

(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2007-128556

(P2007-128556A)

(43) 公開日 平成19年5月24日(2007.5.24)

(51) Int. Cl.	F I	テーマコード (参考)
G06F 15/78 (2006.01)	G06F 15/78 510P	5B062
G06F 1/04 (2006.01)	G06F 1/04 301C	5B079

審査請求 有 請求項の数 17 O L (全 27 頁)

(21) 出願番号	特願2007-27444 (P2007-27444)	(71) 出願人	590000879
(22) 出願日	平成19年2月7日(2007.2.7)		テキサス インストルメンツ インコーポ レイテッド
(62) 分割の表示	特願平9-6718の分割		アメリカ合衆国テキサス州ダラス, ノース セントラルエクスプレスウェイ 135 OO
原出願日	平成9年1月17日(1997.1.17)	(74) 代理人	100066692
(31) 優先権主張番号	010136		弁理士 浅村 皓
(32) 優先日	平成8年1月17日(1996.1.17)	(74) 代理人	100072040
(33) 優先権主張国	米国 (US)		弁理士 浅村 肇
		(74) 代理人	100094673
			弁理士 林 拓三
		(74) 代理人	100091339
			弁理士 清水 邦明

最終頁に続く

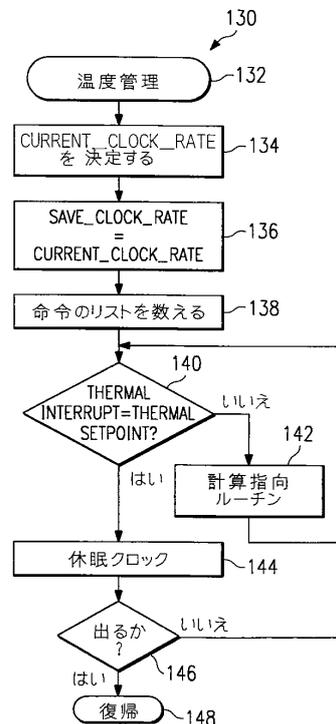
(54) 【発明の名称】 CPUの動作特性に応じてコンピュータの動作を制御する方法と装置

(57) 【要約】

【課題】 ポータブルコンピュータのCPUの動的動作特性を検出して活動レベルを予測し、電力節約や温度管理を動的に行う方法と装置を提供する。

【解決手段】 CPUが第1クロック速度で動作中に(134)少なくとも1つの動的CPU動作特性を検出する(140)方法と回路を含む。この装置(130)は、少なくとも1つの検出されたCPUの動的動作特性が前記少なくとも1つの動的動作特性に関連する所定の設定点に対して所定の関係を確立する(140)ことにより、設定点割り込み条件が存在するかどうかを決定する(140)。設定点割り込み条件が存在する場合は、第1クロック速度に対してクロック速度を制御する(144)。設定点割り込み条件が存在しない場合は、割り込み条件を決定してクロック速度を制御する上記ステップを繰り返す。またこの方法と装置(130)はCPUが計算指向状態にあるかどうかを決定する(142)。

【選択図】 図5



【特許請求の範囲】**【請求項 1】**

検出されたプロセッサ動作特性を制御する方法であって、

- (a) 第 1 のクロック速度で動作するプロセッサの少なくとも 1 つの動的動作特性を検出し、
 - (b) 前記少なくとも 1 つの動的動作特性に関連する所定の設定点に対して所定の関係を前記少なくとも 1 つの動的動作特性が確立することにより、設定点割り込み条件が存在することを決定し、
 - (c) 前記設定点割り込み条件が存在する場合は、前記第 1 のクロック速度を変更し、
 - (d) 前記設定点割り込み条件が存在しない場合は、前記プロセッサが計算束縛状態にあるかを決定する
- ステップを含む、方法。

10

【請求項 2】

請求項 1 に記載の方法において、前記プロセッサが計算束縛状態にある場合は、前記計算束縛状態中に前記割り込み条件が存在するかをさらに決定する、方法。

【請求項 3】

請求項 1 に記載の方法において、前記プロセッサが計算束縛状態にある場合は、前記少なくとも 1 つの動的動作特性が前記計算束縛状態中の前記割り込み条件内に存在するかを決定し、前記計算束縛状態中に前記割り込み条件が存在する場合は、前記割り込み条件を変更する、方法。

20

【請求項 4】

請求項 1 に記載の方法において、前記プロセッサが計算束縛状態にある場合は、前記計算束縛状態中に前記プロセッサが実行すべき計算束縛命令の量を決定することにより前記計算束縛状態の期間を決定する、方法。

【請求項 5】

請求項 1 に記載の方法において、前記プロセッサが計算束縛状態にある場合は、前記計算束縛状態中に前記プロセッサが実行すべき計算束縛命令の量を決定することにより前記計算束縛状態の期間を決定し、前記計算束縛状態中に前記割り込み条件が存在する場合は前記割り込み条件を変更する、方法。

30

【請求項 6】

請求項 1 に記載の方法において、前記プロセッサが計算束縛状態にある場合は、前記計算束縛状態の期間を決定し、さらに前記計算束縛状態中の前記割り込み条件の存在を避けるために前記第 1 のクロック速度を変更する、方法。

【請求項 7】

検出されたプロセッサ動作特性を制御する装置であって、

- (a) 第 1 のクロック速度で動作するプロセッサの少なくとも 1 つの動的動作特性を検出する検出回路と、
 - (b) 前記少なくとも 1 つの動的動作特性に関連する所定の設定点に対して所定の関係を前記少なくとも 1 つの動的動作特性が確立することにより、設定点割り込み条件が存在することを決定するために前記プロセッサ上で動作する設定点命令と、
 - (c) 前記設定点割り込み条件が存在する場合に、前記第 1 のクロック速度を変更するための制御命令と、
 - (d) 前記プロセッサが計算束縛状態にある場合に、前記プロセッサが計算束縛状態にあるかを決定するための計算束縛決定命令と
- を含む、装置。

40

【請求項 8】

請求項 7 に記載の装置において、前記プロセッサが計算束縛状態にある場合に、前記計算束縛状態中に前記割り込み条件が存在するかを前記計算束縛決定命令がさらに決定する、装置。

【請求項 9】

50

請求項 7 に記載の装置において、前記設定点命令は、前記少なくとも 1 つの動的動作特性が前記計算束縛状態中の前記割り込み条件内に存在するかを決定する命令をさらに含み、前記計算束縛状態中に前記割り込み条件が存在する場合に前記割り込み条件を変更するための変更命令をさらに含む、装置。

【請求項 10】

請求項 7 に記載の装置において、前記プロセッサが計算束縛状態にある場合に、前記計算束縛状態中に前記プロセッサが実行すべき計算束縛命令の量を決定することにより前記計算束縛状態の期間を決定する、装置。

【請求項 11】

請求項 7 に記載の装置において、前記計算束縛決定命令は、前記計算束縛状態中に前記プロセッサが実行すべき計算束縛命令の量を決定することにより前記計算束縛状態の期間をさらに決定し、さらに、前記計算束縛状態中に前記割り込み条件が存在する場合に前記割り込み条件を変更する、装置。

10

【請求項 12】

請求項 7 に記載の装置において、前記プロセッサが計算束縛状態にある場合に、前記計算束縛決定命令は、前記計算束縛状態の期間をさらに決定し、さらに、前記計算束縛状態中の前記割り込み条件の存在を避けるために前記第 1 のクロック速度を変更する、装置。

【請求項 13】

請求項 1 ~ 6 のいずれかに記載の方法において、前記プロセッサは中央処理ユニット (CPU) である、方法。

20

【請求項 14】

請求項 7 ~ 12 のいずれかに記載の装置において、前記プロセッサは中央処理ユニット (CPU) である、装置。

【請求項 15】

請求項 1 ~ 6 のいずれかに記載の方法において、前記プロセッサは計算装置内にある、方法。

【請求項 16】

請求項 1 ~ 6 のいずれかに記載の方法において、前記ステップ (a) ~ (d) は計算装置内で発生する、方法。

【請求項 17】

請求項 7 ~ 12 のいずれかに記載の装置を含むシステム。

30

【発明の詳細な説明】

【技術分野】

【0001】

この発明は実時間のコンピュータ中央処理装置 (CPU) 制御に関し、より詳しくは、中央処理装置 (CPU) 内の実時間活動レベルから生じる CPU 温度や温度の変化や電力消費などの動的動作特性を検出し、これに基づいて CPU のクロック速度を調整する装置と方法に関する。

【背景技術】

【0002】

パーソナルコンピュータ産業の初期の発展段階では、移動可能なポータブルコンピュータは非常に人気があった。初期のポータブルコンピュータは大きな電源を用いており、実際は小型のデスクトップパーソナルコンピュータであった。しかし現在のポータブルコンピュータはデスクトップパーソナルコンピュータにくらべて小型で軽く、しかもユーザはデスクトップコンピュータと同じソフトウェアを用いることができる。

40

【0003】

第 1 世代の「ポータブル」コンピュータは交流電力だけで動作した。より新しい次世代のパーソナルコンピュータは電池を使っているので真の意味でポータブルである。このように可搬性が向上したのは、ディスプレイ技術や、ディスクの記憶容量や、要素の重量および容積技術が発達したためである。

50

【 0 0 0 4 】

このような発達によってポータブルコンピュータの大きさは減少したが、動作上の限界はまだ存在する。たとえば、現在のポータブルコンピュータが用いるソフトウェアはデスクトップパーソナルコンピュータが用いるものと一般に同じである。したがって、電池で動作するポータブルコンピュータに特有の機能的限界や考慮すべき点があるにもかかわらず、ポータブルコンピュータはデスクトップコンピュータの機能をすべて備えなければならない。ポータブルコンピュータは一方ではデスクトップコンピュータと同じデータ流れ速度を持つが、他方では電力供給が短時間に限られ、またデスクトップコンピュータにくらべて熱放散能力が限られている。現在のポータブルコンピュータは用いるソフトウェアの異なるプラットフォームに対して特別の対応策を持たない。オペレーティングシステム（たとえばMS-DOS）や、基本入出力システム（BIOS）ソフトウェアや、第三者の応用ソフトウェアは、ポータブルコンピュータでもデスクトップコンピュータでも実質的に同じである。これは特に、CPUの異なる動的動作特性をソフトウェアシステムが処理する方法において当てはまる。

10

【 0 0 0 5 】

ソフトウェア技術者やプログラマがより高機能のソフトウェアシステムを開発するに従って、デスクトップコンピュータはシステム性能の実質的にすべての分野において性能を高めることができる。演算能力を高めたCPUの採用から始まってメモリ容量の増加やディスクドライブの高速高性能化まで、デスクトップコンピュータの動作能力に対する要求は急速に高まっている。ポータブルコンピュータのメーカーは、このような要求に何とかついてゆくしかない。

20

【 0 0 0 6 】

しかし現在まで、ポータブルコンピュータは交流電力でだけか、または大きくて重い電池で動作している。デスクトップコンピュータや新しいソフトウェアの性能要求に追従するため、ポータブルコンピュータの中には高価な要素を用いて必要電力を削減しているものがある。それでも、コンピュータ用の重い電池は長時間にわたって電力を供給することができない。つまり従来のポータブルコンピュータのユーザは、第三者の応用ソフトウェアシステムが期待する性能を出すためには、電源配線から得た交流電流を用いるか、電池で短時間だけ動作するかしかない。

【 発明の開示 】

30

【 発明が解決しようとする課題 】

【 0 0 0 7 】

電力を節約する、すなわち電池で長く動作できるポータブルコンピュータを設計するため、ユーザがコンピュータを使用していない間はポータブルコンピュータの電力消費を減らすようにしたポータブルコンピュータ電源方式もある。またある設計では、キーボードを用いていないときはコンピュータのディスプレイを消すことにより電力を節約する。このような方式はたしかに電力を節約するが、コンピュータの動作を妨げるので実用的でないか、またはコンピュータの中で最大の電力消費が起こったときに電力を節約することができない。これまでのポータブルコンピュータの電力節約方式は、意味のある仕事にユーザがコンピュータを使っているときは電力を節約することができない。

40

【 0 0 0 8 】

しかしポータブルコンピュータの電力供給方式の問題はこれだけではない。現在のポータブルコンピュータシステムには、CPUの動的動作特性に応えるような知的なシステムがない。CPUの動的動作特性は、CPU温度や、温度の変化や、電力消費などの広範囲の特性を含む。電池の電力の節約や最適な使用を管理する方法や装置は存在しない。CPUが複雑な命令を計算するときに、高速で動作する高密度の電子回路の温度を適切に監視して管理する方法や装置は存在しない。さらに、CPUを入出力回路から実際上切り離れたときに、CPUの動的動作特性から望ましくない影響が起こらないようにする方法も装置も存在しない。起こり得る望ましくない状態または影響とは、CPUが過度に電力を消費することや、入出力機能を含まない大量の命令をCPUが実行するときに過大な温度に

50

達することなどである。

【課題を解決するための手段】

【0009】

CPU内の活動レベルを予測し、この予測を用いてCPUの動的動作特性を自動的に管理する装置と方法が必要である。

【0010】

ユーザが自動的な活動レベルの予測を修正し、この修正した予測を用いてCPUの動的動作特性を自動的に管理することができる、フィードバックの方法と装置が必要である。

【0011】

CPUの動的動作特性の管理に応じてクロック速度を実時間で減少させたり回復させたりしてCPUを休止の期間から完全な処理速度に戻し、しかもソフトウェアプログラムやユーザには透明であるような装置と方法がさらに必要である。

【0012】

さらに、動的動作特性に基づいて、CPUの活動レベルに従ってCPUを休止させてよいまたは休止させなければならないかどうかを決定し、この決定に従ってハードウェアセクタを起動するような装置と方法が必要である。CPUを休止させずなわち休眠させてよい場合は、ハードウェアセクタは休眠クロックレベルの振動を与える。CPUを活動させなければならない場合は、ハードウェアセクタは高速クロックレベルの振動を与える。

【0013】

この発明は、CPUの活動状態だけでなく、オペレータと動作中の応用ソフトウェアプログラムの活動を調べる。この発明はこの活動を実時間でサンプリングし、CPUの性能レベルを調整して、電力節約や、コンピュータ電力および温度条件や、また他の任意の適当な所望の動的動作特性を管理する。これらの調整はCPUサイクル内で行うので、ユーザは性能や他のソフトウェアプログラムの動作が変わったことには気が付かない。

【0014】

この発明の好ましい実施態様は検出したCPUの動的動作特性を制御する方法と装置を与えるもので、少なくとも1台のCPUの動的動作特性を検出するステップを含む。この方法と装置は、動作特性を検出するときにCPUのクロック速度を検出する。少なくとも1つの動的動作特性が前記少なくとも1つの動的動作特性に関連する所定の設定点に対して所定の関係を確認することにより設定点割り込み条件が存在することを決定することが、この発明の方法の次のステップである。設定点割り込み条件が存在する場合は、この方法と装置はCPUのクロック速度を制御して所定の設定点を調整するか、または別の方法でCPUの動的動作特性を管理する。他方、設定点割り込み条件が存在しない場合は、この方法と装置は上記の決定および制御のステップを繰り返し、CPUの動的動作特性を効果的に管理する。

【0015】

したがってオペレーティングシステム/BIOSの第三者ソフトウェアのオペレータがコンピュータを使用していないときは、この発明は電力を節約するために、必要になるまでCPUを急速に止めるかまたは遅くして、動作特性をより低い状態に移行させる。必要になると、このシステムは性能が変わったことに気付かれずにCPU動作を急速に完全に回復する。「遅い」モードから完全な動作への切り替えを行うのにはユーザが要求する必要がなく、またコンピュータが「作動可能」状態に戻るのを待っているときにコンピュータの動作が遅れることはない。

【0016】

この発明の技術的利点は、CPUの動作特性を予見すなわち予測できることである。この発明は、命令の数を数えてその種類を決定することにより、入出力がほとんどない計算指向モードにCPUが入るかどうかを決定するので、CPUの遅れやCPUへの入出力は起こらない。この予見により、関連する所定の設定点に1つ以上のCPUの動的動作特性が達するかどうかを、計算指向モード中に決定することができる。達するようであれば、

10

20

30

40

50

この発明はクロック速度を調整したり、所定の設定点を修正したり、またはその他のCPU動作の変更を行うことにより、特定の所定の設定点に達して悪い結果が生じることがないようにする。

【0017】

この発明のさらに別の技術的利点は、単一CPUコンピュータの場合だけでなく、多数のまたは並列のCPUコンピュータシステムの場合も、CPUの動的動作特性の変化に対応できることである。実際に、この発明により1組の並列CPUを構成して、CPUの動的動作特性に関して起こる割り込み条件に応じて1台以上のCPUを調整することにより、すべてのCPUの間で所望の平均クロック速度を維持することができる。

【0018】

以下の説明は、この発明の電力節約に関する。しかしこの発明は電力節約または消費の他に、他のCPUの動的動作特性を制御する方法と装置をさらに含む。

【0019】

任意のシステムのコンピュータ活動の期間を調べる場合、CPUおよび関連する要素には利用率がある。たとえばユーザがキーボードからデータを入力する場合は、キーを打つ間の時間はCPUサイクルに比べて非常に長い。この間にコンピュータは、報告書を印刷するなど多くのことを行うことができる。報告書を印刷している間でも、クロック/カレンダー表示の背景を更新するなど別の動作を行う時間がある。それでもなお、CPUが用いられていない空き時間がほとんど常にある。この発明はこの空き時間にCPUを止めまたは遅くして実時間の電力節約を行い、コンピュータの電池の寿命を延ばす方法を提供する。

【0020】

この発明の好ましい実施態様では、MS-DOSだけでなく、OS/2、UNIX（登録商標）、Windows（登録商標）などのオペレーティングシステムやアップルコンピュータのオペレーティングシステムの下でのCPUの動的動作特性を制御するためには、ハードウェアとソフトウェアの組み合わせを必要とする。この実施態様は任意のシステムで動作するので、実現する際にはシステム毎にわずかな違いはあるものの、この発明の範囲はMS-DOSまたはその他の特定のオペレーティングシステムの下で動作するコンピュータシステムに限定されないことに注意していただきたい。

【0021】

この発明の好ましい実施態様では、コンピュータシステムの要素を遅くしまたは停止させることにより電力消費を減らし、動作温度を下げ、またはその他の方法で動的動作特性を制御する。もちろん制御または変化の量はシステムによって異なってよい。したがってこの実施態様では、クロックを止めれば（しかし実際にはクロックを止められないCPUもある）クロックを遅くするだけより電力の消費を減らすことができる。

【0022】

一般に、毎秒のCPU動作（すなわち命令）の数はCPUクロックにほぼ比例すると考えてよい。

$$\text{命令/秒} = \text{命令/サイクル} * \text{サイクル/秒} \quad (1)$$

【0023】

簡単のために、同じ命令を繰り返し実行して毎秒の命令が一定であると仮定すると、上の関係は次のようになる。

$$F_q = K_1 * Clk \quad (2)$$

ただし、 F_q は毎秒の命令数、 K_1 はサイクル毎の命令の単位を持つ値、 Clk は毎秒のサイクル数を表す。したがって大まかに言えば、実行速度はCPUクロックの周波数と共に増加する。

【0024】

任意の瞬間にコンピュータが使う電力量もCPUクロックの周波数に、したがってCPUの実行速度に関係する。一般にこの関係は次の式で表される。

$$P = K_2 + (K_3 * Clk) \quad (3)$$

10

20

30

40

50

ただし、Pは電力（ワット）、 K_2 はワットの単位を持つ定数、 K_3 はワット秒/サイクルの数を表す定数、Clkは毎秒のCPUクロックのサイクル、である。式（3）から、CPUクロック周波数が増加すると所定の時刻にCPUが消費する電力量も増加することが分かる。

【0025】

所定の期間Tをn個の間隔に分けて、各間隔中の電力Pが一定になるようにすることができる。また全期間中にCPUが消費するエネルギーの量Eは次式で表される。

$$E = P(1) T_1 + P(2) T_2 + \dots + P(n) T_N \quad (4)$$

さらに、CPUクロックClkは「オン」か「オフ」の2つの状態だけと仮定する。ここで「オン」状態はCPUが動作する最大周波数でのCPUクロックの状態を表し、「オフ」状態は最小周波数でのクロックの状態を表す。この最小クロック速度は、自分のクロックを止めることができるCPUではゼロである。

10

【0026】

CPUクロックが常に「オン」という条件では前式の各P(i)は等しく、全エネルギーは

$$\begin{aligned} E(\max) &= P(\text{オン}) * (T_1 + T_2 + \dots + T_N) \\ &= P(\text{オン}) * T \end{aligned} \quad (5)$$

である。ただし、P(オン)はクロックが「オン」状態の時に消費する電力を表し、P(オフ)はクロックが「オフ」のときに消費する電力を表す。これは、CPUの動的動作特性を制御しない場合の、コンピュータの最大電力消費を表す。間隔1から4の一部でCPUクロックが「オフ」の場合は、各間隔は2つの電力レベルを含む。クロックが「オン」の時間間隔をすべて合計してT(オン)とし、「オフ」の間隔を合計してT(オフ)とすると、次式が得られる。

20

$$T = T(\text{オン}) + T(\text{オフ}) \quad (6)$$

【0027】

ところで、CPUが期間Tの間に使うエネルギーは次式のようにになる。

$$E = [P(\text{オン}) * T(\text{オン})] + [P(\text{オフ}) * T(\text{オフ})] \quad (7)$$

【0028】

これらの条件の下で、CPUが消費する全エネルギーは時間間隔T(オフ)を多くすることにより減少させることができる。したがってクロックが「オフ」状態にある期間を制御することにより、この実施態様はCPUが使うエネルギーの量を減らす。期間T中のT(オフ)の期間を多数の間隔に分けると、各間隔の幅がゼロに近づくに従ってエネルギー消費は最大値になることがこの実施態様から分かる。逆に、T(オフ)の間隔の幅を大きくするとエネルギー消費は減少する。

30

【実施例】

【0029】

「オフ」の間隔とCPUが通常休止している期間とを一致させると、この実施態様は性能が低下したことをユーザに気付かれずに全エネルギー消費をE(最大)状態から減少させて、コンピュータシステムを動作させることができる。T(オフ)間隔とCPU休止期間とを一致させるために、この実施態様は、閉ループ、たとえば図1に示す閉ループ10、を用いてCPU活動レベルの幅T(オフ)間隔を決定する。図1ではCPU活動レベルをステップ12で決定する。このレベルが直前の決定より増加している場合は、判断14により流れ図はステップ16に進み、この発明はT(オフ)間隔を減少させる。ステップ16から流れはステップ12に進んで、CPU活動レベルを決定する。他方、CPU活動レベルが直前の決定より減少している場合は、この発明はステップ18に示すようにT(オフ)間隔を増加させ、流れはステップ12に進んで再びCPU活動レベルを決定する。このようにして、閉ループ10は絶えずT(オフ)間隔を調整して、CPU活動レベルを合わせる。

40

【0030】

どのオペレーティングシステムにも2つの重要な論理の要点がある。すなわち、(1)

50

オペレーティングシステム内のアイドルすなわち「何もしない」ループと(2)通常、応用ソフトウェアが必要とするサービスに利用できるオペレーティングシステム要求チャンネルである。これらの論理の要点と調和する論理を用いることにより、この実施態様は応用ソフトウェアが行うまたは将来行う予定の活動要求の種類を評価することができる。この種の情報を持っているので、この実施態様はフィードバックと制御を行ってCPUの動的動作特性を管理することができる。たとえば活動要求の種類が分かると、電力節約を開始し、スライス期間の決定を開始することができる。スライス期間は時間軸のT(オン)対T(オフ)の数を活動レベルによって計算したものである。

【0031】

この実施態様では、CPUの活動レベルを決定するのに1つの仮定を行う。この実施態様が行う仮定は、サービスを必要とするソフトウェアプログラムは通常さらに別のサービスを必要とすることと、サービス要求の間の期間を用いて、コンピュータ上を走って電力などのCPUの動的動作特性を制御する任意の応用ソフトウェアの活動レベルを決定したり節約のためのスライス数を決めたりしてよいことである。

10

【0032】

たとえば、電力節約スライス中に、すなわちT(オフ)中に、この実施態様がCPUに割り込みを行うと、CPUは割り込みソフトウェアに向かう前に割り込みを受けたルーチンの状態を保存する。電力節約ソフトウェアはこの節約スライス中に動作していたので、この実施態様は活動的な電力節約ループに制御を返す。前記活動的な電力節約ループはCPUクロックを監視するだけであって、電力節約モードから出る条件が存在することを決定してT(オフ)状態からT(オン)状態に出る。図1に関して上に説明したように、処理の流れは監視した活動レベルに従って次の電力節約状態の間隔を調整する。ある実現ではハードウェア論理によってT(オフ)からの自動的な出口を作り、電力節約ループから自動的に出て、T(オン)間隔中にCPU命令を実行する。

20

【0033】

図2から図4は、この実施態様の活動的な電力モニタ機能を示す。動作を説明すると、CPUのリードオンリーメモリ(ROM)が記憶するプログラムから、またはランダムアクセスメモリ(RAM)内にプログラムを記憶する外部装置から、CPUは活動的な電力モニタ機能20をインストールする。CPUが活動的な電力モニタ機能20をロードすると、活動的な電力モニタ機能20の処理の流れは初期化ステップ22に進んで、システム割り込みの初期化や、ユーザ構成の設定や、システム/応用の特定の初期化のステップを行う。アイドルすなわち「何もしない」機能のためのハードウェアまたはソフトウェア割り込みが発生すると、アイドル分岐24を実行する。これについては図3aに詳細に示す。アイドルすなわち「何もしない」ループ(すなわち計画的な休止状態)に入るCPUはこの種類の割り込みを発生する。オペレーティングシステムや、入出力サービス要求や、応用プログラムや、内部オペレーティングシステム機能によって起こるソフトウェアまたはハードウェア割り込みが発生すると、流れ図の活動分岐26を実行する。これについては図4に詳細に示す。

30

【0034】

プログラムはたとえばディスク入出力機能や、読み出しや、印刷や、ロードや、その他のサービスの要求であってよい。選択した分岐が何であっても、復帰ベクトル28は処理の流れを最終的にCPUのオペレーティングシステムに戻す。好ましい実施態様は、CPUが活動的な電力モニタ機能20をROM内のプログラムによりロードする場合は1度だけ、またCPUがモニタ機能20を外部の装置のRAMからロードする場合はパワーアップ中は毎回、初期化分岐(図2に詳細に示す)を実行する。活動的な電力モニタ機能20の初期化分岐22が完全にモニタ機能を行った後、処理の流れはCPU活動の種類に従ってアイドル分岐24か活動分岐26に分岐する。オペレーティングシステムが電力節約モードに入ると、CPU活動の種類に従ってアイドル分岐24か活動分岐26を選択する。すなわち、計画的な休止中の電力節約ではアイドル分岐24を選択し、CPU活動中の電力節約では活動分岐26を選択する。

40

50

【0035】

さらに図2の初期化分岐22を詳細に見ると、すべてのシステム割り込みと変数を初期化した後、処理の流れはステップ30で、POWER_LEVELパラメータをDEFAULT_LEVELに等しく設定する。POWER_LEVELの入力をユーザが制御するオペレーティングシステムでは、ステップ32でプログラムはUSER_LEVELパラメータをすでに選択したかどうか調べる。USER_LEVELパラメータがゼロより小さいかまたはMAXIMUM_LEVELより大きい場合は、システムはDEFAULT_LEVELパラメータ値を用いる。そうでない場合は、初期化分岐22はステップ34に進んで、POWER_LEVELパラメータ値をUSER_LEVELパラメータに等しく設定する。

10

【0036】

この発明のこの実施態様では、初期化分岐22はステップ36で変数IDLE_TICKをゼロに、変数ACTIVITY_TICKをゼロに設定する。MS/DOSを用いる場合は、IDLE_TICK変数は「何もしない」ループ内の割り込みの数である。ACTIVITY_TICK変数は、活動割り込みによる割り込みの数を表す。この割り込みの数はCPU活動レベルを決定する。チックのカウントは次の割り込みのデルタ時間 T_i を表す。IDLE_TICKは、ソフトウェア割り込みが起こってこの期間に書きするのでなければ、チックの間の一定デルタ時間 T_i である。すなわち、ソフトウェア割り込みは割り込みの間のデルタ時間を再プログラムすることができる。

20

【0037】

IDLE_TICK変数とACTIVITY_TICK変数をゼロに設定した後で、初期化分岐22は設定ステップ38に進み、ここで初期化分岐22はシステムに特有の詳細を用いて応用に特有の構成の微同調を行い、設定ステップ38で活動的な電力モニタ機能20を初期化する。次に初期化分岐22はステップ40で割り込み入出力にハードウェアへの命令を準備する。これは次の割り込みでそのハードウェアが制御を受け持つてよいことを示す。次に初期化分岐22は復帰ベクトル28を経て、オペレーティングシステムに、または活動的な電力モニタ20を呼んだ元の機能に出る。

【0038】

図3aは、活動的な電力モニタ機能20のアイドル分岐24を詳細に示す。CPUの計画的な休止期間に応じて、活動的な電力モニタ機能20は、アイドル分岐24に入ることができるかどうか知るためにまず活動割り込みAが現在使用中かどうか決定する。判断42でパラメータBUSY_Aが再入フラグBUSY_FLAGの値に等しい場合は、現在CPUは使用中であって休眠にはできないことを意味する。したがって活動的な電力モニタ機能20はすぐ復帰Iステップ44に進んでルーチンから出る。復帰Iステップ44は、前のオペレーティングシステムのアイドルベクトル割り込みに戻って通常の処理を行うための間接ベクトルである。CPUは、活動的な電力モニタ機能20に入る前にこの間接ベクトルを記憶する。言い換えると、復帰Iステップ44によりすぐ前の連鎖ベクトルに割り込み復帰する。

30

【0039】

判断42でBUSY_A割り込みフラグが使用中でないと決定した場合は、活動的な電力モニタ機能20はステップ46で、BUSY_IDLE割り込みフラグがBUSY_FLAGに等しいかどうか調べる。等しければ、活動的な電力モニタ機能はすでにアイドル分岐24にあることを示し、したがってCPU割り込みがあってはならない。BUSY_IがBUSY_FLAGに等しい場合は、アイドル分岐24の処理の流れはステップ44に進む。

40

【0040】

しかしBUSY_A再入フラグもBUSY_I再入フラグもセットされていない場合は、再入保護のために、ステップ48でアイドル分岐24はBUSY_Iフラグをセットする。すなわちBUSY_IをBUSY_FLAGと同じ値にする。ステップ50で、IDLE_TICK変数を1だけ増やす。IDLE_TICK変数はあるT(オフ)間隔の前

50

のT（オン）の数である。アイドル割り込みおよび設定割り込みの数とCPU活動レベルによりIDLE_TICKの値が決まる。IDLE_TICK変数を1だけ増やして事象を平滑化する。これにより、限界(critical)入出力活動は平滑化を制御することができる。

【0041】

ステップ52で、活動的な電力モニタ機能20はIDLE_TICK変数が所定の一定値IDLE_MAXTICKSに等しいかどうか調べる。IDLE_MAXTICKSは初期化分岐22の設定ステップ38で初期化する定数の1つで、所定のCPUでは一定である。この実施態様では、一定のパラメータIDLE_MAXTICKSにより活動レベルの自己同調ができる。変数IDLE_TICKが定数IDLE_MAXTICKSに等しくない場合は、アイドル分岐24はステップ54でBUSY_Iフラグをクリアし、復帰I間接ベクトル44に進んで処理の流れから出る。しかしIDLE_TICK変数がIDLE_MAXTICKSに等しい場合は、ステップ56でIDLE_TICK変数をIDLE_START_TICKS定数（ゼロまたは非ゼロ）に等しくする。

【0042】

一般に、特定のCPUがそのクロックを停止してしかも正しく機能することができる場合は、IDLE_START_TICKSはゼロである。そうでない場合は、IDLE_START_TICKSは所定のCPUに適した所定の値を取る。このステップは、活動的な電力モニタ機能20が残りの休眠機能を何度実行してよいかを指定することにより、この実施態様の自己同調の態様を決定する。IDLE_START_TICKSをIDLE_MAXTICKSから1を引いた値に等しく設定することにより、この実施態様は連続的なT（オフ）間隔を得る。判断58で、アイドル分岐24はPOWER_LEVELパラメータのレベルを調べる。POWER_LEVELパラメータがゼロの場合は、アイドル分岐24はステップ54でBUSY_Iフラグをクリアし、復帰I間接ベクトル44に進んで、制御をオペレーティングシステムに返す。したがって、オペレーティングシステムは活動的な電力モニタ機能20に入る前に行っていた仕事を続けてよい。

【0043】

しかし、判断58でPOWER_LEVELパラメータの値がゼロでない場合は、アイドル分岐24は割り込みマスクがあるかどうか決定する。システム/応用ソフトウェアはINTERRUPT_MASK変数を設定して、活動的な電力モニタ機能20に割り込みを行ってよいかどうかを決定する。判断60で割り込みマスクがNOT_AVAILABLEと決定した場合は、アイドル分岐24はBUSY_I再入フラグをクリアして制御をオペレーティングシステムに返し、CPUは活動的な電力モニタ機能20に入る前に実行していた仕事を続ける。オペレーティングシステムも応用ソフトウェアも、割り込みマスクをNOT_AVAILABLE値すなわちフラグと同じ値に設定することにより、連続的なT（オン）状態を作るようにT（オン）間隔を設定することができる。

【0044】

割り込みマスクがAVAILABLEの場合は、活動的な電力モニタ機能20は電力節約サブルーチン62に進み、ハードウェア状態が確立したT（オフ）の1期間中、電力節約サブルーチン62を十分に実行する。たとえば好ましい実施態様では、可能な最長間隔は恐らく18ミリ秒で、これは実時間クロックからの2つのチックすなわち割り込みの間の最長時間である。電力節約サブルーチン62の間は、CPUクロックは速度を落として休眠クロックレベルになる。

【0045】

限界入出力動作によりT（オン）間隔が開始すると、アイドル分岐24の割り込みは引き続き別の限界入出力要求に対して作動可能である。CPUが限界入出力で使用中になると、利用可能なT（オフ）間隔は少なくなる。逆に限界入出力要求が減少してその間の時間間隔が増加すると、利用可能なT（オフ）間隔は多くなる。アイドル分岐24は、活動割り込みからのフィードバック（CPU活動レベルが低下するにつれてT（オフ）間隔を増やす）に基づく自己同調システムを与える。

10

20

30

40

50

【 0 0 4 6 】

図 3 b と関連する説明によって示すように、活動的な電力モニタ機能 2 0 が電力節約サブルーチン 6 2 を完了すると、ステップ 5 4 で B U S Y _ I 再入フラグをクリアし、流れは復帰ベクトル 4 4 を経て、活動的な電力モニタ機能 2 0 を要求した元のオペレーティングシステムに戻る。

【 0 0 4 7 】

図 3 b をより詳細に参照すると、流れ図 8 0 は電力節約サブルーチン 6 2 の説明である。活動的な電力モニタ機能 2 0 はステップ 8 2 で入出力ハードウェア高速クロックを決定する。電力節約サブルーチン 6 2 の次のステップ 8 4 で、C U R R E N T _ C L O C K _ R A T E を適当な高速クロックに等しく設定して、多レベルの高速クロックを持つ C P U にこの値を保存する。したがって、ある C P U が 1 2 M H z と 6 M H z の高速クロックを持つ場合は、活動的な電力モニタ機能 2 0 は、活動的な電力モニタ機能 2 0 が電力を減少させる前に、どちらの高速クロックで C P U を制御するかを決めなければならない。これにより C P U は、目覚めたときに C P U を適切な高速クロックで再設定することができる。ステップ 8 4 で、電力節約サブルーチンは S A V E _ C L O C K _ R A T E 変数を、ステップ 8 2 で決定した C U R R E N T _ C L O C K _ R A T E の値に等しくする。C P U の高速クロックが 1 個だけの場合は S A V E _ C L O C K _ R A T E 8 4 は使わない。次に活動的な電力モニタ機能 2 0 は休眠クロックステップ 8 6 に進み、パルスハードウェアセレクタ (図 7 に示す) に送ってクロック周波数を下げまたは止めることにより、C P U クロックを休眠させる。入出力ポートハードウェア休眠クロックの振動は、C P U クロックの通常の振動よりはるかに低い。

【 0 0 4 8 】

この時点で 2 つの事象のどちらかが起こる。すなわちシステム / 応用割り込みか、実時間クロック割り込みが起こる。システム / 応用割り込み 8 8 が起こる場合は、活動的な電力モニタ機能 2 0 は割り込みルーチン 9 0 に進む。電力節約ルーチン 6 2 はこの割り込みをすぐ処理してステップ 9 2 で入出力割り込みを準備し、判断 9 4 に戻って、割り込みがあるかどうか決定する。この場合は割り込みがあったので、ステップ 9 6 で S A V E _ C L O C K _ R A T E の値を用いて、どちらの高速クロックで C P U を戻すかを決定する。電力節約サブルーチン 6 2 の流れ図 8 0 は復帰ベクトル 9 8 で終了する。

【 0 0 4 9 】

しかし判断 9 4 でシステム / 応用割り込みがないと決定した場合は、電力節約サブルーチン 6 2 は判断 9 4 で実時間クロック割り込みが起こるまで待ち続ける。実時間割り込みが起こると、電力節約サブルーチン 6 2 は C P U を、記憶した S A V E _ C L O C K _ R A T E に再設定する。休眠クロック速度が停止クロック速度でなかった場合、すなわち休眠クロック速度がゼロでなかった場合は、制御は遅いクロックで進み、電力節約サブルーチン 6 2 は判断 9 4 とステップ 8 8 から 9 2 を通る割り込みループを実行する。他方、制御が休眠クロック速度ゼロで進む場合は、電力節約サブルーチン 6 2 は、判断 9 4 とステップ 8 8 から 9 2 を含む割り込みループを一度だけ実行した後 C P U クロックを S A V E _ C L O C K _ R A T E 9 6 に戻し、その後流れは復帰ベクトル 9 8 に進む。

【 0 0 5 0 】

次に図 4 において流れ図 1 0 0 は、システム / 応用の活動要求がオペレーティングシステムサービス要求の割り込みによりトリガする、活動分岐 2 6 の処理の流れを示す。活動分岐 2 6 は再入保護から始まる。ステップ 1 0 2 で活動的な電力モニタ機能 2 0 は、B U S Y _ I 変数が B U S Y _ F L A G 値に等しく設定されているかどうかを決定する。等しい場合は、活動的な電力モニタ機能 2 0 システムは既にアイドル分岐 2 4 にあり、割り込みできないことを意味する。B U S Y _ I が B U S Y _ F L A G 値に等しい場合は、処理の流れは復帰ベクトル 4 4 に進んで、活動的な電力モニタ機能 2 0 から出る。復帰ベクトル 4 4 は元の活動ベクトル割り込みに戻って通常の処理を行うための間接ベクトルである。復帰ベクトル 4 4 は割り込みベクトルにより C P U 動作を通常の位置に戻す。これは要求されたサービスをオペレーティングシステムが行った後に動作する。

【 0 0 5 1 】

B U S Y _ I フラグ変数が B U S Y _ F L A G 値に等しくない場合は、活動的な電力モニタ機能 2 0 がアイドル分岐 2 4 にアクセスしていないことを意味する。活動的な電力モニタ機能 2 0 は判断 1 0 4 で、B U S Y _ A フラグが B U S Y _ F L A G 値に等しいかどうか決定する。等しければ、活動分岐 2 6 はこの時点で制御をコンピュータシステムに戻す。これは、すでに活動分岐 2 6 に入っているので割り込みできないことを意味する。B U S Y _ A フラグがセットされていない場合は、すなわち B U S Y _ A が B U S Y _ F L A G 値に等しくない場合は、活動的な電力モニタ機能 2 0 はステップ 1 0 6 で B U S Y _ A パラメータを B U S Y _ F L A G 値に等しく設定し、実行中の割り込みを防ぐ。

【 0 0 5 2 】

判断 1 0 8 で、活動分岐 2 6 は P O W E R _ L E V E L パラメータ値を決定する。P O W E R _ L E V E L パラメータ値がゼロの場合は、活動的な電力モニタ機能 2 0 はまずステップ 1 1 0 で B U S Y _ A 再入フラグをクリアした後、活動分岐 2 6 から出る。しかし P O W E R _ L E V E L パラメータ値がゼロでない場合は、活動分岐 2 6 は次に、入出力ハードウェアの C U R R E N T _ C L O C K _ R A T E パラメータの値を決定する。図 3 b の電力節約サブルーチンのステップ 8 4 でそうであったように、所定の C P U に多レベルの高速クロックがある場合は、図 4 の活動分岐 2 6 はステップ 1 1 2 で C U R R E N T _ C L O C K _ R A T E パラメータ値を用いる。多レベルでない場合は、C U R R E N T _ C L O C K _ R A T E パラメータ値は常に C P U の高速クロックに等しい。ステップ 1 1 2 で C U R R E N T _ C L O C K _ R A T E パラメータ値を決定した後、ステップ 1 1 4 で I D L E _ T I C K パラメータ値を一定の S T A R T _ T I C K S に等しく設定する。S T A R T _ T I C K S 定数は、前に決定した C U R R E N T _ C L O C K _ R A T E に設定する。活動的な現在の高速クロックが T (オフ) 間隔を設定する。

【 0 0 5 3 】

活動分岐 2 6 の次のステップ判断 1 1 6 では、要求があったことを決定する。要求は、特定の種類の必要なサービスのためにコンピュータ上を走る応用ソフトウェアからの入力である。判断 1 1 6 で、活動分岐 2 6 は要求が限界入出力かどうか決定する。限界入出力は、T (オン) が T (オフ) より大きくなるまで絶えず T (オン) を長くする。要求が限界入出力の場合は、活動的な電力モニタ機能 2 0 はまずステップ 1 1 0 で B U S Y _ A 再入フラグをクリアした後、活動分岐 2 6 から出る。

【 0 0 5 4 】

他方、要求が限界入出力でない場合は、活動分岐 2 6 はステップ 1 1 8 で A C T I V I T Y _ T I C K パラメータを 1 だけ増やす。処理の流れは判断 1 2 0 に進み、A C T I V I T Y _ T I C K パラメータ値が A C T I V I T Y _ M A X T I C K S 一定値に等しいかどうか決定する。判断 1 2 0 のチェックにより限界入出力からの平滑化が可能になり、次の A C T I V I T Y _ T I C K の T (オン) 間隔中にシステムを別の限界入出力に備えさせる。A C T I V I T Y _ T I C K パラメータ値が A C T I V I T Y _ M A X T I C K S 一定値に等しくない場合は、活動分岐 2 6 の処理の流れはステップ 1 1 0 で B U S Y _ A 再入フラグをクリアした後で、復帰 I ベクトル 4 4 に進む。他方、A C T I V I T Y _ T I C K が一定値 A C T I V I T Y _ M A X T I C K S に等しい場合は、ステップ 1 2 2 で、活動分岐 2 6 は A C T I V I T Y _ T I C K パラメータを一定値 L E V E L _ M A X T I C K S に設定する。活動分岐 2 6 は L E V E L _ M A X T I C K S 値を、判断 1 0 8 で決定する特定の電力レベルに設定する。

【 0 0 5 5 】

判断 1 2 4 で、活動分岐 2 6 は割り込みマスクが存在するかどうか決定する。システム / 応用ソフトウェアは割り込みマスクを設定する。割り込みマスクを N O T _ A V A I L A B L E フラグ値に設定すると連続的な T (オン) 状態を作る。これは、現在割り込みが利用できないことを意味する。この場合は、活動分岐 2 6 の処理の流れはステップ 1 1 0 に進み、B U S Y _ A 再入フラグをクリアして、復帰 I ベクトル 4 4 に進む。しかし割り込みマスクが A V A I L A B L E の場合は、活動分岐 2 6 は判断 1 2 6 で、判断 1 1 6 で

10

20

30

40

50

識別した要求が遅い I/O_INTERRUPT だったかどうか決定する。遅い I/O_INTERRUPT 要求は、入出力装置が「作動可能」になるまで遅れてよい。「作動可能にする」動作中に連続的な T (オフ) 間隔が始まって電力の節約を続ける。

【0056】

したがって、要求が遅い I/O_INTERRUPT でない場合は、活動分岐 26 の処理の流れはステップ 110 に進んで BUSY_A 再入フラグをクリアし、続いて復帰 I ベクトル 44 に進む。しかし要求が遅い I/O_INTERRUPT で、かつ入出力装置が「作動可能」になるまでにまだ時間がある場合は、活動分岐 26 は判断 128 で、入出力要求が完了したかどうか決定する。したがって本質的に、判断 128 は関連する入出力装置が作動可能かどうか決定する。入出力装置が作動可能でない場合は、活動的な電力モニ
タ機能 20 は T (オフ) を長くする。これにより CPU は、遅い入出力装置が作動可能になるまで待たされすなわち休眠させられて、電力を節約する時間がある。したがって活動分岐 26 は、図 2c と関連する説明で詳細に示した電力節約サブルーチン 62 に入る。判断 128 で入出力要求が完了したと決定すると、処理の流れはステップ 110 に進んで BUSY_A 再入フラグをクリアし、復帰 I ベクトル 44 を経て、制御はオペレーティングシステムに戻る。

10

【0057】

この発明のこの実施態様は活動的な電力監視を行うだけでなく、他の CPU の動的動作特性を管理する方法と装置も提供する。他の動的動作特性とは、たとえば望ましくない電磁周波数放射線の放出や、この発明に従って繰り返し調整しても CPU が所定の温度帯域
内で動作を継続することができない、などである。詳しく言うと、図 5 に示すように、この実施態様は積極的に温度を管理する方法や、ユーザに透明で図 3b の電力節約サブルーチン 62 と同様な方法で動作する、温度制御用の実時間フィードバックループを含む装置を提供する。

20

【0058】

図 5 は、ステップ 132 から始まる温度管理サブルーチンの流れ図 130 を示す。温度管理サブルーチン 132 を実現するには、活動的な電力モニタ機能 20 を修正して、関連する CPU およびコンピュータシステムにより電力消費だけでなく他の CPU の動的動作特性を監視できるよう、より一般的な活動的なモニタ機能にする。従って本質的に、活動的な電力モニタ機能 20 がアイドル分岐 24 と活動分岐 26 により電力節約サブルーチン
62 を呼ぶのと同様に、より一般的な活動的なモニタ機能は、温度管理サブルーチン 132 に CPU の温度制御動作を行わせるときは温度管理サブルーチン 132 を呼ぶ。しかし重要な差は、電力節約サブルーチン 62 を呼ぶのは活動レベルが低いときであるのに反して、この実施態様は、温度条件が CPU の温度限界に近付いたときまたは近づきそうなときに温度管理サブルーチン 132 を呼ぶ点である。

30

【0059】

詳しく言うと、この実施態様の活動的なモニタ機能は、BUSY_A パラメータが BUSY_FLAG 一定値に等しいときに、活動分岐 26 内で温度管理サブルーチン 132 を呼ぶ。それ以後の処理の流れは温度管理サブルーチン 132 に示すように進む。

【0060】

図 5 をより詳細に参照すると、ステップ 132 は温度管理サブルーチンの最初のステップである。温度管理サブルーチン 132 はステップ 134 で CURRENT_CLOCK_RATE パラメータの値を決定する。次のステップ 136 で、CPU が多レベルの高速クロックを持つ場合は、CURRENT_CLOCK_RATE パラメータを適当な高速クロックに等しく設定し、この値を保存する。したがって、ある CPU が 12MHz と 6MHz の高速クロックを持つ場合は、活動的なモニタ機能は CPU 内の過度の温度または温度変化条件に応じるために、活動的なモニタ機能がどの高速クロックで CPU を制御するかを決定しなければならない。これにより、CPU は目覚めたときに適当な高速で動作を再設定することができる。ステップ 136 で、温度管理サブルーチン 132 はステップ 134 で決定した値に SAVE_CLOCK_RATE パラメータ値を設定する。この実

40

50

施態様では、CPUの高速クロックが1個だけの場合はSAVE_CLOCK_RATEステップ136を用いない方がよい。

【0061】

温度管理サブルーチン132はステップ138に進み、CPUが処理する予定の命令の数を数える。この実施態様は、関連するコンピュータシステムのROMが記憶するルックアップテーブルを含んでよい。ルックアップテーブルは、入出力を要求する命令と入出力なしのCPU動作に関する命令とを区別する。

【0062】

命令のリストを数えるステップ138の次に、判断140はTHERMAL_INTERRUPTパラメータ値が一定値THERMAL_SETPOINTに等しいかどうか決定する。これは温度割り込みの必要性を示す。温度割り込みは、温度条件が望ましくない場合にCPUの動作を止めるか変えるための割り込みである。THERMAL_INTERRUPTパラメータの値がTHERMAL_SETPOINTの値に等しくない場合は、温度管理サブルーチン132の処理の流れは計算指向サブルーチン142に進む。他方、THERMAL_INTERRUPTがTHERMAL_SETPOINTに等しい場合は処理の流れはステップ144に進み、ここでCPUクロックはより低い休眠クロック速度に移行する。処理の流れは休眠クロック速度ステップ144から判断146に進み、処理の流れが温度管理サブルーチン132から出るかどうか調べる。処理の流れが温度管理サブルーチン132から出ない場合は、処理の流れは判断140に戻り、引き続き温度割り込み条件が存在するかどうか調べる。

10

20

【0063】

CPUが温度管理サブルーチン132から出る場合は、処理の流れは判断146から復帰ベクトル148に進む。処理の流れはここから前に説明した活動的なモニタ機能に戻る。これには種々の場合がある。たとえば、活動的なモニタ機能が図3aのアイドル分岐24に飛ぶなど。温度割り込み条件が存在する限り、温度管理サブルーチン132はステップ144のより遅い休眠クロック速度でCPUを動作させる。

【0064】

温度管理サブルーチン132の処理の流れが計算指向サブルーチン142に向かう場合は、処理の流れは図6に示す計算指向サブルーチン流れ図150のようになる。計算指向サブルーチンはステップ142で始まり、処理の流れは判断152に進んで、CPUが計算指向状態にあるかどうか調べる。CPUが計算指向状態になれば、処理の流れは復帰ベクトル154に進む。復帰ベクトル154は、温度管理サブルーチン132の処理の流れをTHERMAL_INTERRUPT判断140に戻す(図5参照)。

30

【0065】

CPUが計算指向状態にある場合は、計算指向サブルーチン142の処理の流れは判断156に進む。この発明が与える技術的な利点は、CPUの動作特性を予見すなわち予測できることである。命令の数を数えてその種類を決定することにより、この実施態様はCPUが計算指向動作モードにあって、入出力はほとんどないか、したがってCPUの減速もCPUへの入出力機能も起こらないかどうか決定する。CPUが計算指向状態にあるかどうかの決定は、命令のリストを数えるステップ138で数えた、CPUが実行する予定の命令の数から得られる。

40

【0066】

この実施態様は、CPUが実行する予定の命令とコンピュータシステムがROMに保持しているルックアップテーブルの要素とを比較する比較器を備える。このルックアップテーブルは、命令の種類が入出力機能を含むものか、入出力機能を含まずにCPUの処理と計算動作だけに関するものかという特徴付けを行う。この実施態様は、入出力機能を含む断続した入出力命令を持たない、連続した計算命令の数についての所定のしきい値を与える。入出力機能を含まない命令の数が所定のしきい値数を超える場合は、この実施態様は、CPUが計算指向状態にあると判断する。

【0067】

50

C P Uが計算指向状態にあると計算指向サブルーチン142が決定すると、処理の流れは判断156に進み、C P Uが計算指向状態中に温度割り込み条件を設定するかどうか調べる。この設定は、C P Uが温度割り込みに達するのを防ぐために、または所望であれば、この温度割り込みを新しい温度割り込みレベルにするために行う。したがって計算指向サブルーチン142が判断156で、計算指向状態中にC P Uが温度割り込みに達すると決定すると、処理の流れは判断158に進む。他方、計算指向状態中に温度割り込み条件が起らない場合は、処理の流れは復帰ベクトル154に進んで、温度管理サブルーチン132内の処理を続ける。

【0068】

たとえばC P Uが計算指向状態中に温度割り込み設定点を必要とはするが、計算指向状態が十分長くないのでC P Uに悪い影響を与えない場合は、この実施態様は一定値THERMAL__SETPOINTを調整することができる。言い換えると、通常の条件下で、C P Uがこの温度に達しても安全の余裕があってC P Uが温度による損傷を受けることがないように、温度割り込み条件を設定する。たとえばC P U温度が一定値THERMAL__SETPOINTに達する理由が分かる場合は、かなりの調整可能な安全余裕がある。計算指向C P Uの外箱の温度の上昇速度に従って、C P Uの温度は既知の予測可能な状態で上昇する。

【0069】

計算指向状態でC P Uが実行する命令の数と種類が分かると、これに基づいて計算指向サブルーチン142は、C P UはC P Uの温度設定点THERMAL__SETPOINTに達するかまたは超えるが、そうなるのはほんの短い期間だけであると決定することができる。計算指向サブルーチン142が決定するこの短い期間ではC P Uの損傷やその恐れはない。この状態に対処するために、計算指向サブルーチン142は判断158で現在のTHERMAL__SETPOINTの値を調整する必要があるかどうか調べる。現在のTHERMAL__SETPOINTの値を調整する必要がある場合は、処理の流れはステップ160に進んで、THERMAL__SETPOINTの値を調整する。ステップ160から、処理の流れは復帰ベクトル154に進む。他方、THERMAL__SETPOINTを現在の値に保つ場合は処理の流れは判断162に進み、THERMAL__INTERRUPTパラメータの値をTHERMAL__SETPOINTの一定値に設定するかどうか調べる。

【0070】

THERMAL__INTERRUPTパラメータの値をTHERMAL__SETPOINTの一定値に設定する場合は処理の流れはステップ164に進み、計算指向サブルーチン142はTHERMAL__INTERRUPTパラメータの値をTHERMAL__SETPOINTの一定値に設定する。次に、処理の流れは復帰ベクトル154に進む。他方、THERMAL__INTERRUPTパラメータの値をTHERMAL__SETPOINTの値に設定しない場合は、処理の流れはそのまま復帰ベクトル154に進む。この後、処理の流れは復帰ベクトル148に進んで、温度管理サブルーチン132に戻る。

【0071】

自己同調は連続フィードバックループの制御システムに固有のものである。この発明のソフトウェアはC P Uの活動が低いとき、したがってこの発明の電力節約態様を起動してよいときを検出することができる。他方この実施態様は、C P Uの活動が高くて温度管理を起動してよい時を決定する。電力節約モニタが起動すると、その間隔内でC P Uクロック動作は最高速度に急速に復帰するのでコンピュータの性能は低下しない。同様に、温度設定点の条件がなくなって温度管理の必要がなくなると、C P U内のC P Uクロック動作は最高速度に急速に復帰して、温度管理の影響はほとんど気付かれない。この最高速度のC P Uクロック動作に急速に復帰させるため、この発明の好ましい実施態様はいくつかの関連するハードウェアを用いる。

【0072】

次に図7は、活動的な電力節約および温度管理用の、この発明に用いる関連ハードウエ

10

20

30

40

50

アを表す簡単化した略図である。CPUがいつでも休眠できる状態にあると決定した場合は、活動的な電力モニタ機能20は入出力ポート(図示せず)に書き込んで休眠線にパルスを出す。休眠線のこのパルスの立ち上がり端で、フリップフロップのQは高に、Q-は低になる。これにより、ANDゲート202および204とORゲート206を含むAND/OR論理は、休眠クロック発振器208から休眠クロック線に出るパルスを選択して、CPUクロックに送って使用する。休眠クロック発振器208は、通常のCPU活動中に用いるCPUクロックより遅いクロックである。

【0073】

フリップフロップ200のQから来る高信号は、休眠クロック発振器208から来るパルスとの論理積をANDゲート202でとり、またフリップフロップ200のQ-からの低信号との論理和をORゲート206でとり、さらに高速クロック発振器210から高速クロック線に出るパルスとの論理積をANDゲート204でとって、CPUクロックを作る。入出力ポートが休眠クロックを指定すると、CPUクロックは休眠クロック発振器208の値に等しくなる。他方、割り込みが起こると割り込み値はフリップフロップ200をクリアし、この割り込みにより、ANDゲート202および204とORゲート206を備えるAND/ORセクタは高速クロック値を選択する。AND/ORセクタは、CPUクロック値を高速クロック発振器210から来る値に戻す。したがってCPUの電力節約動作中にシステム内で割り込みを検出すると、CPU動作を最高クロック速度に回復してから割り込みを受けて処理する。

【0074】

任意のシステムの各CPU以外の必要な関連ハードウェアは、用いるオペレーティングシステムや、CPUが止められるか、などによって異なってよいことに注意していただきたい。しかし、多くの使用可能なポータブルコンピュータシステム内でこの発明が活動的に電力を節約しまたは温度条件を管理するのに、この発明の範囲はシステムに特有の必要な変更によって制限されるものではない。例として、2つの実際の実施態様を図8と図9に示して以下に説明する。

【0075】

現在、多くのVLSI設計ではCPU速度のクロック切り替えが可能である。ゼロクロックまたは遅いクロックから速いクロックに切り替える論理は、ユーザがキーボード命令により速度を変える論理と同じである。このような切り替え論理を活動的なモニタ機能の論理に追加することにより、割り込みを検出するとすぐ高速クロックに戻すことができる。この簡単な論理が、CPUに割り込みを行って割り込みを最高速度で処理できるようにするのに必要なハードウェア支援の鍵である。

【0076】

MS-DOSの下で電力消費および温度を含むCPUの動的動作特性を管理する方法は、MS-DOSアイドルループトラップを用いて、「何もしない」ループにアクセスする。アイドルループは、アイドル状態すなわち低活動状態にある応用ソフトウェアおよびオペレーティングシステムの動作に特殊なアクセスを行うためのもので、システム内の任意の部分で活動レベルを決定するには慎重な調査が必要である。活動レベルを決定するには、割り込み21Hサービス要求からのフィードバックループを用いる。活動レベルの予測は割り込み21H要求から決定し、これから、この発明はCPUを「休眠させる」(すなわち、遅くするかまたは止める)ためのスライス期間を設定する。追加の機能により、ユーザは割り込み21Hの活動レベルに従ってスライスを変更することができる。

【0077】

図8は、インテル80386などのシステム用の実際の休眠ハードウェアの実現の略図である。ちなみに、インテル80386CPUはクロックを止めることはできない。アドレス可能バス220およびアドレスバス222からデマルチプレクサ224にCPU入力を与える。デマルチプレクサ224の出力は、SLEEPCSを経てORゲート226および228の入力になる。ORゲート226および228の他の入力は、それぞれ入出力書き込み制御線と入出力読み出し制御線である。これらのゲートの出力は、NORゲート

230の出力と共にDフリップフロップ232に入ってポートを復号する。「INTR」は入出力ポート(周辺装置)からNORゲート230に入る割り込み入力で、これにより論理ハードウェアは切り替わって高速クロックに戻る。次に、フリップフロップ232の出力はORゲート226からの出力と共に3状態バッファ234に入る。これにより、バッファ234はポート上にあるものを読み戻すことができる。読み出し/書き込み入出力ポート(周辺装置)は、上に述べたすべてのハードウェアを用いて電力節約「休眠」動作を選択する。出力「SLOW-」は図2から図4の「SLEEP」と同じもので、後で説明するフリップフロップ236に inputs する。

【0078】

休眠クロック発振器238の出力は、Dフリップフロップ240および242により2つの遅いクロックに分割される。図8に示す特定の実現では、16MHzの休眠クロック発振器238は4MHzと8MHzクロックに分割される。ジャンパーJ1により、どちらのクロックを「休眠クロック」にするかを選択する。

【0079】

この特定の実現では、高速クロック発振器244は32MHz発振器である。ただしこの特定の速度はこの発明では必要条件ではない。32MHzの発振器は抵抗器(この実現では33オーム)に直列であり、抵抗器は2個のコンデンサ(10pF)に直列である。この発振の出力をDフリップフロップ246および248のクロックに接続する。

【0080】

Dフリップフロップ236、246、248は同期したフリップフロップである。Dフリップフロップ236および246は図2から図4の活動的な電力モニタ機能20の単純化した休眠ハードウェアには示さなかったが、これらのフリップフロップにより、クロック切り替えはクロックの端だけで起こる。図8に示すように、図7のフリップフロップ200と同様に、CPUが休眠する(「FASTEN-」)か目覚めるか(「SLOWEN-」)かに従って、フリップフロップ248の出力はORゲート250かORゲート252を起動する。

【0081】

ORゲート250および252とANDゲート254は、図7のAND/ORセレクタと機能的に同じものである。これらにより、「SLOWCLK」(遅いクロック、また休眠クロックとも言う)が高速クロック(入力線に32MHzと指定されている)かを選択する。この実現では、ジャンパーJ1に従って遅いクロックは4MHzか8MHzであり、高速クロックは32MHzである。ANDゲート254の出力(図のATUCLK信号)はCPUクロックの速度を設定する。これは、図7のCPUクロックと同じものである。

【0082】

次に図9は、インテル80286などのシステムの、別の実際の休眠ハードウェア実現の略図を示す。ちなみに、インテル80286はクロックを止めることができる。速度の切り替えにはウエスタン・デジタルFE3600VLSIを特殊な外部装置PAL256と共に用い、割り込みが来るとCPUを目覚めさせる割り込みゲートを制御する。この発明のソフトウェア電力節約態様により、割り込み受け付けが起こるのを監視する。これは割り込み後の次の $P(i) \quad T_i$ 間隔を起動する。

【0083】

CPUへの割り込み要求があると、システムは通常の動作に戻る。CPUへの割り込み要求INTRQにより、PALはRESCPU線にWAKE_UP信号を出してFE3001に送り、FE3001はCPUとDMAクロックを使用可能にして、システムを通常の状態に戻す。これは、図7の「INTERRUPT-」と同じものである。状態機械の混乱を避けるために割り込み要求は同期をとっているため、割り込み信号INTDETはサイクルが活動化しているときだけ検出される。RESCPU信号の立ち上がり端によりFE3001は目を覚まし、システム全体を休眠モードから解放する。

【0084】

10

20

30

40

50

386SXで実現する場合は、この実施態様のCPUの動的動作特性機能を実行する外部のハードウェアとソフトウェアループが異なるだけである。割り込みが起こると、ソフトウェアループは割り込みを実行する前に外部ハードウェアを高速クロックに切り替える。電力節約ソフトウェアに戻ると、活動的なモニタ機能は高速クロックサイクルを検出して、高速クロック動作のハードウェアをリセットする。

【0085】

OS/2で実現する場合は、優先度の低い背景動作で走るスレッド(THREAD)としてプログラムされた「何もしない」ループを用いる。スレッドを起動すると、割り込みが発生してCPUを元のクロック速度に戻すまでは、CPUは休眠すなわち低速度クロック動作になる。

10

【0086】

この発明の好ましい実施態様では割り込みを用いてCPUを目覚めさせたが、システム内のまたはシステムに与えられる任意の周期的活動を用いても同じ機能を実現することができる。

【0087】

この発明の電力節約態様と温度管理態様との違いは、温度管理態様ではCPUの温度を検出する必要があることである。これはサーミスタまたはその他の直接検出機構を用いて行い、温度の測定値を温度管理サブルーチン132に与える。サーミスタまたはその他の温度検出装置はこの技術でよく知られている。さらに、検出された動作特性は、特定の応用で関心のあるCPUの動的動作特性に従って、他の直接読み取り装置すなわちセンサにより得ることができる。

20

【0088】

要約すると、この発明は検出されたCPUの動的動作特性を制御する方法を与えるもので、少なくとも1つのCPUの動的動作特性を検出するステップを含む。この発明は、少なくとも1つの動的動作特性を検出したときのCPUのクロック速度を検出する。この方法と装置は、少なくとも1つのCPUの動的動作特性が前記少なくとも1つのCPUの動的動作特性に関連する所定の設定点に対して所定の関係を確認することにより、設定点割り込み条件が存在することを決定する。割り込み条件が存在する場合は、この発明は、検出したクロック速度に対してクロック速度を制御するステップを含む。他方、設定点割り込み条件が存在しない場合は、この発明は、上記の決定および制御ステップを繰り返すためのステップと回路を含む。

30

【0089】

この発明の別の優れた特徴は、電力消費や温度などの検出されたCPUの動的動作特性に従って単一のCPUを制御するだけでなく、所定のコンピュータシステムを支援する多数のCPUを制御することである。たとえば二重CPUコンピュータシステムでは、この発明は検出されたCPUの動的動作特性の制御を調整して、コンピュータシステムの性能を全体的に所望のレベルに維持する。たとえば、両方のCPUに与えられた計算指向サブルーチン142の決定に従って、一方のCPUは計算指向モードにあり、二重CPUコンピュータの他方のCPUは計算指向ではなく、多くの入出力機能を含む命令を実行しているとする。計算指向CPUの外箱温度の上昇係数によって現在のクロック速度ではCPU温度が上昇することが分かると、この発明は計算指向CPUのクロックを減速させて、温度の設定点に達しないようにする。コンピュータシステムの性能を一定にする必要があるかどうかに従って、この発明は他のCPUのクロック速度を上げて、平均のまたは全体のクロック速度を一定に保ち、コンピュータシステムの性能レベルを均一にする。

40

【0090】

この発明の多重CPU実施態様の一態様を例示する図10は、多重CPU開始ステップ282で始まる多重CPU流れ図280を示す。多重CPU開始ステップ282は、まず判断284で1個以上のCPUを制御する必要があるかどうか調べる。1個以上のCPUを制御する場合は、処理の流れは判断286に進む。そうでない場合は、処理の流れは復帰ベクトル288に進んで、CPUは多重CPUサブルーチン282を実行する前に行っ

50

ていた動作に戻る。

【0091】

判断286で、多重CPUサブルーチン282は多重CPUを調整して管理するかどうか調べる。調整しない場合は処理の流れは復帰ベクトル288に進む。調整する場合は処理の流れはステップ290に進む。ステップ290は、多重CPUのそれぞれの制御を導くベクトルの役目をする。検出されたCPUの動的動作特性により制御されるCPU毎に、処理の流れは活動的なモニタサブルーチン292に進む。活動的なモニタサブルーチン292は本質的に、活動的な電力モニタ機能20や、温度管理サブルーチン132のモニタ機能などの活動的なモニタ機能や、電力節約と温度管理を調整するモニタ機能や、1つ以上の検出されたCPUの動的動作特性に応じてCPUを制御する他のモニタ機能またはサブルーチンと同じである。

10

【0092】

多重CPUサブルーチン282の処理の流れがCPU毎に進むと、判断294で、少なくとも1つの調整されたCPUに設定点割り込み条件があるかどうか調べる。このような設定点関連の割り込み条件がない場合は処理の流れは判断294に戻って、設定点関連の割り込み条件があるかどうかの調査を続ける。このような割り込み条件があると処理の流れは判断296に進んで、影響を受けないCPUのクロック速度を多重CPUサブルーチン282が制御または調整する希望または予定があるか調べる。影響を受けないCPUを調整しない場合は、判断296から処理の流れは復帰ベクトル288に進む。調整する場合は、処理の流れはステップ298に進み、影響を受けないCPUのクロック速度を調整して、コンピュータシステム全体のクロック速度を一定に保つかまたは所望の値を得る。

20

【0093】

この発明の好ましい実施態様のいくつかの実現を示して説明したが、当業者は種々の修正や別の実施態様を考えることができる。したがって、この発明は特許請求の範囲によってだけ限定されるものである。

【0094】

以上の説明に関してさらに以下の項を開示する。

(1) 検出されたCPUの動的動作特性を制御する方法であって、

(a) コンピュータ内の第1クロック速度で動作する、コンピュータ内の中央処理装置(CPU)の少なくとも1つの動的動作特性を検出し、

30

(b) 前記少なくとも1つの動的動作特性が前記少なくとも1つの動的動作特性に関連する所定の設定点に対して所定の関係を確立することにより、設定点割り込み条件が存在することを決定し、

(c) 前記設定点割り込み条件が存在する場合は、前記クロック速度を前記第1クロック速度に対して制御し、

(d) 前記設定点割り込み条件が存在しない場合は、ステップ(b)と(c)を繰り返す、

ステップを含む、検出されたCPUの動的動作特性を制御する方法。

【0095】

(2) 検出されたCPUの動的動作特性を制御する前記方法はCPUの動的動作特性を管理する方法を含み、さらに、前記少なくとも1つの動的動作特性を検出する前記ステップは前記CPUの動作温度を検出するステップを含み、またさらに前記決定するステップは割り込み条件の存在を決定するステップを含み、前記動作温度は所定の温度設定点に対して所定の関係を確立する、第1項記載の検出されたCPUの動的動作特性を制御する方法。

40

【0096】

(3) 前記少なくとも1つの動的動作特性を検出する前記ステップは前記CPUの動作温度を検出するステップを含み、またさらに前記決定するステップは前記CPUが計算指向状態にある期間中に前記動作温度が所定の温度設定点に近いことを決定するステップを含む、第1項記載の検出されたCPUの動的動作特性を制御する方法。

50

(4) 前記CPUが計算指向状態にあるかどうかを決定し、前記計算指向状態中に前記割り込み条件が存在するかどうかをさらに決定する、ステップをさらに含む、第1項記載の検出されたCPUの動的動作特性を制御する方法。

【0097】

(5) (a)前記CPUが計算指向状態にあるかどうかを決定し、(b)前記計算指向状態中に前記少なくとも1つの動的動作特性に前記割り込み条件が存在するかどうかを決定し、(c)前記計算指向状態中に前記割り込み条件が存在する場合は前記割り込み条件を修正する、ステップをさらに含む、第1項記載の検出されたCPUの動的動作特性を制御する方法。

(6) 前記CPUの電力節約が可能かどうかを決定し、前記CPUの電力節約が可能な場合は、前記電力節約から生じる前記割り込み条件および制御信号の存在に従って前記クロック速度を前記保存されたクロック速度に対して制御する、ステップをさらに含む、第1項記載の検出されたCPUの動的動作特性を制御する方法。

【0098】

(7) 前記CPUが計算指向状態にあるかどうかを決定し、前記計算指向状態中に前記CPUが実行する予定の計算指向命令の量を決定することにより前記計算指向状態の継続時間を決定する、ステップをさらに含む、第1項記載の検出されたCPUの動的動作特性を制御する方法。

(8) 前記CPUが計算指向状態にあるかどうかを決定し、前記計算指向状態中に前記CPUが実行する予定の計算指向命令の量を決定することにより前記計算指向状態の継続時間を決定し、さらに前記計算指向状態中に前記割り込み条件が存在する場合は前記割り込み条件を修正する、ステップをさらに含む、第1項記載の検出されたCPUの動的動作特性を制御する方法。

【0099】

(9) 前記CPUが計算指向状態にあるかどうかを決定し、前記計算指向状態の継続時間を決定し、さらに前記クロック速度を前記保存されたクロック速度に対して修正して前記計算指向状態中に前記割り込み条件が存在しないようにする、ステップをさらに含む、第1項記載の検出されたCPUの動的動作特性を制御する方法。

(10) 前記CPUの前記少なくとも1つのCPUの動的動作特性を検出する前記ステップは、サーミスタ検出装置を用いて前記CPUの動作温度を検出するステップを含む、第1項記載の検出されたCPUの動的動作特性を制御する方法。

【0100】

(11) 前記所定の関係と前記所定の設定点を前記コンピュータのメモリ位置内に記憶するステップをさらに含む、第1項記載の検出されたCPUの動的動作特性を制御する方法。

(12) 前記所定の関係と前記所定の設定点を前記コンピュータのレジスタを備えるメモリ位置内に記憶するステップをさらに含む、第1項記載の検出されたCPUの動的動作特性を制御する方法。

(13) 前記所定の関係と前記所定の設定点を前記コンピュータに関連し前記CPU以外の回路を備えるメモリ位置内に記憶するステップをさらに含む、第1項記載の検出されたCPUの動的動作特性を制御する方法。

【0101】

(14) 第2項記載の方法であって、実時間電力節約を調整し前記CPU内の検出された動的動作特性を制御する方法をさらに含み、前記実時間電力節約法は前記検出されたCPU動的動作特性の前記制御法に関連して動作し、

(e) 前記CPUの電力節約が可能かどうかを決定し、

(f) 前記CPUの電力節約が可能な場合は、前記CPUの現在のクロック速度を決定し、

(g) 前記CPUに与えられている現在のクロック速度を減らすか止めるようハードウェアセレクタに示し、

(h) 電力節約割り込みが起こったかどうか決定し、

(i) 前記電力節約割り込みが起こらなかった場合はこのステップ(i)を繰り返し、前記電力節約割り込みが起こるまで前記CPUは前記電力節約モードに留まり、

(j) 前記電力節約割り込みが起こった場合は、前記決定された現在のクロック速度を前記CPUに回復するよう前記ハードウェアセクタに示す、

ステップを含む、実時間電力節約を調整し前記CPU内の検出された動的動作特性を制御する方法。

【0102】

(15) 前記CPUに与えられている現在のクロック速度を減らすか止めるようハードウェアセクタに示す前記ステップ(g)は、

10

前記ハードウェアセクタへの通信線を通して前記ハードウェアセクタに電力節約のCPU命令をパルスで与え、

前記電力節約のCPU命令に基づいて前記ハードウェアセクタにより電力節約クロックを選択し、

前記電力節約クロックから前記CPUにパルスを送って前記CPUを電力節約モードにする、

ステップをさらに含む、第14項記載の実時間電力節約を調整し前記CPU内の検出された動的動作特性を制御する方法。

【0103】

(16) コンピュータ内の中央処理装置(CPU)の電力節約が可能かどうかを決定する前記ステップ(e)は、

20

前記CPUがすでに前記電力節約モードにあるかどうか調べ、

前記CPUがまだ前記電力節約モードにない場合は、前記CPUが前記電力節約モードになる前に前記CPUを目覚めさせる電力節約割り込みが使用可能かどうか決定し、

電力節約割り込みが使用可能な場合は、前記CPUが前記電力節約モードに入ることを妨げる、

ステップをさらに含む、第14項記載の実時間電力節約を調整し前記CPU内の検出された動的動作特性を制御する方法。

【0104】

(17) コンピュータ内の中央処理装置(CPU)の電力節約が可能かどうかを決定する前記ステップ(e)は、

30

前記CPUが要求を受けたかどうかを調べ、

前記CPUが要求を受けた場合は、前記要求は限界入出力かどうかを決定し、

前記要求が限界入出力の場合は前記CPUが前記電力節約モードに入らないようにし、

前記要求が限界入出力でない場合は、前記CPUが前記電力節約モードに入る前に前記CPUを目覚めさせる電力節約割り込みが使用可能かどうか決定し、

前記CPUが前記電力節約モードに入らないようにし、

前記要求は前記入出力装置が作動可能になるまで遅れを持つ入出力からかどうかを決定し、

前記要求が遅れを持つ入出力からでない場合は前記CPUが前記電力節約モードに入らないようにする、

40

ステップをさらに含む、第14項記載の実時間電力節約を調整し前記CPU内の検出された動的動作特性を制御する方法。

【0105】

(18) 複数のCPUの検出された動的動作特性を制御する方法であって、

(a) 計算システムの複数の中央処理装置(CPU)の少なくとも1つの動的動作特性を検出し、

(b) 各前記複数のCPUの第1クロック速度を検出し、

(c) 各前記複数のCPU毎に少なくとも1つの設定点割り込み条件が存在することを決定し、1つ以上の前記少なくとも1つの検出された動的動作特性は前記少なくとも1

50

つの動的動作特性の対応する1つに関連する所定の設定点に対する所定の関係確立し、
(d) 少なくとも1つの割り込み条件が存在する場合は、前記少なくとも1つの設定点割り込み条件に対応する前記複数のCPUの1つの前記第1クロック速度に対して少なくとも前記クロック速度を制御し、

(e) 前記複数のCPUのどれにも割り込み条件が存在しない場合はステップ(b)と(c)を繰り返す、
ステップを含む、複数のCPUの検出された動的動作特性を制御する方法。

【0106】

(19) 前記割り込み条件に対応する前記CPU以外の少なくとも1つの前記複数のCPUを制御して、前記計算システムが前記複数のCPUの間でほぼ一定の平均クロック速度を保持するようにするステップをさらに含む、第18項記載の複数のCPUの検出された動的動作特性を制御する方法。

【0107】

(20) 検出されたCPUの動的動作特性を制御する装置であって、
CPU活動検出器と、
CPUクロックと、

少なくとも第1パルスで、また第2パルスを第2速度で与える少なくとも1個の発振器と、

前記第1パルスと前記第2パルスを選択的に制御し選択するハードウェアセレクタであって、前記ハードウェアセレクタはさらに前記第1パルスと前記第2パルスの前記選択された方を前記CPUクロックに送り、また少なくとも1つの検出されたCPUの動的動作特性が前記検出されたCPUの動的動作特性に関連する所定の設定点に対して所定の関係確立することにより存在する、設定点割り込み条件に応じる、ハードウェアセレクタと

、
検出されたCPUの活動を前記CPU活動検出器から受け、さらに前記ハードウェアセレクタに送るパルスを発生して前記ハードウェアセレクタがどのパルスを選択すべきかを指定する、CPU休眠管理装置と、
を備える、検出されたCPUの動的動作特性を制御する装置。

【0108】

(21) 検出されたCPUの動的動作特性を制御する装置であって、

(a) コンピュータ内の第1クロック速度で動作する中央処理装置(CPU)の少なくとも1つの動的動作特性を検出する検出回路と、

(b) 前記少なくとも1つの動的動作特性が前記少なくとも1つの動的動作特性に関連する所定の設定点に対して所定の関係確立することにより、設定点割り込み条件が存在することを決定する、前記CPU上で動作する設定点命令と、

(c) 前記設定点割り込み条件が存在する場合は、前記クロック速度を前記第1クロック速度に対して制御する制御命令と、

(d) 前記設定点割り込み条件が存在しない場合は、ステップ(b)と(c)を繰り返す繰り返し命令と、

を備える、検出されたCPUの動的動作特性を制御する装置。

【0109】

(22) CPUの動的動作特性を管理する管理命令をさらに含み、また前記温度検出回路はさらに、前記CPUの動作温度を含む前記少なくとも1つの動的動作特性を検出する動的検出回路を備え、またさらに前記設定点命令は割り込み条件の存在を決定する命令をさらに含み、前記動作温度は所定の温度設定点に対して所定の関係確立する、第21項記載の検出されたCPUの動的動作特性を制御する装置。

【0110】

(23) 前記検出回路は前記CPUの動作温度を検出する回路をさらに備え、またさらに前記設定点命令は前記CPUが計算指向状態にある期間中に前記動作温度が所定の温度設定点に近いことを決定する命令をさらに含む、第21項記載の検出されたCPUの動的

10

20

30

40

50

動作特性を制御する装置。

(24) 前記CPUが計算指向状態にあるかどうかを決定し、前記計算指向状態中に前記割り込み条件が存在するかどうかをさらに決定する、計算指向決定命令をさらに含む、第21項記載の検出されたCPUの動的動作特性を制御する装置。

【0111】

(25) 前記CPUが計算指向状態にあるかどうかを決定する計算指向決定命令をさらに含み、前記設定点命令は前記計算指向状態中に前記少なくとも1つの動的動作特性に前記割り込み条件が存在するかどうかを決定する命令をさらに含み、前記計算指向状態中に前記割り込み条件が存在する場合は前記割り込み条件を修正する修正命令をさらに含む、第21項記載の検出されたCPUの動的動作特性を制御する装置。

10

(26) 前記CPUの電力節約が可能かどうかを決定し、前記CPUの電力節約が可能な場合は、前記電力節約から生じる前記割り込み条件の存在と制御信号に従って前記クロック速度を前記保存されたクロック速度に対して制御する、電力節約命令をさらに含む、第21項記載の検出されたCPUの動的動作特性を制御する装置。

【0112】

(27) 前記CPUが計算指向状態にあるかどうかを決定し、前記計算指向状態中に前記CPUが実行する予定の計算指向命令の量を決定することにより前記計算指向状態の継続時間を決定する、計算指向決定命令をさらに含む、第21項記載の検出されたCPUの動的動作特性を制御する装置。

(28) 前記CPUが計算指向状態にあるかどうかを決定し、前記計算指向状態中に前記CPUが実行する予定の計算指向命令の量を決定することにより前記計算指向状態の継続時間を決定し、さらに前記計算指向状態中に前記割り込み条件が存在する場合は前記割り込み条件を修正する、計算指向決定命令をさらに含む、第21項記載の検出されたCPUの動的動作特性を制御する装置。

20

【0113】

(29) 前記CPUが計算指向状態にあるかどうかを決定し、前記計算指向状態の継続時間を決定し、さらに前記クロック速度を前記保存されたクロック速度に対して修正して前記計算指向状態中に前記割り込み条件が存在しないようにする、計算指向決定命令をさらに含む、第21項記載の検出されたCPUの動的動作特性を制御する装置。

(30) 前記検出回路はサーミスタ検出装置を備える、第21項記載の検出されたCPUの動的動作特性を制御する装置。

30

(31) 前記所定の関係と前記所定の設定点を前記コンピュータのメモリ位置内に記憶するメモリ回路をさらに備える、第21項記載の検出されたCPUの動的動作特性を制御する装置。

【0114】

(32) 検出されたCPUの動的動作特性を制御する方法と装置(130)であって、CPUが第1クロック速度で動作中に(134)少なくとも1つの動的CPU動作特性を検出する(140)方法と回路を含む。装置(130)は、少なくとも1つの検出されたCPUの動的動作特性が前記少なくとも1つの動的動作特性に関連する所定の設定点に対して所定の関係を確立する(140)ことにより、設定点割り込み条件が存在するかどうかを決定する(140)。設定点割り込み条件が存在する場合は、回路と命令は第1クロック速度に対してクロック速度を制御する(144)。設定点割り込み条件が存在しない場合は、回路と命令は割り込み条件を決定してクロック速度を制御する上記ステップを繰り返す。またこの方法と装置(130)はCPUが計算指向状態にあるかどうかを決定する(142)。この動作は、実時間の電力節約装置および方法(20)と共に、この発明の特に優れた機能である。

40

【図面の簡単な説明】

【0115】

【図1】この発明の好ましい実施態様の自己同調状態を述べる流れ図。

【図2】この発明の実施態様で用いる活動的な電力節約モニタを説明する流れ図。

50

【図3】この発明の実施態様で用いる活動的な電力節約モニタを説明する流れ図。

【図4】この発明の実施態様で用いる活動的な電力節約モニタを説明する流れ図。

【図5】この発明が用いる温度管理法の簡単化した流れ図。

【図6】この発明が用いる計算指向決定ステップの流れ図。

【図7】この発明の実施態様が用いる、活動的な電力節約に関連するハードウェアを表す簡単化した略図。

【図8】この発明の一実施態様の休眠ハードウェアの略図。

【図9】この発明の別の実施態様の休眠ハードウェアの略図。

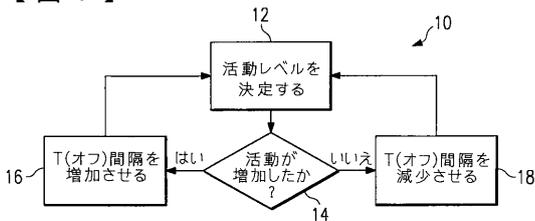
【図10】この発明の別の実施態様の多重CPUの動的動作特性の制御機能の流れ図。

【符号の説明】

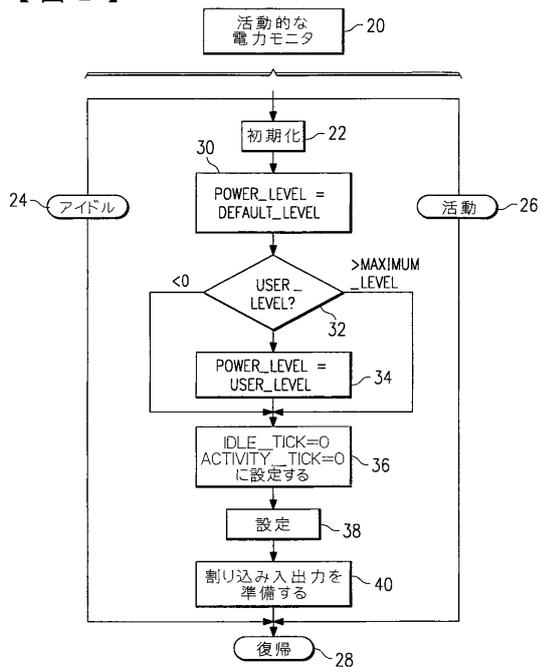
【0116】

- 130 CPUの動的動作特性の制御装置
- 134 現在のクロック速度の決定
- 140 設定点割り込み条件の存在の決定
- 142 CPUが計算指向状態にあることの決定
- 144 休眠クロック速度の制御

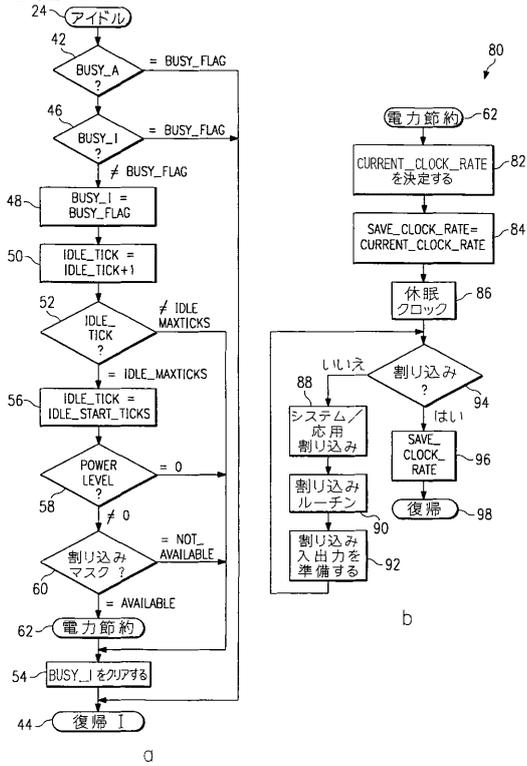
【図1】



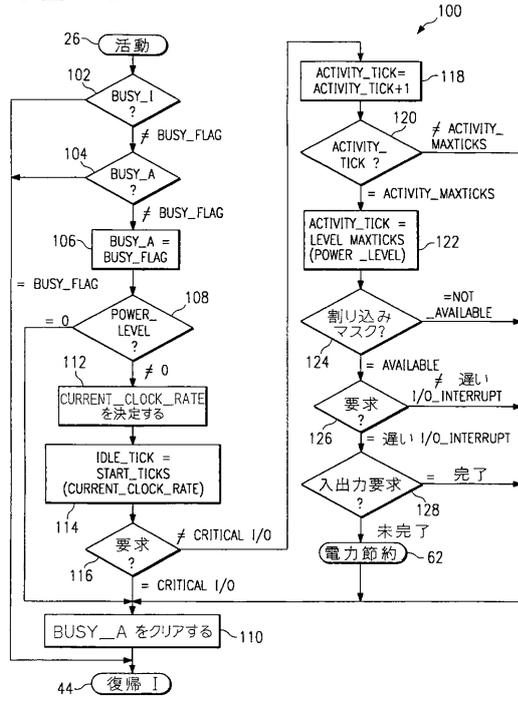
【図2】



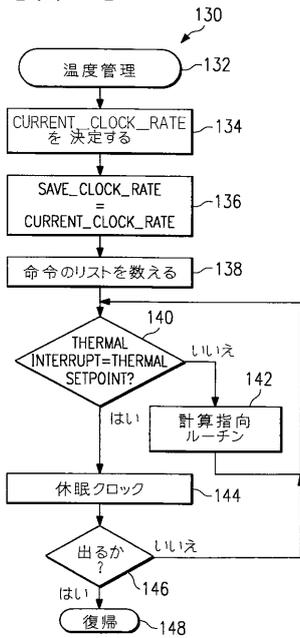
【図3】



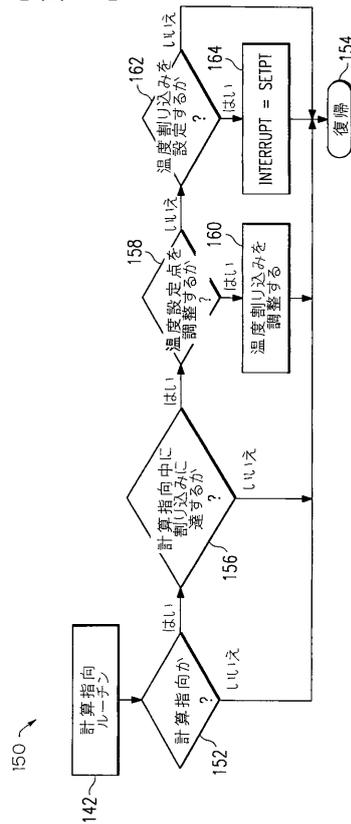
【図4】



【図5】



【図6】



フロントページの続き

(72)発明者 ラボーフン エフ.ワッツ

アメリカ合衆国テキサス州テンプル、サマ - ウッド ドライブ 5302

Fターム(参考) 5B062 AA05 HH02 HH06

5B079 AA02 BA01 BB01 BC01