御され、追跡データが追跡バッファ108内で決して上書きされないように、管理プログラムにตอบสนองしてタイマー120と124をプログラムすることができる。好ましくは、追跡バッファ108は先入れ先出し(FIFO)デバイスとして実装される。

30

【0032】

上述しているように、好ましくは、装置100はデータポートを備えており、該データポートを通じて追跡データが追跡バッファから移動されて、外部に記録され解析されることもできる。このデータポートは追跡バッファ108と外部メモリ(ダブルデータライト(DDR)メモリなど)との間に結合されているサウスブリッジを備える。好ましくは、DDR上のデータサイズは変更可能であり、変更されたサイズに達すると、サウスブリッジは抽出した追跡データのDDRへの書込みをストップする。これに関連して、好ましくは、読出し制御回路126が追跡バッファ108から追跡データの抽出をストップするように、サウスブリッジは好ましくは割り込みを開始し、この割り込みはパフォーマンスモニタ104へと返送される。この機能がサポートされない場合、追跡データがDDRでラップアラウンド、つまり、最新の追跡データが最初に書き込まれた追跡データを上書きすることになり、重要な追跡データが損失される。別の構造では、データをGB速度で、YDRAMなどの特殊用途のDRAMへ送信できる、専門化したサウスブリッジが用いら

40

50

れてもよい。

【0033】

図11を参照し、かつ、本発明の好ましい実施形態によれば、上述した本発明の態様を実施するのに適した他の形態の構造のコンピューティング装置は、複数のプロセッサ202A-D、関連するローカルメモリ204A-D、及びバス208により相互接続されているメインメモリ（又は共有メモリ）を備えている。一例としては4台のプロセッサ202が例示されているが、本発明の精神と範囲を逸脱することなく、いずれの台数のプロセッサを用いることができる。図示していないが、装置200は、上述したパフォーマンスモニタ104も備えている。

【0034】

プロセッサ202は、システムメモリ206からデータを要求し、所望の結果を得るためにそのデータを操作することができる、従来のいずれの技術を用いて実装されうる。好ましくは、ローカルメモリ204は対応のプロセッサ202と同じチップに配置される。しかし、ハードウェアキャッシュメモリ機能を実装するため、好ましくはオンチップの又はチップ外のハードウェアキャッシュ回路、キャッシュレジスタ、キャッシュメモリコントローラなどが無いことが好ましいために、ローカルメモリ204は好ましくはハードウェアキャッシュメモリではない。他の実施形態では、ローカルメモリ204はキャッシュメモリ及び/又は採用されうる追加のキャッシュメモリであってよい。チップ上のスペースには限りがある場合があるため、ローカルメモリ204はシステムメモリ206よりも相当サイズが小さいことがある。プロセッサ102は、好ましくは、プログラムを実行したりデータを操作するために、データアクセスリクエストを出して、バス208経由でシステムメモリ206からローカルメモリ204にデータ（プログラムデータを含みうる）をコピーする。データアクセスを容易にするためのメカニズムは、好ましくはメモリインターフェース110にあるダイレクトメモリアクセスコントローラ（DMAC）を用いて実装される。

【0035】

以下に本明細書で説明している1つ以上の特徴を実行するのに適した、マルチプロセッサシステムのための好ましいコンピュータアーキテクチャを説明する。1つ以上の実施形態によれば、マルチプロセッサシステムは、ゲームシステム、家庭用端末、PCシステム、サーバーシステム、及びワークステーションなどのメディアリッチアプリケーションを、スタンドアロン処理、及び/又は分散処理するために動作することができる、シングルチップソリューションとして実装されうる。ゲームシステムや家庭用端末などのいくつかのアプリケーションでは、リアルタイムの演算処理は必須である。例えば、リアルタイムの分散ゲームアプリケーションでは、ユーザーにリアルタイムの経験をしていると思わせる程速く、1つ以上のネットワークイメージの復元、3Dコンピュータグラフィック、オーディオ生成、ネットワーク通信、物理的シミュレーション、及び人工知能処理が実行される必要がある。従って、マルチプロセッサシステムの各プロセッサは、短時間で、かつ予測可能時間でタスクを完了する必要がある。

【0036】

このために、また、本コンピュータアーキテクチャによれば、マルチプロセッシングコンピュータシステムの全プロセッサは、共通の演算モジュール（あるいはセル）から構成される。この共通の演算モジュールは、構造が一貫しており、また好ましくは、同じ命令セットアーキテクチャを採用している。マルチプロセッシングコンピュータシステムは、1つ以上のクライアント、サーバー、PC、モバイルコンピュータ、ゲームマシン、PDA、セットトップボックス、電気器具、デジタルテレビ、及びコンピュータプロセッサを使用する他のデバイスから形成されうる。

【0037】

複数のコンピュータシステムもまた、所望に応じてネットワークのメンバーとなりうる。一貫モジュール構造により、マルチプロセッシングコンピュータシステムによるアプリケーション及びデータの効率的な高速処理が可能になる。またネットワークが採用される

10

20

30

40

50

場合は、ネットワーク上にアプリケーション及びデータの高速送信が可能にする。この構造はまた、大きさや処理能力が様々なネットワークのメンバーの構築を単純化し、また、これらのメンバーが処理するアプリケーションの準備を単純化する。

【0038】

図12を参照すると、基本的な処理モジュールはプロセッサエレメント(PE)500である。PE500はI/Oインターフェース502、プロセッシングユニット(PU)504、及び複数のサブプロセッシングユニット508、すなわち、サブプロセッシングユニット508A、サブプロセッシングユニット508B、サブプロセッシングユニット508C、及びサブプロセッシングユニット508Dを備えている。ローカル(あるいは内部)PEバス512は、データ及びアプリケーションを、PU504、サブプロセッシングユニット508、及びメモリアンターフェース511間に送信する。ローカルPEバス512は、例えば従来のアーキテクチャを備えることができ、又は、パケット-スイッチネットワークとして実装されうる。パケットスイッチネットワークとして実装される場合は、更なるハードウェアが必要であるものの、利用可能な帯域幅を増やす。

10

【0039】

PE500はデジタル論理回路を実装するよう様々な方法を用いて構成されうる。しながら、好ましくは、PE500はシリコン基板に相補性金属酸化膜半導体(CMOS: Complementary Metal Oxide Semiconductor)を用いた単一の集積回路として好適に構成される。基板の他の材料には、ガリウムヒ素、ガリウムアルミニウムヒ素、及び、様々なドーパントを採用している他の、いわゆる、III-B化合物を含む。PE500はまた、高速単一磁束量子(RSFQ: Rapid Single-flux-Quantum)論理回路などの超電導材料を用いて実装されうる。

20

【0040】

PE500は高帯域のメモリ接続516を介して、共有(メイン)メモリ514と密接に結合するよう構成できる。好ましくは、メモリ514はダイナミックランダムアクセスメモリ(DRAM: Dynamic Random Access Memory)であるが、メモリ514は例えば、スタティックランダムアクセスメモリ(SRAM: Static Random Access Memory)、磁気ランダムアクセスメモリ(MRAM: Magnetic Random Access Memory)、光メモリ、ホログラフィックメモリなどとして、他の方法を用いて実装されうる。

【0041】

PU504とサブプロセッシングユニット508は好ましくは、それぞれダイレクトメモリアクセス(DMA)の機能を備えたメモリフローコントローラ(MFC: Memory Flow Controller)と結合されており、該コントローラはメモリアンターフェース511と共に、PE500のDRAM514とサブプロセッシングユニット508、PU504間のデータ転送を促進する。DMAC及び/又はメモリアンターフェース511は、サブプロセッシングユニット508及びPU504に一体化して、別個に配置されうる。更に、DMACの機能及び/又はメモリアンターフェース511の機能は、1つ以上の(好ましくは全ての)サブプロセッシングユニット508及びPU504に統合することができる。また、DRAM514はPE500と統合されていても、別個に配置されていてもよいことに留意されたい。例えば、DRAM514は、実例で示しているように、チップ外に配置しても、あるいは一体化してオンチップ配置としてもよい。

30

40

【0042】

PU504はデータ及びアプリケーションをスタンドアロン処理できる標準プロセッサなどでありうる。作動時、PU504は、好ましくは、サブプロセッシングユニットによるデータ及びアプリケーション処理をスケジューリングし、調整を行う。サブプロセッシングユニットは好ましくは、単一命令複数データ(SIMD: Single Instruction Multiple Data)プロセッサである。PU504の管理下、サブプロセッシングユニットは並列で、かつ独立して、これらのデータ及びアプリケーション処理を行う。PU504は好ましくは、RISC(Reduced Instruction Set Computing)技術を採用しているマイクロプロセッサアーキテクチャであるパワーPC(PowerPC)コアを用いて実装され

50

る。RISCは、単純な命令の組合せを用いて、より複雑な命令を実行する。従って、プロセッサのタイミングは、単純で高速の動作に基づくものであり、マイクロプロセッサがより多くの命令を所定のクロック速度で実行できるようにする。

【0043】

PU504はサブプロセッシングユニット508により、データ及びアプリケーション処理をスケジューリングし調整を行う、メインプロセッシングユニットの役割を果たしているサブプロセッシングユニット508のうち、1つのサブプロセッシングユニットにより実装されうる。更に、プロセッサエレメント500内には1つ以上の実装されたPUが存在しうる。

【0044】

本モジュール構造によれば、特定のコンピュータシステムにおけるPE500の数は、そのシステムが要求する処理能力に基づく。例えば、サーバーにおけるPE500の数は4、ワークステーションにおけるPE500の数は2、PDAにおけるPE500の数は1とすることができる。特定のソフトウェアセルの処理に割当てられるPE500のサブプロセッシングユニット数は、セル内のプログラムやデータの複雑度や規模により決定される。

【0045】

図13は、サブプロセッシングユニット(SPU)508の好ましい構造および機能を例示している。SPU508アーキテクチャは好ましくは、多目的プロセッサ(平均して高性能を広範なアプリケーションに実現するように設計されているもの)と、特殊目的プロセッサ(高性能を単一のアプリケーションに実現するように設計されているもの)間の間隙を埋める。SPU508は、ゲームアプリケーション、メディアアプリケーション、ブロードバンドシステムなどに高性能を実現するように、また、リアルタイムアプリケーションのプログラマーに高度な制御を提供するように設計される。SPU508は、グラフィックジオメトリパイプライン、サーフェスサブディビジョン、高速フーリエ変換、画像処理キーワード、ストリームプロセッシング、MPEGのエンコード/デコード、エンクリプション、デクリプション、デバイスドライバの拡張、モデリング、ゲーム物理学、コンテンツ制作、音響合成及び処理が可能である。

【0046】

サブプロセッシングユニット508は2つの基本機能ユニットを有し、それらはSPUCOA510A及びメモリフローコントローラ(MFC)510Bである。SPUCOA510Aはプログラムの実行、データ操作、などを行い、一方でMFC510BはシステムのSPUCOA510AとDRAM514の間のデータ転送に関連する関数を実施する。

【0047】

SPUCOA510Aはローカルメモリ550、命令ユニット(IU: Instruction Unit)552、レジスタ554、1つ以上の浮動小数点実行ステージ556、及び1つ以上の固定小数点実行ステージ558を有している。ローカルメモリ550は好ましくは、SRAMなどの、シングルポートのランダムメモリアクセスを用いて実装される。殆どのプロセッサはキャッシュの導入により、メモリへのレイテンシを小さくする一方で、SPUCOA510Aはキャッシュより小さいローカルメモリ550を実装している。更に、リアルタイムアプリケーション(及び本明細書に述べているように、他のアプリケーション)のプログラマーたちに一貫した、予測可能なメモリアクセスレイテンシを提供するため、SPU508A内のキャッシュメモリアーキテクチャは好ましくない。キャッシュメモリのキャッシュヒット/ミスという特徴のために、数サイクルから数百サイクルまでの、予測困難なメモリアクセス時間が生じる。そのような予測困難性により、例えばリアルタイムアプリケーションのプログラミングに望ましい、アクセス時間の予測可能性が低下する。DMA転送をデータの演算処理にオーバーラップさせることで、ローカルメモリSRAM550においてレイテンシの隠蔽を実現しうる。これにより、リアルタイムアプリケーションのプログラミングが制御しやすくなる。DMAの転送に関連するレイテンシと命令のオーバーヘッドが、キャッシュミスにサービスしているレイテンシのオーバーヘッドを超

10

20

30

40

50

過していることから、DMAの転送サイズが十分に大きく、十分に予測可能な場合（例えば、データが必要とされる前にDMAコマンドが発行される場合）に、このSRAMのローカルメモリ手法による利点が得られる。

【0048】

サブプロセッシングユニット508のうちの、所定の1つのサブプロセッシングユニット上で実行しているプログラムは、ローカルアドレスを使用している関連のローカルメモリ550を参照する。しかしながら、ローカルメモリ550のそれぞれの場所はまた、システムのメモリマップ全体内に実アドレス（RA：Real Address）も割当てられる。これにより、特権ソフトウェア（あるいはプリビレッジソフトウェア：Privilege Software）はローカルメモリ550をプロセスの有効アドレス（EA：Effective Address）にマッピングする、ローカルメモリ550と別のローカルメモリ550間のDMA転送を促進する。PU504はまた、有効アドレスを用いてローカルメモリ550に直接アクセスすることができる。好ましい実施形態では、ローカルメモリ550は556キロバイトのストレージを有し、またレジスタ552の容量は128×128ビットである。

10

【0049】

SPUコア504Aは、好ましくは、論理命令がパイプライン式で処理される、プロセッシングパイプラインを用いて実装される。パイプラインは命令が処理される任意の数のステージに分けられうるが、一般にパイプラインは1つ以上の命令のフェッチ、命令のデコード、命令間の依存性チェック、命令の発行、及び、命令の実行ステップを有している。これに関連して、IU552は命令バッファ、命令デコード回路、依存性チェック回路、及び命令発行回路、を有する。

20

【0050】

命令バッファは、好ましくは、ローカルメモリ550と結合され、また、フェッチされる際に一時的に命令を記録するよう動作可能な複数のレジスタを備えている。命令バッファは好ましくは、全ての命令が一つのグループとしてレジスタから出て行く、つまり、実質的に同時に出て行くように動作する。命令バッファはいずれの大きさでありうるが、好ましくは、2あるいは3レジスタよりは大きくないサイズである。

【0051】

一般に、デコード回路は命令を壊し、対応する命令の関数を実施する論理的マイクロオペレーションを生成する。例えば、論理的マイクロオペレーションは、算術論理演算、ローカルメモリ550へのロード及びストアオペレーション、レジスタソースオペランド、及び/又は即値データオペランドを特定しうる。デコード回路はまた、ターゲットレジスタアドレス、構造リソース、機能ユニット、及び/又はバスなど、命令がどのリソースを使用するかを示しうる。デコード回路はまた、リソースが要求される命令パイプラインステージを例示した情報を与えることができる。命令デコード回路は好ましくは、命令バッファのレジスタ数に等しい数の命令を実質的に同時にデコードするように動作する。

30

【0052】

依存性チェック回路は、所定の命令のオペランドがパイプラインの他の命令のオペランドに依存しているかどうかを判断するために試験を行う、デジタル論理回路を含む。その場合、所定の命令はそのような他のオペランドが（例えば、他の命令が実行の完了を許容することにより）アップデートされるまで、実行されることができない。依存性チェック回路は好ましくは、デコーダ回路112から同時に送られる複数の命令の依存性を判断する。

40

【0053】

命令発行回路は浮動小数点実行ステージ556、及び/または固定小数点実行ステージ558へ命令を発行するように動作することができる。

【0054】

レジスタ554は好ましくは、128エントリのレジスタファイルなどの、相対的に大きな統一レジスタファイルとして実装される。これにより、レジスタが足りなくなる状態を回避するよう、レジスタリネーミングを必要としない、深くパイプライン化された高周

50

波数の実装品が可能になる。一般に、リネーミング用ハードウェアは、処理システムのかなりの割合の領域と電力を消費する。その結果、ソフトウェアのループ展開、又は他のインターリーブ技術によりレイテンシがカバーされると、最新のオペレーションが実現される。

【0055】

好ましくは、SPUコア510Aはスーパースカラアーキテクチャであり、これにより1つ以上の命令がクロックサイクル毎に発行される。SPUコア510Aは好ましくは、命令バッファから送られる同時命令の数、例えば2~3命令(各クロックサイクル毎に2命令あるいは3命令が発行されることを意味する)に対応する程度まで、スーパースカラとして動作する。所望の処理能力に応じて、多数の、あるいは少数の浮動小数点実行ステージ556と、固定小数点実行ステージ558が採用される。好ましい実施形態では、浮動小数点実行ステージ556は1秒あたり320億の浮動小数点演算速度で演算し(32GFLOPS)、また、固定小数点実行ステージ558は演算速度が1秒あたり320億回(32GOPS)となっている。

10

【0056】

MFC510Bは、好ましくは、バスインターフェースユニット(BIU: Bus Interface Unit)564、メモリ管理ユニット(MMU: Memory Management Unit)562、及びダイレクトメモリアクセスコントローラ(DMAC: Direct Memory Access Controller)560を備えている。DMAC560は例外として、MFC510Bは好ましくは、低電力化設計とするため、SPUコア510Aやバス512と比べて半分の周波数で(半分の速度で)実行する。MFC510Bはバス512からSPU508に入力されるデータや命令を処理するように動作することができ、DMACに対しアドレス変換を行い、また、データコヒーレンシーに対しヌープオペレーションを提供する。BIU564はバス512とMMU562及びDMAC560間にインターフェースを提供する。従って、SPU508(SPUコア510A及びMFC510Bを含む)及びDMAC560は、バス512と物理的に、及び/又は論理的に結合されている。

20

【0057】

MMU562は、好ましくは、メモリアクセスのために、実アドレスに有効アドレスを変換するように動作することができる。例えば、MMU562は、有効アドレスの上位ビットを実アドレスビットに変換しうる。しかしながら下位のアドレスビットは好ましくは変換不能であり、また、実アドレスの形成及びメモリへのアクセスリクエストに使用する場合には、ともに論理的及び物理的なものと考えられる。1つ以上の実施形態では、MMU562は、64ビットのメモリ管理モデルに基づいて実装され、また、4K-、64K-、1M-、及び16M-バイトのページサイズを有する 2^{64} バイトの有効アドレススペースと、256MBのセグメントサイズを提供しうる。MMU562は好ましくは、DMAコマンドに対し、 2^{65} バイトまでの仮想メモリ、 2^{42} バイト(4テラバイト)までの物理メモリをサポートするように動作することができる。MMU562のハードウェアは、8-エントリでフルアソシエイティブのSLBと、256-エントリと、4ウェイセットアソシエイティブのTLBと、TLBに対してハードウェアTLBのミスハンドリングに使用される4x4リプレースメント管理テーブル(RMT: Replacement Management Table)と、を含む。

30

40

【0058】

DMAC560は、好ましくは、SPUコア510Aや、PU504、及び/又は他のSPUなどの、1つ以上の他のデバイスからのDMAコマンドを管理するように動作することができる。DMAコマンドには3つのカテゴリが存在し、それらは、プットコマンド、ゲットコマンド、及びストレージ制御コマンドである。プットコマンドは、ローカルメモリ550から共有メモリ514へデータを移動させるよう動作する。ゲットコマンドは、共有メモリ514からローカルメモリ550へデータを移動させるよう動作する。また、ストレージ制御コマンドには、SLIコマンドと同期化コマンドが含まれる。この同期化コマンドは、アトミックコマンド(atomic command)、信号送信コマンド、及び専用バリ

50

アコマンドを有しうる。DMAコマンドに回答して、MMU562は有効アドレスを実アドレスに変換し、実アドレスはBIU564へ送られる。

【0059】

SPUコア510Aは、好ましくは、DMAC560内のインターフェースと通信(DMAコマンド、ステータスなどを送る)するために、チャンネルインターフェース及びデータインターフェースを使用する。SPUコア510Aはチャンネルインターフェースを介して、DMAC560のDMAキューへDMAコマンドを送る。DMAコマンドがDMAキューに存在すると、そのコマンドはDMAC560内の発行及び完了論理により処理される。DMAコマンドに対する全てのバストランザクションが終了すると、完了信号がチャンネルインターフェースを越えて、SPUコア510Aへ送られる。

10

【0060】

図14はPU504の一般的な構造及び機能を例示している。PU504は2つの基本的な機能ユニットを有しており、それらはPUコア504Aとメモリフローコントローラ(MFC)504Bである。PUコア504Aは、プログラム実行、データ操作、マルチプロセッサマネージメント関数などを実施し、一方でMFC504Bはシステム100のPUコア504Aとメモリスペース間のデータ転送に関連する機能を実行する。

【0061】

PUコア504AはL1キャッシュ570、命令ユニット572、レジスタ574、1つ以上の浮動小数点実行ステージ576、及び1つ以上の固定小数点実行ステージ578を有することができる。L1キャッシュは、共有メモリ106、プロセッサ102、又はMFC504Bを介してメモリスペースの他の部分、から受信したデータに対するデータキャッシングの機能を提供する。PUコア504Aが好ましくはスーパーパイプラインとして実装されるので、命令ユニット572は好ましくは、フェッチ、デコード、依存性チェック、発行、などを含む、多くのステージを備えた命令パイプラインとして実装される。PUコア504はまた好ましくは、スーパースカラ構成であり、一方で1つ以上の命令がクロックサイクル毎に命令ユニット572から発行される。高度な処理(演算)能力を実現するために、浮動小数点実行ステージ576と固定小数点実行ステージ578は、パイプライン構成で複数のステージを有する。所望の処理能力に応じて、多数の、あるいは少数の浮動小数点実行ステージ576と、固定小数点実行ステージ578が採用される。

20

【0062】

MFC504Bは、バスインターフェースユニット(BIU)580、L2キャッシュメモリ、キャッシュ不可能なユニット(NCU: Non-Cachable Unit)584、コアインターフェースユニット(CIU: Core Interface Unit)586、及びメモリ管理ユニット(MMU)588を備えている。殆どのMFC504Bは、低電力化設計とするために、PUコア504Aとバス108と比べて、半分の周波数(半分の速度)で実行する。

30

【0063】

BIU580はバス108とL2キャッシュ582とNCU584論理ブロック間にインターフェースを提供する。このために、BIU580はバス108上で、十分にコヒーレントなメモリオペレーションを実施するために、マスタデバイスとして、また同様にスレーブデバイスとして機能する。マスタデバイスとして、BIU580はL2キャッシュ582とNCU584のために機能するため、バス108へロード/ストアリクエストを供給する。BIU580はまた、バス108へ送信されうるコマンドの合計数を制限するコマンドに対し、フロー制御機構を実装しうる。バス108のデータオペレーションは、8ビット要するように設計され、そのために、BIU580は好ましくは128バイトキャッシュラインを有するように設計され、また、コヒーレンシーと同期化の粒度単位は128KBである。

40

【0064】

L2キャッシュメモリ582(及びサポートハードウェア論理回路)は、好ましくは、512KBのデータをキャッシュするように設計されている。例えば、L2キャッシュ582はキャッシュ可能なロード/ストア、データプリフェッチ、命令フェッチ、命令プリ

50

フェッチ、キャッシュオペレーション、及びバリアオペレーションを処理しうる。L2 キャッシュ582は好ましくは8ウェイのセットアソシエティブシステムである。L2 キャッシュ582は6つのキャストアウトキュー(6つのRCマシンなど)と一致する6つのリロードキューと、8つ(64バイト幅)のストアキューを備えうる。L2 キャッシュ582はL1 キャッシュ570において、一部の、あるいは全てのデータのコピーをバックアップするように動作しうる。この点は、処理ノードがホットスワップである場合に状態を回復するのに便利である。この構成により、L1 キャッシュ570が少ないポート数でより速く動作することができ、また、より速くキャッシュツーキャッシュ転送ができる(リクエストがL2 キャッシュ582でストップしうるため)。この構成はまた、キャッシュコピーレンシー管理をL2 キャッシュメモリ582へ送るための機構も提供しうる。 10

【0065】

NCU584は、CIU586、L2 キャッシュメモリ582、及びBIU580と連動しており、通常は、PUコア504Aとメモリシステム間のキャッシュ不可能なオペレーションに対して、キューイング/バッファリング回路として機能する。NCU584は好ましくは、キャッシュ抑制ロード/ストア、バリアオペレーション、及びキャッシュコピーレンシーオペレーションなどの、L2 キャッシュ582により処理されないPUコア504Aとの全ての通信を処理する。NCU584は好ましくは、上述の低電力化目的を満たすように、半分の速度で実行されうる。

【0066】

CIU586は、MFC504BとPUコア504Aの境界に配置され、実行ステージ576、578、命令ユニット572、及びMMUユニット588からのリクエストに対し、また、L2 キャッシュ582及びNCU584へのリクエストに対し、ルーティング、アービトレーション、及びフロー制御ポイントとして機能する。PUコア504A及びMMU588は好ましくはフルスピードで実行され、一方でL2 キャッシュ582及びNCU584は2:1の速度比で動作することができる。従って、周波数の境界がCIU586に存在し、その機能の一つは、2つの周波数ドメイン間でリクエストの送信及びデータのりロードを行いながら、周波数の差を適切に処理することである。 20

【0067】

CIU586は3つの機能ブロックを有しており、それらは、ロードユニット、ストアユニット、及びリロードユニットである。更に、データプリフェッチ関数がCIU586により実施され、また好ましくは、ロードユニットの機能部である。CIU586は、好ましくは、(i) PUコア504AとMMU588からのロード及びストアリクエストを受け、(ii) フルスピードのクロック周波数をハーフスピードに変換する(2:1のクロック周波数変換)、(iii) キャッシュ可能なリクエストをL2 キャッシュ582へ送り、キャッシュ不可能なリクエストをNCU584へ送る、(iv) L2 キャッシュ582に対するリクエストとNCU584に対するリクエストを公正に調停する、(v) ターゲットウィンドウでリクエストが受信されてオーバーフローが回避されるように、L2 キャッシュ582とNCU584に対する転送のフロー制御を提供する、(vi) ロードリターンデータを受信し、そのデータを実行ステージ576、578、命令ユニット572、又はMMU588へ送る、(vii) スヌープリクエストを実行ステージ576、578、命令ユニット572、又はMMU588へ送る、(viii) ロードリターンデータとスヌープトラフィックを、ハーフスピードからフルスピードへ変換する、ように動作可能である。 30

【0068】

MMU588は、好ましくはPUコア540Aに対して、第2レベルのアドレス変換機能などによりアドレス変換を行う。第1レベルの変換は好ましくは、MMU588よりも小型で高速でありうる、別々の命令及びデータERAT(Effective to Real Address Translation)アレイにより、PUコア504Aにおいて提供されうる。

【0069】

好ましい実施形態では、PUコア504は、64ビットの実装品で、4-6GHz、1 50

0 F 0 4で動作する。レジスタは好ましくは64ビット長(1つ以上の特殊用途のレジスタは小型でありうるが)であり、また、有効アドレスは64ビット長である。命令ユニット570、レジスタ572、及び実行ステージ574と576は好ましくは、(RISC)演算技術を実現するために、PowerPCステージ技術を用いて実装される。

【0070】

本コンピュータシステムのモジュール構造に関する更なる詳細は、米国特許第6,526,491号に解説されており、該特許は参照として本願に組込まれる。

【0071】

本発明の少なくとも1つの更なる態様によれば、上述の方法及び装置は、図面において例示しているような、適切なハードウェアを利用して実現されうる。そのようなハードウェアは標準デジタル回路などのいずれの従来技術、ソフトウェア、及び/又はファームウェアプログラムを実行するよう動作可能ないずれの従来のプロセッサ、プログラム可能なROM(PROM)、プログラム可能なアレイ論理デバイス(PAL:Programmable Array Logic)などの、1つ以上のプログラム可能なデジタルデバイスあるいはシステム、を用いて実装されうる。更に、図示している装置は、特定の機能ブロックに分割されて示されているが、そのようなブロックは別々の回路を用いて、及び/あるいは1つ以上の機能ユニットに組み合わせて実装されうる。更に、本発明の様々な態様は、輸送及び/又は配布のために、(フロッピーディスク、メモリチップ、などの)適切な記憶媒体に記録されうる、ソフトウェア、及び/又はファームウェアプログラムを通じて実装されうる。

10

【0072】

以上のように、本発明によれば、少なくとも1つのプロセッサから受信したプログラムステータスデータから追跡データを生成する方法及び装置100が提供され、この少なくとも1つのプロセッサは、プログラムカウンタから取得したアドレスによるアドレッシング命令によりソフトウェアを実行するように動作できる。また、追跡バッファに追跡データを記録する方法及び装置100が提供され、この追跡データは、(i)各々のプログラム実行イベントの総カウント数、(ii)様々な時点でプログラムカウンタから取得したアドレス、及び(iii)総カウント数もアドレスも含まないプログラムのパフォーマンス情報を含むブックマークデータ、が含まれる。

20

本明細書において、具体的な実施形態を用いて本発明を記載したが、これらの実施形態は本発明の原理及び用途の例を例示したものに過ぎないことを理解されたい。このため、添付の請求の範囲に記載した本発明の趣旨及び範囲から逸脱することなく、これら例示的な実施形態を種々に変更したり、上記以外の構成を考案し得ることが理解されよう。

30

【図面の簡単な説明】

【0073】

【図1】本発明の1つ以上の態様による複数のサブプロセッサとパフォーマンスモニタを備えたマルチプロセッシングシステムの構造を例示したブロック図。

【図2】本発明の1つ以上の態様による図1のマルチプロセッシングシステムのより詳細なブロック図。

【図3】本発明の1つ以上の態様による図1のプロセッシングシステムのパフォーマンスモニタの一部の他の構成のブロック図。

40

【図4】本発明の1つ以上の態様による図1のプロセッシングシステムのパフォーマンスモニタのバッファに記録されるプログラムデータをパッキングするデータフォーマット構造の概念図。

【図5】本発明の1つ以上の態様によるパフォーマンスモニタのバッファに記録されるプログラムデータをパッキングする他のデータフォーマット構造の概念図。

【図6】本発明の1つ以上の態様によるパフォーマンスモニタのバッファに記録されるプログラムデータをパッキングする更に他のデータフォーマット構造の概念図。

【図7】本発明の1つ以上の態様によるパフォーマンスモニタのバッファに記録されるプログラムデータをパッキングする更に他のデータフォーマット構造の概念図。

【図8】本発明の1つ以上の態様によるパフォーマンスモニタのバッファに記録されるプ

50

ログラムデータをパッキングする更に他のデータフォーマット構造の概念図。

【図 9】本発明の 1 つ以上の異なる態様によりパフォーマンスモニタのバッファに記録される追跡データの一例を例示した概念図。

【図 10】本発明の 1 つ以上の異なる態様により図 1 のパフォーマンスモニタの一般的な実装品を例示したブロック図。

【図 11】本発明の 1 つ以上の異なる態様により他のマルチプロセッシングシステムの構造を例示したブロック図。

【図 12】本発明の 1 つ以上の異なる態様によりマルチプロセッサシステムの実装に使用される好適なプロセッサエレメント (P E) を例示した説明図。

【図 13】本発明の 1 つ以上の異なる態様により、図 12 のシステムの一般的なサブプロセッシングユニット (S P U) の構造を例示した説明図。 10

【図 14】本発明の 1 つ以上の異なる態様により、図 12 のシステムのプロセッシングユニット (P U) の構造を例示した説明図。

【符号の説明】

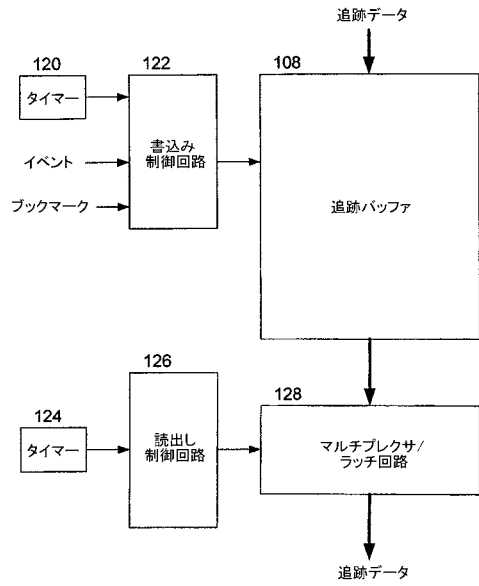
【 0 0 7 4 】

1 0 0	処理システム
1 0 2	プロセッサ
1 0 2	各プロセッサ
1 0 4	パフォーマンスモニタ回路
1 0 6	共有メモリ
1 0 8	追跡バッファ
1 1 0	マルチプレクサ
1 1 2	カウンタ
1 1 4	ブロック
1 1 6 A	レジスタ
1 2 0	タイマー
1 2 2	書込み制御回路
1 2 4	タイマー
1 2 6	読出し制御回路
2 0 2	プロセッサ
2 0 6	システムメモリ
2 0 8	バス
5 0 0	プロセッサエレメント
5 0 8	サブプロセッシングユニット
5 1 4	共有メモリ
5 5 0	ローカルメモリ

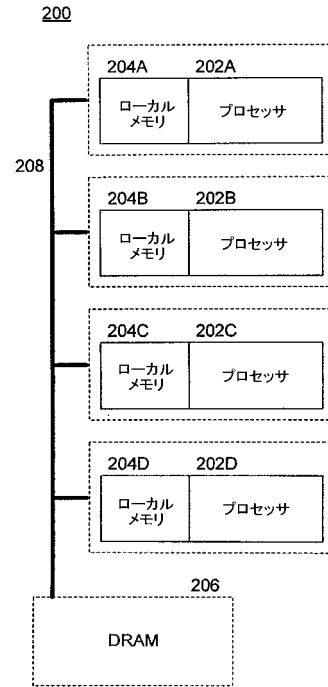
20

30

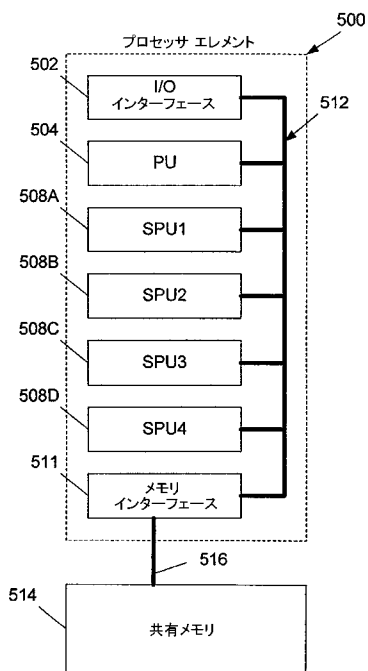
【図10】



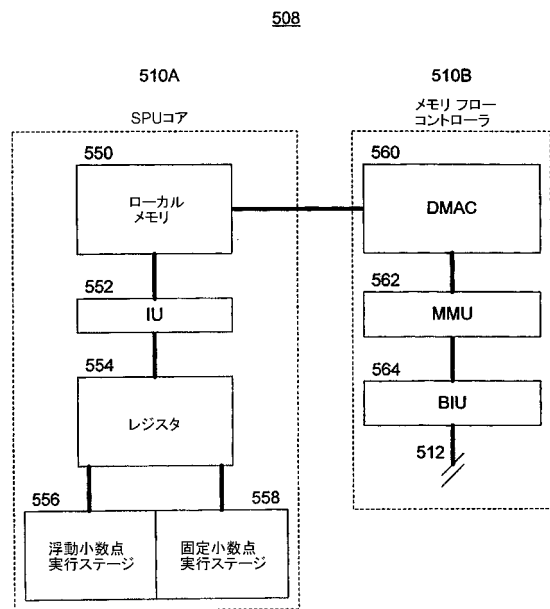
【図11】



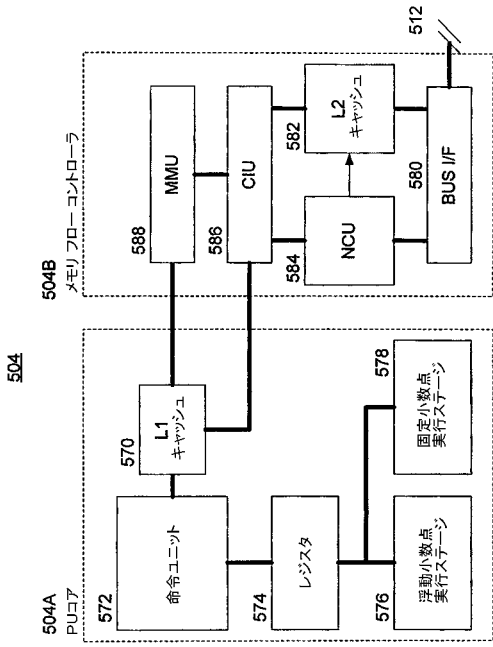
【図12】



【図13】



【 図 1 4 】



フロントページの続き

(72)発明者 大澤 正紀

東京都港区南青山二丁目6番21号 株式会社ソニー・コンピュータエンタテインメント内

Fターム(参考) 5B042 GA14 GA15 GA23 GB02 GC08 GC16 HH20 HH30 MA08 MB03

MC03 MC06 MC21 MC23 MC29 MC40