

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第5465215号  
(P5465215)

(45) 発行日 平成26年4月9日(2014.4.9)

(24) 登録日 平成26年1月31日(2014.1.31)

(51) Int.Cl.	F I				
<b>G06F 9/50</b>	<b>(2006.01)</b>	G06F	9/46	465D	
<b>G06F 1/26</b>	<b>(2006.01)</b>	G06F	9/46	465A	
<b>G06F 1/32</b>	<b>(2006.01)</b>	G06F	1/00	330C	
<b>G06F 1/04</b>	<b>(2006.01)</b>	G06F	1/00	332Z	
		G06F	1/04	301C	

請求項の数 8 外国語出願 (全 26 頁) 最終頁に続く

(21) 出願番号	特願2011-123762 (P2011-123762)	(73) 特許権者	591003943 インテル・コーポレーション
(22) 出願日	平成23年6月1日(2011.6.1)		アメリカ合衆国 95054 カリフォルニア州・サンタクララ・ミッション カレッジ ブレーバード・2200
(62) 分割の表示	特願2009-232857 (P2009-232857) の分割	(74) 代理人	110000877 龍華国際特許業務法人
原出願日	平成17年9月28日(2005.9.28)	(72) 発明者	グロハウスキー、エドワード アメリカ合衆国、95118 カリフォルニア州、サン ノゼ、エール ドライブ 5565
(65) 公開番号	特開2011-210275 (P2011-210275A)	(72) 発明者	シェン、ジョン アメリカ合衆国、95121 カリフォルニア州、サン ノゼ、クウェイル プラフ プレイス 2303
(43) 公開日	平成23年10月20日(2011.10.20)		最終頁に続く
審査請求日	平成23年6月30日(2011.6.30)		
(31) 優先権主張番号	10/952,627		
(32) 優先日	平成16年9月28日(2004.9.28)		
(33) 優先権主張国	米国 (US)		
前置審査			

(54) 【発明の名称】 利用可能な並列性の量に従って1命令当たりのエネルギーを変化させるための方法及び装置

(57) 【特許請求の範囲】

【請求項1】

高い性能を有する第1の組のコアおよび低い性能を有する第2の組のコアを含む、複数の非対称型コアと、前記高い性能を有する第1の組のコアおよび前記低い性能を有する第2の組のコアに接続された、スレッドマイグレーションロジックと、を有するプロセッサであって、

前記スレッドマイグレーションロジックは、オペレーティングシステムのスケジューラに  
応答するものであり、前記オペレーティングシステムは、現在の実行スレッドを有する  
コアの数を追跡するためのものであり、

前記スレッドマイグレーションロジックは、モニタされたそれぞれのコアのクロック  
当たりの電力消費に応じたプロセッサ自身の一命令当たりのエネルギーに基づいて、全チ  
ップ電力を予め定められた電力バジェット内に維持するように、前記高い性能を有する第  
1の組のコアの第1のコアと、前記低い性能を有する第2の組のコアの第2のコアとの間  
で、スレッドをマイグレーションする、プロセッサ。

【請求項2】

前記スレッドマイグレーションロジックは、前記第1の組のコアおよび第2の組のコア  
に関連付けられた電源電流によるものを含む電力消費の大きさに応じてマイグレーション  
する、請求項1に記載のプロセッサ。

【請求項3】

前記スレッドマイグレーションロジックは、マイグレーションをトリガする特定の時間

間隔を発生する周期タイマに応じてマイグレーションする、請求項 1 に記載のプロセッサ。

【請求項 4】

高い性能を有する第 1 の組のコアおよび低い性能を有する第 2 の組のコアを含む、複数の非対称型コアと、前記高い性能を有する第 1 の組のコアおよび前記低い性能を有する第 2 の組のコアに接続された、スレッドマイグレーションロジックと、を有するプロセッサであって、

前記スレッドマイグレーションロジックは、モニタされた前記第 1 の組のコアおよび第 2 の組のコアに関連付けられた電源電流によるものを含む電力消費の大きさに応じて、全チップ電力を予め定められた電力バジェット内に維持するように、前記高い性能を有する第 1 の組のコアの第 1 のコアと、前記低い性能を有する第 2 の組のコアの第 2 のコアとの間で、スレッドをマイグレーションし、

前記電力消費の大きさは、コアが実行しているか停止しているかを反映する値に、コアが実行状態にあるときに 1 クロックあたりに費やされるエネルギー量を乗算することによって求められる、プロセッサ。

【請求項 5】

高い性能を有する第 1 の組のコアおよび低い性能を有する第 2 の組のコアを含む、複数の非対称型コアと、前記高い性能を有する第 1 の組のコアおよび前記低い性能を有する第 2 の組のコアに接続された、スレッドマイグレーションロジックと、を有するプロセッサであって、

前記プロセッサの各組のコアの属性を監視するためのモニタロジックをさらに備え、前記属性は 1 クロックサイクルあたりにリタイヤされる命令の数であり、

前記スレッドマイグレーションロジックは、1 クロックサイクルあたりにリタイヤされる命令の前記数に、1 クロックサイクルあたりの命令ごとに費やされるエネルギー量を乗算することによって求められる電力消費の大きさに応じて、全チップ電力を予め定められた電力バジェット内に維持するように、前記高い性能を有する第 1 の組のコアの第 1 のコアと、前記低い性能を有する第 2 の組のコアの第 2 のコアとの間で、スレッドをマイグレーションする、プロセッサ。

【請求項 6】

高い性能を有する第 1 の組のコアおよび低い性能を有する第 2 の組のコアを含む、複数の非対称型コアと、前記高い性能を有する第 1 の組のコアおよび前記低い性能を有する第 2 の組のコアに接続された、スレッドマイグレーションロジックと、を有するプロセッサであって、

前記スレッドマイグレーションロジックは、モニタされた前記第 1 の組のコアおよび第 2 の組のコアに関連付けられた電源電流によるものを含む電力消費の大きさに応じて、全チップ電力を予め定められた電力バジェット内に維持するように、前記高い性能を有する第 1 の組のコアの第 1 のコアと、前記低い性能を有する第 2 の組のコアの第 2 のコアとの間で、スレッドをマイグレーションし、

前記電力消費の大きさは、1 クロックサイクルあたりにリタイヤされる命令の数に、1 命令あたり、かつ、1 クロックあたりに費やされるエネルギー量を乗算することによって求められる、プロセッサ。

【請求項 7】

高い性能を有する第 1 の組のコアおよび低い性能を有する第 2 の組のコアを含む、複数の非対称型コアと、前記高い性能を有する第 1 の組のコアおよび前記低い性能を有する第 2 の組のコアに接続された、スレッドマイグレーションロジックと、を有するプロセッサであって、

前記プロセッサの各組のコアの属性を監視するためのモニタロジックをさらに備え、前記属性は前記コアの機能ブロックのアクティブステータスであり、

前記スレッドマイグレーションロジックは、前記機能ブロックがアクティブであるのか又はアイドルであるのかを反映する値に、前記機能ブロックがアクティブであるときに 1

10

20

30

40

50

クロックあたりに費やされるエネルギー量を乗算することによって求められる電力消費の大きさに応じて、全チップ電力を予め定められた電力バジェット内に維持するように、前記高い性能を有する第1の組のコアの第1のコアと、前記低い性能を有する第2の組のコアの第2のコアとの間で、スレッドをマイグレーションする、プロセッサ。

【請求項8】

高い性能を有する第1の組のコアおよび低い性能を有する第2の組のコアを含む、複数の非対称型コアと、前記高い性能を有する第1の組のコアおよび前記低い性能を有する第2の組のコアに接続された、スレッドマイグレーションロジックと、を有するプロセッサにおいて、

前記スレッドマイグレーションロジックは、オペレーティングシステムのスケジューラに 10 応答するものであり、前記オペレーティングシステムは、現在の実行スレッドを有するコアの数を追跡するためのものであり、

モニタされたそれぞれのコアのクロックあたりの電力消費に応じたプロセッサ自身の1命令あたりのエネルギーのメトリックに基づいて、全チップ電力を予め定められた電力バジェット内に維持するように、前記低い性能を有する第2の組の複数のコアに含まれる第2のコアから前記高い性能を有する第1の組の複数のコアに含まれる第1のコアへ、前記スレッドマイグレーションロジックを介して、スレッドの実行をマイグレートすることを含む方法。

【発明の詳細な説明】

【技術分野】

【0001】

本開示は、包括的には、変化するスカラー資源所要量及び並列資源所要量を有するプログラムを実行することができるマイクロプロセッサに関し、より具体的には、複数のコアを使用するマイクロプロセッサに関する。

【背景技術】

【0002】

コンピュータワークロードは、いくつかの実施の形態では、本来的に並列性をほとんど有さない（大部分がスカラーである）ものから、かなりの量の並列性を有する（大部分が並列である）ものまで連続的に及んでおり、この性質は、ソフトウェアのセグメントごとに变化し得る。一般的なスカラーワークロードには、ソフトウェア開発ツール、オフィス生産性スイート、及びオペレーティングシステムカーネルルーチンが含まれる。一般的な並列ワークロードには、3Dグラフィックス、メディア処理、及び科学的応用が含まれる。スカラーワークロードは、1クロックあたりの命令数（IPC）が0.2～2.0の範囲で命令をリタイヤさせることができるのに対して、並列ワークロードは、4IPC～数千IPCの範囲のスループットを達成することができる。後者の高いIPCは、命令レベルの並列性及びスレッドレベルの並列性の使用を通じて取得可能な場合がある。

【0003】

従来技術のマイクロプロセッサは、多くの場合、スカラー性能又は並列性能のいずれかを主目的として設計されている。高いスカラー性能を達成するには、多くの場合、実行待ち時間をできるだけ削減することが望ましい。有効待ち時間を削減するためのマイクロアーキテクチャ技法には、投機的実行、分岐予測、及びキャッシュが含まれる。高いスカラー性能の追求の結果、パイプラインマイクロプロセッサは、大きなアウトオブオーダー、投機的性の高い、深いパイプラインマイクロプロセッサとなっている。高い並列性能を達成するには、できるだけ大きな実行スループット（帯域幅）を提供することが望ましい場合がある。スループットを増大させるためのマイクロアーキテクチャ技法には、幅の広いスーパー scaler 処理、単一命令複数データ命令、チップレベルマルチプロセッシング、及びマルチスレッド化が含まれる。

【発明の開示】

【発明が解決しようとする課題】

【0004】

10

20

30

40

50

問題は、スカラータスク及び並列タスクの双方においてうまく動作するマイクロプロセッサを構築しようとする時に発生し得る。1つの問題は、短い待ち時間を達成するのに必要とされる設計技法が、場合により、高いスループットを達成するのに必要とされる設計技法と非常に異なるという認識から発生し得る。

【0005】

本発明は、添付図面の図に、限定ではなく一例として示されている。添付図面において、同じ参照符号は同様の構成要素を指す。

【発明を実施するための最良の形態】

【0006】

以下の説明は、ソフトウェアプログラムにおいて利用可能な並列性の量に従って各命令を処理するのに費やされるエネルギー量を変化させるための技法を記載している。以下の説明では、本発明のより完全な理解を提供するために、ロジック実施態様、ソフトウェアモジュール割り当て、バス及び他のインターフェースシグナリング技法、オペレーションの詳細等の多数の具体的な詳細が述べられている。しかしながら、本発明は、このような具体的な詳細がなくても実施できることが当業者には十分理解されよう。それ以外の場合には、制御構造、ゲートレベルの回路、及び完全なソフトウェア命令シーケンスは、本発明を分かりにくくしないために詳細に示されていない。当業者は、含まれた説明によって、必要以上の実験を行うことなく適切な機能を実施することができる。特定の実施態様では、本発明は、Intel（登録商標）社によって生産されたようなPentium（登録商標）互換プロセッサのマルチコアの実施態様の形で開示されている。しかしながら、本発明は、Itaniumプロセッサファミリー互換プロセッサ、X-Scale（登録商標）ファミリー互換プロセッサ、他のベンダ又は設計者のプロセッサアーキテクチャのいずれかからの多種多様な異なる汎用プロセッサのいずれか等の、他の種類のプロセッサにおいても実施することができる。加えて、いくつかの実施の形態は、グラフィックス、ネットワーク、画像、通信、他の任意の既知のタイプのプロセッサ又は別の方法で利用可能なタイプのプロセッサ等の、専用プロセッサを含むこともできるし、専用プロセッサとすることもできる。

【0007】

電力効率は、1ワット当たりについての1秒当たりの命令数（IPS）の点で測定することができる。IPS/ワットのメトリックは、1命令当たりのエネルギーと等価である。より正確には、IPS/ワットは、以下のように、1命令当たりのエネルギーの逆数に比例する。

【0008】

【数1】

$$(IPS)/(Watt) = (Instructions)/(Joule)$$

(1)

【0009】

1命令当たりのエネルギーのメトリックの重要な特性は、命令を処理するのに必要とされる時間とは独立であるということである。これによって、1命令当たりのエネルギーは、スループット性能にとって役立つメトリックとなる。

【0010】

マイクロプロセッサの電力消費の近似解析は、命令が処理される度に充電又は放電されるコンデンサとしてマイクロプロセッサをモデル化することにより行うことができる（簡単にするために、漏れ電流及び短絡スイッチング電流は無視することができる）。この前提により、1命令当たりのエネルギーは、2つの事項、すなわち、（フェッチからリタイヤへ）各命令を処理するためにトグルされる静電容量及び電源電圧にのみ依存することが

できる。コンデンサに標準的に適用される既知の公式

【 0 0 1 1 】

【 数 2 】

$$E = C V^2 / 2$$

( 2 )

【 0 0 1 2 】

をマイクロプロセッサにも同様に適用することができる。Eは、1つの命令を処理するのに必要とされるエネルギーであり、Cは、その1つの命令を処理する際にトグルされる静電容量であり、Vは、電源電圧である。

10

【 0 0 1 3 】

マイクロプロセッサは、たとえば100ワット等の一定の電力バジェット内で動作することができる。或る時間にわたって平均化されたマイクロプロセッサの電力消費は、マイクロプロセッサ又はソフトウェアが何を行うかにかかわらず、この電力バジェットを超えるべきではない。この目的を達成するために、マイクロプロセッサは、或る種の動的な温度管理を組み込むことができる。同様に、チップレベルマルチプロセッサは、そのマルチプロセッサが、1クロックあたりの命令数 (IPC) がたとえば0.2個で命令をリタイヤしているのか、又は20IPCで命令をリタイヤしているのかにかかわらず、一定の電力バジェット内に留まるようにそのアクティビティを調整 (又はスロットル) することができる。良好な性能を発揮するために、チップレベルマルチプロセッサは、自身のMIPS /ワットを変化させることができるべきである。すなわち、自身のエネルギー / 命令を、この例では100 : 1の範囲にわたって変化させることができるべきである。

20

【 0 0 1 4 】

高いスカラー性能及び高いスループット性能の双方を達成できるマイクロプロセッサを設計するための1つの手法は、ソフトウェアにおいて利用可能な並列性の量又は利用可能であると推定される並列性の量に従って、各命令を処理するのに費やされるエネルギー量を動的に変化させることである。換言すれば、並列性の量が少ない場合には、マイクロプロセッサは、すべての利用可能なエネルギーを、少数の命令を処理することに費やすことができ、並列性の量がより多い場合には、マイクロプロセッサは、各命令を処理することにほとんどエネルギーを費やすことができない。これは、

30

【 0 0 1 5 】

【 数 3 】

$$P = (EPI) \times (IPS)$$

( 3 )

【 0 0 1 6 】

として表すことができる。ここで、Pは、一定の電力バジェットであり、EPIは、リタイヤされた1命令当たりの平均エネルギーであり、IPSは、すべてのプロセッサコアにわたって1秒あたりにリタイヤされた命令の総数である。この実施の形態は、全マルチプロセッサチップ電力をほぼ一定のレベルに維持しようと試みるものである。

40

【 0 0 1 7 】

相補型金属酸化膜半導体 (CMOS) の電圧及び周波数のスケーリングは、1命令あたりのエネルギーの比について異なった比を達成するのに使用することができる。一実施の形態では、ロジックが、所望の性能レベル及び電力レベルに従って、マイクロプロセッサの電源電圧及びクロック周波数を調和して変化させる。チップレベルマルチプロセッサの全電力消費を一定の電力バジェット内に維持するために、電圧及び周波数のスケーリング

50

を以下のように動的に適用することができる。スレッドの並列性が低い段階では、最良のスカラー性能を得るために、高い電源電圧及び高い周波数を使用して、少数のコアを実行することができる。スレッドの並列性が高い段階では、最良のスループット性能を得るために、低い電源電圧及び低い周波数を使用して、多数のコアを実行することができる。アクティブでないコアの電力消費は低いことが望ましい場合があるので、ダイナミックスリープトランジスタ (dynamic sleep transistor) やボディバイアス等の漏れ制御技法を使用することができる。

#### 【 0 0 1 8 】

次に図 1 を参照すると、一実施の形態による、電圧及び周波数によって構成可能なコアを含むプロセッサの概略図が示されている。コア 1 1 2 0、コア 2 1 3 0、コア 3 1 4 0、及びコア 4 1 5 0 が示されているが、他の実施の形態では、プロセッサのコアは 4 つよりも多くすることもできるし、少なくすることもできる。これらのコアの 1 つ又は複数は、電圧制御回路及びクロック周波数制御回路を有することができる。図 1 は、コア 1 1 2 0 が、電圧制御回路 1 2 2 及び周波数制御回路 1 2 4 を所有することを明示的に示しているが、他のコアも、同等の回路を同様に有することもできるし、電圧制御ロジック及び周波数制御ロジックを、特定のコアに直接関連付けられていない別個のロジックとすることもできる。

#### 【 0 0 1 9 】

スロットルモジュール 1 1 0 は、情報を収集して、実行しているソフトウェアプログラムに存在する並列性の量又はその量の推定値を求めるのに使用することができる。一実施の形態では、並列性の量は、サポートされる同時スレッドの個数とすることができる。他の実施の形態では、1 秒あたりにリタイヤされる命令の総数又はマルチスレッド化された投機的実行をサポートできる分岐命令の個数等の、他のメトリックを使用して、並列性の量を表すことができる。スロットルモジュール 1 1 0 は、オペレーティングシステムによって提供された情報を利用して、並列性の量を求めるのを助けることができる。他の実施の形態では、スロットルモジュール 1 1 0 は、プロセッサ内のハードウェアロジック及びプロセッサのコアを使用してこの並列性の量を求めることを行うことができる。この並列性の量を求めることは、連続的に行うこともできるし、定期的に行うこともできる。

#### 【 0 0 2 0 】

スロットルモジュール 1 1 0 は、プログラムにおける並列性の量を求めるごとに、信号線 1 1 2、1 1 4、1 1 6、及び 1 1 8 を介してコア 1 2 0、1 3 0、1 4 0、1 5 0 に、それらの電圧及びクロック周波数を変更するように指示することができる。一実施の形態では、信号線 1 1 2、1 1 4、1 1 6、及び 1 1 8 は、コアをオン又はオフにすることにもできるし、コアを収容する電力泉 (power well) からの電力を除去することにもできる。他の実施の形態では、コアは、クロックゲーティング技法又は命令スタベーション (instruction starvation) 技法によってオフにすることもできる。一実施の形態では、現在のスレッドレベルの並列性の量が以前の量を超え、その超える量がしきい値を超える場合、スロットルモジュールは、各コアの電圧及びクロック周波数を減少させながらより多数のコアでスレッドを実行することによって、より多数のスレッドの実行への移行を開始することができる。以前にオフにされていたコアをオンにして、より多数のスレッドをサポートすることができる。同様に、現在のスレッドレベルの並列性の量が以前の量を下回り、その下回る量がしきい値を超える場合、スロットルモジュールは、いくつかのコアの電圧及びクロック周波数を増加させながらより少数のこれらのコアでスレッドを実行することによって、より少数のスレッドの実行への移行を開始することができる。以前にオンにされていたいくつかのコアは、より少数のスレッドをサポートするのもはや必要とされる場合はないので、それらのコアをオフにすることができる。

#### 【 0 0 2 1 】

一実施の形態では、異なるマイクロアーキテクチャを使用して或る範囲の性能及び電力を補うことができる単一命令セットアーキテクチャ (ISA) 異種マルチコアマイクロプロセッサを設計することが可能な場合がある。一実施の形態では、チップレベルマルチブ

10

20

30

40

50

ロセッサを2つのタイプのプロセッサコアから構築することができる。これら2つのタイプのコアは、ラージコア及びスモールコアと呼ぶことができる。これら2つのタイプのコアは、同じ命令セットアーキテクチャを実施することができ、キャッシュコヒーレンシを使用して共有メモリを実施することができ、それらのマイクロアーキテクチャのみが異なることができる。他の実施の形態では、2つのタイプのコアは、類似の命令セットアーキテクチャを実施することができる。すなわち、スモールコアは、ラージコアの命令セットのサブセットを実施することができる。ラージコアは、アウトオブオーダーでスーパースカラーの深いパイプラインマシンとすることができるのに対して、スモールコアは、インオーダーのスカラーの短いパイプラインマシンとすることができる。Intel Pentium 4 プロセッサ及びIntel i486 プロセッサは、これら2つのクラスのコアの代表的なものである。他の実施の形態では、ほぼ同様又は同一の命令セットアーキテクチャを実行する3つ以上のクラスのコア又は3つ以上の性能レベルのコアを使用することができる。

10

#### 【0022】

一実施の形態では、チップレベルマルチプロセッサは、1つのラージコア及び25個のスモールコアを含む。これら2つのタイプのコアは、電力消費が25:1の比を有し、スカラー性能が5:1の比を有し、5:1レンジの1命令当たりのエネルギーを有する。チップレベルマルチプロセッサ又はこの実施の形態は、以下のように動作することができる。スレッドレベルの並列性が低い段階では、ラージコアが、最良のスカラー性能を得るために実行することができる。スレッドレベルの並列性が高い段階では、複数のスモールコアが、最良のスループット性能を得るために実行することができる。

20

#### 【0023】

どの瞬間においても、マイクロプロセッサは、1つのラージコア又は25個のスモールコアのいずれかを実行することができる。利用可能なソフトウェアスレッドの個数は、時間と共に変化するので、非対称型マルチプロセッサは、ラージコアとスモールコアとの間でスレッドをマイグレーションできるべきである。スレッドマイグレーションロジックが、この機能をサポートするために実施される場合がある。

#### 【0024】

実際には、ラージコアをオフにスイッチングした時点におけるスループット性能の不連続を低減するために、ラージコアと同時に少数のスモールコアを実行することを可能にすることが望ましい場合がある。前の例では、ラージコアをオフにスイッチングすると共に2つのスモールコアをオンにスイッチングすることに起因して、3ユニットのスループットの不連続が生じる場合がある。たとえば、電源が、5個までのスモールコアをラージコアと同時に実行することを短時間の間サポートする場合に、失われる全スループットの割合を削減するために、この同時実行を可能にすることによって、不連続を、より多数の実行スレッドで発生するように移動させることができる。

30

#### 【0025】

今日のマイクロプロセッサを代表する2つのタイプのコアを使用すると、4:1レンジの1命令当たりのエネルギーが達成可能である。将来のマイクロプロセッサがさらに高いレベルのスカラー性能を発揮し続けると、可能な1命令当たりのエネルギーのレンジは、おそらく6:1、又は、この比よりもはるかに優れた比に増加すると予想することができる。

40

#### 【0026】

次に図2を参照すると、一実施の形態による、処理電力及び電力消費によって選択可能なコアを含むプロセッサの概略図が示されている。このプロセッサは、少数のラージコアであるAコアを含むことができ、また、より多数のスモールコアであるコアBも含むことができる。Aコア1 220、Aコア2 222、及びBコア1 230~60 262が示されているが、他の実施の形態では、プロセッサに存在するAコアは、2つよりも多くすることもできるし、少なくすることもでき、プロセッサに存在するBコアは、60個よりも多くすることもできるし、少なくすることもできる。

50

## 【 0 0 2 7 】

スロットルモジュール 2 1 0 は、この場合も、情報を収集して、実行しているソフトウェアプログラムに存在する並列性の量を求めるのに使用することができる。一実施の形態では、並列性の量は、サポートされる同時スレッドの個数とすることができる。他の実施の形態では、上述したように、他のメトリックを使用して、並列性の量を表すことができる。スロットルモジュール 2 1 0 は、オペレーティングシステムによって提供される情報を利用して、並列性の量を求めることを助けることができる。他の実施の形態では、スロットルモジュール 2 1 0 は、プロセッサ内のハードウェアロジック及びプロセッサのコアを使用してこの並列性の量を求めることを行うことができる。この並列性の量を求めることは、連続的に行うこともできるし、定期的に行うこともできる。

10

## 【 0 0 2 8 】

利用可能なソフトウェアスレッドの個数は、時間と共に変化するもので、図 1 のプロセッサは、ラージ A コアとスモール B コアとの間でスレッドをマイグレーションできるスレッドマイグレーションロジック 2 1 2 を含むことができる。ラージ A コアをオフにスイッチングした時点におけるスループット性能の不連続を低減するために、ラージ A コアと同時に少数のスモール B コアを実行することを可能にすることが望ましい場合がある。失われる全スループットの割合を削減するために、たとえば、5 個までのスモールコアをラージコアと同時に実行することを可能にすることによって、不連続を、より多数の実行スレッドで発生するように移動させることができる。

20

## 【 0 0 2 9 】

スロットルモジュール 2 1 0 は、プログラムにおける並列性の量を求めるごとに、信号線 2 2 4 ~ 2 6 6 を使用して A コア及び B コアの電源投入又は電源切断を開始することができる。一実施の形態では、現在の並列性の量が以前の量を超え、その超える量がしきい値を超える場合、スロットルモジュール 2 1 0 は、スレッドマイグレーションロジック 2 1 2 を使用して、より多数の B コアで実行できる、より多数のスレッドの実行への移行を開始することができる。以前にオフにされていた B コアをオンにして、それら多数のスレッドをサポートすることができ、オンにされているどの A コアもオフにすることができる。同様に、現在の並列性の量が以前の量を下回り、その下回る量がしきい値を超える場合、スロットルモジュールは、より少数の A コアでスレッドを実行することによって、より少数のスレッドの実行への移行を開始することができる。以前にオンにされていた B コアは、それら少数のスレッドをサポートするのにもはや必要とされる場合はないので、それらの B コアをオフにすることができ、A コアをオンにして、それら少数のスレッドをサポートすることができる。上述したように、ラージコアをオフにスイッチングした時点におけるスループット性能の不連続を低減するために、A コアと同時に少数の B コアを実行することを可能にすることが望ましい場合がある。

30

## 【 0 0 3 0 】

一実施の形態では、スロットルモジュールは、フィードバックループを必要としない方法で実施することができる。この点で、スロットルの制御動作（たとえば、どのタイプのいくつのコアがスレッドを実行するかの決定）が、入力値（たとえば、スレッドのコアの割り当て及び構成）に影響を与えるために戻されることはない。この実施の形態では、各 A コア 2 2 0、2 2 2 は、B コア 2 3 0 ~ 2 6 2 の中の 2 5 個と同じ電力量を消費できるものと仮定することができる。他の実施の形態では、異なる電力消費の比を使用することができる。プロセッサは、自身の全電力バジェットを 2 つの部分に分割することができる。各部分について、その電力バジェットにより、1 つの A コア及び 5 個までの B コアが同時に動作することを可能にすることもできるし、A コアが動作せず、3 0 個までの B コアが同時に動作することを可能にすることもできる。他の実施の形態では、電力バジェットは、他の方法で複数の部分に分割することができる。

40

## 【 0 0 3 1 】

一実施の形態では、表 I に従って、実行スレッド（R T）の個数を或る数量の A コア（Q A C）及び或る数量の B コア（Q B C）に割り当てることができる。

50



## 【 0 0 3 2 】

## 【表 1】

表 I

R T	Q A C	Q B C	R T	Q A C	Q B C
0	0	0	2 9	1	2 8
1	1	0	3 0	1	2 9
2	2	0	3 1	1	3 0
3	2	1	3 2	1	3 1
4	2	2	3 3	1	3 2
...			3 4	1	3 3
1 0	2	8	3 5	1	3 4
1 1	2	9	3 6	1	3 5
1 2	2	1 0	3 7	0	3 7
1 3	1	1 2	3 8	0	3 8
1 4	1	1 3	3 9	0	3 9
1 5	1	1 4	4 0	0	4 0
...			...		
2 8	1	2 7	6 0	0	6 0

10

## 【 0 0 3 3 】

実行スレッドの個数が増加し、新しいスレッドが（一実施の形態では、プロセッサ間割り込みを介して）起動されると、スロットルモジュールは、現在の実行スレッドの個数を求めることができる。現在の実行スレッドの個数に応じて、上記表 I に従い、新しいスレッドを A コア又は B コアのいずれかに割り当てることができる。この実施の形態では、1 2 個のスレッドから 1 3 個のスレッドに増加した時、又は 3 6 個のスレッドから 3 7 個のスレッドに増加した時等の特定の場合に、A コアで実行されている既存のスレッドは、B コアにおける実行にマイグレーションされる。このマイグレーションが完了すると、マイグレーションされた既存のスレッド及び新しいスレッドの双方を起動することができる。この理由により、この実施の形態では、新しいスレッドは、起動の際に遅延を示す場合がある。

20

## 【 0 0 3 4 】

同様のプロセスは、実行スレッドの個数が減少した時にも発生し得る。特定のスレッドが終了し、そのコアが停止されると、さまざまな方法を使用して、残りのスレッドの 1 つを B コアにおける実行から A コアにおける実行へ、場合によりマイグレーションすることができる。これは、たとえば、実行スレッドの個数を 1 3 個のスレッドから 1 2 個のスレッドへ、又は 3 7 個のスレッドから 3 6 個のスレッドへ削減する時に発生する可能性がある。一実施の形態では、周期タイマを使用して、特定の時間間隔に 1 度だけマイグレーションを可能にすることができる。これによって、スレッドが高速に作成されて終了される場合に、スレッドマイグレーションがあまりにも頻繁に行われることを有利に防止することができる。影響を受けるスレッドは、特定の時間間隔までの間は、B コアで引き続き実行し続けることができる。

30

## 【 0 0 3 5 】

一実施の形態では、スロットルモジュールは、A コアから B コアへのマイグレーションを、ソフトウェアにトランスペアレントに行うことができる。スロットルモジュールのスレッドマイグレーションメカニズムは、論理コアを物理コアへマッピングするための表と、コアマイグレーションが必要とされていることを信号で伝えるための割り込みと、コアのプロセッサ状態をコピーするためのマイクロコード又は配線ロジックと、プロセッサのコア間の相互接続ネットワークとを含むことができる。論理コアの個数は、B コアの個数と等しくすることができる。

40

## 【 0 0 3 6 】

別の実施の形態では、スロットルモジュールは、A コアから B コアへのマイグレーションを、ソフトウェアにトランスペアレントでない方法で行うことができる。このスレッドマイグレーションは、オペレーティングシステムのスケジューラが行うことができる。オペレーティングシステムは、現在の実行スレッドを有するコアの個数を追跡して、新しい

50

スレッドをコアに割り当て、AコアからBコア（又はBコアからAコアへ）スレッドをマイグレーションすることができる。ソフトウェアスレッドマイグレーションは、ハードウェア実施態様について上述したものと同等の機能を使用することができる。一実施の形態では、スロットルモジュールの動作は、アプリケーションプログラムにはトランスペアレントとすることができるが、オペレーティングシステムにはトランスペアレントでないものとすることができる。

【0037】

電力消費を調節するための1つの代替的な方法は、ロジックブロックのサイズ又は機能を調整することとすることができる。たとえば、大きなアレイサイズが必要とされない場合に、可変サイズのスケジューラ、キャッシュ、変換索引バッファ（TLB）、分岐予測、及び他のオプションの性能回路を使用して、スイッチング容量（したがって、エネルギー）を削減することができる。アレイを動的にサイズ変更することに加えて、実行ユニット、パイプライン段、及び他のオプションの性能回路を動的に無効化することによって、コアの性能をより小さなコアの性能に低下させるラージコアを設計することも可能である。これらの技法は、まとめて適応型処理として既知である場合がある。

10

【0038】

チップレベルマルチプロセッサの一実施の形態は、以下のように動作することができる。スレッドの並列性が低い段階では、少数のコアが、良好なスカラー性能を得るために、各コアにおける第1の一組の（たとえば、すべて又は多くの）利用可能なオプションの性能回路を使用して実行することができる。スレッドの並列性が高い段階では、良好なスループット性能を得るために、各コアにおいて使用するオプションの性能回路をより少なくして多くのコアを動作させることができる。

20

【0039】

アレイサイズの削減及び実行ユニットの無効化の最終結果は、1命令当たりにトグルされる静電容量を削減することになる場合がある。しかしながら、スイッチング容量は、最初からより小さなコアを設計するほど多く削減されない場合がある。未使用の実行ハードウェアはゲート制御でオフにできるが、コアの物理サイズは変化せず、したがって、まだアクティブであるハードウェアブロックに関連したワイヤの長さは、スモールコアにおける長さよりも依然として長いままである場合がある。

【0040】

1命令当たりの削減可能なエネルギーの推定は、大きなアウトオブオーダーマイクロプロセッサの配置図（floorplan）を調べて、そのプロセッサを小さなインオーダーマシンに変換するためにオプションの性能回路をいくつオフにできるかを求めることにより行うことができる（ブロックは物理的に移動できないことに留意されたい）。オフにされたプロセッサコア面積の割合をその後定量化することができる。この定量化によって、スイッチング容量の削減を近似することができる。式（2）から、1命令当たりのエネルギーは、スイッチング容量の量にほぼ比例する。

30

【0041】

粗い推定は、場合により、スイッチング容量の50%までをオフにできるということになり、その結果、1命令当たりのエネルギーの1倍から2倍の削減となる。いくつかの実施の形態では、クロックゲーティングに加えて、ダイナミックスリープランジスタ及びボディバイアス等の漏れ制御技法を使用することにより、1命令当たりに消費されるエネルギーの削減を容易にすることができる。

40

【0042】

次に図3を参照すると、一実施の形態による、オプションの性能回路によって構成可能なコアを含むプロセッサの概略図が示されている。この図3のプロセッサは、コア1 320、コア2 370、コア3 380、及びコア4 390の4つのコアを含むことができる。他の実施の形態では、4つよりも多くのコア又は少ないコアを使用することができる。コア1 320は、さまざまなオプションの性能回路を示している。スケジューラ A 334は、オンにされると性能を高めることができるオプションのスケジューラ B

50

336に接続することができる。実行ユニット1 340は、オンにされると性能を高めることができるオプションの実行ユニット2~4の342、344、346に接続することができる。レベル0(L0)キャッシュA 322は、オンにされると性能を高めることができるL0キャッシュB 324に接続することができる。TLB A 326は、オンにされると性能を高めることができるTLB B 328に接続することができる。再配列バッファ(ROB)338は、可変の個数のラインを有することができる、すなわち、完全にオフしてアウトオブオーダー実行を防止することができる。最後に、他のパイプライン段330とは別個のプリフェッチ段332は、電源が投入されると、投機的フェッチを行うことができる。他の実施の形態では、他のオプションの性能回路を使用することができる。

10

#### 【0043】

スロットルモジュール310は、この場合も、情報を収集して、実行しているソフトウェアプログラムに存在する並列性の量を求めるのに使用することができる。スロットルモジュール310は、図1及び図2に関して上述したものと同様のものとすることができる。スロットルモジュール310は、プログラムにおける並列性の量を求めるごとに、信号線312、314、316、及び318を介してコア320、370、380、及び390に、電源が投入又は切断されるオプションの性能回路の個数を変更するように指示することができる。一実施の形態では、信号線312、314、316、及び318は、コア320、370、380、及び390をオン又はオフにするのに使用することもできる。一実施の形態では、現在の並列性の量が以前の量を超え、その超える量がしきい値を超える場合、スロットルモジュールは、各コアでオンにされているオプションの性能回路を減少させながらより多数のこれらのコアでスレッドを実行することによって、より多数のスレッドの実行への移行を開始することができる。以前にオフにされていたコアをオンにして、より多数のスレッドをサポートすることができる。同様に、現在の並列性の量が以前の量を下回り、その下回る量がしきい値を超える場合、スロットルモジュールは、いくつかのコアでオンにされているオプションの性能回路を増加させながらより少数のこれらのコアでスレッドを実行することによって、より少数のスレッドの実行への移行を開始することができる。以前にオンにされていたいくつかのコアは、より少数のスレッドをサポートするのにもはや必要とされる場合はないので、それらのコアをオフにすることができる。

20

30

#### 【0044】

たとえば予測ミスした分岐に続く命令といった投機ミスした命令により浪費されるエネルギーを削減するためのさまざまな形態の投機制御が提案されている。追加されるエネルギーは、投機ミスした命令を処理するためにトグルされる静電容量に起因する。投機ミスした命令の結果は廃棄することができるが、エネルギーはすでに費やされている。このエネルギーは、次の正しく投機(リタイヤ)された命令にこのエネルギーを充電することによって考慮することができる。

#### 【0045】

パイプラインゲーティングは、1つ又は複数の信頼性の低い分岐予測によって廃棄される可能性のある命令でパイプラインが満たされることを回避するための技法である。投機制御を使用する一実施の形態では、チップレベルマルチプロセッサは、以下のように動作することができる。スレッドの並列性が低い段階では、良好なスカラー性能を得るために、投機の量を増加させて少数のコアを実行することができる。スレッドの並列性が高い段階では、良好なスループット性能を得るために、各コアにおける投機の量を減少させて多くのコアを実行することができる。

40

#### 【0046】

スケジューラ及び再配列バッファのエントリーの個数を削減することによって、投機できる命令の個数も削減されるので、可変サイズコア技法と投機制御との間には或る重複が存在する。キャッシュ、TLB、分岐予測器等の他のプロセッサ資源のサイズは、可能な投機の量にそれほど大きな影響を与えることはできない。

50

## 【0047】

次に図4を参照すると、本開示の一実施の形態による、オプションの投機回路によって構成可能なコアを含むプロセッサの概略図が示されている。この図4のプロセッサは、コア1 420、コア2 470、コア3 480、及びコア4 490の4つのコアを含むことができる。他の実施の形態では、4つよりも多くのコア又は少ないコアを使用することができる。コア1 420は、さまざまなオプションの投機回路を示している。プリフェッチ段430は、命令を投機的にフェッチするのに使用することができる。分岐予測器434は、プログラムの分岐の投機的実行をサポートするのに使用することができる。他の予測器436は、実施の形態によっては、ループ予測器とすることもできるし、他の形態の投機的実行をサポートするための他の形態の予測器とすることもできる。他の実施の形態では、さらに別のオプションの投機回路を使用することができる。

10

## 【0048】

スロットルモジュール410は、この場合も、情報を収集して、実行しているソフトウェアプログラムに存在する並列性の量を求めるのに使用することができる。スロットルモジュール410は、一般的に、図1、図2、及び図3に関して上述したように動作することができる。スロットルモジュール410は、プログラムにおける並列性の量を求めるごとに、信号線412、414、416、及び418を介してコア420、470、480、及び490に、電源が投入又は切断されるオプションの投機回路の個数を変更するように指示することができる。一実施の形態では、信号線412、414、416、及び418は、コア420、470、480、及び490をオン又はオフにするのに使用することもできる。一実施の形態では、現在の並列性の量が以前の量を超え、その超える量がしきい値を超える場合、スロットルモジュールは、各コアでオンにされているオプションの投機回路を減少させながらより多数のこれらのコアでスレッドを実行することによって、より多数のスレッドの実行への移行を開始することができる。以前にオフにされていたコアをオンにして、より多数のスレッドをサポートすることができる。同様に、現在の並列性の量が以前の量を下回り、その下回る量がしきい値を超える場合、スロットルモジュールは、いくつかのコアでオンにされているオプションの投機回路を増加させながらより少数のこれらのコアでスレッドを実行することによって、より少数のスレッドの実行への移行を開始することができる。以前にオンにされていたいくつかのコアは、より少数のスレッドをサポートするのに必要とされる場合はないので、それらのコアをオフにすることができる。

20

30

## 【0049】

図1、図2、図3、及び図4に関して上述した技法のそれぞれについて、スロットルモジュールは、マルチプロセッサの動作を調整して、全チップ電力を一定の電力バジェット内に維持することができる。スロットルモジュールハードウェアは、1秒当たりリタイヤされる命令の総数に反比例する、1命令当たりのエネルギー量を変化させることによって、式(3)を満たすことができる。過電力状況に応じて、いくつかの実施の形態では、スロットルモジュールは、以下の動作、すなわち、電圧及び周波数を下げること(図1の電圧及び周波数のスケールリングの場合)、ラージコアからスモールコアへスレッドをマイグレーションすること(図2の非対称型コアの場合)、オプションの性能回路部の容量を削減すること(図3の可変サイズのコアの場合)、又は、投機の量を削減すること(図4の投機制御の場合)の1つ又は複数を取ることができる。

40

## 【0050】

各場合において、たとえ正味のスループットが増加しても、ソフトウェアはより多くのスレッドを同時に実行するようにハードウェアに依頼するので、既存のスレッドはより遅くなるという独特の特性を有するが、ソフトウェアプログラムは、プロセッサを大きな対称型チップレベルマルチプロセッサとみなすことができる。この手法によれば、今日の共有メモリマルチプロセッサプログラミングモデル用に記述されたソフトウェアを、変更することなく、引き続き実行することができる。

## 【0051】

50

次に図5を参照すると、本開示の一実施の形態による、コア、及び、スロットルモジュールの細部を含むプロセッサの概略図が示されている。数量Mのプロセッサコア1502~M508は、実施の形態によっては、電圧及び周波数によって構成可能な図1のコア、処理電力及び電力消費が変化する図2のラージコア及びスモールコア、オプションの性能回路によって構成可能な図3のコア、オプションの投機回路によって構成可能な図4のコア、又は、これらのタイプの一部又は全部の混在したものとすることができる。他の実施の形態では、他のタイプのコアを使用することができる。

【0052】

モニタ1512~M518は、関連するコア及び補助的なプロセッサ回路部の1つ又は複数の属性を監視することができる。いくつかの実施の形態では、これらの属性には、コアが実行されているのか又は停止されているのか、1クロックサイクル当たりリタイヤされる命令の瞬間の個数、コアの所与の機能ブロックがアクティブであるのか又はアイドルであるのか、コア内の機能ブロック以外のプロセッサの所与の機能ブロックがアクティブであるのか又はアイドルであるのか、並びに、コアの電源電流及び温度（又はダイの温度）が含まれ得る。コア内の機能ブロック以外のプロセッサの機能ブロックには、共有キャッシュ又はメモリコントローラ等の回路が含まれ得る。電源電流は、電流センサ抵抗器及びアナログ/デジタル(A/D)変換器を介して測定することができる。温度は、サーマルダイオード及びA/D変換器を介して測定することができる。各モニタ1512~M518の出力は、これらの属性のすべて又は或る部分の監視値とすることができる。

【0053】

電力化変換(convert-to-power)ロジック530は、モニタ1512~M518から1つ又は複数の監視された値を受け取ることができる。さまざまな実施の形態では、これら監視された値は、電力消費の大きさに変換することができる。監視された値が、コアが実行されているのか又は停止されているのかを反映する一実施の形態では、監視された値は、コアが実行（又は停止）状態にある間に1クロック当たり費やされる平均エネルギー量をコアの実行（又は停止）状態に乗算することによって、電力消費に変換することができる。これらの積は、その後、プロセッサのすべてのコアについて合計することができる。監視された値が、1クロックサイクル当たりリタイヤされる命令の瞬間の個数を反映する一実施の形態では、監視された値は、その命令数に、1クロック当たりの命令ごとに費やされる平均エネルギー量を乗算することによって電力消費に変換することができる。これらの積は、その後、プロセッサのすべてのコアについて合計することができる。監視された値が、コアの所与の機能ブロックがアクティブであるのか若しくはアイドルであるのか（又はいくつかの実施の形態では、コア内の機能ブロック以外のプロセッサの所与の機能ブロックがアクティブであるのか若しくはアイドルであるのか）を反映する一実施の形態では、監視された値は、ブロックがアクティブ（又はアイドル）状態である間に1クロック当たり費やされる平均エネルギー量をアクティブ（又はアイドル）状態に乗算することによって電力消費に変換することができる。これらの積は、その後、各コアのすべてのブロックについて合計することができ、そして、プロセッサのすべてのコアについて合計することができる。コアの内部にないブロックが考慮される場合、その積を合計に含めることができる。これらの実施の形態のそれぞれにおいて、結果の合計は、電源電圧の2乗をクロック周波数に乗算したものに比例する値を乗算することによって、電圧及び周波数について調整することができる。

【0054】

監視された値が各コアの電源電流を反映する一実施の形態では、監視された値は、電源電流に電源電圧を乗算することによって電力消費に変換することができる。

【0055】

監視された値がコア又はダイの温度を反映する一実施の形態では、図5の回路を使用して、ほぼ一定の電力消費を維持する代わりに、ほぼ一定のコア又はダイの温度を維持することができる。

10

20

30

40

50

## 【0056】

計算差ロジック534は、電力化変換ロジック530の変換された電力消費（又はダイの温度）出力と所望の電力消費値（又は所望のダイの温度値）との間の差を計算するのに使用することができる。この差は、「誤差」値として使用することができる。

## 【0057】

積分ロジック538は、一実施の形態では、計算差ロジック534によって供給された誤差値の時間にわたる積分に比例する積分値を提供することができる。一実施の形態では、この積分は、アキュムレータを使用することによって計算することができる。このアキュムレータは、加算器及びレジスタを使用して実施することができる。アキュムレータは、オーバフローを回避するために上限及び下限の双方で飽和させることができる。一実施の形態では、アキュムレータの出力は、たとえば、2～6ビットの整数ビット及び2～16ビットの小数ビットを有する固定小数点の2進数とすることができる。アキュムレータの最上位ビットを調べることが可能である。これによって、出力をゆるやかに変化させるという特性を有利に与えることができる。積分ロジック538の使用によって、低速フィードバックループ550が時間と共に誤差値を0に削減しようとするのを助けることができる。積分ロジック538の使用は、誤差値が電力である場合に最も有利であり得る。その理由は、瞬間電力消費は、ほぼクロックサイクルごとに変化し得るからである。

10

## 【0058】

他の実施の形態では、積分ロジック538を、誤差値に比例する値を与えるロジックと取り替えることができる。このロジックは、誤差値が温度である場合に有利となり得る。別の実施の形態では、このロジックは、誤差値の時間微分又は上記値のすべての或る線形結合に比例した値を与えることができる。

20

## 【0059】

サンプルロジック544は、特定の実施の形態で使用することができる。たとえ積分ロジック538からの積分値が理論上は図5のロジックの速度に比べてゆるやかに変化する場合があっても、その集積値は、各クロックサイクルと共に変化する場合がある。いくつかの制御技法は、システムコストをほとんど伴うことなく、クロックサイクルごとにそれらの動作を変化させることができる。この場合、サンプリングは必要とされない場合がある。他の制御技法では、積分ロジック538の出力が遷移点に近くなった時に、各クロックと共に振動する値を低減することを助けるのにサンプリングを使用することが有利な場合がある。

30

## 【0060】

一実施の形態では、一定サンプリング技法をサンプルロジック544に使用することができる。一定数Nのクロックサイクルごとに1回値をサンプリングすることができる。これによって、値が、Nクロックサイクルごとに1回よりも高速に変化することを防止することができる。しかしながら、この技法は、Nクロックサイクルよりも短い期間の間制御を行うことを可能にすることができず、積分が、Nクロックサイクルの期間と比較して遅い場合に、最もよく機能することができる。

## 【0061】

別の実施の形態では、ヒステリシス技法をサンプルロジック544に使用することができる。サンプルロジック544の出力は、入力値と出力値との間の差がしきい値Tを超えた時にのみ変化することができる。この出力値は、差がTを超えた時に入力値と取り替えることができ、そうでない時、出力値は変わらないままにすることができる。

40

## 【0062】

低速フィードバックループ550を使用する実施の形態の場合、制御ロジック548を使用することができる。コアが、電源電圧及び周波数を変化させることによって電力消費を変化させることができる図1の実施の形態等の一実施の形態では、制御ロジック548は、電源電圧及び周波数の変更を指示することができる。これらの実施の形態では、電源電圧を変更することは、100マイクロ秒等の比較的長い時間を要する場合があることに留意すべきである。一実施の形態では、表又は論理ブロックを使用して、電源電圧及び周

50

波数を設定することができる。表 I I は、制御値 ( C V ) のさまざまな値についてのこのような表又は論理ブロックの一例を与えている。

【 0 0 6 3 】

【表 2】

C V	ボルト	周波数 ( G H z )
0	1. 4	4
1	1. 4	4
2	1. 4	4
3	1. 2	3. 3
4	1. 1	2. 9
5	1. 0	2. 5
6	1. 0	2. 5
7	0. 9	2. 2

10

【 0 0 6 4 】

スレッドをラージコア及びスモールコアに割り当てることができる図 2 の実施の形態等の一実施の形態では、制御ロジック 9 4 8 は、制御値に従って、ソフトウェアに対してトランスペアレントにスレッドをコアに割り当てることができ、且つ、スレッドをマイグレーションすることができる。スレッドマイグレーションメカニズムは、「論理」プロセッサコアを「物理」プロセッサコアにマッピングするための表又は論理ブロックを含むことができる。また、スレッドマイグレーションメカニズムは、マイグレーションが要求されていることを信号で伝えるための割り込みと、プロセッサ状態をコピーするためのマイクロコード又は同等のロジックと、さまざまなプロセッサコア間の相互接続ネットワークとを含むこともできる。ソフトウェアに見える論理プロセッサの個数は、実行されているスモールコアの個数と等しくすることができる。電力バジェットが許す場合には、スモールコアの代わりにラージコアを使用することができる。ラージコアのこのような使用を制御値の関数として配分する ( budget ) のに使用される表又は論理ブロックは、一実施の形態では、表 I I I に与えるものとするすることができる。

20

【 0 0 6 5 】

【表 3】

C V	ラージコアの数量
0	2
1	1
2	0

30

【 0 0 6 6 】

ここで、ラージコアを使用できない時にスモールコアでスレッドを実行するという簡単な仮定を行うことができる。

【 0 0 6 7 】

資源を動的に変化させることができるコアにスレッドを割り当てることができる図 3 の実施の形態等の一実施の形態では、制御ロジック 5 4 8 は、コアの機能ユニットの数量又は容量を変更できる制御値を与えることができる。実施の形態によっては、コアの容量を変化させるのに、アレイを満たすか若しくはライトバックする必要がある場合、又は、パイプラインを補充する必要がある場合に、この変更を使用される時間は、10 マイクロ秒の範囲となる場合がある。コアの機能ユニットの使用を制御値の関数として配分するのに、表又は論理ブロックを使用することができる。機能ユニットのこのような使用を制御値の関数として配分するのに使用される表又は論理ブロックは、一実施の形態では、表 I V に与えるものとするすることができる。ここで、機能ユニットは、コアの実行ユニットとすることができる。

40

【 0 0 6 8 】

【表 4】

表 I V

C V	実行ユニット
0	4
1	4
2	4
3	2
4	2
5	1
6	1
7	1
8	1

10

【 0 0 6 9 】

投機の量を動的に変化させることができるコアにスレッドを割り当てることができる図 4 の実施の形態等の一実施の形態では、制御ロジック 5 4 8 は、コアで許可される投機の量を変更できる制御値を与えることができる。コアにおける投機の量の使用を制御値の関数として配分するのに表又は論理ブロックを使用することができる。投機の量を制御値の関数として調整するのに使用される表又は論理ブロックは、一実施の形態では、表 V に与えるものとして与えることができる。ここで、投機の量は、コアにおいて、非投機的な命令よりも先に投機的に実行される命令の個数によって与えることができる。

【 0 0 7 0 】

【表 5】

20

表 V

C V	投機的命令
0	1 2 8
1	1 2 8
2	1 2 8
3	8 5
4	6 4
5	5 1
6	4 2
7	3 6
8	3 2

30

【 0 0 7 1 】

多くの実施の形態では、プロセッサコアは、上記属性のいくつかを含むことができる。たとえば、プロセッサは、電圧及び周波数のスケールアップが可能なコアを有することができる、また、調整可能な量の投機的実行が可能なコアも有することができる。一実施の形態では、図 2 のラージ A コア及びスモール B コアも、図 1 に示すような電圧及び周波数のスケールアップを有することができる。一実施の形態について、電圧及び周波数は一定とすることができる、ラージ A コアは高電圧で実行するように構成され、スモール B コアは低電圧で実行するように構成される。別の実施の形態では、電圧及び周波数の調整範囲が、ラージ A コアとスモール B コアとの間で異なったものとして与えることができる。ラージ A コアとスモール B コアとの間のマイグレーションを援助するために、スモール B コアが最初に、高電圧及び高周波数で実行されている場合に、ラージ A コアからスモール B コアへスレッドをマイグレーションすることができる。その B コアの電圧及び周波数は、その後、ラージ A コアが実行されなくなると、下げることができる。

40

【 0 0 7 2 】

この場合も、スレッドをラージ A コア又はスモール B コアに割り当てるのに表又は論理ブロックを使用することができる、さらに、スレッドをコアの電圧及び周波数に割り当てるのに表又は論理ブロックを使用することができる。制御値の関数としてこれらの調整を行うのに使用される表又は論理ブロックは、一実施の形態では、表 V I に与えるものとして

50



ることができる。

【0073】

【表6】

表V1

C V	ラージコアの数 量	電圧	周波数
0	2	1.4	4
1	2	1.4	4
2	1	1.4	4
3	1	1.4	4
4	0	1.4	4
5	0	1.2	3.3
6	0	1.1	2.9
7	0	1.0	2.5
8	0	1.0	2.5
9	0	0.9	2.2

10

【0074】

低速フィードバックループ550を巡って幾分一定の利得を維持することが有利な場合がある。この幾分一定の利得は、このような表又は論理ブロックの各段階の電力効果を同様の大きさにすることにより達成することができる。

20

【0075】

特定の実施の形態では、前述した低速フィードバックループ550と併せて高速フィードバックループ560を使用することができる。一実施の形態では、たとえクロックスロットルを行うことが、1命令当たりのエネルギー量に影響を与えることができなくても、クロックスロットル540を使用することができる。クロックスロットル540を使用することは、上述した低速フィードバックループ550技法の1つ又は複数と組み合わせて使用されると有利な場合がある。

【0076】

特定の場合に、待ち時間の短い高速フィードバックループ560を適用して、過電力状況を防止することができ、この適用は、低速フィードバック550が応答時間を有するまでしか効力を持続することができない。高速フィードバックループ960が実施されない実施の形態では、プロセッサ及びその電力供給システムは、低速フィードバックループ550の応答時間の間持続する過電力状況をハンドリングするように設計されるべきである。高速フィードバックループ560が追加される場合、過電力状況は、高速フィードバックループ560の応答時間よりも長く持続することはできない。この応答時間は、10ナノ秒の範囲にすることができる。

30

【0077】

クロックスロットル540は、その入力として計算差ロジック534で計算された誤差値を使用して、特定のデューティサイクルでコアクロックのオン及びオフをゲート制御するように動作することができる。1つの簡単な実施の形態では、誤差値が一定のしきい値を超えると、所与のサイクル数の間、クロックを停止することができる。この計算は、各クロックサイクルにおいて行うことができ、完全にパイプライン化することができる。この所与のサイクル数は、その電力を、(低速フィードバックループ550が応答することを可能にすることができる)所望の電力よりも多少大きい電力に制限するように選ぶことができるが、最大電力よりも大きくするように選ぶことはできない。

40

【0078】

別の実施の形態では、クロックスロットル550は、計算差ロジック535によって供給された誤差値の大きさ(たとえば、プロセッサが、電力しきい値をどの程度上回って現在実行されているか)に応じて、デューティサイクルを調節することができる。一実施の

50

形態では、デューティサイクル及び誤差値を表 V I I のように関係付けることができる。

【 0 0 7 9 】

【表 7】

表 V I I			
誤差値	デューティサイクル	誤差値	デューティサイクル
- 2	1 0 0 %	7	1 0 0 %
- 1	1 0 0 %	8	9 4 %
0	1 0 0 %	9	8 8 %
1	1 0 0 %	1 0	8 1 %
2	1 0 0 %	1 1	7 5 %
3	1 0 0 %	1 2	6 9 %
4	1 0 0 %	1 3	6 3 %
5	1 0 0 %	1 4	5 6 %
6	1 0 0 %	1 5	5 0 %

10

【 0 0 8 0 】

次に図 6 を参照すると、本開示の一実施の形態による、異なるコア構成への移行を示すフローチャートが示されている。このプロセスは、ブロック 6 1 0 で開始する。ブロック 6 1 0 は、さまざまなソフトウェアスレッドを利用可能なコアに割り当てることができる。一実施の形態では、スレッドの個数が求められ、各スレッドは 1 つのコアに割り当てられる。実施の形態によっては、この個数を求めることは、オペレーティングシステム等のソフトウェアが行うことができる。他の実施の形態では、この個数を求めることは、ハードウェアロジックが行うこともできるし、ハードウェア及びソフトウェアを組み合わせたものが行うこともできる。ブロック 6 1 4 において、各コアによって消費される電力量が監視され、観測された電力消費と所望の電力消費との間の誤差値が計算される。さまざまな実施の形態では、図 5 に関して上述した監視技法のいずれも使用することができる。次に、ブロック 6 1 8 において、ブロック 6 1 4 の誤差値は、図 5 に関して上述したような積分技法若しくはサンプリング技法又はその双方を受けることができる。実施の形態によっては、ブロック 6 1 8 はオプションとすることができる。ブロック 6 1 8 の出力は、或る期間の間の誤差値が、コアの構成を変更する性能ペナルティを正当とするのにほぼ十分であると判断できるまでコアの構成変更を防止できる制御値とすることができる。

20

30

【 0 0 8 1 】

判定ブロック 6 2 2 において、時間中の誤差値がコアの構成の変更を正当とするのに十分であるかどうかの判断を行うことができる。差が十分でない場合、プロセスは、N O の経路を通過して判定ブロック 6 2 2 を出ることができ、プロセスは繰り返す。一方、差が十分である場合、プロセスは、Y E S の経路を通過して判定ブロック 6 2 2 を出ることができる。

【 0 0 8 2 】

ブロック 6 2 6 において、電圧及び周波数を制御値に従って変更することができる。一実施の形態では、電圧及び周波数は、上記の表 I I で示したように制御値について変更することができる。一実施の形態では、さまざまなコアの電圧及び周波数をすべて同様に変更することができる。他の実施の形態では、電圧及び周波数をさまざまなコアにおいて異なって変更することができる。いずれの場合にも、プロセスはブロック 6 1 0 で繰り返す。

40

【 0 0 8 3 】

次に図 7 を参照すると、本開示の一実施の形態による、異なるコア構成への移行を示すフローチャートが示されている。このプロセスは、ブロック 7 1 0 で開始する。ブロック 7 1 0 は、さまざまなソフトウェアスレッドを利用可能なコアに割り当てることができる。一実施の形態では、スレッドの個数が求められ、各スレッドは 1 つのコアに

50

割り当てられる。実施の形態によっては、この個数を求めることは、オペレーティングシステム等のソフトウェアが行うことができる。他の実施の形態では、この個数を求めることは、ハードウェアロジックが行うこともできるし、ハードウェア及びソフトウェアを組み合わせたものが行うこともできる。ブロック 714 において、各コアによって消費される電力量が監視され、観測された電力消費と所望の電力消費との間の誤差値が計算される。さまざまな実施の形態では、図 5 に関して上述した監視技法のいずれも使用することができる。次に、ブロック 718 において、ブロック 714 の誤差値は、図 5 に関して上述したような積分技法若しくはサンプリング技法又はその双方を受けることができる。実施の形態によっては、ブロック 718 はオプションとすることができる。ブロック 718 の出力は、或る期間の間の誤差値が、コアの構成を変更する性能ペナルティを正当とするのにほぼ十分であると判断できるまでコアの構成変更を防止できる制御値とすることができる。

10

**【 0084 】**

判定ブロック 722 において、時間中の誤差値がコアの構成の変更を正当とするのに十分であるかどうかの判断を行うことができる。差が十分でない場合、プロセスは、NO の経路を通過して判定ブロック 722 を出ることができ、プロセスは繰り返す。一方、差が十分である場合、プロセスは、YES の経路を通過して判定ブロック 722 を出ることができる。

**【 0085 】**

ブロック 726 において、制御値に従ってコア間にスレッドを再割り当てすることができる。一実施の形態では、スレッドは、上記の表 III で示したように、制御値についての特定の個数のラージコアに再割り当てすることができる。一実施の形態では、ラージコアの個数は、表 III のように与えられ、ラージコアで実行できないスレッドは、スモールコアに再割り当てすることができる。他の実施の形態では、スレッドをコアに再割り当てするための他の方式を使用することができる。いずれの場合にも、プロセスはブロック 710 で繰り返す。

20

**【 0086 】**

次に図 8 を参照すると、本開示の一実施の形態による、異なるコア構成への移行を示すフローチャートが示されている。このプロセスは、ブロック 810 で開始する。ブロック 810 は、さまざまなソフトウェアスレッドを利用可能なコアに割り当てることを含むことができる。一実施の形態では、スレッドの個数が求められ、各スレッドは 1 つのコアに割り当てられる。実施の形態によっては、この個数を求めることは、オペレーティングシステム等のソフトウェアが行うことができる。他の実施の形態では、この個数を求めることは、ハードウェアロジックが行うこともできるし、ハードウェア及びソフトウェアを組み合わせたものが行うこともできる。ブロック 814 において、各コアによって消費される電力量が監視され、観測された電力消費と所望の電力消費との間の誤差値が計算される。さまざまな実施の形態では、図 5 に関して上述した監視技法のいずれも使用することができる。次に、ブロック 818 において、ブロック 814 の誤差値は、図 5 に関して上述したような積分技法若しくはサンプリング技法又はその双方を受けることができる。実施の形態によっては、ブロック 818 はオプションとすることができる。ブロック 818 の出力は、或る期間の間の誤差値が、コアの構成を変更する性能ペナルティを正当とするのにほぼ十分であると判断できるまでコアの構成変更を防止できる制御値とすることができる。

30

40

**【 0087 】**

判定ブロック 822 において、時間中の誤差値がコアの構成の変更を正当とするのに十分であるかどうかの判断を行うことができる。差が十分でない場合、プロセスは、NO の経路を通過して判定ブロック 822 を出ることができ、プロセスは繰り返す。一方、差が十分である場合、プロセスは、YES の経路を通過して判定ブロック 822 を出ることができる。

**【 0088 】**

50

ブロック 8 2 6 において、上記の表 I V で与えるように、制御値に従った量の、コアのオプションの回路部をオン又はオフにすることができる。一実施の形態では、上記の表 I V に示すような制御値について、電源をオン又はオフにされる実行ユニットの個数を変更することができる。他の実施の形態では、他のオプションの性能回路を、制御値に従ってオン又はオフにすることができる。これらの他のオプションの性能回路には、スケジューラ、キャッシュ、変換索引バッファ、スケジューラ、及び再配列バッファが含まれ得る。いずれの場合にも、プロセスはブロック 8 1 0 で繰り返す。

#### 【 0 0 8 9 】

次に図 9 を参照すると、本開示の一実施の形態による、異なるコア構成への移行を示すフローチャートが示されている。このプロセスは、ブロック 9 1 0 で開始する。ブロック 9 1 0 は、さまざまなソフトウェアスレッドを利用可能なコアに割り当てることができる。一実施の形態では、スレッドの個数が求められ、各スレッドは 1 つのコアに割り当てられる。実施の形態によっては、この個数を求めることは、オペレーティングシステム等のソフトウェアが行うことができる。他の実施の形態では、この個数を求めることは、ハードウェアロジックが行うこともできるし、ハードウェア及びソフトウェアを組み合わせたものを行うこともできる。ブロック 9 1 4 において、各コアによって消費される電力量が監視され、観測された電力消費と所望の電力消費との間の誤差値が計算される。さまざまな実施の形態では、図 5 に関して上述した監視技法のいずれも使用することができる。次に、ブロック 9 1 8 において、ブロック 9 1 4 の誤差値は、図 5 に関して上述したような積分技法若しくはサンプリング技法又はその双方を受けることができる。実施の形態によっては、ブロック 9 1 8 はオプションとすることができる。ブロック 9 1 8 の出力は、或る期間の間の誤差値が、コアの構成を変更する性能ペナルティを正当とするのにほぼ十分であると判断できるまでコアの構成変更を防止できる制御値とすることができる。

#### 【 0 0 9 0 】

判定ブロック 9 2 2 において、時間中の誤差値がコアの構成の変更を正当とするのに十分であるかどうかの判断を行うことができる。差が十分でない場合、プロセスは、N O の経路を通過して判定ブロック 9 2 2 を出ることができ、プロセスは繰り返す。一方、差が十分である場合、プロセスは、Y E S の経路を通過して判定ブロック 9 2 2 を出ることができる。

#### 【 0 0 9 1 】

ブロック 9 2 6 において、コアにおける投機の量を、上記の表 V で与えるように制御値に従って変化させることができる。一実施の形態では、投機的に実行される命令の個数を、上記の表 V に示すような制御値について変更することができる。他の実施の形態では、他の方法を使用して、制御値に従って投機の量を変化させることができる。投機の量を変化させるためのこれらの他の方法は、プリフェッチユニット、分岐予測器、又は別の形態の予測器の電源をオン又はオフにすることを含むことができる。いずれの場合にも、プロセスはブロック 9 1 0 で繰り返す。

#### 【 0 0 9 2 】

次に図 1 0 A 及び図 1 0 B を参照すると、本開示の 2 つの実施の形態による、スロットル及び複数のコアを有するプロセッサを含むシステムの概略図が示されている。図 1 0 A のシステムは、プロセッサとメモリと入出力デバイスとがシステムバスによって相互接続されているシステムを一般的に示しているのに対して、図 1 0 B のシステムは、プロセッサとメモリと入出力デバイスとが複数のポイントツーポイントインターフェースによって相互接続されているシステムを一般的に示している。

#### 【 0 0 9 3 】

図 1 0 A のシステムは、1 つ又はいくつかのプロセッサを含むことができ、明確にするためにこれらプロセッサのうちの 2 つのプロセッサ 4 0、6 0 のみが図 1 0 A に示されている。プロセッサ 4 0、6 0 は、レベル 1 キャッシュ 4 2、6 2 を含むことができる。図 1 0 A のシステムは、システムバス 6 とのバスインターフェース 4 4、6 4、1 2、8 を

10

20

30

40

50

介して接続されているいくつかの機能を有することができる。一実施の形態では、システムバス6は、Intel（登録商標）社によって製造されたPentium（登録商標）クラスのマイクロプロセッサと共に利用されるフロントサイドバス（FSB）とすることができる。他の実施の形態では、他のバスを使用することができる。実施の形態によっては、メモリコントローラ34及びバスブリッジ32は、まとめてチップセットと呼ばれる場合がある。いくつかの実施の形態では、チップセットの機能は、図10Aの実施の形態で示したようなものとは異なり、複数の物理チップに分割することができる。

【0094】

メモリコントローラ34は、プロセッサ40、60がシステムメモリ10及び基本入出力システム（BIOS）消去可能プログラブル読み出し専用メモリ（EPROM）36から読み出し及び書き込みを行うことを可能にすることができる。いくつかの実施の形態では、BIOS EPROM36は、フラッシュメモリを利用することができる。メモリコントローラ34は、メモリ読み出しデータ及びメモリ書き込みデータをシステムバス6のバスエージェントへ運ぶこと、及び、当該バスエージェントから運ぶことを可能にするためのバスインターフェース8を含むことができる。また、メモリコントローラ34は、高性能グラフィックスインターフェース39を通じて、高性能グラフィックス回路38と接続することもできる。特定の実施の形態では、高性能グラフィックスインターフェース39は、アドバンスドグラフィックスポートAGPインターフェースとすることができる。メモリコントローラ34は、システムメモリ10からのデータを、高性能グラフィックスインターフェース39を通じて高性能グラフィックス回路38へ向けることができる。

【0095】

図10Bのシステムも、1つ又はいくつかのプロセッサを含むことができ、明確にするためにこれらプロセッサのうち2つのプロセッサ70、80のみが示されている。プロセッサ70、80は、それぞれ、メモリ2、4と接続するためのローカルメモリコントローラハブ（MCH）72、82を含むことができる。プロセッサ70、80は、ポイントツーポイントインターフェース回路78、88を使用して、ポイントツーポイントインターフェース50を介してデータを交換することができる。プロセッサ70、80は、それぞれ、ポイントツーポイントインターフェース回路76、94、86、98を使用して、個別のポイントツーポイントインターフェース52、54を介してチップセット90とデータを交換することができる。チップセット90も、高性能グラフィックスインターフェース92を介して高性能グラフィックス回路38とデータを交換することができる。

【0096】

図10Aのシステムでは、バスブリッジ32が、システムバス6とバス16との間のデータ交換を可能にすることができる。バス16は、実施の形態によっては、業界標準アーキテクチャ（ISA）バス又は周辺機器相互接続（PCI）バスとすることができる。図10Bのシステムでは、チップセット90は、バスインターフェース96を介してバス16とデータを交換することができる。いずれのシステムにも、バス16上にさまざまな入出力I/Oデバイス14が存在することができる。これらの入出力I/Oデバイスには、いくつかの実施の形態では、低性能グラフィックスコントローラ、ビデオコントローラ、及びネットワーク接続コントローラが含まれる。別のバスブリッジ18は、実施の形態によっては、バス16とバス20との間のデータ交換を可能にするのに使用することができる。バス20は、実施の形態によっては、小型コンピュータシステムインターフェース（SCSI）バス、統合ドライブエレクトロニクス（IDE）バス、又はユニバーサルシリアルバス（USB）バスとすることができる。追加のI/Oデバイスをバス20と接続することもできる。これらの追加のI/Oデバイスには、マウスを含むキーボード/カーソル制御デバイス22、音声I/O24、モデム及びネットワークインターフェースを含む通信デバイス26、並びにデータストレージデバイス28が含まれ得る。ソフトウェアコード30は、データストレージデバイス28に記憶することができる。実施の形態によっては、データストレージデバイス28は、固定磁気ディスク、フロッピーディスクドライブ、光ディスクドライブ、光磁気ディスクドライブ、磁気テープ、又はフラッシュメモリ

を含む不揮発性メモリとすることができる。

【0097】

上記明細書では、本発明をその特定の実施の形態に関して説明してきた。しかしながら、添付の特許請求の範囲で述べられているような本発明のより広い精神及び範囲から逸脱することなく、それら実施の形態に対してさまざまな修正及び変更を行えることは明らかである。本明細書及び図面は、したがって、限定的な意味ではなく例示の意味とみなされるべきである。

【図面の簡単な説明】

【0098】

【図1】一実施の形態による、電圧及び周波数によって構成可能なコアを含むプロセッサの概略図である。 10

【図2】一実施の形態による、処理電力及び電力消費によって選択可能なコアを含むプロセッサの概略図である。

【図3】一実施の形態による、オプションの性能回路によって構成可能なコアを含むプロセッサの概略図である。

【図4】本開示の一実施の形態による、オプションの投機回路によって構成可能なコアを含むプロセッサの概略図である。

【図5】本開示の一実施の形態による、コア、及び、スロットルの細部を含むプロセッサの概略図である。

【図6】本開示の一実施の形態による、異なるコア構成への移行を示すフローチャートである。 20

【図7】本開示の別の実施の形態による、異なるコア構成への移行を示すフローチャートである。

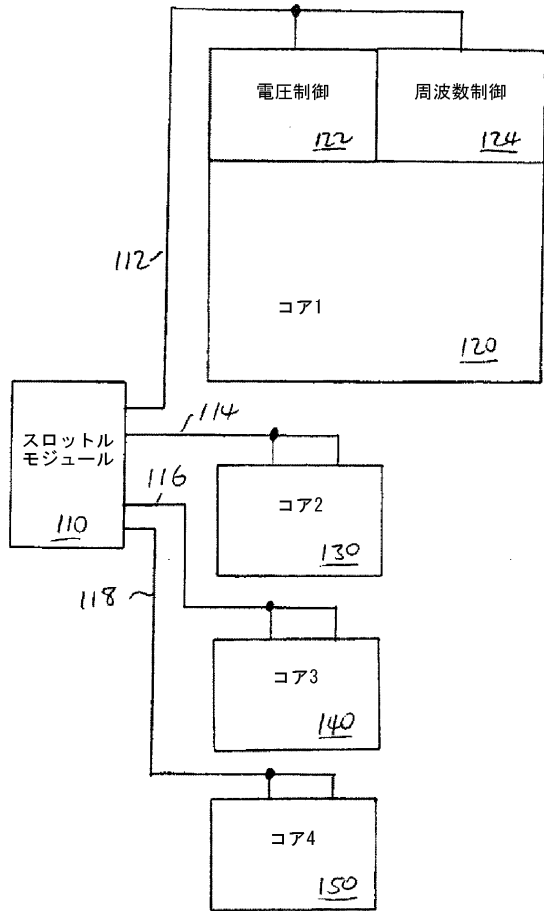
【図8】本開示の別の実施の形態による、異なるコア構成への移行を示すフローチャートである。

【図9】本開示の別の実施の形態による、異なるコア構成への移行を示すフローチャートである。

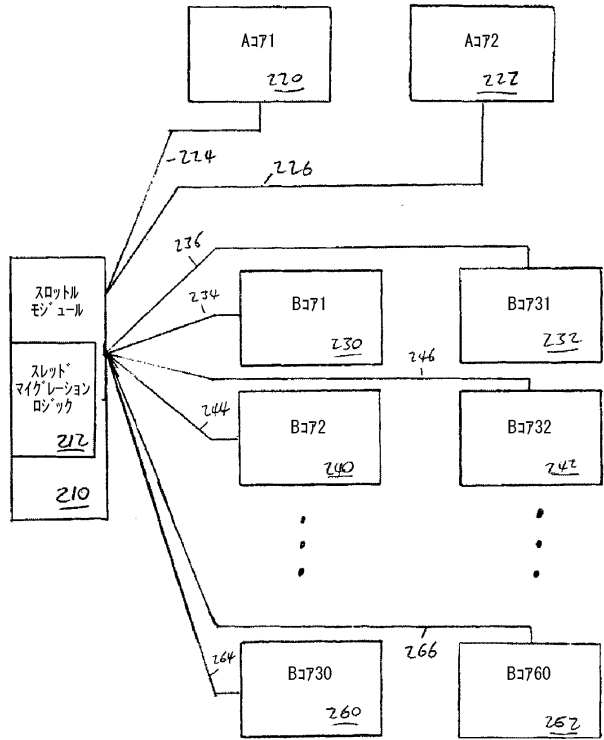
【図10A】本開示の一実施の形態による、スロットル及び複数のコアを有するプロセッサを含むシステムの概略図である。

【図10B】本開示の別の実施の形態による、スロットル及び複数のコアを有するプロセッサを含むシステムの概略図である。 30

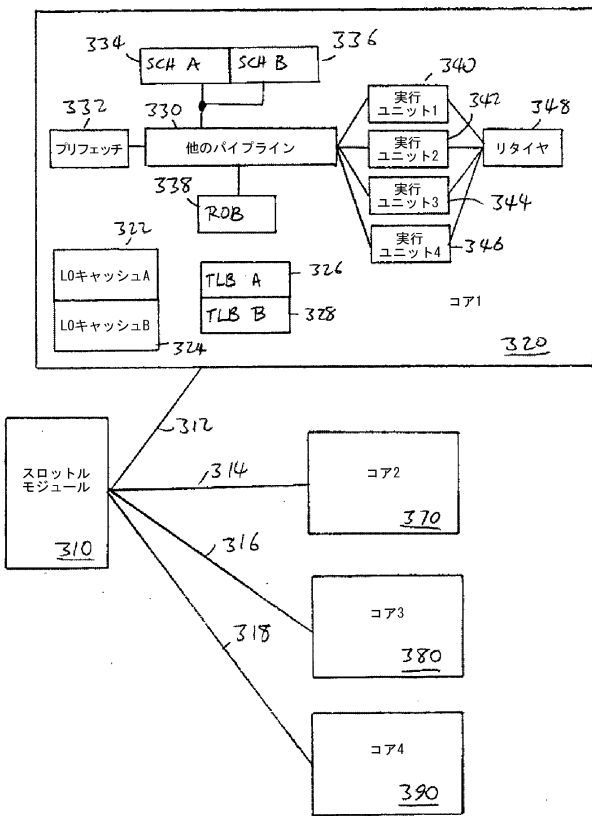
【図1】



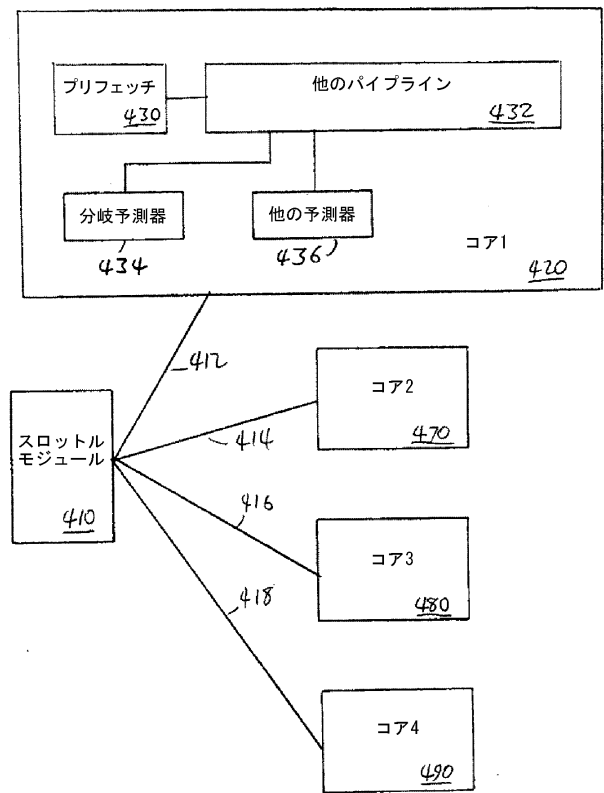
【図2】



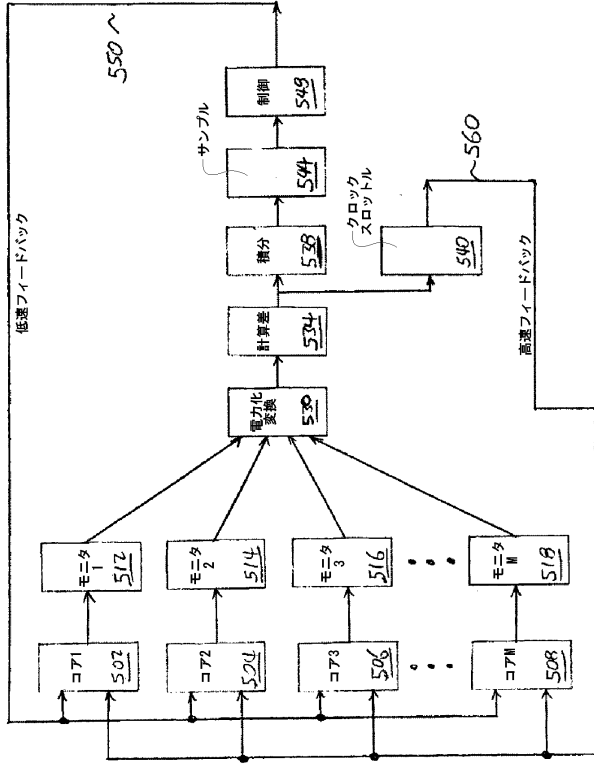
【図3】



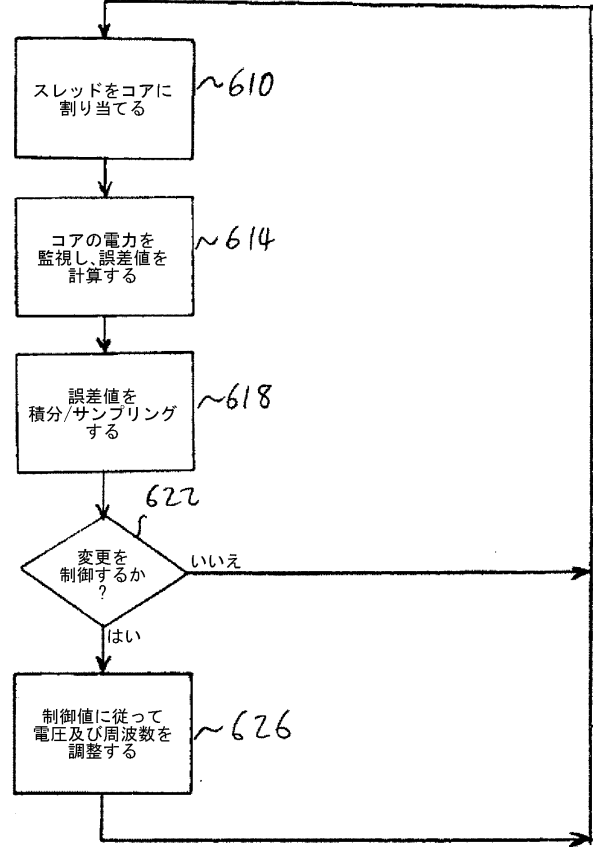
【図4】



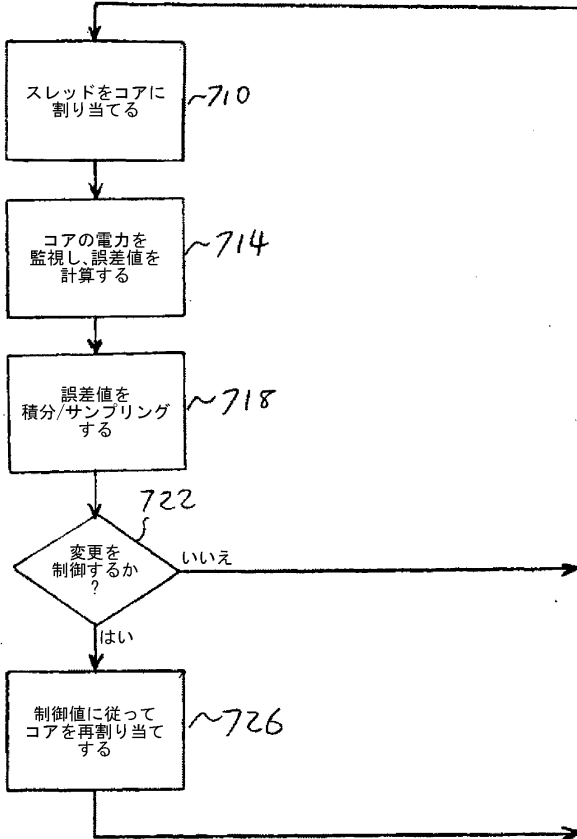
【図5】



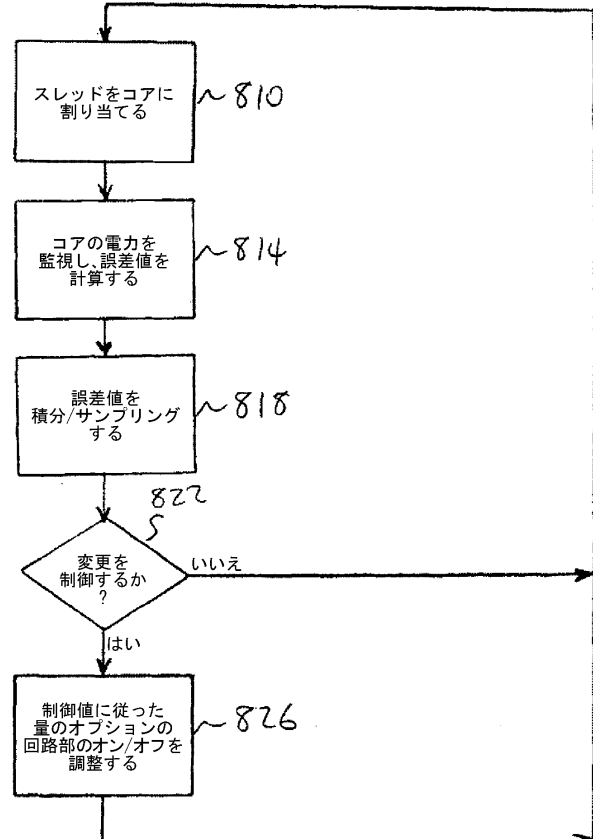
【図6】



【図7】

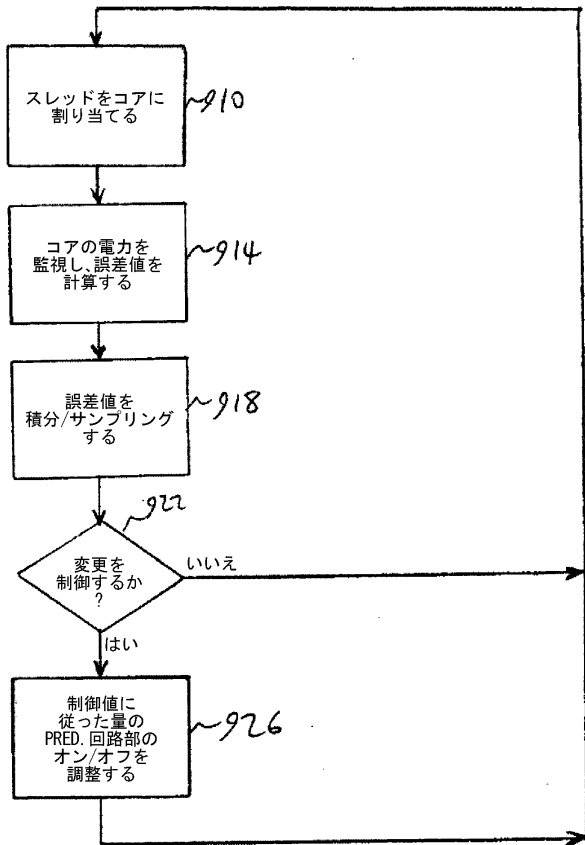


【図8】

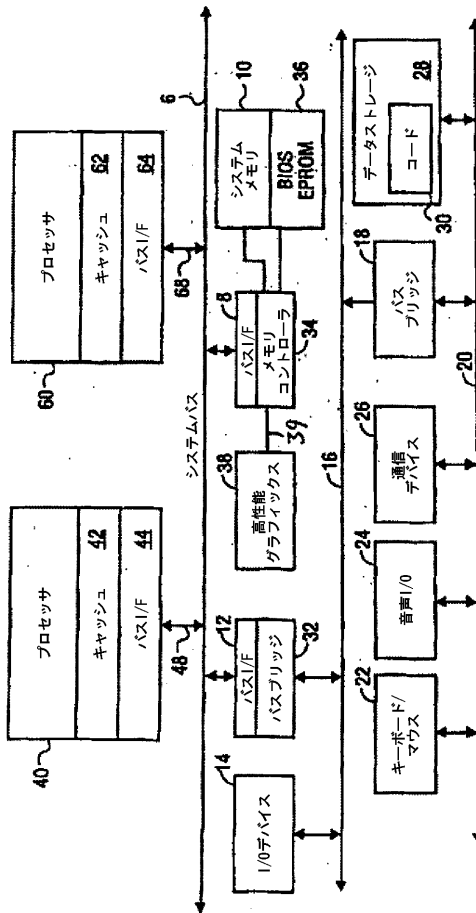




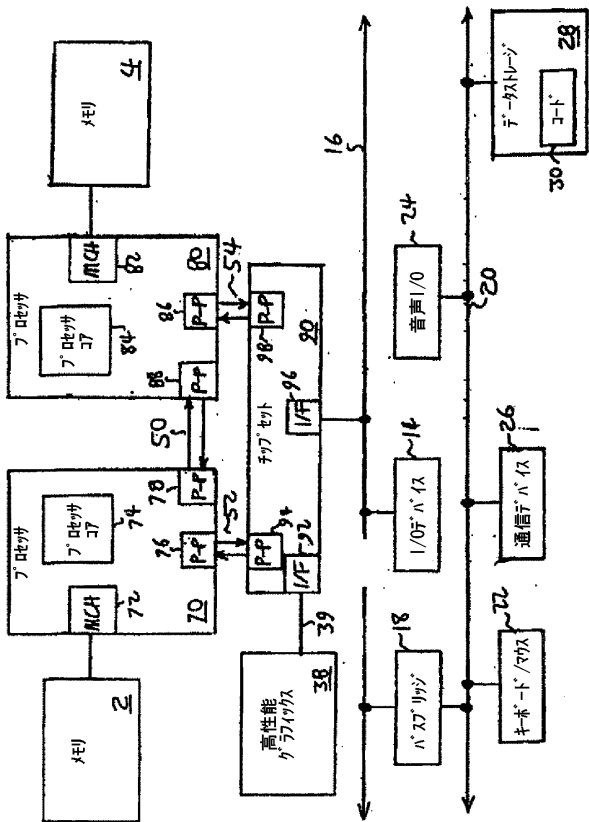
【図9】



【図10A】



【図10B】



## フロントページの続き

(51)Int.Cl. F I  
G 0 6 F 1/00 3 3 4 G

- (72)発明者 ワン、ホン  
アメリカ合衆国、95051 カリフォルニア州、サンタ クララ、ミッション カレッジ ブールバード 2200
- (72)発明者 オレンスタイン、ドロン  
イスラエル国、34604 ハイファ、イガル アロン ストリート 12
- (72)発明者 シーファー、ガッド  
イスラエル国、34604 ハイファ、アインシュタイン ストリート 33エー
- (72)発明者 ローネン、ロニー  
イスラエル国、34747 ハイファ、ハードユフ ストリート 11/11
- (72)発明者 アンナバラム、ムラリ  
アメリカ合衆国、95054 カリフォルニア州、サンタ クララ、ピスタ クラブ サークル  
ナンバー 208、1537

審査官 清木 泰

- (56)参考文献 特開2002-108958(JP,A)  
米国特許出願公開第2003/0110012(US,A1)  
特開平08-030472(JP,A)  
特開平11-338719(JP,A)  
米国特許第06411156(US,B1)  
米国特許出願公開第2001/0003207(US,A1)  
国際公開第99/017186(WO,A1)  
米国特許出願公開第2003/0117175(US,A1)  
Rakesh Kumar et al., A Multi-Core Approach to Addressing the Energy-Complexity Problem  
in Microprocessors, Proceedings of the Workshop on Complexity-Effective Design(WCED)  
, 2003年 6月

- (58)調査した分野(Int.Cl., DB名)  
G 0 6 F 9 / 4 6 - 9 / 5 4  
G 0 6 F 1 5 / 1 6 - 1 5 / 1 7 7  
G 0 6 F 1 / 2 6 - 1 / 3 2  
G 0 6 F 1 / 0 4 - 1 / 1 4