

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第5154682号
(P5154682)

(45) 発行日 平成25年2月27日 (2013. 2. 27)

(24) 登録日 平成24年12月14日 (2012. 12. 14)

(51) Int. Cl.	F I				
G06F 1/32	(2006.01)	G06F	1/00	332Z	
G06F 9/50	(2006.01)	G06F	9/46	462Z	
G06F 1/04	(2006.01)	G06F	1/04	301C	

請求項の数 19 (全 13 頁)

(21) 出願番号	特願2011-202792 (P2011-202792)	(73) 特許権者	593096712
(22) 出願日	平成23年9月16日 (2011. 9. 16)		インテル コーポレーション
(62) 分割の表示	特願2007-522581 (P2007-522581) の分割		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 ブールバード 2200
原出願日	平成17年7月15日 (2005. 7. 15)	(74) 代理人	100070150
(65) 公開番号	特開2012-69115 (P2012-69115A)		弁理士 伊東 忠彦
(43) 公開日	平成24年4月5日 (2012. 4. 5)	(74) 代理人	100091214
審査請求日	平成23年9月16日 (2011. 9. 16)		弁理士 大貫 進介
(31) 優先権主張番号	10/899, 674	(74) 代理人	100107766
(32) 優先日	平成16年7月27日 (2004. 7. 27)		弁理士 伊東 忠重
(33) 優先権主張国	米国 (US)	(72) 発明者	ナヴェー, アロン
			イスラエル国 47204 ラマハ ハシ ヤロン ウシシュキン ストリート 97

最終頁に続く

(54) 【発明の名称】 マルチコアプロセッサにおける電力管理調整

(57) 【特許請求の範囲】

【請求項 1】

各コアが対応する独立リソース設定を有し、第1コアと第2コアとを含む複数のコアにより共有される共有リソースに対して、互いに異なる対応する第1動作要求と第2動作要求とを含む複数の対応する整合しない動作要求を提供する複数のコアと、

前記複数のコアに接続され、前記複数の整合しない動作要求と、電力セービングポリシーと高パフォーマンスポリシーとの何れが制御するのに最も重要であるものとして特定されているかに基づき、前記複数のコアにより共有される共有リソースに対する共有リソース設定を調整する調整ロジックと、

を有するプロセッサであって、

前記調整ロジックは、

電力状態遷移テーブルに基づき前記第1動作要求を第1目標値に変換し、

前記電力状態遷移テーブルに基づき前記第2動作要求を第2目標値に変換し、

前記第1目標値と前記第2目標値とを比較することによって前記第1動作要求と前記第2動作要求とを比較し、

前記第2目標値が前記第1目標値より大きい場合、前記第2目標値を前記共有リソース設定として選択する、

ことによって、前記複数の整合しない動作要求に基づき前記共有リソース設定を調整するプロセッサ。

【請求項 2】

前記共有リソース設定は、前記複数のコアにより共有されるバス周波数設定と電圧設定とを有し、

前記独立リソース設定は、独立クロックロットル設定とアーキテクチャロットル設定とを有する、請求項 1 記載のプロセッサ。

【請求項 3】

前記第 1 コアは、前記第 1 動作要求が充足されることを可能にする前記第 1 コアの対応する独立リソース設定に対する調整値を選択する、請求項 2 記載のプロセッサ。

【請求項 4】

前記調整ロジックは、前記共有リソース設定から得られる訂正された動作要求を前記第 1 コアに通知するよう動作可能であり、

前記第 1 コアは、前記訂正された動作要求に基づき前記調整値を選択する、請求項 3 記載のプロセッサ。

【請求項 5】

前記第 2 コアは、より小さな第 2 目標値に対応する変更された動作要求を提供し、

前記調整ロジックは、前記変更された動作要求に基づき前記共有リソース設定を調整し、

前記第 1 コアは、前記調整された共有リソース設定に基づき、前記第 1 コアの独立リソース設定を調整し、

前記共有リソース設定は、前記複数のコアにより共有されるバス周波数設定と電圧設定とを有し、

前記独立リソース設定は、独立クロックロットル設定とアーキテクチャロットル設定とを有する、請求項 1 記載のプロセッサ。

【請求項 6】

前記独立リソース設定は、独立クロックロットル設定とアーキテクチャロットル設定とを含むセットから選択される設定を有する、請求項 1 記載のプロセッサ。

【請求項 7】

前記共有リソース設定は、電圧設定と周波数設定とを含むセットから選択される設定を有する、請求項 1 記載のプロセッサ。

【請求項 8】

前記複数の動作要求は、複数の電力要求を有する、請求項 1 記載のプロセッサ。

【請求項 9】

前記複数の動作要求は、複数のパフォーマンス要求を有する、請求項 1 記載のプロセッサ。

【請求項 10】

各コアが対応する独立リソース設定を有し、第 1 コアと第 2 コアとを含む複数のコアにより共有される共有リソースに対して、互いに異なる対応する第 1 動作要求と第 2 動作要求とを含む複数の対応する整合しない動作要求を提供するステップと、

前記複数の整合しない動作要求と、電力セービングポリシーと高パフォーマンスポリシーとの何れが制御するのに最も重要であるものとして特定されているかに基づき、前記複数のコアにより共有される共有リソースの共有リソース設定を調整するステップと、

を有する方法であって、

前記共有リソース設定を調整するステップは、

電力状態遷移テーブルに基づき前記第 1 動作要求を第 1 目標値に変換し、

前記電力状態遷移テーブルに基づき前記第 2 動作要求を第 2 目標値に変換し、

前記第 1 目標値と前記第 2 目標値とを比較することによって前記第 1 動作要求と前記第 2 動作要求とを比較し、

前記第 2 目標値が前記第 1 目標値より大きい場合、前記第 2 目標値を前記共有リソース設定として選択する、

10

20

30

40

50

ことを有する方法。

【請求項 1 1】

前記共有リソース設定は、前記複数のコアにより共有されるバス周波数設定と電圧設定とを有し、

前記独立リソース設定は、独立クロックスロットル設定とアーキテクチャスロットル設定とを有する、請求項 1 0 記載の方法。

【請求項 1 2】

前記共有リソース設定から得られる訂正された動作要求を前記第 1 コアに通知するステップと、

前記訂正された動作要求に基づき前記調整値を選択するステップと、
をさらに有する、請求項 1 0 記載の方法。

10

【請求項 1 3】

前記独立リソース設定により前記共有リソース設定を調整するステップは、独立クロックスロットル設定とアーキテクチャスロットル設定とを含むセットから選択される設定により前記共有リソース設定を調整する、請求項 1 1 記載の方法。

【請求項 1 4】

前記調整するステップは、電圧設定と周波数設定とを含むセットから選択される設定により調整する、請求項 1 0 記載の方法。

【請求項 1 5】

共有周波数設定を有するクロックソースと、

前記クロックソースに接続される請求項 1 乃至 9 何れか一項記載のプロセッサと、
を有する計算システム。

20

【請求項 1 6】

第 1 プロセッサコアから第 1 動作要求を通知するステップと、

第 2 プロセッサコアから第 2 動作要求を通知するステップと、

前記第 1 動作要求を第 1 目標値に変換するステップと、

前記第 2 動作要求を第 2 目標値に変換するステップと、

前記第 1 動作要求と前記第 2 動作要求とを比較するステップと、

前記第 2 動作要求が前記第 1 動作要求より大きい場合、前記第 2 目標値を共有リソース設定として選択するステップと、

前記第 1 動作要求が充足されることを可能にする独立リソース設定に対して調整値を選択するステップと、

前記第 2 プロセッサコアから変更された動作要求を通知するステップと、

前記変更された動作要求に基づき前記共有リソース設定を調整するステップと、

前記調整された共有リソース設定に基づき前記独立リソース設定を調整するステップと

、

を有する方法。

【請求項 1 7】

前記共有リソース設定は、電圧設定と周波数設定とを含むセットから選択される設定を含む、請求項 1 6 記載の方法。

40

【請求項 1 8】

前記独立リソース設定は、独立クロックスロットル設定とアーキテクチャスロットル設定とを含むセットから選択される設定を含む、請求項 1 6 記載の方法。

【請求項 1 9】

前記第 1 動作要求と前記第 2 動作要求とは、合成された最適状態絶対電力、電力パーセンテージ、パフォーマンスインデックス及び / 又は合成された最適状態絶対パフォーマンスの少なくとも 1 つである、請求項 1 6 記載の方法。

【発明の詳細な説明】

【技術分野】

【0 0 0 1】

50

本発明の1以上の実施例は、一般に電力管理に関する。より詳細には、特定の実施例は、マルチコアプロセッサにおける電力管理動作の調整に関する。

【背景技術】

【0002】

より多くのトランジスタとより高い周波数を備えた先進的なプロセッサに対するトレンドが増大し続けるに従って、コンピュータ設計者及び製造者は、しばしば対応する電力消費の増大に直面する。さらに、より高速かつ小型のコンポーネントを提供する製造技術は、同時に漏れ電力の増大をもたらさう。特にモバイル計算環境では、電力消費の増大はオーバーヒートを発生させる可能性があり、パフォーマンスに悪影響を与え、バッテリー寿命を大きく低減させる可能性がある。

10

【0003】

いくつかの現在のモバイル計算システムは、プロセッサベース電力管理スキームを実現することにより、バッテリー寿命の低減に対する懸念に取り組んでいる。例えば、より一般的なアプローチの1つは、高いパフォーマンスが必要とされない、又は所望されないときには、プロセッサのクロック周波数とコア電圧を低下させることによって、動的にプロセッサの電力消費を低下させるというものである。電力管理はまた、プロセッサがチップセット、メモリサブシステム、入出力(I/O)デバイスなどの他のコンポーネントと通信するのに利用するバスの周波数を招請することによって向上させることが可能である。他のアプローチは、電力を節約するため、プロセッサに対するクロックをスイッチオン・オフ又は「ゲート」オン・オフ(すなわち、クロックの「スロットル」)することである。

プロセッサベース電力管理のさらなる具体例は、内部アレイ、実行ユニットなどのプロセッサのアーキテクチャコンポーネントをスロットルすることである。いくつかの技術は、電力消費をさらに低下させるため、上記アプローチの各種組み合わせを利用している。これらの電力管理スキームはある状況下においては許容であるものであるが、いくつかの問題が残っている。

20

【発明の概要】

【発明が解決しようとする課題】

【0004】

1つの問題は、各プロセッサ上に複数のコアを備えるシステムの複数のプロセッサを含むことが可能なより複雑なプロセッサアーキテクチャに対する近年のトレンドに関するものである。このような場合、電力関連リソース及び関連するコントロールのいくつかは、プロセッサコア間で共有される一方、他のリソース及びコントロールは、所与のコアに専用とされるかもしれない。専用のリソース及びコントロールは、他のコアのリソース及びコントロールと独立したものである一方、共有されたリソース及びコントロールは、各コアの状態に依存する。しかしながら、従来の電力管理スキームは、これらのケースにおいては良好には機能しないかもしれない。なぜなら、それらはシングルプロセッサ環境に対して構成されているためである。

30

【0005】

例えば、そのようなスキームの1つは、オペレーティングシステム(OS)レベルでの各種電力状態の間のプロセッサの遷移を制御する。将来的なOSの実現形態は、コア単位ベースにより電力を管理することが可能となるかもしれないが、OSがコア間で調整を行うことが可能となるか否かは明らかではない。さらに、OSの実現形態がコア単位の電力管理を実現したとしても、依然として効率性の問題が残る。特に、OSは、利用可能なパフォーマンス/電力制御機構のすべてを認識していないかもしれない、それらの間のトレードオフをバランスさせることができないかもしれない。さらに、このような複雑なアーキテクチャを調整するためソフトウェアに依存することは、OSのオーバーヘッドを増大させ、ソフトウェア計算複雑さに増大させる可能性がある。マルチコアプロセッサのソフトウェアベースの電力管理は、相互依存する各コアにおいて行われる可能性がある迅速な状態変化のため、タイミングの観点から実現することが困難であるかもしれない。

40

【0006】

50

本発明の課題は、上記課題に鑑み、マルチコアプロセッサにおける電力管理動作を調整するのに好適な技術を提供することである。

【課題を解決するための手段】

【0007】

上記課題を解決するため、本発明の一特徴は、各コアが対応する独立リソース設定を有し、第1コアと第2コアとを含む複数のコアにより共有される共有リソースに対して、互いに異なる対応する第1動作要求と第2動作要求とを含む複数の対応する整合しない動作要求を提供する複数のコアと、前記複数のコアに接続され、前記複数の整合しない動作要求と、電力セービングポリシーと高パフォーマンスポリシーとの何れが制御するのに最も重要であるものとして特定されているかに基づき、前記複数のコアにより共有される共有リソースに対する共有リソース設定を調整する調整ロジックとを有するプロセッサであって、前記調整ロジックは、電力状態遷移テーブルに基づき前記第1動作要求を第1目標値に変換し、前記電力状態遷移テーブルに基づき前記第2動作要求を第2目標値に変換し、前記第1目標値と前記第2目標値とを比較することによって前記第1動作要求と前記第2動作要求とを比較し、前記第2目標値が前記第1目標値より大きい場合、前記第2目標値を前記共有リソース設定として選択することによって、前記複数の整合しない動作要求に基づき前記共有リソース設定を調整するプロセッサに関する。

10

【発明の効果】

【0008】

本発明によると、マルチコアプロセッサにおける電力管理動作を調整するのに好適な技術を提供することができる。

20

【図面の簡単な説明】

【0009】

【図1】図1は、本発明の一実施例によるプロセッサの一例のブロック図である。

【図2】図2は、本発明の一実施例による電力管理アーキテクチャの一例を示す図である。

【図3】図3は、本発明の一実施例によるパフォーマンス状態遷移テーブルの一例を示す図である。

【図4】図4は、本発明の一実施例によるシステムの一例を示すブロック図である。

【図5】図5は、本発明の一実施例による電力管理方法の一例のフローチャートである。

30

【図6】図6は、本発明の一実施例による独立リソース設定により共有リソース設定を調整する処理の一例のフローチャートである。

【図7】図7は、本発明の一実施例による共有リソース設定を選択する処理の一例のフローチャートである。

【図8】図8は、本発明の一実施例によるリソース設定を調整する処理の一例のフローチャートである。

【発明を実施するための形態】

【0010】

図1は、第1コア12と、第2コア14と、第1コア12及び第2コア14に動作可能に接続されるハードウェア調整ロジック16とを有するプロセッサ10を示す。ここでは、「第1」及び「第2」という用語は、説明の簡単化のためだけに利用される。さらに、ここではデュアルコアコンフィギュレーションが図示されているが、プロセッサ10のコアの個数は、本発明の実施例の趣旨及び範囲から逸脱することなく、容易に増やすことが可能である。図示されたコア12と14のそれぞれは、論理プロセッサとして十分機能的なものであり、従来の実行ユニット、レベル1(L1)キャッシュなどを有する。従って、図示されたデュアルコアコンフィギュレーションは、従来シングルコアプロセッサに対して大きなパフォーマンスの向上をもたらす。

40

【0011】

プロセッサ10は、電力の大きな低下を可能にするいくつかの機能/リソースを有する。例えば、プロセッサ10は、高いパフォーマンスが必要とされない、又は所望されない

50

ときには、プロセッサクロック周波数及び電源電圧（コア電圧など）を低下させることが可能な機能を有するかもしれない。他の電力節約機能は、バス（図示せず）のクロック周波数を調整するものであるかもしれない。さらなる他の機能は、クロックをスロットル（又は「ゲート」）オン・オフすることが可能であり、これにより、電力消費を低下させることが可能である。クロックスロットル処理は、プロセッサ10の各部分に又はパッケージレベルにおいて実行することが可能である。さらなる他の電力節約機能は、電力を低下及び/又は温度を低下させるため、コア12及び14のアーキテクチャコンポーネントをスロットル処理することに関するものであるかもしれない。

【0012】

上記機能のそれぞれは電力管理の観点から特に効果的であるが、それらのいくつかは、コア12及び14により共有されるリソースを利用し、他のものは、コア12及び14により独立に制御されるリソースに関するものである。図示されたプロセッサ10は、一般には双方のシナリオを動的にサポートすることが可能なアーキテクチャを有し、従来の電力管理スキームに対していくつかの効果を提供する。

10

【0013】

例えば、図示されたプロセッサ10は、コア12及び14の両者により共有される電源電圧設定及びプロセッサクロック周波数設定を有するが、独立したクロックスロットル設定は、どの頻度によりクロックが第1コア12のみに印加されるか制御するものであるかもしれない。従って、クロック周波数及び電源電圧設定は、共有リソース設定22としてみなすことができ、独立したクロックスロットル設定は、独立したリソース設定24としてみなすことができる。同様に、第2コア14はまた、独立したリソース設定26により表される独立したクロックスロットル設定を有するかもしれない。

20

【0014】

独立したクロックスロットル処理に加えて、コア12及び14は、それらの内部アーキテクチャの各種機能ブロックを独立にスロットル処理することが可能であるかもしれない。例えば、独立したリソース設定24と26は、内部アレイ、リオーダバッファ（ROB）、リザベーションステーション（RS）テーブル、パラレルユニット、実行ユニットなどのアーキテクチャコンポーネントのスロットル処理を行うことができる。

【0015】

図示された実施例では、第1コア12が第1動作要求18を提供し、第2コア14が第2動作要求を提供する。調整ロジック16は、動作要求18及び20と共有リソース設定22を調整することができる。以下においてより詳細に説明されるように、動作要求18及び20は、電力ポリシーやパフォーマンスポリシーなどのポリシーに従って生成することが可能であり、コア12及び14のそれぞれの上で実行されるオペレーティングシステム（OS）のプロダクトであるかもしれない。例えば、第1コア12は、20%だけ電力消費を低減させることが可能なユーザ/ソフトウェア選択可能な電力目標を有するかもしれない。この場合、第1動作要求18は、8%の電力レベルを示す電力要求とすることができる。あるいは、第2コア12は、第2動作要求20が100%の電力レベルを示すように、フルパワーを要求する動作モードにあるかもしれない。具体的な数値は、単なる例示として利用される。

30

40

【0016】

動作要求18と20はまた、以下でより詳細に説明されるように、パフォーマンスレベルやパフォーマンス状態遷移テーブルへのインデックスなどのパフォーマンス要求を特定することができる。ハードウェア調整ロジック16がパフォーマンス又は電力ポリシーに基づき機能することを可能にすることによって、プロセッサ10は、他のものについてはおそらく妥協しながら、ソフトウェアが制御するのに現在最も重要なファクタとなるものに集中することができる。複数の機能と共通のポリシー（パフォーマンス又は電力）にわたって調整するとき、パフォーマンス/電力ポリシーが有用となりうるが、他のアプローチもまた利用可能である。例えば、実際のリソース設定を動作要求に組み込むことはまた、許容可能なアプローチであるかもしれない。

50

【 0 0 1 7 】

すでに説明したように、調整ロジック 1 6 は、第 1 動作要求 1 8 と第 2 動作要求 2 0 に従って、共有リソース設定 2 2 を選択することが可能である。この結果は、動作要求 1 8 及び 2 0 を充足するかもしれない、又はしないかもしれない動作状態である。例えば、第 1 動作要求 1 8 が共有リソース設定 2 2 により充足されない場合、第 1 コア 1 2 は、第 1 動作要求 1 8 が充足されることを可能にする第 1 独立リソース設定 2 4 に対する変更 / 調整された値を選択する。あるいは、第 2 動作要求 2 0 が共有リソース設定 2 2 により充足されない場合、第 2 コア 1 4 は、第 2 動作要求 2 0 が充足されることを可能にする第 2 独立リソースに対して変更 / 調整された値を選択する。従って、調整ロジック 1 6 は、実際に実現される調整された状態を反映する各動作要求 1 8 及び 2 0 と共有リソース設定 2 2 との間の相違に基づき、結果としての独立リソース設定 2 4 及び 2 6 を決定するかもしれない。簡単に言えば、独立リソース設定 2 4 及び 2 6 は、1 以上のコア 1 2 と 1 4 によって調整可能な共有リソース設定 2 2 に係る電力 / パフォーマンス不足を補償することが可能である。

10

【 0 0 1 8 】

従って、一致又は整合しない (m i s m a t c h e d) 動作要求の上記例では、調整ロジック 1 6 は、第 2 コア 1 4 がより高い電力レベルを要求しているという事実を考慮し、この要求を充足するため、調整された共有リソース設定 2 2 を選択するであろう。従って、共有リソース設定 2 2 は、両方のコア 1 2 と 1 4 に対して 1 0 0 % の電力の動作状態をもたらす、第 1 コア 1 2 は依然として 8 % の動作要求 1 8 を有することとなるであろう。第 1 コア 1 2 はまだ第 1 動作要求 1 8 を充足していないため、第 1 コア 1 2 は、第 1 動作要求 1 8 が充足されることを可能にする第 1 の独立リソース設定 2 4 に対する相関 / 調整された値を選択する。例えば、第 1 コア 1 2 は、所望される 2 0 % の電力低下を実現するため、その独立クロックスロットル設定を増やすかもしれない。従って、共有リソース設定 2 2 を独立リソース設定 2 4 及び 2 6 により調整することによって、調整ロジック 1 6 は、マルチコア環境における電力を効率的に管理することが可能である。

20

【 0 0 1 9 】

調整ロジック 1 6 は、第 1 動作要求 1 8 を第 1 目標値に変換し、第 2 動作要求 2 0 を第 2 目標値に変換することによって共有リソース設定 2 2 を選択することが可能である。当該目標値は、共有リソース設定 2 2 の潜在的な数値を表す。調整ロジック 1 6 は、その後、第 1 目標値と第 2 目標値を比較し、より高いものを共有リソース設定 2 2 として選択するかもしれない。また、調整ロジック 1 6 は単に動作要求 1 8 及び 2 0 を比較し、より高いものを選択することも可能であるということに留意すべきである。従って、調整ロジック 1 6 は、共有リソース設定 2 2 の目標値及び / 又は動作要求に対する「最大値検出装置」として機能することが可能である。

30

【 0 0 2 0 】

あるいは、調整ロジック 1 6 は、「双方のリクエストの最小値」の調整ポリシーが適切である場合、より小さな値を選択するようにしてもよい。このような状況は、共有クロックスロットル処理の場合、又は OS がコントロールにおいて “ F o r c e _ M i n ” フラグを設定することを決定する場合に生じる可能性がある。ここで、“ F o r c e _ M i n ” フラグは、双方のコアの各スロットル処理リクエストの最小のものが選択されたものであることを保証する M I N I M U M 動作ポイントポリシーを実行する必要があることを調整ロジック 1 6 に通知することが可能となる。簡単化のため、ここで説明される具体例のいくつかは、最大値としての調整ポリシーを表す。しかしながら、本発明の実施例に係る効果はまた最小ポリシーに適用可能であるということに留意すべきである。

40

【 0 0 2 1 】

第 2 コア 1 4 が以降において、より低い目標値に対応する変更された動作要求を提供する場合、調整ロジック 1 6 は、この変更された動作要求に基づき共有リソース設定 2 2 を調整することが可能である。このような場合、第 1 コア 1 2 は、調整された共有リソース設定に基づき、独立リソース設定 2 4 を調整することが可能である。一致しない動作要求

50

の上述した例では、第1コア12は、その独立クロックロットル設定を低減するかもしれない(それはもはや不要となるため)。調整ロジックをハードウェアにより実現することによって、迅速な状態変化に対応するため高速なOSレスポンス時間が、システムに要求されることがなくなる。独立リソース設定24及び26が共有リソース設定22より効率的であると知られている場合、上述した処理は、独立リソース設定24と26がまず選択されるように、独立リソース設定24と26を補償する共有リソース設定22と逆転されてもよい。

【0022】

図2及び3は、上述した変換をそれぞれ実現するのに利用可能なハードウェア調整ロジックとパフォーマンス状態遷移テーブル34を実現するのに利用可能な電力管理アーキテクチャ58を示す。アーキテクチャ58とテーブル34は、電力を管理するのに大変有用となる可能性があるが、他のアプローチもまた利用可能である。図示された例では、各プロセッサコアに対するフォースミニマム(force minimum)特性、パフォーマンスインデックス、パフォーマンスレベル及び/又は電力レベルなどの動作要求特性を規定するパフォーマンス状態(P状態)起動コマンド60が、パフォーマンス状態遷移テーブル34にアクセスするのに利用される。1つの起動コマンド60しか示されていないが、複数の起動コマンドが同時にテーブル34に適用可能である。

【0023】

図示された例では、動作要求は、合成された最適状態絶対電力(Abs-Pwr)42や電力パーセンテージ(Pwr%)44などの電力要求として規定することが可能である。動作要求はまた、パフォーマンスインデックス(Pindex)50や合成された最適状態絶対パフォーマンス(Perf)52などのパフォーマンス要求として規定することが可能である。相対的に複雑なハードウェア調整をサポートするため、共有リソース設定が各種機能において統合され、実際の動作要求が「グローバル」レベルに指定されてもよいということが理解できる。従って、図示された例では、起動コマンド60は、合成された最適状態を表す動作要求を含む。

【0024】

上述されたように、リソース設定の目標値は、Speedstep(登録商標)機能(Shared Feature)が、クロック周波数設定(Frequency)36とコア電圧設定(V_{cc})38として示される共有リソース設定に対する制御を提供する機能によって特定することができる。他方、TM1機能(Independent Feature)は、独立クロックロットル設定(Throttle%)40として示される独立リソース設定に対する制御を提供するかもしれない。動作要求及び目標値が調整ロジックにより受け付けされると、それらは、適切なグローバル設定の選択のため互いに比較することができる。

【0025】

他の例では、第1コアが88%のパフォーマンスレベルをリクエストする場合、調整ロジックは、テーブルの合成された最適状態部分へのインデックスするため、88%のパフォーマンス「動作要求」を利用することが可能である。1300MHz/1.008Vの目標周波数/電圧値が、テーブルから抽出され、第1共有リソースコントロールブロック64に送信するためコントロールバス62(62a~62e)に配置することができる。従って、図示された第1共有リソースコントロールブロック64は、テーブル34に示される共有機能に対応する。第1コアリクエストと同時に、第2コアは、100%のパフォーマンスレベルにおいて動作するためのリクエストなどの動作要求を提供するかもしれない。従って、調整ロジックは、100%のパフォーマンスを利用して、テーブルの合成された最適状態部分にインデックスし、テーブルから1700MHz/1.233Vの目標周波数/電圧値を抽出し、それをコントロールバス66に配備することが可能である。

【0026】

図示されたシステムが「最大検出」モードである場合(すなわち、Force_Minが設定されていない)、第1共有リソースコントロールブロック64は、これら2つの動

10

20

30

40

50

作要求を比較し、適切なグローバル設定が100%のパフォーマンスであると判断する。従って、1700MHz / 1.233Vの目標周波数 / 電圧値が、共有リソース設定68として選択されるかもしれない。図示された例では、共有リソース設定68が、クロックソース及び / 又は電圧ソースに印加され、共有リソースに共有リソース設定68により規定される適切な周波数及び / 又はコア電圧において動作するよう指示する。第1共有リソースコントロールブロック64はまた、実際の動作状態計算72を生成する状態計算ロジック84に動作状態信号70を送信する。動作状態信号70は、共有リソース設定68から生じるパフォーマンス及び / 又は電力状態を特定する。従って、本例では、動作状態信号70が100%のパフォーマンスレベルを特定することとなる。

【0027】

さらに、第1共有リソースコントロールブロック64は、共有リソース設定68に基づき、訂正された動作要求及び / 又は目標値を生成し、訂正された目標値をコントロールバス62bに配備することができる。ここで、訂正された目標値は、第2共有リソースコントロールブロック76が次の共有リソースに対する共有リソース設定80を決定することを可能にする。訂正された値は、選択された動作レベルに対応し、テーブル34などのテーブルから抽出することができる(1つの共有機能しかテーブルには図示されていないが)。図示された例では、共有リソース設定80は、周波数設定とすることができる。第2共有リソースコントロールブロック76はまた、コントロールバス78から他の1以上のプロセッサコアの目標値及び / 又は動作要求を収集し、当該セットにおける最大値を決定する。この最大値は、クロックが適切な共有リソース設定により動作するよう指示する共有リソース設定80を表す。動作状態信号82はまた、状態計算ロジック84に送信される。さらなるリソースコントロールブロックが必要な場合、訂正された動作要求及び / 又は目標値が生成され、コントロールバス62cに配備される。

【0028】

共有リソース設定がすべて完了すると、電力管理アーキテクチャ58は、独立リソースコントロールブロック86を利用して独立リソース設定88を生成するため設けられ、独立リソース設定88は、当該プロセッサコアがコントロールバス62dを介し訂正された動作要求に反映されるさらなる電力の節約を実現することを可能にする。訂正された動作要求は、共有リソースコントロールブロックの最後から取得することができる。88%のパフォーマンスレベルを必要とする第1コアの上記例では、独立リソースコントロールブロック86は、88%のパフォーマンスを提供する0.125の独立クロックロットルの独立リソース設定を選択するかもしれない。図示されたP12のグローバル設定はまた、0.125の独立クロックロットルと合成される場合、全体として57%のパフォーマンスを提供する共有電圧 / 周波数設定を通じて67%のパフォーマンスの低下を含む。しかしながら、ここで説明されるアプローチは、共有リソースがすでに選択及び固定されているときには、独立リソース設定を利用することによりパフォーマンス数を補償する。独立リソースコントロールブロック86はまた、実際の動作状態計算72を生成するのに利用するため、動作状態信号92を生成することが可能である。独立リソース設定88が、訂正された動作要求を完全には充足することができない場合、他の訂正された動作要求が、コントロールバス62eを介し次の独立リソースコントロールブロック(図示せず)に送信可能である。

【0029】

次に図4を参照するに、共有周波数設定22a'によるクロックソース30と、共有周波数設定22c'によるバス11と、共有電圧設定22b'による電圧ソース32と、マルチコアプロセッサ10'とを有するシステム28の一例が示される。システム28は、電力消費及びオーバーヒートが特に問題となる、ノートブックパーソナルコンピュータ(PC)、携帯情報端末(PDA)、無線「スマート」フォンなどモバイル計算システムの一部とすることが可能である。図示されたプロセッサ10'は、第1コア12'と、第2コア14'と、ハードウェア調整ロジック16'とを有する。第1コア12'は第1動作要求(図示せず)を提供し、第2コア14'は第2動作要求(図示せず)を提供する。調

10

20

30

40

50

整ロジック 16' は、これらの動作要求及び第 1 コア 12' と第 2 コア 14' のそれぞれの独立リソース設定 24' と 26' により共有周波数設定 22' (22a' ~ 22c') を調整する。

【0030】

図 5 は、何れか利用可能なハードウェア及び / 又はソフトウェアプログラミング技術を利用してマルチコアプロセッサにより実現することが可能である。例えば、方法 96 の 1 以上の部分が、固定された機能ハードウェア、特定用途向け集積回路 (ASIC)、マシン可読媒体に格納されるマイクロコード命令セット又は上記の何れかの組み合わせにより実現することが可能である。特に、図示された方法 96 は、処理ブロック 98 において第 1 プロセッサコアから第 1 動作要求を提供する。第 2 動作要求は、ブロック 100 において第 2 プロセッサコアから提供される。ブロック 102 は、これら動作要求により共有リソース設定を調整する。共有リソース設定は、第 1 及び第 2 動作要求に基づき独立リソース設定によりすでに調整されている。独立リソース設定は、第 1 又は第 2 プロセッサコアに専用のものであるとすることが可能である。

10

【0031】

次に図 6 を参照するに、共有リソース設定を調整するための 1 つのアプローチが、ブロック 102' においてより詳細に示される。特に、ブロック 104 は、第 1 及び第 2 動作要求に従って、共有リソース設定を選択する。ブロック 106 は、動作要求が充足されることを可能にする独立リソース設定に対する調整された値を選択する。

20

【0032】

図 7 は、ブロック 104' における共有リソース設定を選択する 1 つのアプローチをより詳細に示す。図示された例では、第 1 動作要求は、ブロック 108 において第 1 目標値に変換され、第 2 動作要求は、ブロック 110 において第 2 目標値に変換される。上述されたように、ブロック 108 及び 110 における変換は、電力状態遷移テーブルを利用することにより実現することができる。ブロック 112 は、第 1 動作要求と第 2 動作要求を比較する。あるいは、目標値自体を比較することが可能である。ブロック 114 において、第 2 動作要求が第 1 動作要求より大きいと判断されると、ブロック 116 は、共有リソース設定として第 2 目標値を選択する。そうでない場合、第 1 目標値は、ブロック 118 において共有リソース設定として選択される。ブロック 120 は、必要に応じて、残りの共有リソースに対して共有リソース設定選択処理を繰り返す。共有リソース設定から得られる訂正された動作要求が、ブロック 122 においてコアに通知される。

30

【0033】

次に図 8 を参照するに、共有リソース設定及び独立リソース設定を更新する処理 124 が示される。図示された例では、第 2 プロセッサコアは、以前には第 1 プロセッサコアより高いレベルを要求していた。従って、第 1 プロセッサコアは、共有リソース設定を補償するため、独立リソース設定を利用している。ブロック 126 は、第 2 プロセッサコアから変更された動作要求を提供する。共有リソース設定は、ブロック 128 において変更された動作要求に基づき調整される。ブロック 130 は、調整された共有リソース設定に基づき、第 1 プロセッサコアの独立リソース設定を調整する。

【0034】

以上より、ここに記載された実施例は、マルチコアプロセッサにおける電力管理についていくつかの特有の効果を提供する。例えば、ハードウェア調整ロジックにおける電力管理の調整は、ソフトウェアオーバヘッドと計算の問題を軽減する。さらに、ハードウェア調整は、比較的迅速であり、OS により調整される電力管理に係るレスポンス時間の問題を軽減する。さらに、パフォーマンス又は電力ポリシーに基づく電力管理によって、プロセッサは、おそらくその他のものに妥協しながら、ソフトウェアが制御する現在最も重要なファクタとなるものに集中することができる。

40

【0035】

当業者は、上記記載から本発明の実施例の広範な技術が各種形態により実現可能であることを理解することが可能である。従って、本発明の実施例がその特定の具体例に関し

50

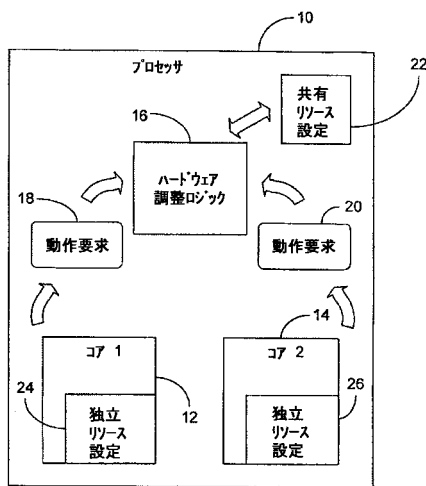
て説明されたが、当該図面、明細書及び以下の請求項を参照することにより、他の改良が当業者に想到すると考えられるため、本発明の実施例の真の範囲はこれに限定されるべきではない。

【符号の説明】

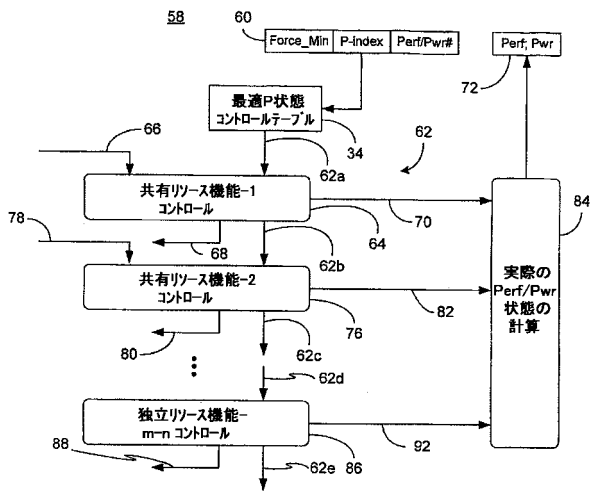
【 0 0 3 6 】

- 1 0 プロセッサ
- 1 2 , 1 4 コア
- 1 6 調整ロジック

【 図 1 】



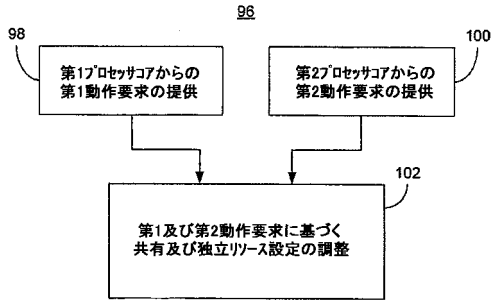
【 図 2 】



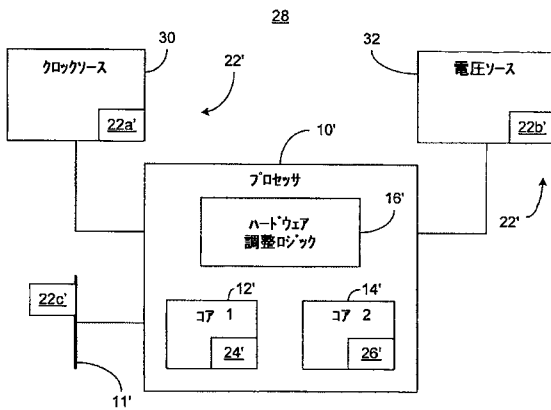
【図3】

合成された最適状態				共有機能				独立機能			
Pindex	Perf	Abs-Pwr	Pwr%	SpeedStep@コントロール		TM1 コントロール		Throttle%	Pwr%	Perf	
				Frequency	Vcc	Vcc	Perf				
0	1	100	100%	1700	1.233	100%	100%	0	1	100%	
1	0.97	84	84%	1600	1.183	84%	97%	0	1	100%	
2	0.94	68	68%	1500	1.133	68%	94%	0	1	100%	
3	0.91	54	54%	1400	1.083	54%	91%	0	1	100%	
4	0.88	41	41%	1300	1.008	41%	88%	0	1	100%	
5	0.85	28	28%	1200	0.933	28%	85%	0	1	100%	
6	0.82	16	16%	1100	0.85	16%	82%	0	1	100%	
7	0.79	15	15%	1000	0.85	15%	79%	0	1	100%	
8	0.76	13	13%	900	0.85	13%	76%	0	1	100%	
9	0.73	12	12%	800	0.85	12%	73%	0	1	100%	
10	0.7	10	10%	700	0.85	10%	70%	0	1	100%	
11	0.67	9	9%	600	0.85	9%	67%	0	1	100%	
12	0.58625	9	8%	600	0.85	9%	67%	0.125	0.9063	88%	
13	0.5025	7	7%	600	0.85	9%	67%	0.25	0.8125	75%	
14	0.41875	6	6%	600	0.85	9%	67%	0.375	0.7188	63%	
15	0.335	6	6%	600	0.85	9%	67%	0.5	0.625	50%	
16	0.1675	4	4%	600	0.85	9%	67%	0.75	0.4375	25%	

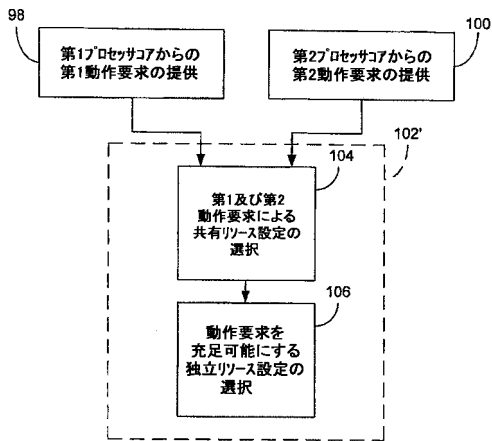
【図5】



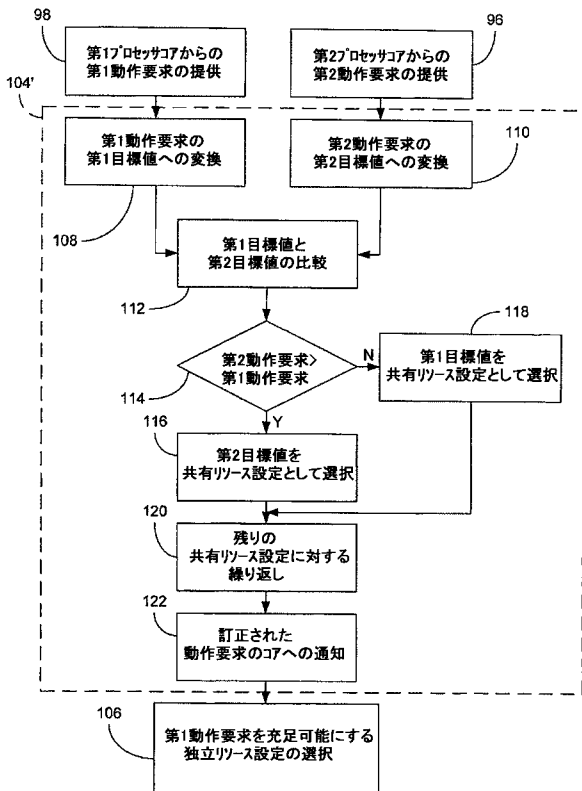
【図4】



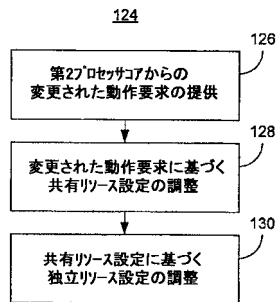
【図6】



【図7】



【図8】



フロントページの続き

(72)発明者 ロテム, エフライム

イスラエル国 34400 ハイファ ヴィソ ストリート 8

(72)発明者 ワイズマン, エリエゼル

イスラエル国 34995 ハイファ アルベルト シュバイツァー ストリート 10

審査官 田中 友章

(56)参考文献 特表2003-523563(JP, A)

特開2004-021574(JP, A)

特開2002-099433(JP, A)

特開2000-066776(JP, A)

国際公開第01/01228(WO, A1)

特開2002-215599(JP, A)

特開2004-078940(JP, A)

(58)調査した分野(Int.Cl., DB名)

G06F 1/32

G06F 1/04

G06F 9/50