

(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2003-140787

(P2003-140787A)

(43) 公開日 平成15年5月16日 (2003.5.16)

(51) Int.Cl. <sup>7</sup>	識別記号	F I	キーワード* (参考)
G 0 6 F 1/32		G 0 6 F 9/46	3 4 0 B 5 B 0 1 1
9/46	3 4 0	1/00	3 3 2 Z 5 B 0 9 8

審査請求 未請求 請求項の数 7 O L (全 9 頁)

(21) 出願番号 特願2001-340706(P2001-340706)

(22) 出願日 平成13年11月6日(2001.11.6)

特許法第30条第1項適用申請有り 2001年5月18日 社団法人電子情報通信学会発行の「電子情報通信学会技術研究報告 信学技報V○1, 101N○, 85」に発表

(71) 出願人 000005108  
株式会社日立製作所  
東京都千代田区神田駿河台四丁目6番地

(71) 出願人 391012327  
東京大学長  
東京都文京区本郷7丁目3番1号

(72) 発明者 桜井 貴康  
東京都世田谷区宮坂2-21-15

(74) 代理人 100072051  
弁理士 杉村 興作 (外1名)

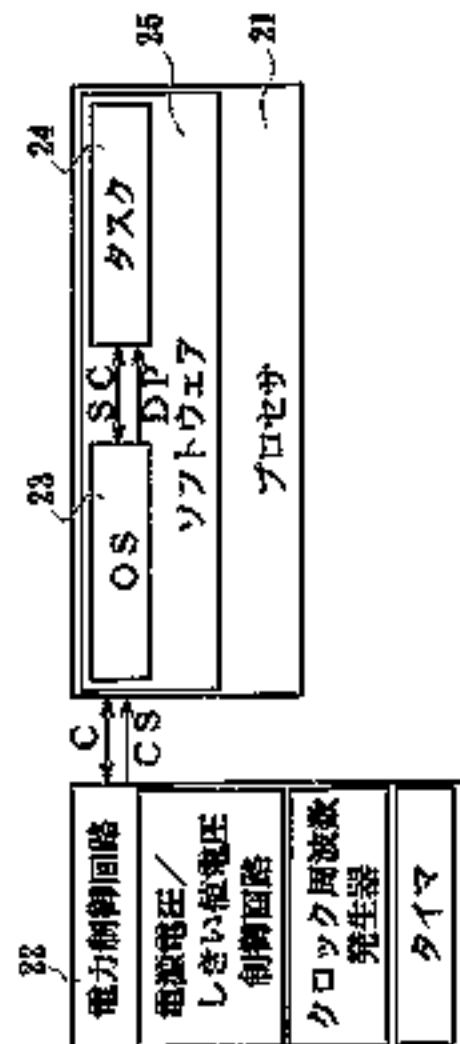
最終頁に続く

(54) 【発明の名称】 電力制御装置及び方法並びに電力制御プログラム

(57) 【要約】

【課題】 複数のタスクが実行される環境でプロセサの消費電力の最適化を行う。

【解決手段】 OS 23は、プロセサ 21上で実行されるタスク 24のうちの実行中のタスク及び実行可能なタスクを識別し、全タスクのうちの最初に起動されるものの起動時刻を実行中のタスクに提供する。プロセサ 21の消費電力の最適化は、起動時刻及び実行中のタスクに関する最悪実行時間に基づいて行われる。



## 【特許請求の範囲】

【請求項1】 プロセッサ上で実行される複数のタスクのうちの実行中のタスク及び実行可能なタスクを識別する手段と、

前記複数のタスクのうち最初に起動されるものの起動時刻を前記実行中のタスクに提供する手段と、

前記起動時刻及び前記実行中のタスクに関する最悪実行時間に基づいて前記プロセッサの消費電力の最適化を行う手段とを具備することを特徴とする電力制御装置。

【請求項2】 プロセッサ上で実行される複数のタスクのうちの実行中のタスク及び実行可能なタスクを識別するステップと、

前記複数のタスクのうち最初に起動されるものの起動時刻を前記実行中のタスクに提供するステップと、

前記起動時刻及び前記実行中のタスクに関する最悪実行時間に基づいて前記プロセッサの消費電力の最適化を行うステップとを具備することを特徴とする電力制御方法。

【請求項3】 プロセッサ上で実行される複数のタスクのうちの実行中のタスク及び実行可能なタスクを識別する機能と、

前記複数のタスクのうち最初に起動されるものの起動時刻を前記実行中のタスクに提供する機能と、

前記起動時刻及び前記実行中のタスクに関する最悪実行時間に基づいて前記プロセッサの消費電力の最適化を行う機能とをコンピュータによって実現させることを特徴とする電力制御プログラム。

【請求項4】 前記実行中のタスク及び実行可能なタスクを識別する機能が、前記複数のタスクの次の起動時刻を順番に保持する機能を有することを特徴とする請求項3記載の電力制御プログラム。

【請求項5】 前記実行中のタスク及び実行可能なタスクを識別する機能が、前記実行中のタスク及び実行可能なタスクを優先度の高い順に保持し、前記実行中のタスクのみが保持されているか否かの情報を前記実行中のタスクに提供する機能を有することを特徴とする請求項3又は4記載の電力制御プログラム。

【請求項6】 前記実行中のタスクを前記プロセッサに割り付けた時刻から現在の時刻までの積算実行時間に関する情報を、前記プロセッサに提供する機能をコンピュータによって実現させることを特徴とする請求項3から5のうちいずれか1項に記載の電力制御プログラム。

【請求項7】 前記積算実行時間及び前記起動時刻から前記実行中のタスクの時間余裕を算出する機能をコンピュータによって実現させることを特徴とする請求項6記載の電力制御プログラム。

## 【発明の詳細な説明】

## 【0001】

【発明の属する技術分野】本発明は、例えばCMOS半導体集積回路によって構成したプロセッサの電力制御装置及び方法並びに電力制御プログラムに関するものであ

る。

## 【0002】

【従来の技術】近年、バッテリーによって駆動されるプロセッサ内蔵の携帯機器、例えば、携帯電話、携帯型オーディオ、携帯型ビデオ、ノートブックパーソナルコンピュータ、PDA(Personal Digital Assistants)の消費電力の最適化、例えば消費電力の低減の要求が、従来に比べて一層厳格になっている。消費電力の最適化に対する厳格な要求を満足させるために、プロセッサに供給すべき可変の電源電圧を制御することによって消費電力の最適化を図ることが特許第3138737号に記載されており、CMOS半導体集積回路のNMOS素子及び/又はPMOS素子のしきい値電圧を制御することによって消費電力の最適化を図ることが特願2000-221676号に記載されている。

【0003】これら特許第3138737号及び特願2000-221676号によれば、図1Aに示すように、電力制御装置は、プロセッサ1と、プロセッサ1の可変電源電圧及び/又はしきい値電圧を制御することによって消費電力の最適化を行う電力制御回路2とを具備する。

【0004】プロセッサ1には、タスク3及びデバイスドライバ4を有するソフトウェア5がインストールされ、図1Aでは示さないI/Oポートを通じて簡単なコマンドを電力制御回路2との間でやりとりし、プロセッサ1の内部クロック周波数と、可変電源電圧及び/又はしきい値電圧V<sub>var</sub>とを制御する制御信号CSを出力する。なお、図1Aでは、電力制御回路2をプロセッサ1と個別に構成した場合について示しているが、電力制御回路2をプロセッサ1と一体に構成してもよい。

【0005】このために、電力制御回路2は、電源電圧/しきい値電圧制御回路6と、クロック周波数発生器7と、タイマ8とを有する。電源電圧/しきい値電圧制御回路6は、可変電源電圧及び/又はしきい値電圧V<sub>var</sub>を生成し、その可変電源電圧及び/又はしきい値電圧V<sub>var</sub>をプロセッサ1に供給する。

【0006】クロック周波数発生器7は、外部クロックf<sub>clk</sub>のk/n倍(nを自然数とし、kをn未満の自然数とする。)となる可変周波数f<sub>var</sub>を生成し、その可変周波数f<sub>var</sub>をプロセッサ1に供給する。可変周波数f<sub>var</sub>をそのように設定することによって、外部デバイスとのやりとりが容易となる。タイマ8は、後に説明するようリアルタイム処理を行う際に使用される。

【0007】可変電源電圧及び/又はしきい値電圧V<sub>var</sub>は、後に説明するよう可変周波数f<sub>var</sub>に応じて設定される。

【0008】タスク3は、図1Bに示すようにユーザプログラム9及び電力制御アルゴリズム10を有する。ユーザプログラム9は、設計者によって任意にプログラムされ、電力制御アルゴリズム10は、2以上の動作速度及びアイドル状態に対応する3以上のモードに従っ

て可変電源電圧及び／又はしきい値電圧 $V_{var}$ と可変周波数 $f_{var}$ とを動的に制御するよう電源電圧／しきい値電圧制御回路6にコマンドCを送信する。なお、電源制御アルゴリズム10の構成及びコマンドCの送信については、後に説明する。

【0009】デバイスドライバ4は、図1Cに示すように可変電源電圧及び／又はしきい値電圧 $V_{var}$ を最適化するための参照可能な関係を有するルックアップテーブル11を有する。

【0010】図2は、電力制御アルゴリズムを詳細に示す図である。この電力制御アルゴリズム10は、締切すなわち今回のタスク3の実行を終了すべき時刻を取得するステップS1と、可変クロック周波数 $f_{var}$ と可変電源電圧及び／又はしきい値電圧 $V_{var}$ とを算出するステップS2と、可変クロック周波数 $f_{var}$ と可変電源電圧及び／又はしきい値電圧 $V_{var}$ とを適用するステップS3と、ユーザプログラム9を実行するステップS4と、プロセッサ1のアイドルリングを行うステップS5とを具える。本明細書中、締切とは、特に言及しない限り、タスク又はスライスの実行を終了すべき時刻を意味するものとする。

【0011】タスク3を画像や音声のようなマルチメディアタスクとした場合、タスク3は、特定の周波数（例えば、60Hz又は44.1kHz）に同期をとっており、一定の起動周期とWCET（Worst Case Execution Time:最悪実行時間）とを有する。

【0012】MPEGに代表されるようなアプリケーションでは、入力画像データによって処理速度が変化することが知られている。具体的には、被写体の動きが顕著である場合や、背景の変化が大きくなるに従って負荷が重くなり、それに対して、画像の変化が小さくなるに従って負荷が軽くなる。しかしながら、画像の変化が大きくなるに従って負荷が無限に大きくなるわけではなく、理論上の最大負荷が決まっている。このような理論上の最大負荷は、タスク3のWCETに対応する。

【0013】実際のタスクでは、最大負荷のデータを処理することはほとんどなく、大抵の場合には、WCET以前にタスク3を終了し、時間余裕が発生する。また、タスク3の起動周期は一般にWCETより長いので、最大負荷のデータを処理しても時間余裕が発生する場合もある。

【0014】このような時間余裕が存在することをタスク3の実行中に検知して、プロセッサ1の可変クロック周波数 $f_{var}$ と可変電源電圧及び／又はしきい値電圧 $V_{var}$ とを制御することによって、消費電力を最適化している。

【0015】したがって、負荷に応じてプロセッサ1の可変クロック周波数 $f_{var}$ と可変電源電圧及び／又はしきい値電圧 $V_{var}$ とを柔軟に変化させるアルゴリズムが重要となる。負荷のデータ依存性が高いので、制御は、ラ

ンタイムで動的に行われるべきであり、コンパイルタイムで静的に行われるべきではない。

【0016】しかしながら、タスク終了後に負荷がWCETより軽かったことが明らかになったとしても、もはや何の処理も施すことができない。また、将来の負荷を予測するのは非常に困難である。そのような事態を解決するために、電力制御アルゴリズム10のステップS1において、アプリケーションスライシングを導入する。

【0017】ここで、アプリケーションスライシングを図3を用いて説明すると、タスク3は、N個のスライス1, 2, 3, . . . , Nに分割される。スライス1, 2, 3, . . . , Nの各々のWCETと、各スライスからスライスNまでのWCETは、解析や測定を通じて取得される。

【0018】ここで、スライス2について考える。スライス2のWCET、スライス3からスライスNまでのWCET及びスライス2の開始からタスク3の次の起動時刻までの時間をそれぞれ $t_2$ 、 $t_{3-N}$ 及び $t_D$ とした場合、単一のタスク3を仮定しているため、今回のタスク3の実行を次の起動時刻までに行う必要があり、タスク3の次の起動時刻がタスク3の締切となる。

【0019】スライス3からスライスN（最終スライス）までの処理が時間 $t_{3-N}$ で確実に行われるので、スライス2を実行するための時間余裕 $t_A$ は $t_D - t_{3-N}$ となる。したがって、理想的には、プロセッサ1の可変クロック周波数 $f_{var}$ を外部クロック $f_{clk}$ の $t_2 / t_A$ 倍まで下げることができる。

【0020】実際には、プロセッサ1のクロック周波数を任意の値に設定するのは困難であるので、可変クロック周波数 $f_{var}$ は、好適には、外部クロック $f_{clk}$ の $t_2 / t_A$ 倍を下回らない最小の $1/n$ 倍に設定される。例えば、 $t_2 / t_A$ が0.48の場合、可変クロック周波数 $f_{var}$ は $f_{clk} / 2$ となる。

【0021】プロセッサ1のクロック周波数を変更する際、ルックアップテーブル11を利用する。図4は、図1のルックアップテーブルを詳細に示す図である。ルックアップテーブル11は、可変クロック周波数 $f_{var}$ と可変電源電圧及び／又はしきい値電圧 $V_{var}$ との関係を有する。これらの関係は、プロセッサ1及び電力制御回路2を実際に計測することによって得られる。

【0022】

【発明が解決しようとする課題】しかしながら、特許第3138737号及び特願2000-221676号では、プロセッサの消費電力の最適化を単一の周期タスクについて適用する場合については記載しているものの、複数のタスクが実行される場合については記載及び示唆されていない。

【0023】本発明の目的は、複数のタスクが実行される環境でプロセッサの消費電力の最適化を行う電力制御装置及び方法並びに電力制御プログラムを提供することである。

10

20

30

40

50

【0024】

【課題を解決するための手段】本発明による電力制御装置は、プロセサ上で実行される複数のタスク（全タスク）のうちの実行中のタスク及び実行可能なタスクを識別する手段と、前記複数のタスクのうちの最初に起動されるものの起動時刻を前記実行中のタスクに提供する手段と、前記起動時刻及び前記実行中のタスクに関する最悪実行時間に基づいて前記プロセサの消費電力の最適化を行う手段とを具えることを特徴とする。

【0025】本発明によれば、実行中のタスクに関する最悪実行時間だけでなく、複数のタスクのうちの最初に起動されるものの起動時刻をも参照してプロセサの消費電力の最適化を行っている。したがって、実行中のタスクが他のタスクの起動時刻に関する情報を取得することができるので、複数のタスクが実行される環境でもプロセサの消費電力の最適化を行うことができる。

【0026】本発明による電力制御方法は、プロセサ上で実行される複数のタスクのうちの実行中のタスク及び実行可能なタスクを識別するステップと、前記複数のタスクのうちの最初に起動されるものの起動時刻を前記実行中のタスクに提供するステップと、前記起動時刻及び前記実行中のタスクに関する最悪実行時間に基づいて前記プロセサの消費電力の最適化を行うステップとを具える。

【0027】本発明によれば、複数のタスクが実行される環境でもプロセサの消費電力の最適化を行うことができる。

【0028】本発明による電力制御プログラムは、プロセサ上で実行される複数のタスクのうちの実行中のタスク及び実行可能なタスクを識別する機能と、前記複数のタスクのうちの最初に起動されるものの起動時刻を前記実行中のタスクに提供する機能と、前記起動時刻及び前記実行中のタスクに関する最悪実行時間に基づいて前記プロセサの消費電力の最適化を行う機能をコンピュータによって実現させることを特徴とする。

【0029】本発明によれば、複数のタスクが実行される環境でもプロセサの消費電力の最適化をコンピュータによって実現することができる。

【0030】消費電力の最適化を好適に行うために、前記実行中のタスク及び実行可能なタスクを識別する機能は、例えば、前記複数のタスクの次の起動時刻を順番に保持する機能及び／又は前記実行中のタスク及び実行可能なタスクを優先度の高い順に保持し、前記実行中のタスクのみが保持されているか否かの情報を前記実行中のタスクに提供する機能を有する。

【0031】消費電力の最適化を更に好適に行うために、前記実行中のタスクを前記プロセサに割り付けた時刻から現在の時刻までの積算実行時間に関する情報を、前記プロセサに提供する機能をコンピュータによって実現させる。

【0032】消費電力の最適化を一層好適に行うために、前記積算実行時間及び前記起動時刻から前記実行中のタスクの時間余裕を算出する機能をコンピュータによって実現させる。

【0033】例えば、電力制御プログラムの上記機能のうち、時間余裕を算出する機能については、タスクのようなソフトウェアによって実現され、それ以外の機能については、ソフトウェアに含まれるオペレーティングシステム（OS）によって実現される。

【0034】

【発明の実施の形態】本発明による電力制御装置及び方法並びに電力制御プログラムの実施の形態を、図面を参照して詳細に説明する。図5は、本発明による電力制御装置のブロック図である。電力制御装置は、プロセサ21と、図1の電力制御回路2と同一構成の電力制御回路22とを具える。

【0035】プロセサ21には、OS23及び複数の周期的なタスク（全周期タスク）24を有するソフトウェア25がインストールされ、図5では示さないI/Oポートを通じて簡単なコマンドCを電力制御回路22との間でやりとりし、プロセサ21の内部クロック周波数 $f_{var}$ と、可変電源電圧及び／又はしきい値電圧 $V_{var}$ とを制御する制御信号CSを出力する。なお、図5では、電力制御回路22をプロセサ21と個別に構成した場合について示しているが、電力制御回路22をプロセサ21と一体に構成してもよい。また、図5では、1個のタスク（周期タスク）24のみを示す。

【0036】図6に示すように、OS23は、起動時刻キュー26と、実行優先度キュー27と、デバイスドライバ28と、スケジューラ29と、システムコールハンドラ30とを有する。各タスク24は、ユーザプログラム31と、電力制御アルゴリズム32とを有する。なお、図5及び6において、DPは後に詳細に説明するプロセサ割当て等を表し、SCはシステムコールを表す。

【0037】起動時刻キュー26は、全てのタスク24の次の起動時刻を起動が行われる順番に保持し、タスク24が起動又は終了される度に更新される。ここで、起動時刻キュー26の先頭のタスク（今後最も早く起動される周期タスク）の起動時刻を、第1起動時刻と称する。

【0038】実行優先度キュー27は、実行中のタスク24及び実行可能なタスク24の実行優先度を実行優先度の高い順番に保持する。デバイスドライバ28は、図1のデバイスドライバ4と同一構成を有する。スケジューラ29は、管理するタスク24のスケジューリングを行う。システムコールハンドラ30は、管理されるタスク24との間で情報の受渡しを行う。

【0039】ユーザプログラム31は、図1のユーザプログラム9と同一構成を有し、電力制御アルゴリズム32は、後に詳しく説明するように図1の電力制御アルゴ

10

20

30

40

50



リズム10とは異なる構成を有する。

【0040】各タスク24は、図7に示すように、スケジューラ29によって状態が遷移される。タスク24がRUN状態にあるとき、タスク24は、現在実行中であり、プロセサ21を占有している。

【0041】タスク24がREADY状態にあるとき、タスク24は、実行可能であるが、更に高い優先度を有する他のタスクが実行中であるためにプロセサ割付を待機していることを意味する。READY状態又はRUN状態にあるタスク24の実行優先度は、高い順で実行優先度キュー27に保持される。実行中のタスク24が終了し又はタスク24が起動された場合には、実行優先度キュー27は更新される。スケジューラ29は、実行優先度キュー27の先頭のタスクをプロセサ21に割り付ける。

【0042】タスク24がDORMANT状態にあるとき、タスク24の実行が既に終了し、タスク24が休止中であり、タスク24の次の起動を待機する。

【0043】スケジューラ29は、単位時間ごとに起動時刻キュー26を調べ、休止中のタスク24を起動させるか否かも判断する。所定のタスク24が起動された場合、スケジューラ29は実行優先度キュー27を更新し、現在実行中のタスク24の優先度と、起動直後のタスク24の優先度とを比較する。

【0044】起動直後のタスク24の優先度が実行中のタスク24の優先度より高い場合、実行中のタスク24はプロセサ21の待機を強要され、実行すべきタスク24が入れ替えられる。入れ替えられたタスク24は、READY状態となり、再びプロセサ21への割付けを待機することになる。

【0045】これまで、スケジューラ29によるタスク24の遷移の基本機能を説明したが、本発明によれば、後に詳しく説明するように、上記基本機能に加えて、実行中のタスク24に締切を提供する。この締切は、実行優先度キュー27及び起動時刻キュー26の状態に基づいている。

【0046】なお、実行優先度キュー27に存在するタスク24の個数及び第1起動時刻については、システムコールハンドラ30によって、実行中のタスク24で参照可能である。

【0047】図8は、図6の電力制御アルゴリズムを詳細に示す図である。電力制御アルゴリズム32は、実行優先度キュー27に存在するタスク24の個数及び第1起動時刻に基づくものである。

【0048】電力制御アルゴリズム32では、まず、ステップS11において、後に詳しく説明するように、システムコールハンドラ30を用いて締切を取得する。一旦締切が提供されると、ステップS12において、可変クロック周波数fvarを算出する。次いで、ステップS13において、可変クロック周波数fvarはシステムコ

ールハンドラ30によってOS23に知らされ、OS23は、デバイスドライバ28の有するルックアップテーブルを参照して、可変電源電圧及び/又はしきい値電圧Vvarを制御する。

【0049】その後、図2のステップS4と同様にユーザプログラムを実行した(ステップS14)後、電力制御アルゴリズム32を終了する。

【0050】図9は、図8の締切取得ルーチンを詳細に示す図である。この場合、ステップS31において、実行優先度キュー27に存在する実行中のタスク24の個数が1であるか否か判断する。

【0051】実行優先度キュー27に存在する実行中のタスク24の個数が1である場合、第1起動時刻と実行中のタスク24のWCETのうちの遅い方を締切として取得し(ステップS32)、本ルーチンを終了する。それに対して、実行優先度キュー27に存在する実行中のタスク24の個数が1でない場合、実行中のタスク24のWCETを締切として取得し(ステップS33)、本ルーチンを終了する。

【0052】実行優先度キュー27に実行中のタスク24しか存在しない場合、すなわち、実行優先度キュー27に存在するタスク24が1個である場合、プロセサ21の割付けを待機するタスク24が他に存在しないことを意味する。その結果、タスク24は、少なくとも第1起動時刻までプロセサ21を独占することができる。このために、この第1起動時刻と実行中のタスク24のWCETのうちの遅い方を締切とした場合でも、タスク24の処理に関して影響がない。

【0053】一方、実行優先度キュー27に実行中のタスク24以外のタスク24が存在する場合、すなわち、実行優先度キュー27に依存するタスク24が2個以上ある場合、締切を、実行中のタスク24のWCET(正確には、タスク全体のWCETから現在までの積算実行時間を減算し、現在の時刻を加算したもの)とすべきである。これによって、システムの実行確実性が維持される。

【0054】本実施の形態の動作を、図10-12も用いて説明する。この場合、複数の周期的なタスク(周期タスク群)24の例として、図10に示すように、実行優先度が1であり、WCETが6単位時間であり、かつ、周期が20単位時間であるタスクAと、実行優先度が2であり、WCETが12単位時間であり、かつ、周期が30単位時間であるタスクBと、実行優先度が3であり、WCETが2単位時間であり、かつ、周期が40単位時間であるタスクCとが存在するものとする。図10において、タスクA、B、Cの各々の矩形はスライスを表し、縦方向に周波数の高さを示し、横方向に時間経過を示す。

【0055】タスクA、B、Cの1スライスは2単位時間を有し、タスクA、B、Cの実際の実行時間は負荷に

依存する。この場合、タスクAの負荷が50%であると仮定し、実行時間をWCETの半分とする。また、タスクB、Cの負荷が100%であると仮定し、実行時間をWCETとする。

【0056】消費電力の最適化を行わないOSのスケジューリングでは、図11Aに示すようになる。それに対して、可変クロック $f_{var}$ として $f_{clk}$ 及び $f_{clk}/2$ が設定されると仮定した本発明でのスケジューリングでは、図11Bに示すようになる。

【0057】本発明でのスケジューリングについて詳細に説明する。時刻0でタスクA、B、Cが一斉に起動されると、図12に示すように、実行優先度が最も高いタスクAが実行される。

【0058】この場合、実行優先度キューにタスクが2以上存在するので、タスクAの1番目のスライスでは、締切は、タスクAの実行の開始からタスクA自身のWCETが経過した後の時刻すなわち時刻6となる。2番目のスライスから最後のスライスまでのWCETが4単位時間であるので、時間余裕は2単位時間となる。

【0059】WCETが2単位時間のスライスを2単位時間の時間余裕で実行するためには、可変周波数 $f_{var}$ として $f_{clk}$ が好適である。可変周波数 $f_{var}$ の値は、システムコールSCとしてOS23に提供され、デバイスドライバ28が有するルックアップテーブルに基づいて可変電源電圧及び/又はしきい値電圧 $V_{var}$ と可変周波数 $f_{var}$ とが設定される。

【0060】タスクAの負荷が50%であるため、本来のWCETでは2単位時間を要した1番目のスライスが1単位時間で終了する。タスクAの2番目のスライスでは、締切は、タスクA自身のWCETから1単位時間の積算実行時間を減算した5単位時間が締切までの時間となる。最終スライスのWCETが2単位時間であるので、時間余裕は3単位時間となる。2番目のスライスのWCETも2単位時間であるので、2番目のスライスも $f_{clk}$ の可変クロック周波数 $f_{var}$ で実行される。

【0061】最後のスライスでは、1番目及び2番目のスライスと状況が異なる。タスクAにおいて最後である3番目のスライスの実行が開始される時刻は2であり、積算実行時間は2単位時間である。3番目のスライスの実行の開始から締切までの時間は、タスクA自身のWCETの6単位時間から積算実行時間の2単位時間を減算した4単位時間である。

【0062】3番目のスライス以後にはタスクAにおいて実行すべきスライスが存在しないため、3番目のスライスの実行の開始から3番目のスライスの締切すなわちタスクAの締切までの4単位時間をそのまま利用することができる。その結果、時間余裕は4単位時間となる。

【0063】したがって、3番目のスライスでは、可変クロック周波数 $f_{var}$ を $f_{clk}/2$ に低減させることができ、これと同時に可変電源電圧及び/又はしきい値電圧

$V_{var}$ も制御することができる。

【0064】タスクAは時刻4で終了し、タスクBが上記手順に基づいて時刻4から時刻16の間で実行され、時刻16において、タスクCの1番目のスライスの実行を開始する。

【0065】時刻16では、タスクCのみが実行優先度キューに存在するので、締切は、次のタスクAが起動する時刻20と、タスクCの実行が開始される時刻16からタスクC自身のWCETである2単位時間が経過した時刻18のうちの遅い方の時刻すなわち時刻20となる。換言すれば、タスクC自身のWCETに比べてタスクCの実行開始から次のタスクAの起動時間までの方が、時間余裕が大きくなる。

【0066】この場合、タスクCは、タスクAの場合とは異なり、最初のスライスであるにもかかわらず、可変クロック周波数 $f_{var}$ を $f_{clk}/2$ まで低減することができる。タスクCが時刻20で終了する。

【0067】本発明によれば、実行可能なタスクの個数が1個である場合、実行中のタスクに第1起動時刻を提供することによって、複数のタスクが実行される環境でもプロセサの消費電力の最適化を行うことができる。具体的には、実行優先度キュー内に1個のタスクのみが存在する場合には、第1起動時刻とタスクの起動開始時刻からタスクのWCETが経過した時刻のうちの遅い方を締切とし、実行中のタスクについて可変クロック周波数 $f_{var}$ と可変電源電圧及び/又はしきい値電圧 $V_{var}$ とを制御して、プロセサの消費電力の最適化を行っているため、複数のタスクが実行される環境でもプロセサの消費電力の最適化を行うことができる。

【0068】本発明は、上記実施の形態に限定されるものではなく、幾多の変更及び変形が可能である。例えば、次の起動時間が検出される場合には、タスクが周期的でなくてもよく、消費電力の最適化を、電源電圧及び/又はしきい値電圧以外の他のパラメータを用いて制御することもでき、上記実施の形態で説明したプロセサ21及び電力制御回路22を、当業者によって他の設計にすることもできる。

【0069】また、電力制御アルゴリズム32を各タスク24にインストールした場合について説明したが、各タスク24が記録された記録された記録媒体（例えば、CD-ROM）から各タスク24を読み出すことによって電力制御を行うこともできる。

【図面の簡単な説明】

【図1】 従来の電力制御装置のブロック図である。

【図2】 図1の電力制御アルゴリズムを詳細に示す図である。

【図3】 アプリケーションスライシングを説明するための図である。

【図4】 図1のルックアップテーブルを詳細に示す図である。

【図5】 本発明による電力制御装置のブロック図である。

【図6】 図5のOS及びタスクを詳細に示す図である。

【図7】 タスク状態の遷移を説明するための図である。

【図8】 図6の電力制御アルゴリズムを詳細に示す図である。

【図9】 図8の締切取得ルーチンを詳細に示す図である。

【図10】 周期タスク群の例を示す図である。

【図11】 消費電力の最適化を行わないスケジューリング及び本発明によるスケジューリングを示す図である。

【図12】 図10の周期タスク群の本発明によるスケジューリングにおけるタスク状態の遷移を説明するための図である。

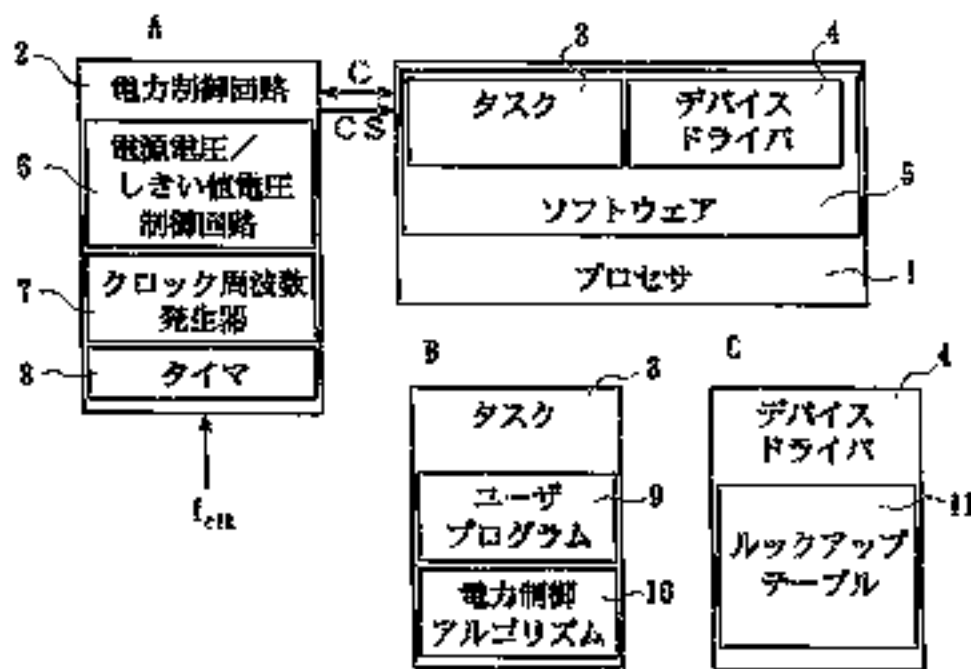
【符号の説明】

- 1、21 プロセサ
- 2、22 電力制御回路

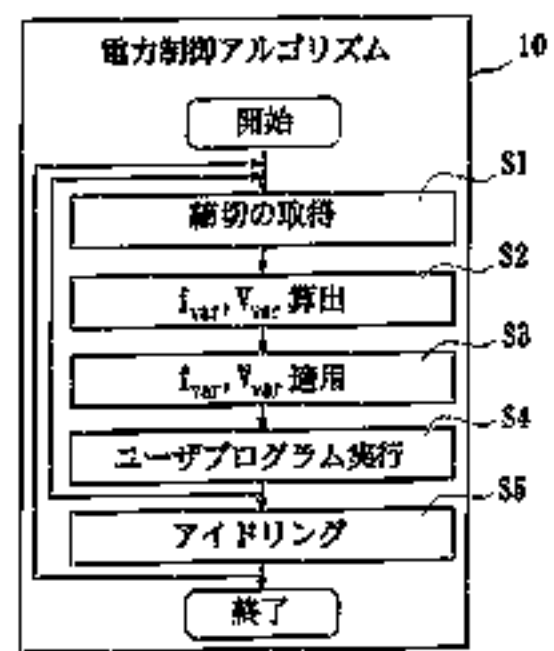
- \*3、24 タスク
- 4、28 デバイスドライバ
- 5、25 ソフトウェア
- 6 電源電圧/しきい値電圧制御回路
- 7 クロック周波数発生器
- 8 タイマ
- 9、31 ユーザプログラム
- 10、32 電力制御アルゴリズム
- 11 ルックアップテーブル
- 10 23 OS
- 26 起動時刻キュー
- 27 実行優先度キュー
- 29 スケジューラ
- 30 システムコールハンドラ
- C コマンド
- CS 制御信号
- DP プロセサ割付け等
- SC システムコール
- clock 外部クロック

\*20

【図1】



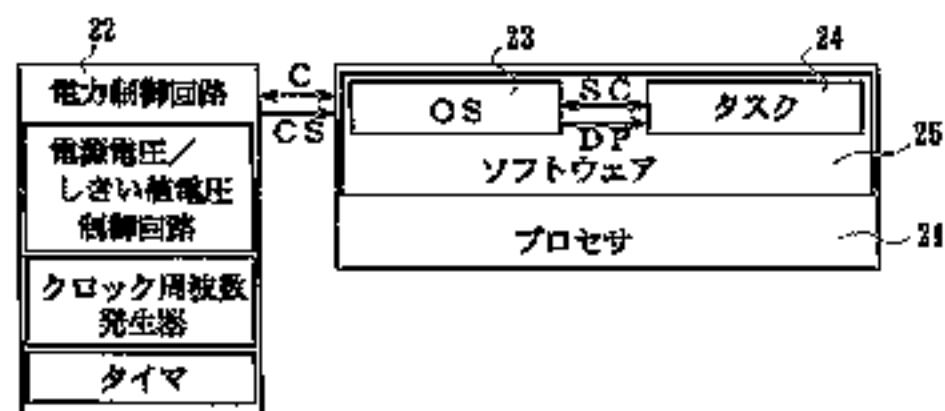
【図2】



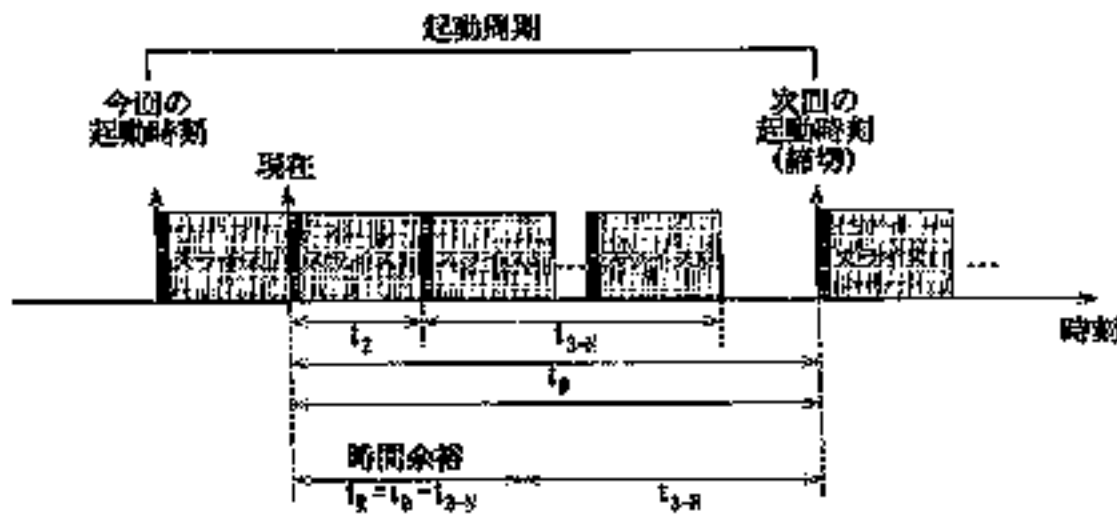
【図4】

$f_{var}$	$V_{var}$
$f_1 = f_{clk}$	$V_1$
$f_2 = f_{clk}/2$	$V_2$
$f_3 = f_{clk}/3$	$V_3$
...	...

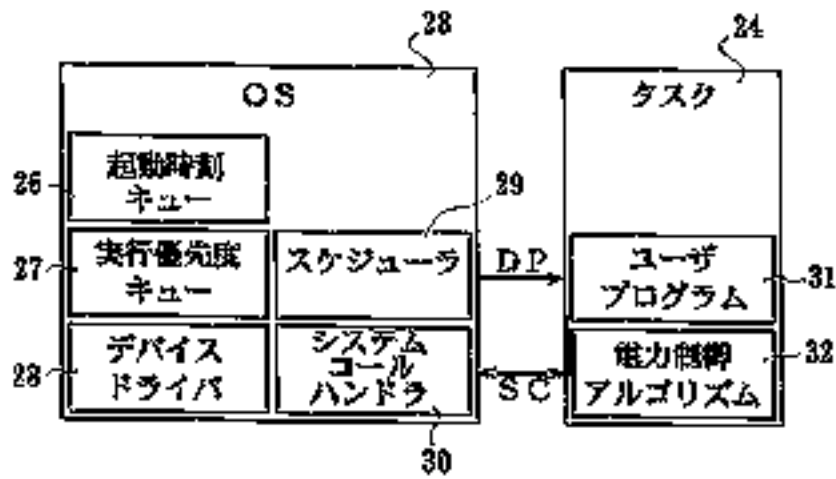
【図5】



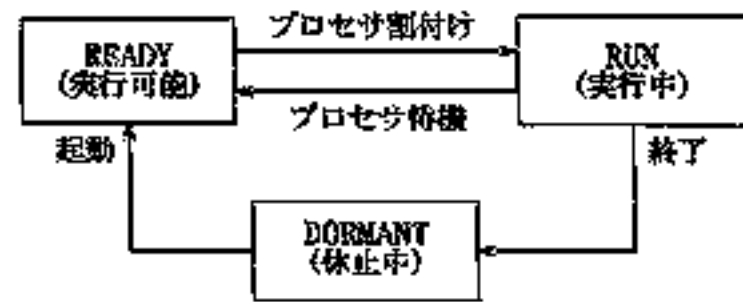
【図3】



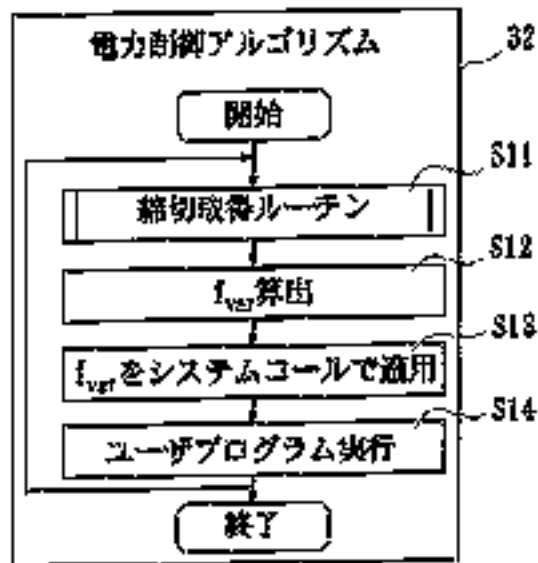
【図6】



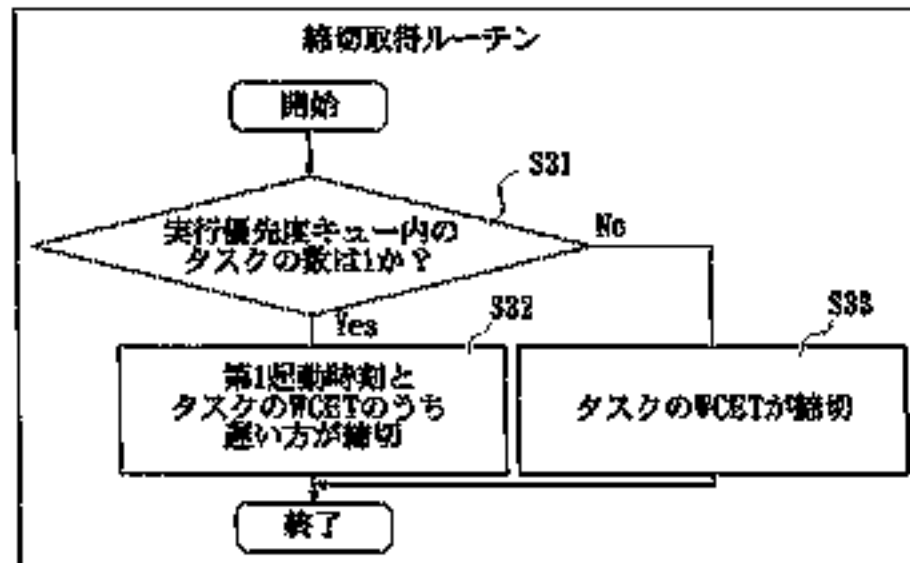
【図7】



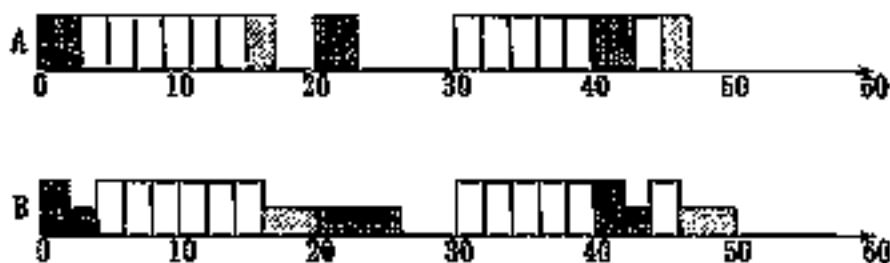
【図8】



【図9】

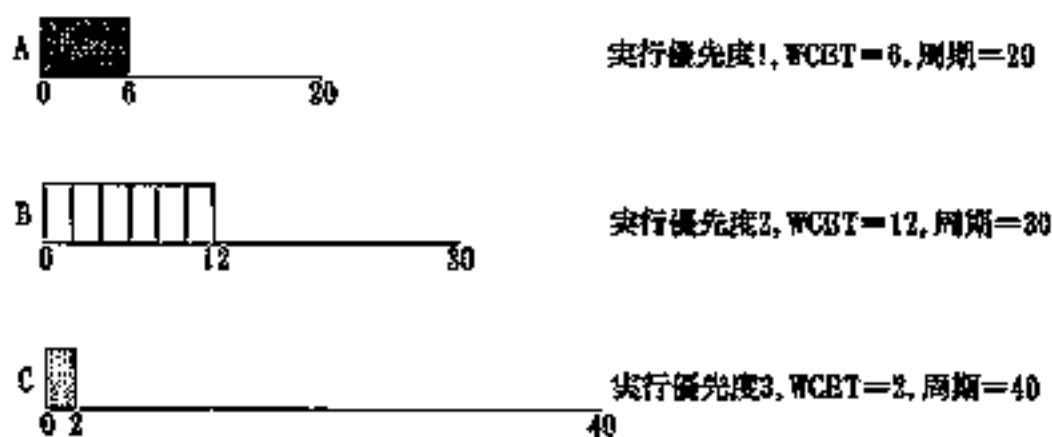


【図11】

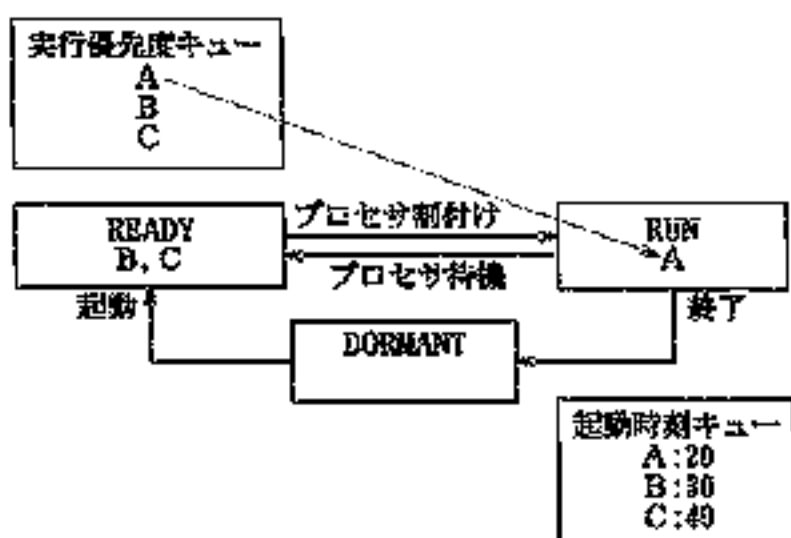




【図10】



【図12】



フロントページの続き

(72)発明者 辛 英洙  
アメリカ合衆国 ニューヨーク州ミルウッド  
ソーミルリヴァーロード324

(72)発明者 川口 博  
東京都江東区越中島3-5-27-209

(72)発明者 相坂 一夫  
東京都国分寺市京恋ヶ窪一丁目280番 株  
式会社日立製作所中央研究所内

(72)発明者 十山 圭介  
東京都国分寺市京恋ヶ窪一丁目280番 株  
式会社日立製作所中央研究所内

(72)発明者 石橋 孝一郎  
東京都国分寺市京恋ヶ窪一丁目280番 株  
式会社日立製作所中央研究所内

Fターム(参考) 5B011 DA06 EA04 EA10 KK02 LL12  
5B098 GA04 GC03 GC05 GD01 GD14