

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第3898789号

(P3898789)

(45) 発行日 平成19年3月28日(2007.3.28)

(24) 登録日 平成19年1月5日(2007.1.5)

(51) Int. Cl.

G06F 9/48 (2006.01)

F I

G06F 9/46 452G

請求項の数 5 (全 17 頁)

(21) 出願番号	特願平8-325421	(73) 特許権者	000005223 富士通株式会社 神奈川県川崎市中原区上小田中4丁目1番1号
(22) 出願日	平成8年12月5日(1996.12.5)	(74) 代理人	100094330 弁理士 山田 正紀
(65) 公開番号	特開平10-171668	(72) 発明者	五味 俊明 神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内
(43) 公開日	平成10年6月26日(1998.6.26)	(72) 発明者	矢崎 昌朋 神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内
審査請求日	平成15年7月24日(2003.7.24)	(72) 発明者	福岡 俊之 神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内

最終頁に続く

(54) 【発明の名称】 周期プロセス負荷制御システムおよび周期プロセス負荷制御方法

(57) 【特許請求の範囲】

【請求項1】

周期的に起動される周期プロセスを含む複数のプロセスを時分割で実行するCPUを備えたマルチプロセスシステムにおける、前記周期プロセスの実行に伴う前記CPUの負荷を制御する周期プロセス負荷制御システムにおいて、

前記周期プロセスは省略可能な処理部を含む複数の処理部と該周期プロセスの処理内容を制御する負荷制御部からなり、

前記省略可能な処理部は実行を省略するか否かを示す実行フラグを持ち、

前記周期プロセスが起動されると前記実行フラグが省略を示している前記省略可能な処理部以外の前記周期プロセス内のすべての前記処理部が実行された後に前記負荷制御部が実行され、

前記負荷制御部は前回動作した時刻と今回動作した時刻との間の時間間隔を計測して過去の複数の時間間隔の積算値からなる負荷評価値を求め、該負荷評価値に基づいて、前記省略可能な処理部の前記実行フラグの内容を設定することを特徴とする周期プロセス負荷制御システム。

【請求項2】

前記負荷制御部が、前記負荷評価値をしきい値と比較し該比較結果に応じて前記省略可能な処理部の前記実行フラグの内容を設定するものであって、

該周期プロセス負荷制御システムが、前記しきい値を変更自在に入力するしきい値入力部を備えたことを特徴とする請求項1記載の周期プロセス負荷制御システム。

10

20

【請求項 3】

前記負荷制御部が、該負荷制御部が求めた過去の複数の時間間隔の積算値を格納するスロットを複数備え、これら複数のスロットに格納された積算値どうしの加算値を前記負荷評価値として用いるものであることを特徴とする請求項 1 記載の周期プロセス負荷制御システム。

【請求項 4】

前記スロットの個数、及び / 又は、1 つのスロットに格納される積算値を構成する時間間隔の数を変更自在に入力するスロット条件入力部を備えたことを特徴とする請求項 3 記載の周期プロセス負荷制御システム。

【請求項 5】

周期的に起動される周期プロセスを含む複数のプロセスを時分割で実行する CPU を備えたマルチプロセスシステムにおける、前記周期プロセスの実行に伴う前記 CPU の負荷を制御する周期プロセス負荷制御方法であって、

前記周期プロセスは省略可能な処理部を含む複数の処理部と該周期プロセスの処理内容を制御する負荷制御部とからなり、

前記省略可能な処理部は実行を省略するか否かを示す実行フラグを持ち、

当該周期プロセス負荷制御方法は、

前記周期プロセスを起動するステップと、

前記実行フラグが省略を示している前記省略可能な処理部以外の前記周期プロセス内のすべての前記処理部を実行するステップと、

前記負荷制御部を実行するステップと、

前記負荷制御部が前回動作した時刻と今回動作した時刻との間の時間間隔を計測するステップと、

過去の複数の前記時間間隔の積算値からなる負荷評価値を求めるステップと、

前記負荷評価値に基づいて、前記省略可能な処理部の前記実行フラグの内容を設定するステップとを含むことを特徴とする周期プロセス負荷制御方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

【0002】

【発明の詳細な説明】

【0003】

【発明の属する技術分野】

本発明は、周期的に起動される周期プロセスを含む複数のプロセスを時分割で実行する CPU を備えたマルチプロセスシステムにおける、周期プロセスの実行に伴う CPU の負荷を制御する周期プロセス負荷制御システムおよび周期プロセス負荷制御方法に関する。

【0004】

【従来の技術】

近年のコンピュータシステムの高度化に伴い、複数のプロセスが時分割で実行されるタイムシェアリングシステムによるマルチプロセスシステムが普及してきている。

タイムシェアリングシステムによるマルチプロセスシステムの場合、基本的に各プロセスは独自に動作し、プロセスの実行切替えもそのタイムシェアリングシステムの構造にまかされており、一つのプロセスが処理に時間がかかるものになると、他のプロセスはそのまま遅れて処理される結果となる。また、タイマイベント等を用いてある特定の時刻に起動することが期待されたプロセスが存在しても、システムの都合（プロセスの優先順位、割り込み等）により指定時刻に起動されずとは限らない。

【0005】

図 15 は、タイマイベントを用いて周期的に繰り返し実行される周期プロセスのみが存在する場合の、その周期プロセスの実行時間を示す図、図 16 は、その周期プロセスの他にも実行されるプロセスが存在する場合の各プロセスの実行時間を示す図である。図 16 に

10

20

30

40

50

において白抜きの部分が注目している周期プロセスの実行時間、斜線の部分が他のプロセスの実行時間である。

【0006】

例えば図15に示すように、タイマイイベントを用いて周期的に繰り返し実行される周期プロセスのみが存在する場合は、そのタイマイイベントにしたがった起動タイミングどおりに起動される。

ところが、複数のプロセスが存在する場合、図16に示すように、周期プロセスの処理の間に別のプロセスに割り込まれたり、優先度の高いプロセスのために起動タイミングが守られず遅れて起動されたり、あるいは他のプロセスの実行が集中したために一周期分の処理が一周期間隔の間に終了せず次の周期に跨ってしまい、その周期における周期プロセスの実行が欠落したりするという問題がある。このようなことから処理が遅延する傾向にある。

10

【0007】

【発明が解決しようとする課題】

このように、タイムシェアリングシステムの場合、例えば映像や音声など高度のリアルタイム性が要求されるプロセスであっても他のリアルタイム性が要求されるプロセスの負荷の影響を受け、要求される時間通りに処理が間に合わず、リアルタイム性が損なわれてしまうことがある。

【0008】

本発明は、上記事情に鑑み、他のプロセスの実行によりCPUの負荷が増えた場合であっても、周期的に起動される周期プロセスのリアルタイム性の損失を緩和する周期プロセス負荷制御システムおよび周期プロセス負荷制御方法を提供することを目的とする。

20

【0009】

【課題を解決するための手段】

上記目的を達成する本発明の周期プロセス負荷制御システムは、周期的に起動される周期プロセスを含む複数のプロセスを時分割で実行するCPUを備えたマルチプロセスシステムにおける、周期プロセスの実行に伴うCPUの負荷を制御する周期プロセス負荷制御システムにおいて、

上記周期プロセスは省略可能な処理部を含む複数の処理部とその周期プロセスの処理内容を制御する負荷制御部からなり、

30

上記省略可能な処理部は実行を省略するか否かを示す実行フラグを持ち、

上記周期プロセスが起動されると実行フラグが省略を示している省略可能な処理部以外の周期プロセス内のすべての処理部が実行された後に負荷制御部が実行され、

上記負荷制御部は前回動作した時刻と今回動作した時刻との間の時間間隔を計測して過去の複数の時間間隔の積算値からなる負荷評価値を求め、その負荷評価値に基づいて、省略可能な処理部の実行フラグの内容を設定することを特徴とする。

【0010】

ここで、上記本発明の周期プロセス負荷制御システムにおいて、上記負荷制御部が、負荷評価値をしきい値と比較しその比較結果に応じて省略可能な処理部の実行フラグの内容を設定するものであって、

40

この周期プロセス負荷制御システムが、上記しきい値を変更自在に入力するしきい値入力部を備えたものであることが好ましい。

【0011】

また、上記本発明の周期プロセス負荷制御システムにおいて、上記負荷制御部が、その負荷制御部が求めた過去の複数の時間間隔の積算値を格納するスロットを複数備え、これら複数のスロットに格納された積算値どうしの加算値を負荷評価値として用いるものであることが好ましい。

その場合に、上記スロットの個数、及び/又は、1つのスロットに格納される積算値を構成する時間間隔の数を変更自在に入力するスロット条件入力部を備えることがさらに好ましい。

50

【 0 0 1 2 】

また、上記目的を達成する本発明の周期プロセス負荷制御方法は、周期的に起動される周期プロセスを含む複数のプロセスを時分割で実行するCPUを備えたマルチプロセスシステムにおける、周期プロセスの実行に伴うCPUの負荷を制御する周期プロセス負荷制御方法であって、

上記周期プロセスは省略可能な処理部を含む複数の処理部とその周期プロセスの処理内容を制御する負荷制御部とからなり、

上記省略可能な処理部は実行を省略するか否かを示す実行フラグを持ち、

この周期プロセス負荷制御方法は、

上記周期プロセスを起動するステップと、

上記実行フラグが省略を示している省略可能な処理部以外の周期プロセス内のすべての処理部を実行するステップと、

上記負荷制御部を実行するステップと、

上記負荷制御部が前回動作した時刻と今回動作した時刻との間の時間間隔を計測するステップと、

過去の複数の時間間隔の積算値からなる負荷評価値を求めるステップと、

上記負荷評価値に基づいて、省略可能な処理部の実行フラグの内容を設定するステップとを含むことを特徴とする。

10

【 0 0 1 3 】

本発明では、周期プロセスを構成する負荷制御部により上記の時間間隔が計測され、その時間間隔の積算値からなる負荷評価値が求められ、その負荷評価値に基づいてその周期プロセス自身の負荷が調整されるものであり、したがってその周期プロセスのリアルタイム性の損失が緩和される。

また、本発明において、上記しきい値入力部を備えると、しきい値を変更することにより、ソフトリアルタイムプロセス部の処理の実行の省略や、ソフトリアルタイムプロセス部を構成する複数の処理の中の負荷の小さい処理を選択することや、ソフトリアルタイムプロセス部の処理の実行の頻度を下げることに伴うデメリットと、リアルタイム性の確保とのバランス点を調整することができる。

20

【 0 0 1 4 】

さらに、本発明において、上記スロット条件入力部を備えると、全体としての負荷の変動に伴う、ソフトリアルタイムプロセス部の処理の実行やその省略の変更ソフトリアルタイムプロセス部を構成する複数の処理の中から選択される処理の変更、あるいはソフトリアルタイムプロセス部の処理の実行の頻度の変更の応答性、すなわち、負荷の短時間的な変動に回答するか、あるいは短時間的な負荷の変動には回答せず中長期的な負荷に回答するかを変更することができ、そのシステムないしそのシステムのユーザに適した応答性を得ることができる。

30

【 0 0 1 5 】

【 発明の実施の形態 】

以下、本発明の実施形態について説明する。

図1は、本発明の周期プロセス負荷制御システムの一実施形態を包含するコンピュータシステムの外觀図である。

40

図1に示すコンピュータシステム100は、CPUやMO（光磁気ディスク）ドライブ装置等が内蔵された本体部101、本体部101からの指示により、表示画面102a上に画像を表示するCRTディスプレイ102、このコンピュータシステム100に種々の情報を入力するためのキーボード103、CRTディスプレイ102の表示画面102a上の任意の位置を指定するマウス104、以下において説明する周期プロセス等を格納しておく記憶部105が備えられている。

【 0 0 1 6 】

周期プロセスを含むプロセス（プログラム）の開発は、この図1に示すコンピュータシステムでも行なうことができ、また、他の同種のコンピュータシステムを用いて開発された

50

ものを可搬型記憶媒体であるMO（光磁気ディスク）110に格納し、そのMO110をこの図1に示すコンピュータシステム100に装填して、このコンピュータシステム100に入力することもできる。またそれと同様に、この図1に示すコンピュータシステム100を用いて開発したプロセスを、MO110を介して他のコンピュータシステムに移すこともできる。

【0017】

図2は、図1に示すコンピュータシステム内に実現された本発明の周期プロセス負荷制御システムの一実施形態を示す図である。

起動イベント発生部10は、周期プロセス20を起動するためのイベントを周期的に発生するものであり、周期プロセス20はそのイベントが発生するたびに実行される。また時計部30は現在時刻を管理しており、周期プロセス20からの要求に応じて現在時刻を周期プロセス20に渡す。

10

【0018】

周期プロセス20は、3つの処理A、処理B、処理Cと、負荷制御部Dとからなる。これらの処理のうち処理Aは、例えば現在時刻に同期した内部状態を保つ必要のある処理であって、起動イベント発生部10がイベントを発生するたびに確実に実行されないで内部の時間が狂ってしまうような処理であり、すなわち、その周期プロセス20の内部状態を所定の状態に維持するためにイベント発生の際に必ず実行する必要のある処理である。

【0019】

一方、処理Bおよび処理Cは、イベント発生の際に実行されるのが望ましいが、イベント発生の際に必ずしも実行されなくても、その後実行されることによりイベント発生の際に必ず実行された場合と同一の状態を回復することができる処理である。ここでは、処理Bと処理Cを比べたとき、処理Bの方が比較的イベント発生の際に実行されるのが好ましい処理であるとする。

20

【0020】

処理B、処理Cは、それぞれ、以下に説明する負荷制御部Dによりセットあるいはリセットされる各実行フラグB__1、C__1を持っている。処理Bの実行にあたっては、先ず実行フラグB__1が参照され、その実行フラグB__1がセットされているときは処理Bが実行され、その実行フラグB__1がリセットされていたときは処理Bの実行は省略される。処理Cについても同様である。

30

【0021】

負荷制御部Dは、前回の時刻を保存しておく時刻保存部D__1、前回の時刻から今回の時刻までの経過時間、ないし過去の複数の経過時間の積算値を保存しておく複数のスロットD__2a、D__2b、D__2cを備え、後述するようにして負荷状態を判断する。また、この負荷制御部Dは判断結果処理部D__3を内蔵しており、この判断結果処理部D__3では、負荷状態の判断結果に応じて、処理B、処理Cの実行フラグB__1、C__1のセット、リセットを行なう。

【0022】

尚、この図2に示す周期プロセス負荷制御システムを構成する周期プロセス20が記憶された状態の記憶部105（図1参照）が本発明の周期プロセス記憶媒体の一実施形態に相当する。この図2に示す周期プロセス20がMO110にダウンロードされたときは、その周期プロセス20を格納した状態にあるMO110も、本発明の周期プロセス記憶媒体の一例に相当する。

40

【0023】

図3は、各負荷状態における周期プロセスの一回あたりの処理量をあらわした模式図である。

起動イベント発生部10によりイベントが発生したときにおいて、低負荷の場合は、図3（1）に示すように、処理A、処理B、処理C、および負荷制御部Dの処理がこの順に実行される。高負荷の場合は、図3（2）に示すように、処理A、処理Bを実行した後、処

50

理 C は省略されて負荷制御部 D の処理が実行される。さらに高負荷の場合は、処理 A を実行した後処理 B , 処理 C 双方が省略されて負荷制御部 D の処理が実行される。このようにして、高負荷の場合の周期プロセス 20 のリアルタイム性の低下が緩和される。

【 0 0 2 4 】

図 4 は、負荷制御部の処理の流れを示すフローチャートである。

負荷制御部 20 の処理の実行が開始すると、先ず時計部 30 からの時刻の取得が行なわれる (ステップ 4 _ 1)。尚、時計部 30 から取得する時刻は現在時刻である必要はなく、プロセスの実行が最初に起動された時刻からの経過時間であってもよく、要するに前回の時刻との差に相当する経過時間を求めることのできる時刻であればよい。

【 0 0 2 5 】

次いで、今回始めてこの周期プロセス 20 の処理が実行されたか否かの判定が行なわれ (ステップ 4 _ 2)、この周期プロセス 20 の処理が始めて行なわれたときは時計部 30 から取得した時刻を時刻保存部 D _ 1 に保存し、かつ通常負荷であると判定し (ステップ 4 _ 3)、判定結果処理部 D _ 3 の処理の実行に移る (4 _ 1 2)。判定結果処理部 D _ 3 の処理については後述する。尚、この周期プロセス 20 が初回に動作する前は、初期状態として、処理 B , C の実行フラグ B _ 1 , C _ 1 がセットされた状態にあり、したがって初回の動作時には処理 B , C が実行され、その後、この図 4 に示す負荷制御部 D の処理に移る。

【 0 0 2 6 】

ステップ 4 _ 2 の判定において周期プロセス 20 の処理が今回始めてではないと判定されるとステップ 4 _ 4 に進み、時計部 30 から今回取得した時刻と時刻保存部 D _ 1 に格納されている前回の時刻との差を求め、前回の実行時から今回の実行時までの経過時間 D T を得 (ステップ 4 _ 4)、時計部 30 から今回取得した時刻を時刻保存部 D _ 1 に格納し (ステップ 4 _ 5)、ステップ 4 _ 4 で求めた経過時間 D T を複数スロット D _ 2 a , D _ 2 b , ... , D _ 2 c のうちの 1 つに保存する (ステップ 4 _ 6)。その保存の際、そのスロットに既に経過時間が保存されているときはそのスロットに既に保存されている経過時間に今回求めた経過時間を加算してそのスロットに保存する。

【 0 0 2 7 】

図 5 は、スロットの経過時間の保存方法を示す模式図である。

ここには、S L O T 1 ~ S L O T 10 の 10 個のスロットが用意されており、各スロットには、5 回の経過時間の積算値が格納される。すなわち、図 5 (1) に示すような S L O T 1 ~ S L O T 10 の全てのスロットが空の状態から出発し、先ず S L O T 1 に順次積算しながら 5 回分の経過時間が順次積算されて格納され、S L O T 1 に 5 回分の経過時間の積算値が格納された後は、同様にして、S L O T 2 に順次積算しながら格納される。

【 0 0 2 8 】

図 5 (2) は、 $5 + 2 = 7$ 回分の経過時間が格納された状態、図 5 (3) は $5 \times 9 + 3 = 48$ 回分の経過時間が格納された状態、図 5 (4) は $5 \times 10 = 50$ 回分の経過時間が格納され丁度満杯になった状態をあらわしている。図 5 (4) に示すように満杯になると一番先に経過時間が格納されたスロット (ここでは S L O T 1) がクリアされ、図 5 (5) に示すように、そのクリアされたスロットに最新の経過時間が格納される。以下このような処理が繰り返される。ただし、以下に説明するようにして「通常」負荷以外の、「高負荷」と判定された場合および「低負荷」と判定された場合は、その判定の時点で S L O T 1 ~ S L O T 10 のスロットが全てクリアされて図 5 (1) の状態に戻り、再度、その後の最初の経過時間から順に格納される。

【 0 0 2 9 】

ここでは、各スロットには複数 (ここでは 5 つ) の経過時間の積算値が格納されるが、スロットの数を増やし各スロットに経過時間を 1 つずつ格納してもよい。ただし、その場合、スロットの数が増え大きなメモリ容量を必要とすることになる。

尚、上記のスロットの数 (図 5 に示す例では 10 個) や 1 つのスロットに積算して格納する経過時間の数 (図 5 に示す例では 5 つ) を固定しておく代わりに、図 1 に示すキーボー

10

20

30

40

50

ド103の操作等によりユーザが自由に設定できるように構成しておくことが好ましい。これらの設定を変更することにより、同一の負荷状態であっても、高負荷であると判定され、あるいは低負荷であると判定される時間間隔が調整される。

【0030】

図4に戻って負荷制御部の処理の流れについての説明を続行する。

ステップ4__7では、複数のスロットに格納された経過時間の積算値どうしが加算されて負荷評価値が求められ、その負荷評価値が高負荷であるか否かを定めるしきい値よりも大きいか否かが判定される。

負荷評価値がその判定値よりも大きいときはステップ4__8に進み、高負荷であると判定し、かつ全てのスロットをクリアして、ステップ4__12の判定結果処理部D__3の処理の実行に移る。

10

【0031】

ステップ4__7における高負荷であるか否かの判定は、経過時間が一回求められるたび、すなわち周期プロセスが一回動作するたびに行なわれる。

ステップ4__7において高負荷ではないと判定されると、ステップ4__9に進み、このステップ4__9では、負荷評価値が低負荷であるか否かを定めるしきい値よりも小さいか否かが判定される。

【0032】

この低負荷であるか否かの判定は、全てのスロットが満杯となった状態、すなわち図5に示す例では、図5(4)の状態において行なわれ、かつ、その後全てのスロットの同時クリアが行なわれないうまま経過時間が5回求められるごとに行なわれる。

20

ステップ4__9において負荷評価値が低負荷であるか否かの判定値よりも小さいと判定されたときはステップ4__10に進み、低負荷であると判定し、かつ全てのスロットをクリアして、ステップ4__12の判定結果処理部D__3の処理に移る。

【0033】

ステップ4__7における高負荷があるか否かの判定を行なう際のしきい値としては、設定周期(起動イベント発生部10によるイベントの発生間隔)をT、複数のスロットによる経過時間DTの最大保存数(図5に示す例では5×10スロット=50)を DT_{MAX} としたとき、例えば設定周期の20%の遅延をリアルタイム性を損う限界とみなした場合、

$$T \times DT_{MAX} \times 1.2 \quad \dots\dots (1)$$

30

が目安となり、ステップ4__9における低負荷であるか否かを判定に用いるしきい値は、

$$T \times DT_{MAX} \quad \dots\dots (2)$$

が目安となる。

【0034】

尚、例えば(1)式、(2)式を目安とした高負荷、低負荷のしきい値を固定的に定められてもよいが、ユーザによるキーボード103の操作等により変更可能に構成しておくことが好ましい。これらのしきい値を変更可能としておくことにより、処理B、処理Cが実行されることによるメリットと、処理B、処理Cの実行によりリアルタイム性が損なわれることによるデメリットとのバランスをユーザが選択することができる。

【0035】

40

判定結果処理部D__3では、ステップ4__8において高負荷であると判定されて判定結果処理部D__3の処理に移ったときは、処理Bの実行フラグB__1と処理Cの実行フラグC__1を見てそれらの実行フラグB__1、C__1の双方がセットされていたとき(すなわち処理Bと処理Cの双方が実行される状態にあるとき)は、処理Cの実行フラグC__1がリセットされ、これにより処理Cの実行は、次回以降、その実行フラグC__1が再びセットされるまでの間、省略される。また、判定結果処理部D__3は、高負荷であると判定された場合において処理Bの実行フラグB__1と処理Cの実行フラグC__1を見て処理Bの実行フラグB__1はセットされており、処理Cの実行フラグC__1がリセットされていたときは、処理Bの実行フラグB__1をリセットする。これは、それよりも以前に処理Cの実行フラグC__1がリセットされており、その後処理Cの実行が省略されたにも拘らずなお高負荷

50

であると判定された場合に処理 B の実行も省略し、これにより負荷をさらに軽減しリアルタイム性を確保するためである。高負荷であると判定されて判定結果処理部 D __ 3 の処理に移った場合において処理 B の実行フラグ B __ 1 と処理 C の実行フラグ C __ 1 の双方が既にリセットされていたときは、処理 A の実行は省略することができないため、特に何も行なわれない。あるいは、例えば図 1 に示す CRT ディスプレイ 1 0 2 の表示画面 1 0 2 a に極めて高負荷である旨の警告を表示してもよい。

【 0 0 3 6 】

一方、ステップ 4 __ 1 0 において低負荷であると判定されてステップ 4 __ 1 2 の判定結果処理部 D __ 3 の処理に移ったときは、上記の高負荷の場合とは逆のフラグ操作が行なわれる。すなわち、低負荷であると判定された場合において処理 B の実行フラグ B __ 1 と処理 C の実行フラグ C __ 1 との双方がリセットされていたときは処理 B の実行フラグ B __ 1 がセットされ、低負荷であると判定された場合において処理 B の実行フラグ B __ 1 がセットされており、かつ処理 C の実行フラグ C __ 1 がリセットされていたときは、処理 C の実行フラグがセットされる。これは、処理 A と処理 B が実行され処理 C の実行が省略されている場合においてそれでも低負荷であった場合に処理 C も実行させるようにするためである。低負荷であると判定された場合において処理 B の実行フラグ B __ 1 と処理 C の実行フラグ C __ 1 の双方がセットされていたときは何も行なわない。

【 0 0 3 7 】

ステップ 4 __ 3、あるいはステップ 4 __ 1 1 において通常負荷であると判定されてステップ 4 __ 1 2 の判定結果処理部 D __ 3 の処理に移ったときは判定結果処理部 4 __ 1 2 では何の処理も行なわれない。すなわちその時の処理 B、処理 C の実行状態あるいは実行の省略の状態が継続される。

尚、判定結果処理部 4 __ 1 2 において処理 B の実行フラグ B __ 1 をセット、リセットするには、それぞれ関数コールで

B __ a c t i v e _ s e t (1)

B __ a c t i v e _ s e t (0)

のように行ない、実行フラグ B __ 1 をセット (' 1 ' を格納) し、あるいはリセット (' 0 ' を格納) する。処理 C の実行フラグ C __ 1 をセット、リセットする場合も、同様に、

C __ a c t i v e _ s e t (1)

C __ a c t i v e _ s e t (0)

のように行なわれる。

【 0 0 3 8 】

図 6 は、図 2 に示す周期プロセス 2 0 の他に実行されるプロセスが存在しない場合の周期プロセスの実行状態を示した図である。

この場合、処理 B、C を含めた周期プロセス全体が起動タイミング (起動イベント発生部 1 0 によるイベント発生のタイミング) に同期して実行される。この場合、経過時間 D T は一定であり、起動タイミングの周期と同一である。

【 0 0 3 9 】

図 7 は、図 2 に示す周期プロセス 2 0 の他にも実行されるプロセスが存在する場合の、それらのプロセスの実行状態を示した図である。

この図 7 に示すように他のプロセスが実行されると周期プロセスの実行途中で他のプロセスが割り込んでくることになり、経過時間 D T が大きくなる。

図 8 は、処理 C の実行が省略された状態のプロセスの実行状態を示した図、図 9 は、処理 B と処理 C の双方の実行が省略された状態のプロセスの実行状態を示した図である。

【 0 0 4 0 】

図 7 に示すように経過時間 D T が大きくなり高負荷であると判定されると、図 8 に示すように処理 C の実行が省略されて、各起動タイミング毎に処理 A、処理 B および負荷制御部 D の処理が実行され、さらに他のプロセスが実行される。

この図 8 に示す状態においてもなお経過時間 D T が大きくなり再度高負荷であると判定されると、今度は図 9 に示すように、処理 C とともに処理 B も省略されて、各起動タイミン

10

20

30

40

50

グ毎に処理 A と負荷制御部 D の処理が実行され、さらに他のプロセスが実行される状態となる。

【 0 0 4 1 】

図 9 に示す状態において経過時間 D T が小さくなり低負荷であると判定されると、図 8 に示す、処理 B が実行される状態に戻り、図 8 に示す状態において経過時間 D T が小さくなり低負荷であると判定されると、図 7 に示す、処理 C が実行される状態に戻る。

図 1 0 は、負荷評価値の推移を示す模式図である。

【 0 0 4 2 】

ここでは簡単のためスロットは 4 個とし、各スロットに各 1 つの経過時間 D T を格納するものとする。

処理 A , B , C の全てが実行されている状態において各スロットへ各経過時間が順次格納され、それら 4 つのスロットに格納された 4 つの経過時間の加算値からなる負荷評価値が求められる。この負荷評価値が時刻 T 1 において高負荷判定のしきい値を越えるとその時刻 T 1 において処理 C が実行から外される。その後も負荷評価値がモニタされ時刻 T 2 において再度高負荷判定のしきい値を越えるとその時刻 T 2 において今度は処理 B が実行から外される。時刻 T 3 では、全てのスロットに経過時間が格納れた状態において低負荷判定のしきい値を越えなかったため処理 B の実行が再開され、時刻 T 4 では、処理 B が実行されており、かつ全てのスロットに経過時間が格納された状態において低負荷判定のしきい値を越えなかったため、処理 C の実行が再開される。

【 0 0 4 3 】

以上のようにして経過時間 D T の遅れに応じて負荷調整が行なわれ、周期プロセスのリアルタイム性が確保される。

図 1 1 は、図 1 に示すコンピュータシステム内に実現された本発明の周期プロセス負荷制御システムの、より具体的な実施形態を示す図である。

図 1 1 に示すように、本実施形態の周期プロセス負荷制御システムはある時間間隔ごとにハードウェア割り込みを発生するタイマ割り込み発生部 4 0 と、その割り込み信号を受け取る C P U 5 0 と、C P U 5 0 が割り込み信号を受けたことで起動されるイベント処理部 6 0 と、イベント処理部 6 0 からタイマイベントを受け取ることで起動する周期プロセス 7 0 と、リアルタイムクロック 8 0 と、メモリ領域 V R A M 9 0 と、V R A M 9 0 の内容を表示する C R T ディスプレイ 1 0 2 と、周期プロセス 7 0 とは独立な T S S プロセス A , B とによって構成される。

【 0 0 4 4 】

ここで周期プロセス 7 0 は、物体移動計算モジュール 7 1 と、物体描画イメージ作成モジュール 7 2 と、物体描画イメージ転送モジュール 7 3 と、負荷制御部 7 4 と、周期プロセス 7 0 内に確保されるイメージバッファ 7 5 とによって構成される。各モジュールの機能について後述する。また、負荷制御部 7 4 内には負荷判断部 7 4 A と判断結果処理部 7 4 B が含まれ、負荷判断部 7 4 A には前回の時刻を記憶しておく時刻記憶部 7 4 1、および 1 0 個のスロット 7 4 2 _ 1 ~ 7 4 2 _ 1 0 が含まれ、物体描画イメージ転送モジュール 7 3 には実行フラグ 7 3 1 が含まれる。

【 0 0 4 5 】

尚、イベント処理部 6 0、周期プロセス 7 0、T S S プロセス A , B はソフトウェア的要素であり、C P U 5 0 が扱うメモリ上に存在する。

本発明との対比では、タイマ割り込み発生部 4 0 と、イベント処理部 6 0 と、タイマ割り込みによってイベント処理部 6 0 を実行する C P U 5 0 との組合せが本発明にいう起動イベント発生部に相当し、周期プロセス 7 0 が本発明にいう周期プロセスに相当し、物体描画イメージ転送モジュール 7 3 が本発明にいうソフトリアルタイムプロセス部に相当し、さらに負荷制御部 7 4 が本発明にいう負荷制御部に相当する。

【 0 0 4 6 】

タイマ割り込み発生部 4 0 は、定期的に割り込み信号を発生するハードウェア割り込み発生部である。C P U 5 0 は、タイマ割り込み発生部 4 0 から割り込み信号を受信すると、

10

20

30

40

50

イベント処理部 60 に対し、タイマ割り込みが発生したことを通知する。イベント処理部 60 は CPU 50 からタイマ割り込みが発生したことの通知を受け、タイマイベント待ちになっているプロセスのイベント発行タイミングになっていれば、タイマイベント待ちになっているプロセス（ここでは周期プロセス 70）にタイマイベントを発行し、そのプロセスを起動する。

【0047】

周期プロセス 70 は、イベント処理部 60 からタイマイベントを受け取ることで実行が開始される。周期プロセス 70 内では、物体移動計算モジュール 71、物体描画イメージ作成モジュール 72、物体描画イメージ転送モジュール 73、負荷制御部 74 がこの順に実行される。これらが一周期分の処理である。物体移動計算モジュール 71 は物体の移動における座標計算等を主に行なう。物体描画イメージ作成モジュール 72 は物体移動計算モジュール 71 の計算結果を受けて、イメージバッファ 75 に、描画イメージを書き込む。物体描画イメージ転送モジュール 73 はイメージバッファ 75 内に作成された描画イメージを VRAM 90 に転送する。描画イメージが VRAM 90 に転送されると、CRT ディスプレイ 102 の表示画面 102a に、その描画イメージが画像として表示される。

10

【0048】

物体描画イメージ転送モジュール 73 は、イメージバッファ 75 に作成された描画イメージの転送に当り、ハードウェアのバスを長時間占有するものであり、一周期分の実行時間の中でも占める割合が大きい。ただし、このモジュールの実行を省いても周期プロセス 70 の内部状態に影響がないことから、実行フラグ 731 を用意して、実行するかしないかを選択できるようにしている。実行フラグ 731 がオンであれば、物体描画イメージ転送モジュール 731 は実行され、実行フラグ 731 がオフであれば物体描画イメージ転送モジュール 73 は実行されない。この実行フラグ 731 は、初期状態ではオンになっている。

20

【0049】

負荷制御部 74 では、まず負荷判断部 74A の処理が実行され、リアルタイムクロック 80 から現在時刻を取得し、時刻保持部 471 が保持していた前回の時刻との差分を算出する。算出した経過時間はスロットに保存される。ここではスロット一つにつき一つの時間間隔が保持される。よって、本実施形態では、スロットが 10 個有るため過去 10 回分の時間間隔が保持されることになる。判断結果処理部 74B では、負荷判断の結果高負荷であると判断されたときは、物体描画イメージ転送モジュール 73 の実行フラグ 731 をオフにし、低負荷であると判断されたときは、その実行フラグ 731 をオンにする。その後、一周期分のプロセスの実行が終了し、次のタイマイベントが届くまで待ちの状態になる。

30

【0050】

ここで、タイマイベントの発行間隔を 100 msec に設定する。このとき、10 個のスロット 742_1, ..., 742_10 にそれぞれ 100 (msec) が格納されるとそれらの加算値（負荷平均値）は 1000 (msec) となる。これに対し、低負荷の基準値を 1010 (msec) とし、高負荷の基準値も 1010 (msec) とする。すなわちその負荷評価値が 1010 を越えるか否かに応じて高負荷であるか低負荷であるかを判定する。ここでは、周期プロセス 70 のほかに TSS プロセス A、B が実行されている環境において、周期プロセス 70 は、ボールを、等速度で CRT ディスプレイ 102 の表示画面 102a の左から右に移動させるよう動画を表示させるものとする。ここでは、周期が乱れないときは 10 秒でその移動が終了するものとする。

40

【0051】

図 12 は実行周期に乱れないときの、CRT ディスプレイ 102 の表示画面 102a 上に表示された画像の時間的推移を示した模式図、図 13 は、実行周期に乱れが生じたにも拘らず仮に負荷制御部 74 を動作させなかった場合、すなわち従来例に相当する表示画像の時間的推移を示した模式図、図 14 は、実行周期に乱れが生じ、負荷制御部 74 を動作させた場合、すなわち本実施形態の場合の表示画像の時間的推移を示した模式図である。

50

図14において、白抜きの部分は物体描画イメージ転送モジュール73の実行も含めて周期プロセス70が実行された時間帯であり、斜線分は負荷制御により物体描画イメージ転送モジュール73の実行が省略されて周期プロセス70が実行された時間帯である。

【0052】

TSSプロセスA、Bの負荷が周期を乱さない程度のものであれば、図12に示すように10秒間で移動表示が完了する。負荷制御部74が動作しない場合、TSSプロセスA、Bの負荷が周期を乱すほどのものであれば、図13のように表示が10秒よりも遅れて終了する。負荷制御部74が動作すれば、図14のように、途中で移動がストップしたように表示される場合が生じるものの、図13の場合より遅れの幅は縮小される。

【0053】

尚、上記実施形態では、オン、オフの二値のみからなる実行フラグを有し、その実行フラグのオン、オフに応じてある特定のモジュール(ソフトリアルタイムプロセス部)を実行させ、あるいは、実行を省略したが、オン、オフの二値ではなく多値からなる実行レベルを設定するようにし、実行レベルに応じて負荷の異なる処理を実行するようにしてもよい。負荷の程度の異なる処理とは、例えば上記の実施形態における物体描画イメージ転送モジュール73においてイメージバッファ75の描画イメージ全体をVRAM90に転送するか、1画素おきの粗いイメージを転送し画像の分解能を犠牲にして負荷を低減させるかといった処理が挙げられる。あるいは描画イメージを転送するときはその描画イメージ全体を転送することとし、ただし一周ごとに必ず転送するか一周おきに転送するか2周期おきに転送するかといった転送の頻度を実行レベルに応じて変更するようにしてもよい。転送の頻度を変更するようにした場合は、多数の周期に跨がって表示が動かないという見苦しさが緩和できる。

【0054】

【発明の効果】

以上説明したように、本発明によれば、タイムシェアリングシステムによるマルチプロセスシステムにおいて周期プロセスの負荷状態を検出し制御することで、時間依存性の高い、例えば映像等を処理するプロセスのリアルタイム性の損失を緩和することができる。特に数時間、数千時間、といったコンピュータ内部で長時間動作し続けるマルチメディアプロセスにおいてその効果が大きい。

【図面の簡単な説明】

【図1】本発明の周期プロセス負荷制御システムの一実施形態を包含するコンピュータシステムの外觀図である。

【図2】図1に示すコンピュータシステム内に実現された本発明の周期プロセス負荷制御システムの一実施形態を示す図である。

【図3】各負荷状態における周期プロセスの一回あたりの処理量をあらわした模式図である。

【図4】負荷制御部の処理の流れを示すフローチャートである。

【図5】スロットの経過時間の保存方法を示す模式図である。

【図6】周期プロセスの他に実行されるプロセスが存在しない場合の周期プロセス実行状態を示した図である。

【図7】周期プロセスの他にも実行されるプロセスが存在する場合の、それらのプロセスの実行状態を示した図である。

【図8】処理Cの実行が省略された状態のプロセスの実行状態を示した図である。

【図9】処理Bと処理Cの双方の実行が省略された状態のプロセスの実行状態を示した図である。

【図10】負荷評価値の推移を示す模式図である。

【図11】図1に示すコンピュータシステム内に実現された本発明の周期プロセス負荷制御システムのより具体的な実施形態を示す図である。

【図12】実行周期に乱れがないときの、CRTディスプレイの表示画面上に表示された画像の時間的推移を示した模式図である。

10

20

30

40

50

【図13】従来例に相当する表示画像の時間的推移を示した模式図である。

【図14】本実施形態の場合の表示画像の時間的推移を示した模式図である。

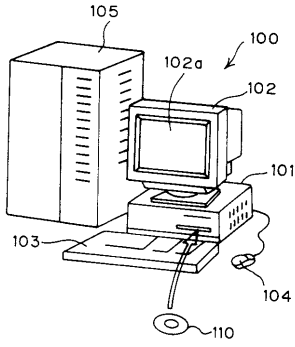
【図15】タイマイイベントを用いて周期的に繰り返し実行される周期プロセスのみが存在する場合の、その周期プロセスの実行時間を示す図である。

【図16】周期プロセスの他にも実行されるプロセスが存在する場合の各プロセスの実行時間を示す図である。

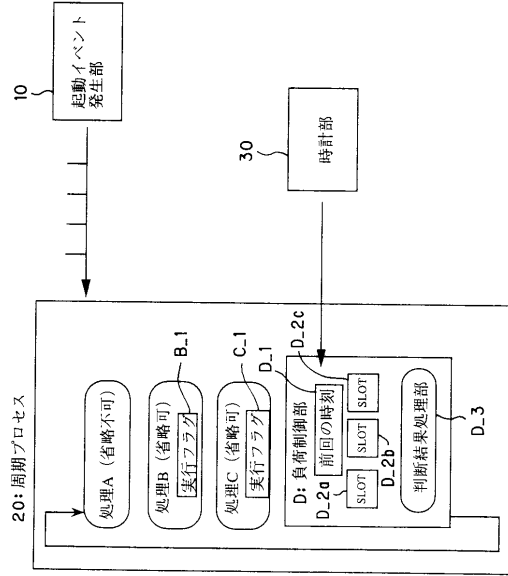
【符号の説明】

10	起動イベント発生部	
20	周期プロセス	
30	時計部	10
40	タイマ割り込み発生部	
50	CPU	
60	イベント処理部	
70	周期プロセス	
71	物体移動計算モジュール	
72	物体描画イメージ作成モジュール	
73	物体描画イメージ転送モジュール	
731	実行フラグ	
74	負荷制御部	
74A	負荷判断部	20
741	時刻記憶部	
742	__1, ..., 742__10	スロット
74B	判断結果処理部	
80	リアルタイムクロック	
90	VRAM	
100	コンピュータシステム	
101	本体部	
102	CRTディスプレイ	
102a	表示画面	
103	キーボード	30
104	マウス	
105	記憶部	
110	MO	

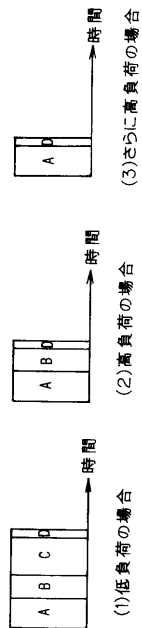
【 図 1 】



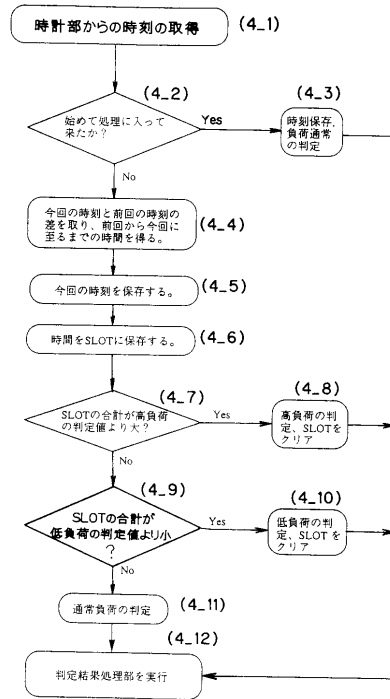
【 図 2 】



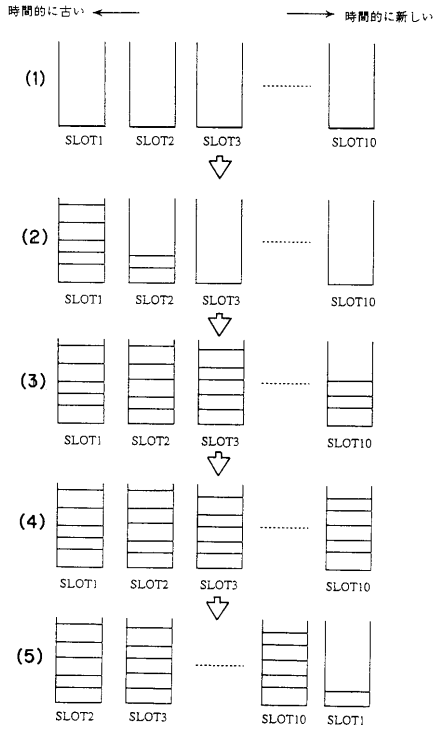
【 図 3 】



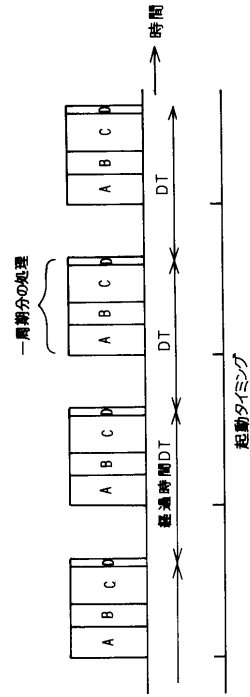
【 図 4 】



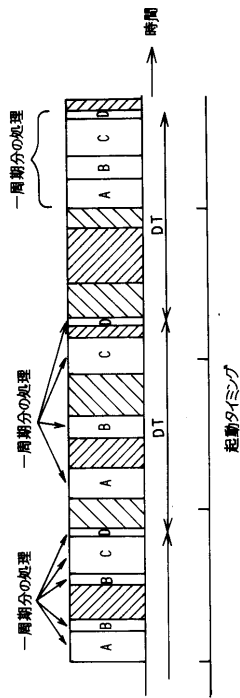
【 図 5 】



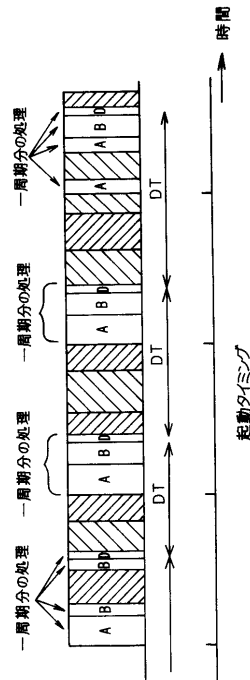
【 図 6 】



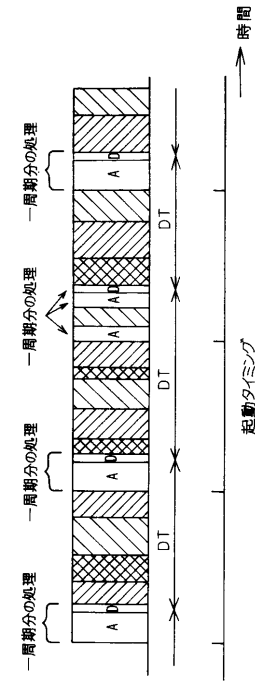
【 図 7 】



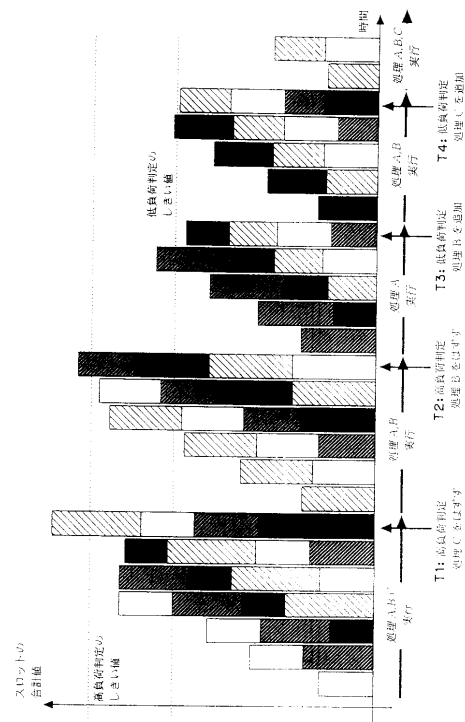
【 図 8 】



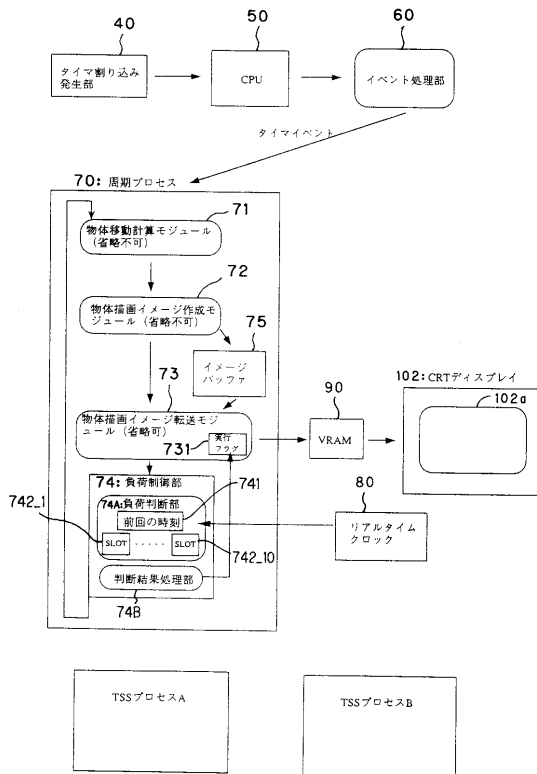
【 図 9 】



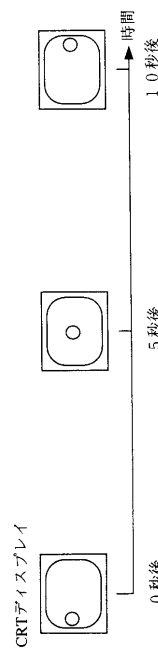
【 図 10 】



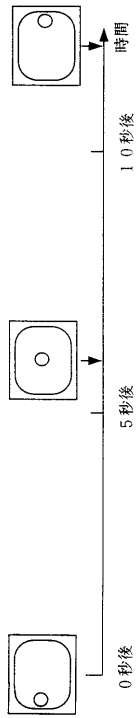
【 図 11 】



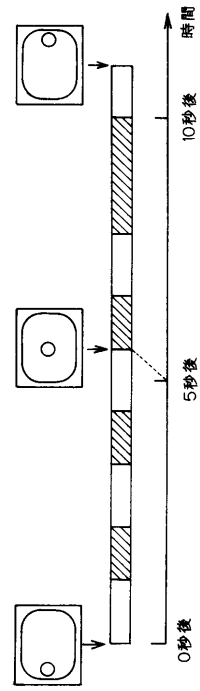
【 図 12 】



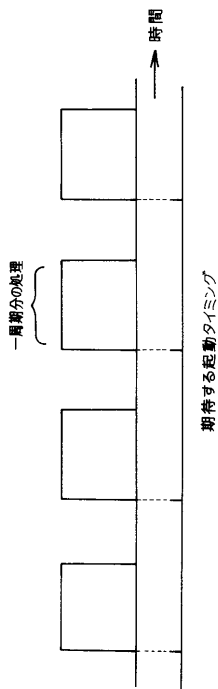
【図 13】



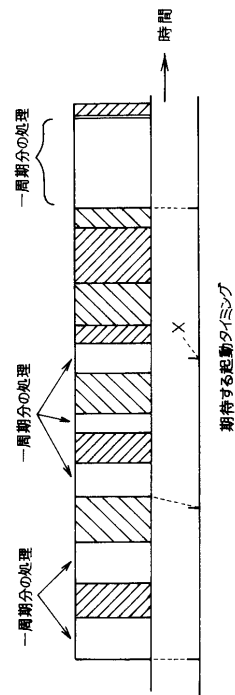
【図 14】



【図 15】



【図 16】



フロントページの続き

- (72)発明者 野田 政秀
神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内
- (72)発明者 村上 公一
神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内

審査官 殿川 雅也

- (56)参考文献 特開平05-012040(JP,A)
特開平10-133890(JP,A)

- (58)調査した分野(Int.Cl., DB名)
G06F 9/46 - 9/54