

(12) 特許協力条約に基づいて公開された国際出願

(19) 世界知的所有権機関
国際事務局

(43) 国際公開日
2012年8月16日(16.08.2012)



(10) 国際公開番号

WO 2012/108058 A1

(51) 国際特許分類:
G06F 9/48 (2006.01)

(21) 国際出願番号: PCT/JP2011/052953

(22) 国際出願日: 2011年2月10日(10.02.2011)

(25) 国際出願の言語: 日本語

(26) 国際公開の言語: 日本語

(71) 出願人(米国を除く全ての指定国について): 富士通株式会社(FUJITSU LIMITED) [JP/JP]; 〒2118588 神奈川県川崎市中原区上小田中4丁目1番1号 Kanagawa (JP).

(72) 発明者; および

(75) 発明者/出願人(米国についてのみ): 鈴木 貴久 (SUZUKI, Takahisa) [JP/JP]; 〒2118588 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 Kanagawa (JP). 山下 浩一郎 (YAMASHITA, Koichiro) [JP/JP]; 〒2118588 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 Kanagawa (JP). 山内 宏真 (YAMAUCHI, Hiromasa) [JP/JP]; 〒2118588 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 Kanagawa (JP). 栗原 康志 (KURIHARA, Koji) [JP/JP]; 〒2118588 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 Kanagawa (JP). 大友 俊也 (OTOMO, Toshiya) [JP/JP]; 〒2118588 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 Kanagawa (JP). 大舘 尚記 (ODATE, Naoki) [JP/JP]; 〒2118588 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 Kanagawa (JP). 平木 哲夫 (HIRAKI, Tetsuo) [JP/JP]; 〒2118588 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 Kanagawa (JP).

(74) 代理人: 酒井 昭徳 (SAKAI, Akinori); 〒1006020 東京都千代田区霞が関3丁目2番5号 霞が関ビルディング20階 酒井総合特許事務所 Tokyo (JP).

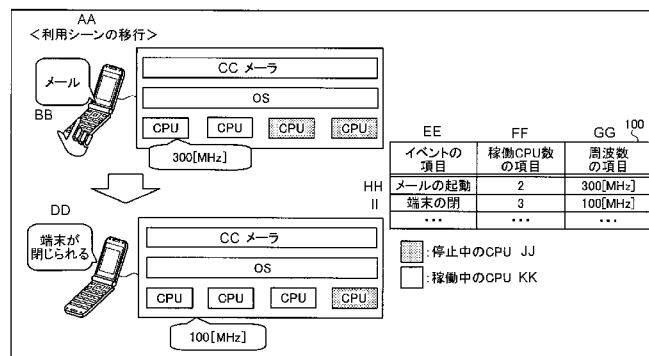
(81) 指定国(表示のない限り、全ての種類の国内保護が可能): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PE, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

[続葉有]

(54) Title: SCHEDULING METHOD, DESIGN SUPPORT METHOD, AND SYSTEM

(54) 発明の名称: スケジューリング方法、設計支援方法、およびシステム

[図1]



AA <Transition of usage scene>
BB Mail
CC Mailer
DD Terminal is closed
EE Event category
FF Number-of-operating-CPU category

GG Frequency category
HH Startup of mail
II Closing of terminal
JJ Stopped CPU
KK Operational CPU

(57) Abstract: In the present invention, a number of operating CPUs and a frequency of a clock are stored in a table for each scene of usage. An OS acquires the number of operating CPUs and the frequency of the clock from the table each time that the scene of usage changes. For example, during mail, the number of operating CPUs is set to 2, and the frequency of the clock supplied to the operating CPUs is set to 300 [MHz]. Then, when the terminal is closed during mail, the OS acquires from the table the number of operating CPUs and the frequency of the clock corresponding to the event in which the terminal is closed. The number of operating CPUs is 3 and the frequency is 100 [MHz]. The OS supplies a clock of the frequency that has been acquired to the CPUs of the number of operating CPUs in order to execute the program being executed with the CPUs of the number of operating CPUs.

(57) 要約:

[続葉有]



WO 2012/108058 A1



(84) 指定国 (表示のない限り、全ての種類の広域保護が可能): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), ユーラシア (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), ヨーロッパ (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI

(BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

添付公開書類:

— 国際調査報告 (条約第 21 条(3))

利用シーンごとに稼働CPU数とクロックの周波数とがテーブルに記憶されている。OSが、利用シーンが切り替わるごとにテーブルから稼働CPU数とクロックの周波数を取得する。たとえば、メール中には、稼働CPU数を2とし、稼働中のCPUに供給するクロックの周波数を300 [MHz] とする。そして、メール中に端末が閉じられると、OSが、テーブルから端末が閉じられるイベントに対応する稼働CPU数とクロックの周波数を取得する。稼働CPU数は3であり、周波数は100 [MHz] である。OSが、稼働CPU数分のCPUに取得した周波数のクロックを供給し、稼働CPU数分のCPUで実行中のプログラムを実行する。

明 細 書

発明の名称：

スケジューリング方法、設計支援方法、およびシステム

技術分野

[0001] 本発明は、プログラムの実行をスケジューリングするスケジューリング方法、およびシステムに関する。また、本発明は、システムの設計を支援する設計支援方法に関する。

背景技術

[0002] 従来、マルチコアプロセッサシステムにおいて、負荷ピーク時の発熱量を抑制させる技術として、単位時間当たりの消費電力が均一になるようにスケジューリングする技術が知られている（たとえば、下記特許文献1参照。）。

[0003] また、従来、マルチコアプロセッサシステムにおいて、消費電力を削減する技術として、CPU（Central Processing Unit）へ供給するクロック周波数と電源電圧を動的に変更するDVFS（Dynamic Voltage Frequency Scaling）が知られている。さらに、マルチコアプロセッサでは、ある処理を複数のCPUに分散して実行することにより、処理時間を高速化することができる。

[0004] そこで、マルチコアプロセッサシステムによる処理時間が稼働CPU数に比例するとして、稼働CPU数ごとに処理時間と消費電力を算出し、最適な稼働CPU数と電源電圧の値とクロックの周波数を決定する技術（以下、「従来技術1」と称する。）が知られている（たとえば、下記特許文献2参照。）。しかしながら、マルチコアプロセッサシステムを用いて処理を並列化すると、実際には並列化オーバーヘッドが存在するため、稼働CPU数に比例して、処理時間を高速化させることができない。

[0005] 図23は、並列化オーバーヘッド例を示す説明図である。並列化オーバーヘッドの要因には大きく二つある。一つはプログラム全体が並列に動作でき

るわけではないためである。たとえば、プログラムには、並列化不可部分と並列化可能部分とがある。たとえば、1 CPUで動かした時の実行時間のうち並列化不可部分が10 [%] があると、CPUが複数あったとしても、並列動作不可能な10 [%] の部分が邪魔して10倍以上の性能は出ないことになる。

[0006] もう一つは、並列処理をおこなう際に一つの処理を分割して複数のCPUでおこなうと、CPUをまたがった処理の間で同期や通信をおこなう必要があるためである（図23では、同期／通信部分である。）。1 CPUの実行では不要だった同期や通信の処理が2 CPU以上で実行する場合には追加される。さらに、2 CPU以上で実行する場合には、一のCPUが並列化不可部分の実行中には他のCPUはアイドル処理となる（アイドル部分）。

[0007] 図24は、並列化オーバーヘッドの影響例を示す説明図である。グラフでは、縦軸が性能であり、横軸がCPU数である。ここでは、稼働CPU数が1のときの性能を1とした場合に、稼働CPU数の増加によってどのくらい性能が向上するかを示している。ここでの性能とは処理時間である。たとえば、処理時間が40 [ms] から20 [ms] になると、性能が2倍になったことを示す。グラフ中の理想値では、稼働CPU数の増加と性能の増加が比例している。しかしながら、グラフ中の実性能では、稼働CPU数が増加すればするほど、性能の向上が鈍化する。

[0008] そこで、アプリケーションを解析して各CPUの空き時間を特定し、最高性能を維持しつつDVFSを用いてCPUへ供給するクロックの周波数を変更することで、消費電力を削減する技術（以下、「従来技術2」と称する。）が知られている（たとえば、下記特許文献3参照。）。たとえば、従来技術2では、処理A、処理B、処理Cの三つの処理があり、処理Cが処理Aと処理Bの結果を利用するため、この二つの処理が終了していないと開始できないとする。このとき、基準周波数では処理Aが5秒、処理Bが10秒で終了するとした場合、処理Aと処理Bを別々のCPUで実行したとして、処理Aを実行するCPUへ供給する処理Aを実行中のクロックの周波数を基準周

波数の半分にしても処理Cの開始時刻は変わらない。よって、DVFSにより消費電力の削減が可能となる。

先行技術文献

特許文献

[0009] 特許文献1：特許第3567354号公報

特許文献2：特開2005-85164号公報

特許文献3：特開2006-293768号公報

発明の概要

発明が解決しようとする課題

[0010] しかしながら、従来技術2では、複数のアプリケーションが同時に実行された場合、どのように空き時間が変化するか不明であるため、クロックの周波数を変更することができない問題点があった。また、携帯電話を例に挙げると、利用者がメールを長時間掛けて作成している最中などには性能はあまり重要ではなく、プロセッサをまたがった同期や通信によるオーバーヘッドを考慮すると並列化を行わない方がかえって効率が良い可能性がある。しかしながら、従来技術2ではそのような点は考慮されておらず、全てのCPUを最大限活用して並列処理を行おうとするため、従来技術2を用いると消費電力が増加する問題点があった。

[0011] 本発明は、上述した従来技術による問題点を解消するため、一定以上の性能を保ちつつ、消費電力を減少させることができるスケジューリング方法、およびシステムを提供することを目的とする。また、本発明は、消費電力が最も少ない稼働CPU数とクロック周波数の組み合わせを容易に特定することができ、システムの設計期間を短縮化することができる設計支援方法を提供することを目的とする。

課題を解決するための手段

[0012] 上述した課題を解決し、目的を達成するため、本発明の一の側面によれば、第1処理から第2処理への変更を検出し、メモリから前記第2処理を実行

するCPUの数と動作周波数とを取得し、前記CPUの数に基づいて、稼働中のCPUを停止しまたは停止中のCPUを起動し、前記第2処理を実行するCPUに前記動作周波数を割り当てるスケジューリング方法、およびシステムが提案される。

- [0013] また、本発明の他の側面によれば、第1処理を実行するときの、第1数個のCPUの第1動作時間と第1停止時間とを測定し、第1最小動作周波数より大きい第1動作周波数を設定し、前記第1動作時間と前記第1停止時間とに基づいて、前記第1動作周波数で動作する前記第1数個のCPUの第1消費電力を算出し、前記第1処理を実行するときの、前記第1数個とは異なる第2数個のCPUの第2動作時間と第2停止時間とを測定して、第2最小動作周波数より大きい第2動作周波数を設定し、前記第2動作時間と前記第2停止時間とに基づいて、前記第2動作周波数で動作する前記第2数個のCPUの第2消費電力を算出し、前記第1消費電力と前記第2消費電力との比較結果に基づいて、前記第1処理実行時のCPUの数を決定する設計支援方法が提案される。

発明の効果

- [0014] 本発明の一の側面によれば、一定以上の性能を保ちつつ、消費電力を減少させることができるという効果を奏する。また、本発明の他の側面によれば、消費電力が最も少ない稼働CPU数とクロック周波数の組み合わせを容易に特定することができるという効果を奏する。

図面の簡単な説明

- [0015] [図1]図1は、本発明の一例を示す説明図である。
- [図2]図2は、マルチコアプロセッサシステムのハードウェア例を示す説明図である。
- [図3]図3は、実施の形態1にかかる設計支援装置のハードウェア例を示すブロック図である。
- [図4]図4は、実施の形態1にかかる設計支援装置300の機能ブロック図を示す説明図である。

[図5] 図5は、測定データの一例を示す説明図である。

[図6] 図6は、DVFS制御情報テーブル600の一例を示す説明図である。

[図7] 図7は、実施の形態1にかかる設計支援装置300による設計支援処理手順の一例を示すフローチャートである。

[図8] 図8は、実施の形態2にかかるマルチコアプロセッサシステム200の機能ブロック図である。

[図9] 図9は、イベントの発生例を示す説明図である。

[図10] 図10は、稼働中のCPUの数が増加する例を示す説明図である。

[図11] 図11は、実施の形態2にかかるOS220による制御処理手順の一例を示すフローチャートである。

[図12] 図12は、設計支援装置300の機能ブロック図を示す説明図である。

[図13] 図13は、稼働時間テーブル例を示す説明図である。

[図14] 図14は、周波数・電力テーブル例を示す説明図である。

[図15] 図15は、実施の形態3にかかる設計支援装置300による設計支援処理手順の一例を示すフローチャートである。

[図16] 図16は、実施の形態4にかかるマルチコアプロセッサシステム200の機能ブロック図である。

[図17] 図17は、複数のプログラムが同時に実行される例を示す説明図である。

[図18] 図18は、合計稼働時間を示す説明図である。

[図19] 図19は、消費電力が最も低くなるCPU数を特定する例を示す説明図である。

[図20] 図20は、稼働CPU数の変更例を示す説明図である。

[図21] 図21は、実施の形態4にかかるOS220による制御処理手順の一例を示すフローチャートである。

[図22] 図22は、実施の形態4にかかるOS220による制御処理手順の一例を示すフローチャートである。

[図23] 図 2 3 は、並列化オーバーヘッド例を示す説明図である。

[図24] 図 2 4 は、並列化オーバーヘッドの影響例を示す説明図である。

発明を実施するための形態

[0016] 以下に添付図面を参照して、本発明にかかるスケジューリング方法、設計支援方法、およびシステムの実施の形態を詳細に説明する。ここで、マルチコアプロセッサシステムにおいて、マルチコアプロセッサとは、コアが複数搭載されたプロセッサである。コアが複数搭載されていれば、複数のコアが搭載された単一のプロセッサでもよく、シングルコアのプロセッサが並列されているプロセッサ群でもよい。なお、本実施の形態では、説明を単純化するため、シングルコアのプロセッサが並列されているプロセッサ群を例に挙げて説明する。

[0017] 図 1 は、本発明の一例を示す説明図である。たとえば、利用シーンごとに稼働 CPU 数とクロックの周波数とがテーブル 100 に記憶されている。ここで、利用シーンとは、たとえば、利用者が動画再生や音楽再生といった特定の処理の起動や、通話やメールの着信といった機器の主要な機能の実行や、機器の開閉などの所定のイベントを実行している状態である。その機器がどのような主要機能を持つかは機器の企画段階で決められるので、主要機能に基づいて機器の利用シーンが決定される。OS が、利用シーンが切り替わるとともにテーブル 100 から稼働 CPU 数とクロックの周波数を取得する。

[0018] たとえば、メール中には、稼働 CPU 数を 2 とし、稼働中の CPU に供給するクロックの周波数を 300 [MHz] とする。そして、メール中に端末が閉じられると、OS が、テーブル 100 から端末が閉じられるイベントに対応する稼働 CPU 数とクロックの周波数を取得する。稼働 CPU 数は 3 であり、周波数は 100 [MHz] である。OS が、稼働 CPU 数分の CPU に取得した周波数のクロックを供給し、稼働 CPU 数分の CPU で実行中のプログラムを実行する。

[0019] ここでは、実施の形態 1 ～実施の形態 4 を用いて詳細に説明する。実施の形態 1, 2 では、利用シーンが変化するとともに、変化後の利用シーンに応じ

たCPU数とクロックの周波数で動作させる例を示す。実施の形態3、4では、複数のプログラムが同時に実行された場合に、実行中のプログラムの組み合わせに応じて最適なCPU数とクロックの周波数を決定する例を示す。

[0020] (実施の形態1)

まず、実施の形態1では、利用シーンに応じたCPU数とクロックの周波数を設計時に特定する例を示す。

[0021] (マルチコアプロセッサシステムのハードウェア例)

図2は、マルチコアプロセッサシステムのハードウェア例を示す説明図である。マルチコアプロセッサシステム200は、CPU201~CPU204と、DVFS制御機構205と、RAM(Random Access Memory)206と、ROM(Read Only Memory)207と、フラッシュROM208と、を有している。さらに、マルチコアプロセッサシステム200は、フラッシュROMコントローラ209と、フラッシュROM210と、ディスプレイ211と、キーボード212と、I/F(Interface)213と、を有している。各部は、バス214を介して接続されている。

[0022] まず、CPU201~CPU204は、それぞれレジスタとコアとキャッシュとを有している。CPU201~CPU204はSMP(Symmetric Multiprocessing)型のOS220を実行する。SMP型のOS220では、OS220の内部処理もOS220上で動くアプリケーションプログラムも一部の例外を除いて、理論的にはすべてのCPUにまたがって実行される。各処理はマルチコアプロセッサ上のいずれかのCPUで実行されるが、どのCPUで実行されるかは意識する必要がない。

[0023] そのため、SMP型のOS220では、稼働中のCPUの数を意識する必要が無く、OS220が稼働中のCPUに適切に処理を割り振るため、ソフトウェアは特に変更を加えなくても様々なCPU数で機器を動作させることができる。実際には、OS220の中核部分だけがCPUごとに独立して動作し、中核部分によりCPU間で通信しながらどのCPUでどの処理をおこ

なうかを決定している。

- [0024] ROM207と、RAM206と、フラッシュROM208と、フラッシュROMコントローラ209と、フラッシュROM210とは、CPU201～CPU204に共有されるメモリである。
- [0025] フラッシュROM208やROM207には、ブートシーケンスが記述されたブートローダなどのプログラムが記憶されている。フラッシュROM208やROM207には、OSなどのシステムソフトウェアやアプリケーションソフトウェアやOS220が制御するテーブルが記憶されている。
- [0026] RAM206は、各CPUのワークエリアとして使用される。フラッシュROMコントローラ209は、各CPUの制御にしたがってフラッシュROM210に対するデータのリード／ライトを制御する。フラッシュROM210は、フラッシュROMコントローラ209の制御で書き込まれたデータを記憶する。データの具体例としては、マルチコアプロセッサシステム200を使用するユーザがI/F213を通して取得した画像データ、映像データなどである。フラッシュROM210は、たとえば、メモリカード、SDカードなどを採用することができる。
- [0027] DVFS制御機構205は、各CPUに電源電圧やクロックを供給する。たとえば、DVFS制御機構205がCPU201～CPU204へ供給可能な電源電圧の値は0.1[V]刻みで1.0[V]～1.6[V]と0[V]である。DVFS制御機構205からCPU201～CPU204へそれぞれ電源配線VDD1～VDD4を介して電源電圧が供給される。DVFS制御機構205がCPU201～CPU204へ供給可能なクロックの周波数は100[MHz]刻みで100[MHz]～500[MHz]と0[Hz]である。DVFS制御機構205からCPU201～CPU204へそれぞれクロック配線CLK1～CLK4を介してクロックが供給される。供給される電源電圧が0[V]またはクロックの周波数が0[Hz]の場合、CPUは停止状態となる。
- [0028] ディスプレイ211は、カーソル、アイコンあるいはツールボックスをは

じめ、文書、画像、機能情報などのデータを表示する。さらに、ディスプレイ 211 は、タッチパネルであり、数字、各種指示などの入力のためのキーを有し、データの入力をおこなってもよい。ディスプレイ 211 は、たとえば、TFT 液晶ディスプレイ 211 などを採用することができる。キーボード 212 は、数字、各種指示などの入力のためのキーを有し、データの入力をおこなう。また、キーボード 212 は、タッチパネル式の入力パッドやテンキーなどであってもよい。

[0029] I/F 213 は、通信回線を通じて LAN (Local Area Network)、WAN (Wide Area Network)、インターネットなどのネットワークに接続され、ネットワークを介して他の装置に接続される。そして、I/F 213 は、ネットワークと内部のインターフェースを司り、外部装置からのデータの入出力を制御する。I/F 213 には、たとえば、モデムや LAN アダプタなどを採用することができる。

[0030] (設計支援装置のハードウェア例)

図 3 は、実施の形態 1 にかかる設計支援装置のハードウェア例を示すブロック図である。設計支援装置 300 は、マルチコアプロセッサシステム 200 を有する機器に接続可能であって、処理時間や消費電力を計測する。図 3 において、設計支援装置 300 は、CPU 301 と、ROM 302 と、RAM 303 と、磁気ディスクドライブ 304 と、磁気ディスク 305 と、光ディスクドライブ 306 と、光ディスク 307 と、ディスプレイ 308 と、I/F 309 と、キーボード 310 と、マウス 311 と、スキャナ 312 と、プリンタ 313 と、を備えている。また、各部はバス 315 によってそれぞれ接続されている。

[0031] ここで、CPU 301 は、設計支援装置 300 の全体の制御を司る。ROM 302 は、ブートプログラムなどのプログラムを記憶している。RAM 303 は、CPU 301 のワークエリアとして使用される。磁気ディスクドライブ 304 は、CPU 301 の制御にしたがって磁気ディスク 305 に対するデータのリード/ライトを制御する。磁気ディスク 305 は、磁気ディス

クドライブ 304 の制御で書き込まれたデータを記憶する。

[0032] 光ディスクドライブ 306 は、CPU 301 の制御にしたがって光ディスク 307 に対するデータのリード／ライトを制御する。光ディスク 307 は、光ディスクドライブ 306 の制御で書き込まれたデータを記憶したり、光ディスク 307 に記憶されたデータをコンピュータに読み取らせたりする。

[0033] ディスプレイ 308 は、カーソル、アイコンあるいはツールボックスをはじめ、文書、画像、機能情報などのデータを表示する。このディスプレイ 308 は、たとえば、CRT、TFT 液晶ディスプレイ、プラズマディスプレイなどを採用することができる。

[0034] I/F 309 は、通信回線を通じて LAN (Local Area Network)、WAN (Wide Area Network)、インターネットなどのネットワーク 314 に接続され、このネットワーク 314 を介して他の装置に接続される。そして、I/F 309 は、ネットワーク 314 と内部のインターフェースを司り、外部装置からのデータの入出力を制御する。I/F 309 には、たとえばモデムや LAN アダプタなどを採用することができる。

[0035] キーボード 310 は、文字、数字、各種指示などの入力のためのキーを備え、データの入力をおこなう。また、タッチパネル式の入力パッドやテンキーなどであってもよい。マウス 311 は、カーソルの移動や範囲選択、あるいはウィンドウの移動やサイズの変更などをおこなう。ポインティングデバイスとして同様に機能を備えるものであれば、トラックボールやジョイスティックなどであってもよい。

[0036] スキャナ 312 は、画像を光学的に読み取り、設計支援装置 300 内に画像データを取り込む。なお、スキャナ 312 は、OCR (Optical Character Reader) 機能を持たせてもよい。また、プリンタ 313 は、画像データや文書データを印刷する。プリンタ 313 には、たとえば、レーザプリンタやインクジェットプリンタを採用することができる。

[0037] (実施の形態 1 にかかる設計支援装置 300 の機能ブロック図)

図 4 は、実施の形態 1 にかかる設計支援装置 300 の機能ブロック図を示す説明図である。設計支援装置 300 は、測定部 401 と、周波数算出部 402 と、消費電力算出部 403 と、決定部 404 と、出力部 405 と、を有している。具体的には、たとえば、測定部 401 ～出力部 405 を有するプログラムが ROM 302、磁気ディスク 305、光ディスク 307 などの記憶装置に記憶されている。CPU 301 が該記憶装置にアクセスして該プログラムを読み出し、該プログラム内にコーディングされている処理を実行することにより、測定部 401 ～出力部 405 の処理が実行される。

[0038] 測定部 401 は、マルチコアプロセッサシステム 200 を有する機器で対象イベントを発生させ、稼働 CPU 数ごとに基準周波数における対象イベントに応じた処理を実行時の稼働時間と延べ空き時間を測定する。空き時間とは OS 220 が各 CPU に割り振る処理が無い状態の時間のことで、マルチコアプロセッサシステム 200 の場合、空き時間はそれぞれの CPU でばらばらに発生する。そのため、測定部 401 は、典型的な処理をしている間に稼働しているすべての CPU で発生する空き時間の合計を延べ空き時間として測定する。マルチコアプロセッサシステム 200 において稼働時間と空き時間は OS 220 が管理しているので、設計支援装置 300 は OS 220 に稼働時間と延べ空き時間を測定させる。

[0039] 図 5 は、測定データの一例を示す説明図である。利用シーン 1 測定データ 501 は、CPU 数の項目と稼働時間の項目と待ち時間の項目とを有している。マルチコアプロセッサシステム 200 の CPU が 4 個であるため、CPU 数の項目には 1 ～ 4 が登録されている。稼働時間の項目には利用シーン 1 での各 CPU 数での稼働時間が登録され、待ち時間の項目には利用シーン 1 での各 CPU 数での空き時間が登録される。

[0040] 利用シーン 2 測定データ 502 は、CPU 数の項目と稼働時間の項目と待ち時間の項目とを有している。マルチコアプロセッサシステム 200 の CPU が 4 個であるため、CPU 数の項目には 1 ～ 4 が登録されている。稼働時

間の項目には利用シーン２での各ＣＰＵ数での稼働時間が登録され、待ち時間の項目には利用シーン２での各ＣＰＵ数での空き時間が登録される。利用シーン１測定データ５０１や利用シーン２測定データ５０２はＲＯＭ３０２、磁気ディスク３０５、光ディスク３０７などの記憶装置に記憶される。

[0041] 図４に戻って、周波数算出部４０２は、測定部４０１により測定された指定ＣＰＵ数での稼働時間と測定部４０１により測定されたＣＰＵ数ごとの稼働時間との比率により、指定ＣＰＵ数での稼働時間を満たすことができる周波数をＣＰＵ数ごとに算出する。ここでは、指定ＣＰＵ数を１とする。具体的には、たとえば、ＣＰＵが、下記式（１）を用いてＣＰＵ数ごとに周波数を算出する。

[0042] 周波数＝基準周波数×測定した稼働時間／指定ＣＰＵ数での稼働時間・・・（１）

[0043] 消費電力算出部４０３は、周波数算出部４０２によりＣＰＵ数ごとに算出された周波数に応じた記憶装置に記憶された単位時間当たりの１ＣＰＵの消費電力を用いて、ＣＰＵ数ごとに消費電力量を算出する。決定部４０４は、消費電力算出部４０３によりＣＰＵ数ごとに算出された消費電力量の中で最小の消費電力量であるＣＰＵ数を対象イベントの稼働ＣＰＵ数に決定する。出力部４０５は、決定部４０４により決定された稼働ＣＰＵ数と算出された周波数とを対象イベントの識別情報に関連付けて出力する。

[0044] 図６は、ＤＶＦＳ制御情報テーブル６００の一例を示す説明図である。ＤＶＦＳ制御情報テーブル６００は、出力結果の一例である。ＤＶＦＳ制御情報テーブル６００には、指定されたイベントごとの稼働ＣＰＵ数と周波数と電源電圧の値が定義されている。ＤＶＦＳ制御情報テーブル６００は、イベントの項目６０１と、稼働ＣＰＵ数の項目６０２と、周波数の項目６０３と、電源電圧の項目６０４と、を有している。

[0045] イベントの項目６０１には、利用シーン移行条件が登録される。ここでは、イベントの項目６０１には、メーラの起動、動画再生ソフトウェアの起動、ブラウザの起動、端末の閉、キーＹの押下が登録されている。稼働ＣＰＵ

数の項目602には稼働CPU数が登録されている。周波数の項目603には稼働CPUへ供給するクロックの周波数が登録されている。電源電圧の項目604には稼働中のCPUへ供給する電源電圧が登録されている。

[0046] (実施の形態1にかかる設計支援装置300による設計支援処理手順)

図7は、実施の形態1にかかる設計支援装置300による設計支援処理手順の一例を示すフローチャートである。まず、設計支援装置300が、稼働CPU数が未決定な利用シーンから任意の利用シーンを設定し(ステップS701)、稼働CPU数=1とし(ステップS702)、稼働CPU数と利用シーンを設定して機器を起動する(ステップS703)。ここで、機器は、上述のマルチコアプロセッサシステム200を有している。設計支援装置300が、稼働時間と延べ空き時間を測定し(ステップS704)、シングルコアでの性能を下回らない最小の周波数を算出する(ステップS705)。

[0047] 設計支援装置300が、算出した周波数でCPUが動作する最小の電源電圧と算出した周波数での消費電力を測定し(ステップS706)、算出した周波数での消費電力量を算出する(ステップS707)。設計支援装置300が、稼働CPU数=稼働CPU数+1とし(ステップS708)、稼働CPU数>全CPU数であるか否かを判断する(ステップS709)。設計支援装置300が、稼働CPU数>全CPU数でないと判断した場合(ステップS709:No)、ステップS702へ戻る。

[0048] 設計支援装置300が、稼働CPU数>全CPU数であると判断した場合(ステップS709:Yes)、消費電力量が最小となるCPU数を決定し(ステップS710)、全利用シーンでCPU数を決定したか否かを判断する(ステップS711)。設計支援装置300が、全利用シーンでCPU数を決定していないと判断した場合(ステップS711:No)、ステップS701へ戻る。

[0049] 設計支援装置300が、全利用シーンでCPU数を決定したと判断した場合(ステップS711:Yes)、決定結果を出力し(ステップS712)

、一連の処理を終了する。出力形式としては、たとえば、ディスプレイ 308 への表示、プリンタ 313 への印刷出力、I/F 309 による外部装置への送信がある。また、RAM 303、磁気ディスク 305、光ディスク 307 などの記憶領域に記憶することとしてもよい。

[0050] (実施の形態 2)

まず、実施の形態 2 では、実施の形態 1 で決定した各利用シーンでの稼働 CPU 数とクロック周波数を用いて、利用シーンが切り替わる都度、該利用シーンに応じた稼働 CPU 数とクロックの周波数で利用シーンに応じた処理を動作させる例を示す。

[0051] (実施の形態 2 にかかるマルチコアプロセッサシステム 200 の機能ブロック図)

図 8 は、実施の形態 2 にかかるマルチコアプロセッサシステム 200 の機能ブロック図である。実施の形態 2 にかかるマルチコアプロセッサシステム 200 は、記憶部 801 と、イベント検出部 802 と、シーン決定部 803 と、DVFS 制御部 804 と、スケジューリング部 805 と、を有している。

[0052] 具体的には、たとえば、イベント検出部 802 ~ スケジューリング部 805 を有するプログラムが ROM 207 などの記憶装置に記憶されている。CPU が該記憶装置にアクセスして該プログラムを読み出し、該プログラム内にコーディングされている処理を実行することにより、イベント検出部 802 ~ スケジューリング部 805 の処理が実行される。ここでは、該プログラムは OS 220 である。

[0053] 記憶部 801 は、複数のイベントのイベントごとに指定数の CPU で実行した場合の性能を保ち、消費電力量が最も少なくなる CPU 数と周波数と電源電圧の値を記憶する。具体的には、たとえば、上述の DVFS 制御情報テーブル 600 が ROM 207 やフラッシュ ROM 208 に記憶されている。

[0054] イベント検出部 802 はイベントを検出し、シーン決定部 803 は、イベント検出部 802 により検出されたイベントが DVFS 制御情報テーブル 6

00に登録されているイベントに含まれているか否かを判断する。イベント検出部802により検出されたイベント（対象イベント）がDVFS制御情報テーブル600に登録されているイベントであると判断された場合、DVFS制御部804は、記憶部801に記憶された対象イベントに関する稼働CPU数を取得する。DVFS制御部804は、記憶部801に記憶された対象イベントに関する周波数を取得する。DVFS制御部804は、取得した稼働CPU数分のCPUへ取得した周波数のクロックを供給する。

[0055] DVFS制御部804は、マルチコアプロセッサのうちの稼働中のCPUの数が稼働CPU数分のCPUより多い場合、稼働中のCPUから稼働CPU数を超える数分のCPUを停止させる。DVFS制御部804は、マルチコアプロセッサのうちの稼働中のCPUの数が稼働CPU数分のCPUより少ない場合、不足分のCPUを起動させる。そして、スケジューリング部805は、稼働CPU数分のCPUで対象イベントに応じた処理を実行する。以上を踏まえて詳細な具体例を説明する。

[0056] 図9は、イベントの発生例を示す説明図である。図9では、CPU201が稼働中であるが、CPU202～CPU204は停止中である。ユーザがメーラを起動すると、OS220が（1）メーラの起動指示を検出する。OS220が、（2）DVFS制御テーブル600のイベントの項目601にメーラの起動が登録されているかを確認する。DVFS制御テーブル600のイベントの項目601にメーラの起動が登録されているため、OS220が利用シーン移行条件を満たしたと判断する。

[0057] OS220が、（3）DVFS制御情報テーブル600からメーラ起動に関する稼働CPU数とクロックの周波数と電源電圧の値とを取得する。OS220が、現在稼働中のCPUの数と取得した稼働CPU数とを比較することで、稼働中のCPUの数に変更が必要か否かを判断する。現在稼働中のCPUの数は1であり、取得した稼働CPU数は2である。よって、不足するCPU数は1である。

[0058] OS220が、稼働中のCPUの数を1つ増加させるため、停止中のCP

Uから起動させるCPUを決定する。ここでは、CPU202が起動するCPUに決定される。OS220が、(4)取得した電源電圧の値と取得したクロックの周波数とをCPU201とCPU202に与えるようにDVFS制御機構205を制御する。

[0059] 図10は、稼働中のCPUの数が増える例を示す説明図である。電源配線VDD1と電源配線VDD2の電源電圧の値が1.3[V]であり、クロック配線CLK1とクロック配線CLK2のクロックの周波数が300[MHz]である。電源配線VDD3と電源配線VDD4の電源電圧の値が0[V]であり、クロック配線CLK3とクロック配線CLK4のクロックの周波数が0[Hz]である。そして、OS220がメーラを起動する。

[0060] (実施の形態2にかかるOS220による制御処理手順)

図11は、実施の形態2にかかるOS220による制御処理手順の一例を示すフローチャートである。まず、OS220が、イベントを検出したか否かを判断する(ステップS1101)。OS220が、イベントを検出していないと判断した場合(ステップS1101:No)、ステップS1101へ戻る。

[0061] OS220が、イベントを検出したと判断した場合(ステップS1101:Yes)、DVFS制御情報テーブル600の利用シーン移行条件をチェックする(ステップS1102)。OS220が、検出したイベントが利用シーン移行条件に合致するか否かを判断し(ステップS1103)、合致しないと判断した場合(ステップS1103:No)、ステップS1112へ移行する。

[0062] OS220が、検出したイベントが利用シーン移行条件に合致すると判断した場合(ステップS1103:Yes)、DVFS制御情報テーブル600から検出したイベントに対応する稼働CPU数、クロックの周波数、電源電圧の値を取得する(ステップS1104)。つぎに、OS220が、稼働CPU数に変更があるか否かを判断し(ステップS1105)、稼働CPU数に変更がないと判断した場合(ステップS1105:No)、ステップS

1 1 1 1 へ移行する。

[0063] OS 2 2 0 が、稼働 CPU 数に変更があると判断した場合（ステップ S 1 1 0 5 : Y e s）、稼働 CPU 数が増加したか否かを判断する（ステップ S 1 1 0 6）。OS 2 2 0 が、稼働 CPU 数が増加したと判断した場合（ステップ S 1 1 0 6 : Y e s）、停止中の CPU から起動させる CPU（起動 CPU）を決定する（ステップ S 1 1 0 7）。OS 2 2 0 が、起動 CPU と稼働中の CPU に取得した電源電圧の値と周波数のクロックとを供給させる制御をおこない（ステップ S 1 1 0 8）、ステップ S 1 1 1 2 へ移行する。

[0064] OS 2 2 0 が、稼働 CPU 数が増加していないと判断した場合（ステップ S 1 1 0 6 : N o）、停止させる CPU（停止 CPU）を決定し（ステップ S 1 1 0 9）、停止 CPU の電源供給、周波数入力を停止させる制御をおこなう（ステップ S 1 1 1 0）。OS 2 2 0 が、稼働中の CPU に取得した電源電圧の値と周波数のクロックとを供給させる制御をおこなう（ステップ S 1 1 1 1）。ステップ S 1 1 0 3 の N o の場合、ステップ S 1 1 0 8、またはステップ S 1 1 1 1 のつぎに、OS 2 2 0 が、検出したイベントに応じた処理を稼働中の CPU で実行し（ステップ S 1 1 1 2）、ステップ S 1 1 0 1 へ移行する。

[0065] （実施の形態 3）

実施の形態 3 では、プログラムごとに基準周波数における CPU 数ごとの処理時間を測定し、周波数ごとに単位時間当たりの 1 コアの消費電力量を測定する例を示す。

[0066] （実施の形態 3 にかかる設計支援装置 3 0 0 の機能ブロック図）

図 1 2 は、設計支援装置 3 0 0 の機能ブロック図を示す説明図である。設計支援装置 3 0 0 は測定部 1 2 0 1 と出力部 1 2 0 2 とを有している。具体的には、たとえば、測定部 1 2 0 1 と出力部 1 2 0 2 を有するプログラムが ROM 3 0 2、磁気ディスク 3 0 5、や光ディスク 3 0 7 などの記憶装置に記憶されている。CPU 3 0 1 が該記憶装置にアクセスして該プログラムを読み出し、該プログラム内にコーディングされている処理を実行することに

より、測定部 1201 と出力部 1202 の処理が実行される。

[0067] 測定部 1201 は、マルチコアプロセッサシステム 200 を有する機器を用いて、プログラムごとに基準周波数における CPU 数ごとの稼働時間を測定する。出力部 1202 は、測定結果とプログラムの識別情報を関連付けて出力する。図 13 に出力例を示す。

[0068] 図 13 は、稼働時間テーブル例を示す説明図である。稼働時間テーブル 1300 は、たとえば、メーラに関する稼働時間が登録されたテーブルであり、CPU 数ごとに該 CPU 数でメーラを実行した際にかかる稼働時間が登録されている。稼働時間テーブル 1310 は、たとえば、ブラウザに関する稼働時間が登録されたテーブルであり、CPU 数ごとに該 CPU 数でメーラを実行した際にかかる稼働時間が登録されている。

[0069] 稼働時間テーブル 1300 は、CPU 数の項目 1301 と、稼働時間の項目 1302 と、空き時間の項目 1303 と、を有している。CPU 数の項目 1301 には 1 ～ 4 までは登録されている。稼働時間の項目 1302 には、CPU 数の項目 1301 に登録された CPU 数ごとに該 CPU 数でメーラを実行した際にかかる単位時間当たりの稼働時間が登録されている。空き時間の項目 1303 には、CPU 数の項目 1301 に登録された CPU 数ごとに該 CPU 数分の CPU でメーラを実行した際に発生する延べ空き時間が登録されている。

[0070] 稼働時間テーブル 1310 は、CPU 数の項目 1311 と、稼働時間の項目 1312 と、空き時間の項目 1313 と、を有している。CPU 数の項目 1311 には 1 ～ 4 までは登録されている。稼働時間の項目 1312 には、CPU 数の項目 1311 に登録された CPU 数ごとに該 CPU 数でブラウザを実行した際にかかる単位時間当たりの稼働時間が登録されている。空き時間の項目 1313 には、CPU 数の項目 1311 に登録された CPU 数ごとに該 CPU 数分の CPU でブラウザを実行した際に発生する延べ空き時間が登録されている。

[0071] 図 12 に戻って、測定部 1201 は、周波数ごとに単位時間当たりの 1 C

P Uの消費電力量を測定する。出力部 1 2 0 3は、測定結果を出力する。図 1 4に出力例を示す。

[0072] 図 1 4は、周波数・電力テーブル例を示す説明図である。周波数・電力テーブル 1 4 0 0は、周波数の項目 1 4 0 1と、電源電圧の項目 1 4 0 2と、1 C P U当たりの消費電力の項目 1 4 0 3と、を有している。周波数の項目 1 4 0 1には、各 C P Uに供給可能なクロックの周波数が登録されている。ここでは、周波数の項目 1 4 0 1には、1 0 0 [M H z]、2 0 0 [M H z]、3 0 0 [M H z]、4 0 0 [M H z]、5 0 0 [M H z] が登録されている。

[0073] 電源電圧の項目 1 4 0 2には、周波数の項目 1 4 0 1に登録された周波数でクロックが供給される際に必要な電源電圧の値が登録されている。たとえば、周波数・電力テーブル 1 4 0 0では、供給されるクロックの周波数が5 0 0 [M H z] の場合、供給される電源電圧の値は1. 6 [V] 以上でなければ、C P Uが動作しないことを示している。

[0074] 1 C P U当たりの消費電力の項目 1 4 0 3には、供給されるクロックの周波数が周波数の項目 1 4 0 1に登録された値で、かつ供給される電源電圧の値が電源電圧の項目 1 4 0 2に登録された値である場合の1 C P U当たりの消費電力の値が登録されている。たとえば、周波数・電力テーブル 1 4 0 0では、C P Uに供給されるクロックの周波数が2 0 0 [M H z] で、C P Uに供給される電源電圧の値が1. 1 [V] である場合、1 C P U当たりの消費電力の値は4 0 [mW] であることを示している。

[0075] (実施の形態 3にかかる設計支援装置 3 0 0による設計支援処理手順)

図 1 5は、実施の形態 3にかかる設計支援装置 3 0 0による設計支援処理手順の一例を示すフローチャートである。まず、設計支援装置 3 0 0が、未測定なプログラムから任意のプログラムを選択し(ステップ S 1 5 0 1)、稼働 C P U数=1とし(ステップ S 1 5 0 2)、稼働 C P U数を設定したプログラムを起動する(ステップ 1 5 0 3)。そして、設計支援装置 3 0 0が、単位時間当たりの稼働時間と延べ空き時間を測定する(ステップ S 1 5 0

4)。

[0076] 設計支援装置300が、稼働CPU数＝稼働CPU数＋1とし（ステップS1505）、稼働CPU数＞全CPU数であるか否かを判断する（ステップS1506）。設計支援装置300が、稼働CPU数＞全CPU数でないと判断した場合（ステップS1506：No）、ステップS1502へ戻る。設計支援装置300が、稼働CPU数＞全CPU数であると判断した場合（ステップS1506：Yes）、全プログラムで測定済か否かを判断する（ステップS1507）。

[0077] 設計支援装置300が、全プログラムで測定済でないと判断した場合（ステップS1507：No）、ステップS1501へ戻る。設計支援装置300が、全プログラムで測定済であると判断した場合（ステップS1507：Yes）、CPUの動作周波数を設定し（ステップS1508）、設定した動作周波数でCPUが動作する最小の電源電圧と消費電力を測定する（ステップS1509）。

[0078] 設計支援装置300が、設定可能な全動作周波数で測定したか否かを判断する（ステップS1510）。設計支援装置300が、設定可能な全動作周波数で測定していないと判断した場合（ステップS1510：No）、ステップS1508へ戻る。設計支援装置300が、設定可能な全動作周波数で測定したと判断した場合（ステップS1510：Yes）、測定結果を出力する（ステップS1511）。

[0079] （実施の形態4）

つぎに、実施の形態4では、複数のプログラムが同時に実行された場合において、1CPUで実行した場合の性能を保ちつつ、最も消費電力が少なくなるCPU数と周波数を特定し、該CPU数と周波数で複数のプログラムを動作させる例を示す。実施の形態4では、実施の形態1～3で説明した構成と同一構成については同一符号を付し、該同一符号が付された構成の詳細な説明を省略する。

[0080] （実施の形態4にかかるマルチコアプロセッサシステム200の機能ブロッ

ク図)

図 16 は、実施の形態 4 にかかるマルチコアプロセッサシステム 200 の機能ブロック図である。マルチコアプロセッサシステム 200 は、記憶部 1601 と、プロセス管理部 1602 と、稼働 CPU 数決定部 1603 と、DVFS 制御部 1604 と、スケジューリング部 1605 と、を有している。

[0081] 具体的には、たとえば、プロセス管理部 1602～スケジューリング部 1605 を有するプログラムが ROM 207 やフラッシュ ROM 208 などの記憶装置に記憶されている。CPU が該記憶装置にアクセスして該プログラムを読み出し、該プログラム内にコーディングされている処理を実行することにより、プロセス管理部 1602～スケジューリング部 1605 の処理が実行される。ここでは、該プログラムは OS 220 である。

[0082] 記憶部 1601 は、プログラムごとに基準周波数における CPU 数ごとの稼働時間を記憶し、周波数ごとに単位時間当たりの 1 CPU の消費電力量を記憶する。具体的には、たとえば、ROM 207 やフラッシュ ROM 208 は、稼働時間テーブル 1300 や稼働時間テーブル 1310 や周波数・電力テーブル 1400 を記憶している。

[0083] プロセス管理部 1602 は、対象プログラムの起動を検出する。稼働 CPU 数決定部 1603 は、対象プログラムと実行中のプログラムを実行する稼働 CPU 数を決定する。稼働 CPU 数決定部 1603 は、抽出部 1611 と、周波数算出部 1612 と、消費電力算出部 1613 と、決定部 1614 と、を有している。抽出部 1611 は、プロセス管理部 1602 により対象プログラムの起動が検出された場合、マルチコアプロセッサで実行中のプログラムの稼働時間を CPU 数ごとに記憶部 1601 から抽出する。抽出部 1611 は、対象プログラムの稼働時間を CPU 数ごとに記憶部 1601 から抽出する。

[0084] 周波数算出部 1612 は、抽出部 1611 により CPU 数ごとに抽出された実行中のプログラムの稼働時間と対象プログラムの稼働時間との合計稼働時間との比率により、指定 CPU 数の合計稼働時間を満たす周波数を CPU

数ごとに算出する。

[0085] 消費電力算出部 1613 は、周波数算出部 1612 により CPU 数ごとに算出された周波数に応じた記憶部 1601 に記憶された単位時間当たりの 1 CPU の消費電力に基づいて、CPU 数ごとに消費電力量を算出する。決定部 1614 は、消費電力算出部 1613 により CPU 数ごとに算出された消費電力量の中で最少の消費電力量である CPU 数を稼働 CPU 数に決定する。

[0086] D V F S 制御部 1604 は、稼働 CPU 数決定部 1603 により決定された CPU 数分の CPU へ、稼働 CPU 数決定部 1603 により算出された周波数のクロックを供給する。D V F S 制御部 1604 は、マルチコアプロセッサのうちの稼働中の CPU の数が決定された CPU 数より多い場合、稼働中の CPU から稼働 CPU 数を超える数分の CPU を停止させる。D V F S 制御部 1604 は、マルチコアプロセッサのうちの稼働中の CPU の数が、決定された CPU 数より少ない場合、不足分の CPU を起動させる。そして、スケジューリング部 1605 は、決定された CPU 数分の CPU で対象プログラムと実行中のプログラムを実行する。以上を踏まえて詳細に説明する。

[0087] 図 17 は、複数のプログラムが同時に実行される例を示す説明図である。マルチコアプロセッサシステム 200 ではブラウザが実行中であり、稼働 CPU 数は 4 である。利用者がメーラを起動すると、OS 220 がメーラの起動指示を検出する。そして、OS 220 が、実行中のプログラムの識別情報を取得する。OS 220 が、該識別情報を取得したプログラムに対応する稼働時間テーブルと起動指示を検出したプログラムに対応する稼働時間テーブルとを取得する。ここでは、メーラに関する稼働時間テーブル 1300 とブラウザに関する稼働時間テーブル 1310 が ROM 207 やフラッシュ ROM 208 などの記憶装置から読み出される。

[0088] 図 18 は、合計稼働時間を示す説明図である。OS 220 が、稼働時間テーブル 1300 と稼働時間テーブル 1310 を用いてメーラとブラウザとの

合計稼働時間をCPU数ごとに算出する。CPU数が1の場合、合計稼働時間は165 [ms] である。CPU数が2の場合、合計稼働時間は125 [ms] である。CPU数が3の場合、合計稼働時間は100 [ms] である。CPU数が4の場合、合計稼働時間は75 [ms] である。

[0089] 図19は、消費電力が最も低くなるCPU数を特定する例を示す説明図である。つぎに、OS220が、各CPU数において、CPU数が1の場合の稼働時間を維持可能なクロックの周波数を算出する。具体的には、たとえば、OS220が、下記式(2)を用いて各CPU数におけるクロックの周波数を算出する。

[0090] 周波数＝基準周波数×各CPU数での全稼働時間／指定CPU数での全稼働時間・・・(2)

[0091] たとえば、ここでは、基準周波数を500 [MHz] とする。CPU数が2の場合、式(2)による算出結果は378 [MHz] であり、クロックの周波数は400 [MHz] となる。CPU数が3の場合、式(2)による算出結果は303 [MHz] であり、クロックの周波数は400 [MHz] となる。CPU数が4の場合、式(2)による算出結果は227 [MHz] であり、クロックの周波数は300 [MHz] となる。

[0092] OS220が、周波数・電力テーブル1400を用いて、各CPU数で実行した場合の消費電力の値を算出する。400 [MHz] の場合の1CPU当たりの消費電力が85 [mW] であるため、CPU数が2の場合、全消費電力は85×2で170 [mW] である。CPU数が3の場合、全消費電力は85×3で255 [mW] である。300 [MHz] の場合の1CPU当たりの消費電力が60 [mW] であるため、CPU数が4の場合、全消費電力は60×4で240 [mW] である。よって、CPU数が1の場合での性能と同一性能で消費電力が最も低くなるCPU数は2であるため、OS220が稼働CPU数を2に決定する。

[0093] 図20は、稼働CPU数の変更例を示す説明図である。OS220が、稼働CPU数を減少させるため、停止させる停止CPUを決定する。ここでは

、CPU203とCPU204を停止CPUとする。OS220が、DVFS制御機構205からCPU203とCPU204へのクロック・電源電圧の供給を停止させる。OS220が、CPU201とCPU202へ供給するクロックの周波数を400 [MHz] に設定し、電源電圧の値を1.4 [V] に設定する。

[0094] (実施の形態4にかかるOS220による制御処理手順)

図21と図22は、実施の形態4にかかるOS220による制御処理手順の一例を示すフローチャートである。まず、OS220が、プロセスの起動または終了を検出したか否かを判断し(ステップS2101)、プロセスの起動および終了を検出していないと判断した場合(ステップS2101: No)、ステップS2101へ戻る。OS220が、プロセスの起動または終了を検出したと判断した場合(ステップS2101: Yes)、稼働中のすべてのプロセスを特定する(ステップS2102)。

[0095] OS220が、特定したプロセスに対応する稼働時間テーブルを参照し(ステップS2103)、各CPU数における稼働時間の全プロセス分の総計を算出する(ステップS2104)。OS220が、周波数・電力テーブル1400から算出した各CPU数での周波数に対応する1CPU当たりの消費電力の値を取得し(ステップS2105)、各CPU数における全消費電力の値を算出する(ステップS2106)。

[0096] OS220が、算出した全消費電力の値が最小となるCPU数を稼働CPU数に決定し(ステップS2107)、稼働CPU数に変更があるか否かを判断する(ステップS2108)。OS220が、稼働CPU数に変更がないと判断した場合(ステップS2108: No)、ステップS2101へ移行する。

[0097] OS220が、稼働CPU数に変更があると判断した場合(ステップS2108: Yes)、稼働CPU数が増加したか否かを判断する(ステップS2109)。OS220が、稼働CPU数が増加したと判断した場合(ステップS2109: Yes)、停止中のCPUから起動させるCPU(起動C

PU)を決定する(ステップS2110)。OS220が、起動CPUと稼働中のCPUに取得した電源電圧の値と周波数とを供給させる制御を行い(ステップS2111)、ステップS2101へ移行する。

[0098] OS220が、稼働CPU数が増加していないと判断した場合(ステップS2109:No)、停止させるCPU(停止CPU)を決定し(ステップS2112)、停止CPUの電源供給、周波数入力を停止させる制御をおこなう(ステップS2113)。OS220が、稼働中有のCPUに取得した電源電圧の値と周波数とを供給させる制御を行い(ステップS2114)、ステップS2101へ移行する。

[0099] 以上実施の形態1、3で説明したように、設計支援方法によれば、イベントごとに消費電力量が最小となる稼働CPU数とクロックの周波数の組み合わせを特定する。これにより、消費電力量が最小となる稼働CPU数とクロックの周波数でシステムを運用することができる。

[0100] また、イベントごとにイベントに応じた処理を1CPUで実行した場合の稼働時間と基準周波数とに基づいてクロックの周波数を算出する。これにより、1CPUで実行した場合の性能を維持することができる。

[0101] 以上実施の形態2で説明したように、スケジューリング方法、およびシステムによれば、所定イベントごとに稼働CPU数とクロックの周波数とをメモリに記憶する。そして、所定イベントが発生する都度、稼働CPU数と周波数を切り替えて、該イベントに応じた処理を実行する。これにより、消費電力が最も少ない稼働CPU数とクロック周波数の組み合わせにより、イベントに応じた処理を実行でき、消費電力を減少させることができる。

[0102] また、稼働中のCPUの数が稼働CPU数よりも多い場合、稼働中のCPUから稼働CPU数を超える数分のCPUを停止させることにより、消費電力が最小となる最適なCPU数でシステムを運用することができる。

[0103] また、稼働中のCPUの数が稼働CPU数分より少ない場合、不足分のCPUを起動させることにより、1CPUで実行した場合の性能を維持することができる。

- [0104] また、所定イベントごとに稼働CPU数とクロックの周波数と電源電圧の値とをメモリに記憶し、所定イベントが発生する都度、稼働CPU数と周波数と電源電圧の値とを切り替えて、該イベントに応じた処理を実行する。消費電力は周波数と電源電圧の二乗に比例する。クロックの周波数を下げることにより、電源電圧を下げるができるため、消費電力量を減少させることができる。
- [0105] 以上実施の形態4で説明したように、スケジューリング方法、およびシステムによれば、プログラムごとに各CPU数分のCPUで実行した場合の稼働時間と、周波数ごとに1CPU当たりの消費電力量を記憶する。そして、複数のプログラムが同時に実行された場合において、1CPUで実行した場合の性能を保ちつつ、最も消費電力が少なくなる最適な稼働CPU数と周波数を特定し、該稼働CPU数と周波数で複数のプログラムを動作させる。これにより、消費電力量を削減することができる。
- [0106] また、稼働中のCPUの数が稼働CPU数よりも多い場合、稼働中のCPUから稼働CPU数を超える数分のCPUを停止させることにより、消費電力が最小となる最適なCPU数でシステムを運用することができる。
- [0107] また、稼働中のCPUの数が稼働CPU数分より少ない場合、不足分のCPUを起動させることにより、1CPUで実行した場合の性能を維持することができる。

符号の説明

- [0108] 207 ROM
 208 フラッシュROM
 401 測定部
 402 周波数算出部
 403 消費電力算出部
 404 決定部
 405 出力部
 801, 1601 記憶部

802 イベント検出部
803 シーン決定部
804, 1604 DVFS制御部
805, 1605 スケジューリング部
1602 プロセス管理部
1603 稼働CPU数決定部
200 マルチコアプロセッサシステム
300 設計支援装置

請求の範囲

- [請求項1] 第1処理から第2処理への変更を検出し、
メモリから前記第2処理を実行するCPUの数と動作周波数とを取得し、
前記CPUの数に基づいて、稼働中のCPUを停止しまたは停止中のCPUを起動し、
前記第2処理を実行するCPUに前記動作周波数を割り当てることを特徴とするスケジューリング方法。
- [請求項2] 前記稼働中のCPUの数が前記CPUの数より多い場合、前記稼働中のCPUから前記CPUの数を超える数分のCPUを停止させること
を特徴とする請求項1に記載のスケジューリング方法。
- [請求項3] 前記稼働中のCPUの数が前記CPUの数分より少ない場合、不足分のCPUを起動させること
を特徴とする請求項1に記載のスケジューリング方法。
- [請求項4] さらに、前記メモリから前記第2処理を実行するCPUの数と動作周波数と電源電圧の値を取得し、
前記CPUの数に基づいて、動作中のCPUを停止しまたは停止中のCPUを起動し、
前記第2処理を実行するCPUに前記動作周波数と前記電源電圧の値を割り当てること
を特徴とする請求項1乃至請求項3のいずれかに記載のスケジューリング方法。
- [請求項5] 第1処理を実行するときの、第1数個のCPUの第1稼働時間と第1停止時間とを測定し、
第1最小動作周波数より大きい第1動作周波数を設定し、
前記第1稼働時間と前記第1停止時間とに基づいて、前記第1動作周波数で動作する前記第1数個のCPUの第1消費電力を算出し、

前記第 1 処理を実行するときの、前記第 1 数個とは異なる第 2 数個の CPU の第 2 稼働時間と第 2 停止時間とを測定し、

第 2 最小動作周波数より大きい第 2 動作周波数を設定し、

前記第 2 稼働時間と前記第 2 停止時間とに基づいて、前記第 2 動作周波数で動作する前記第 2 数個の CPU の第 2 消費電力を算出し、

前記第 1 消費電力と前記第 2 消費電力との比較結果に基づいて、前記第 1 処理実行時の CPU の数を決定すること

を特徴とする設計支援方法。

[請求項6] 前記第 1 最小動作周波数は、前記第 1 処理を 1 個の CPU で処理するときの稼働時間と前記第 1 稼働時間と所定動作周波数とに基づいて算出され、

前記第 2 最小動作周波数は、前記第 1 処理を 1 個の CPU で処理するときの稼働時間と前記第 2 稼働時間と前記所定動作周波数とに基づいて算出されること

を特徴とする請求項 5 に記載の設計支援方法。

[請求項7] 複数の CPU と、
複数の処理のそれぞれについて処理を実行する CPU の数と動作周波数とを記憶するメモリと、

前記 CPU の数に基づいて、動作中の CPU を停止しまたは停止中の CPU を起動するスケジューリング部と、

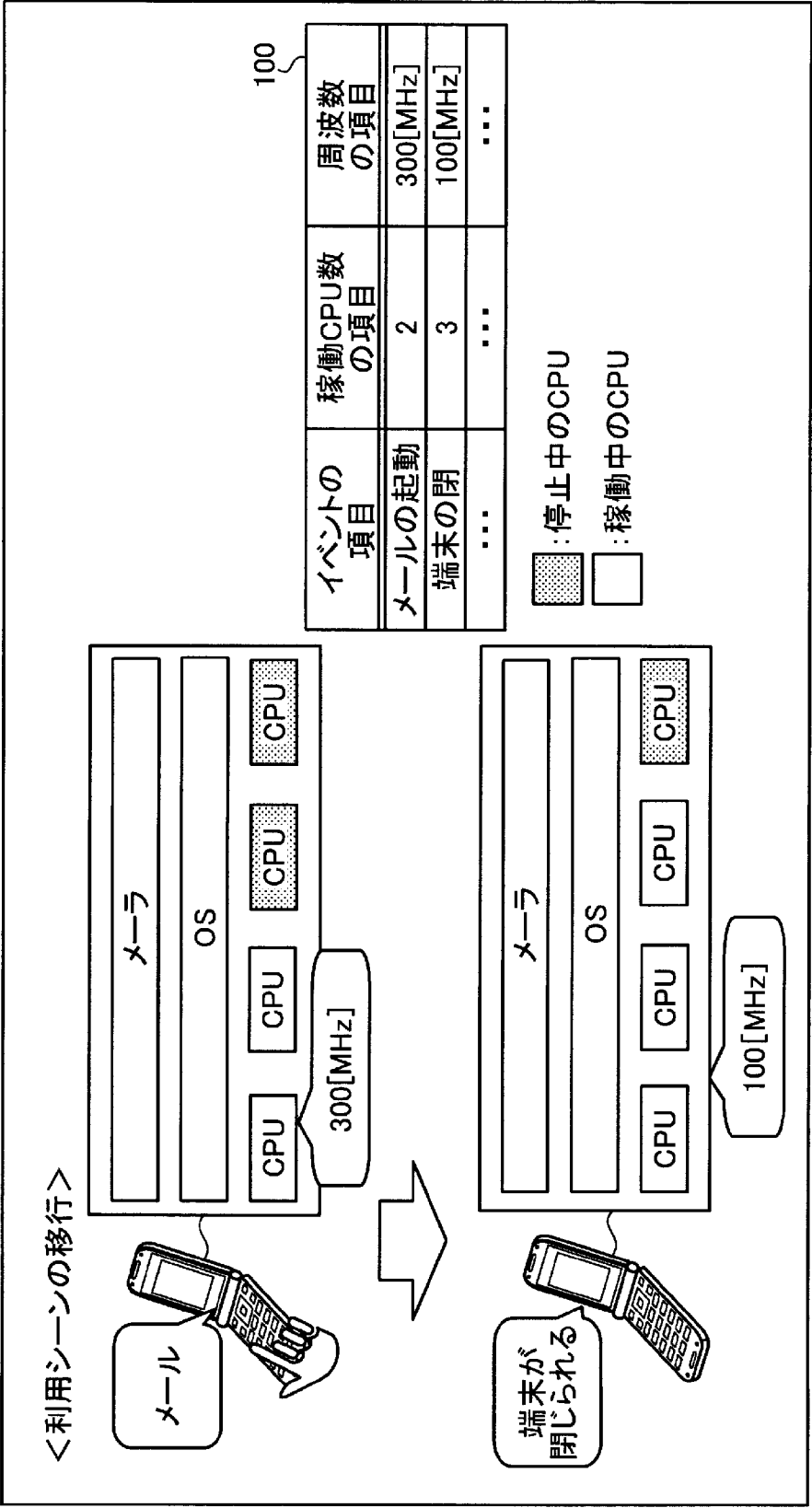
前記 CPU の数分の CPU に前記動作周波数を設定する制御部と、
を含むことを特徴とするシステム。

[請求項8] 前記スケジューリング部は、
前記稼働中の CPU の数が前記 CPU の数より多い場合、前記稼働中の CPU から前記 CPU の数を超える数分の CPU を停止すること
を特徴とする請求項 7 に記載のシステム。

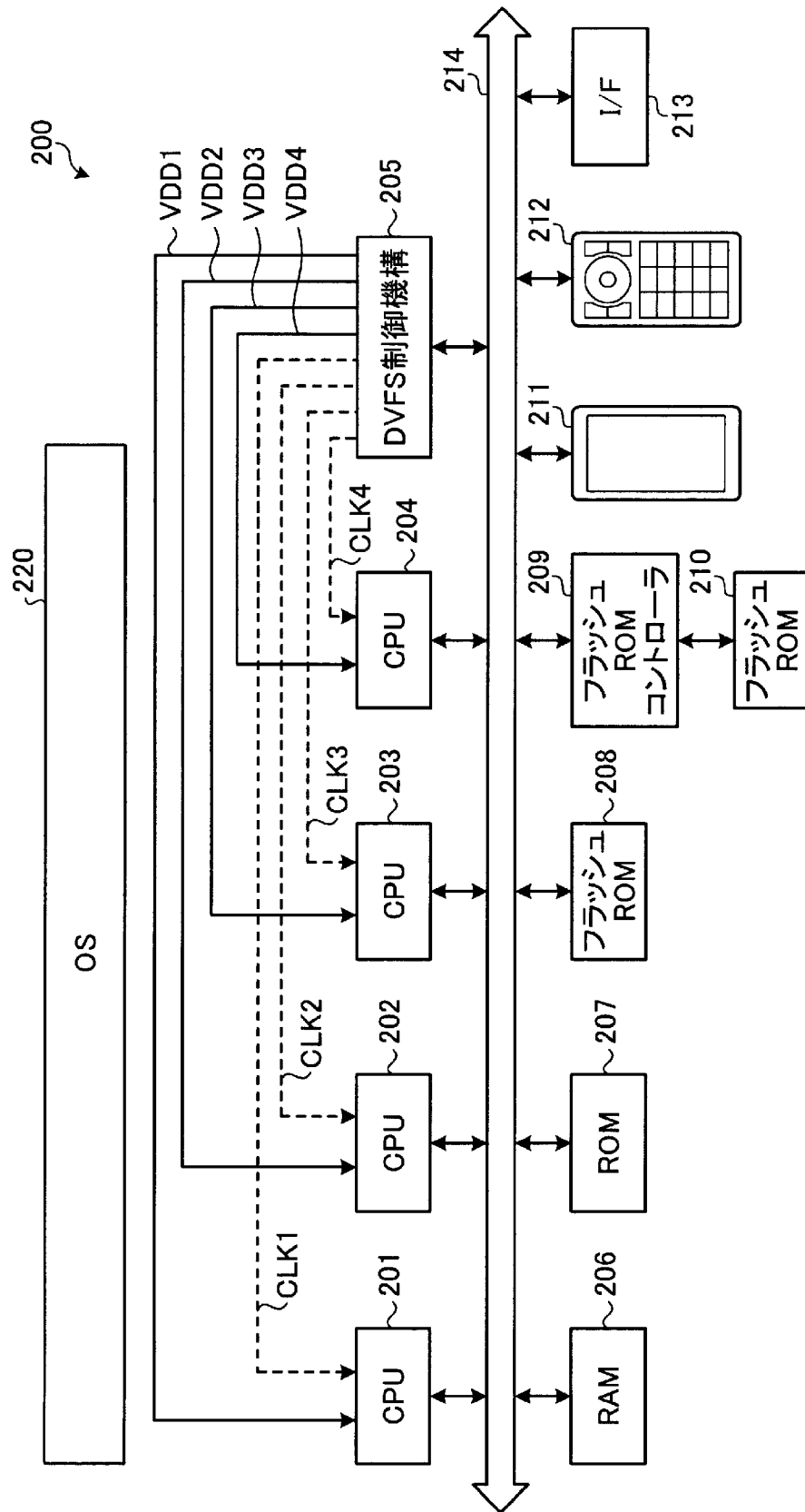
[請求項9] 前記スケジューリング部は、
前記稼働中の CPU の数が前記 CPU の数分より少ない場合、不足

分のCPUを起動すること
を特徴とする請求項7に記載のシステム。

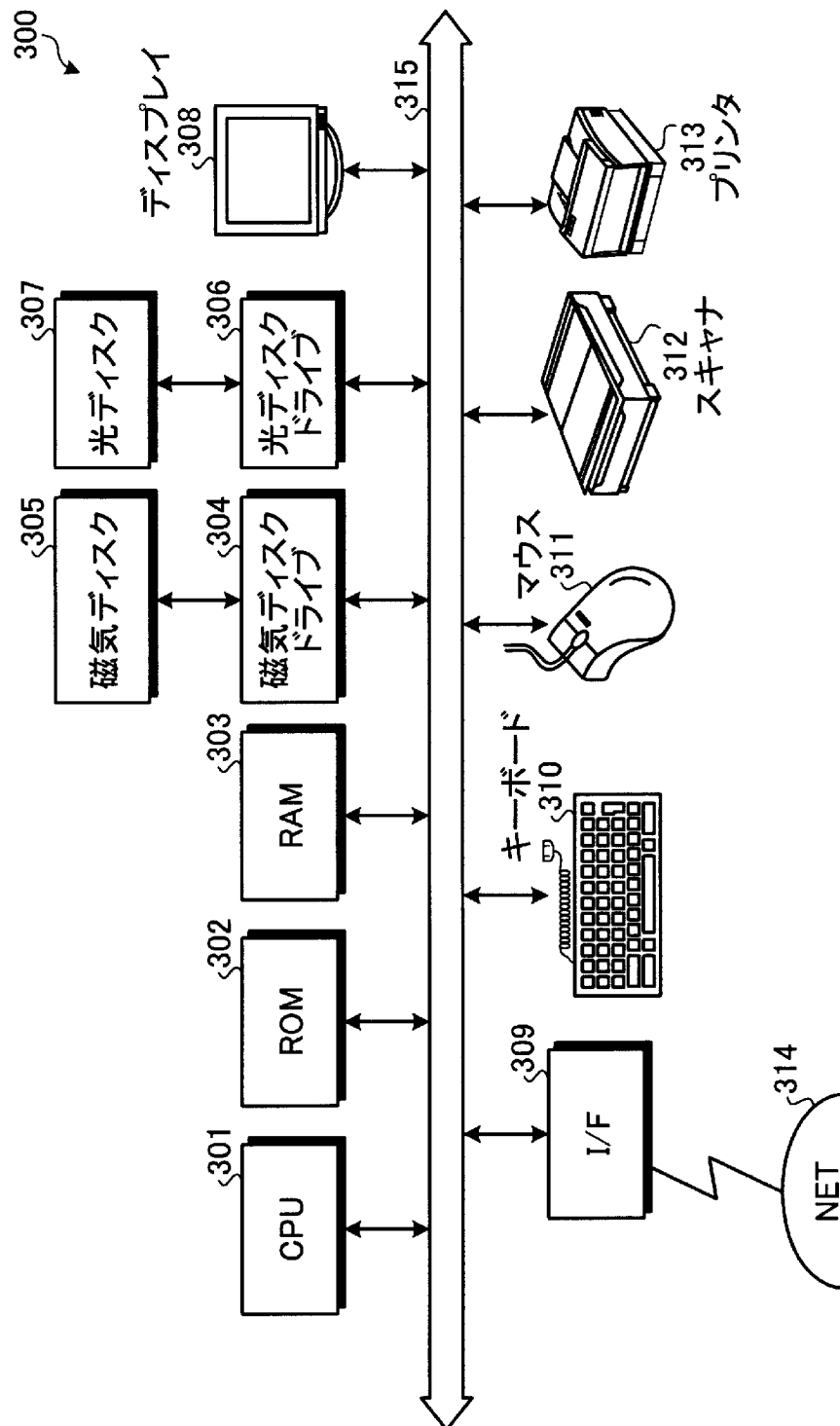
[図1]



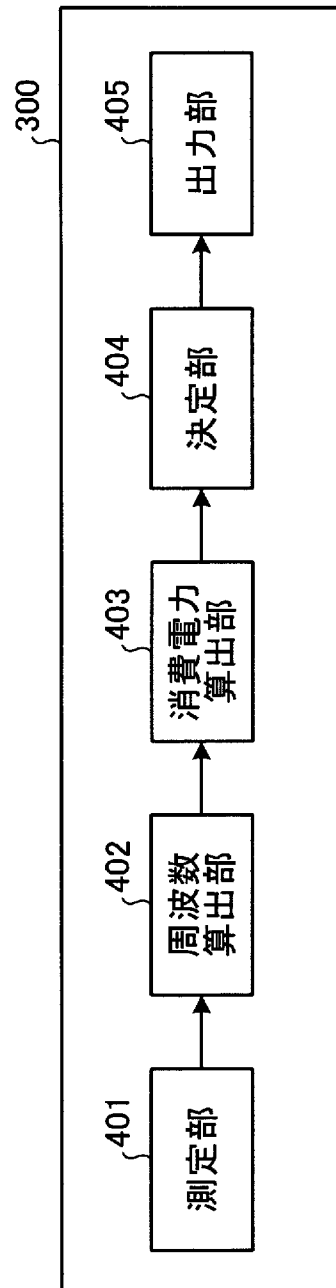
[図2]



[図3]



[図4]



[図5]

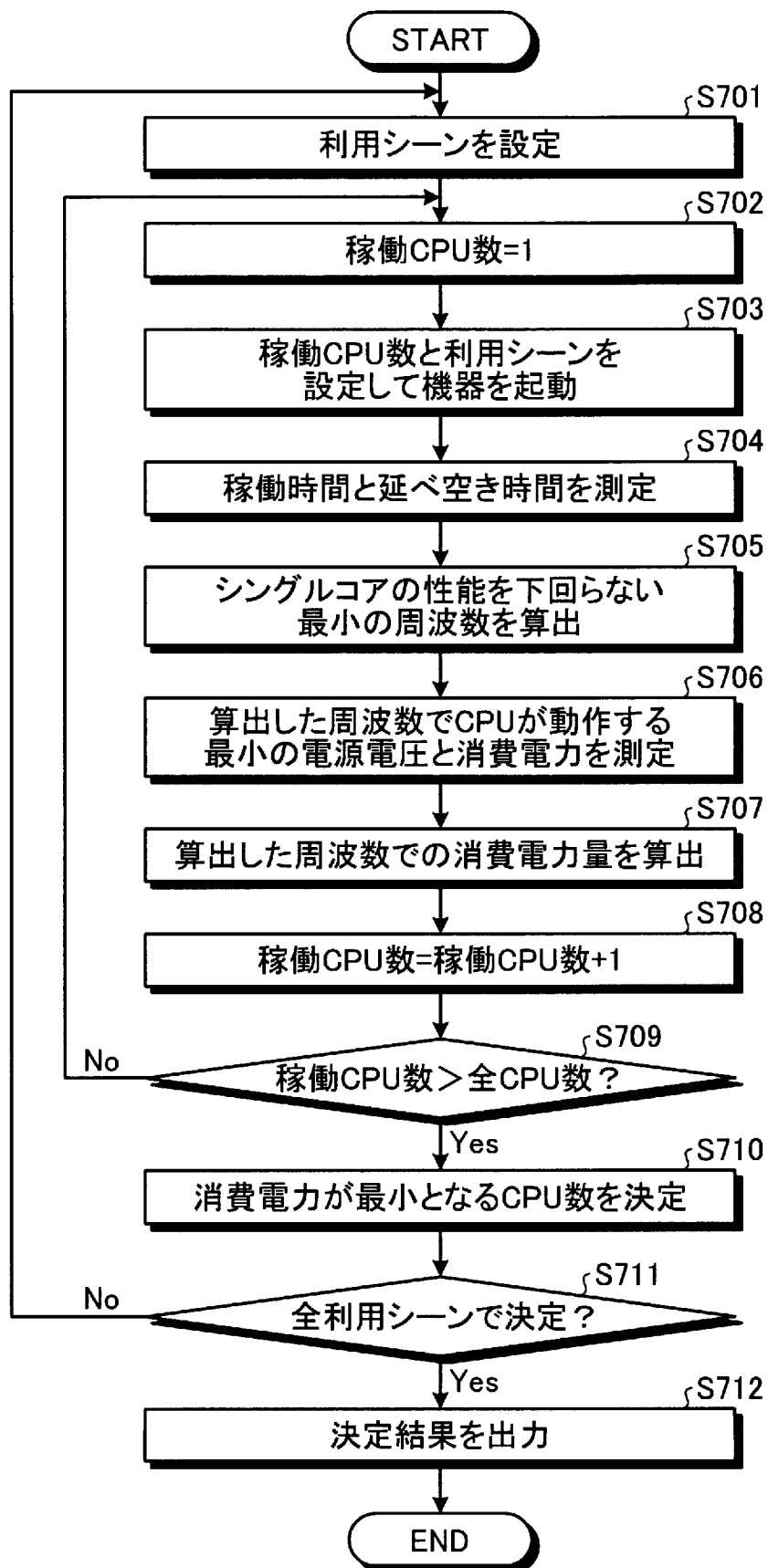
501		
CPU数の項目	稼働時間の項目	待ち時間の項目
1	100[s]	0[s]
2	60[s]	10[s]
3	50[s]	35[s]
4	40[s]	40[s]

502		
CPU数の項目	稼働時間の項目	待ち時間の項目
1	100[s]	0[s]
2	60[s]	10[s]
3	50[s]	35[s]
4	40[s]	40[s]

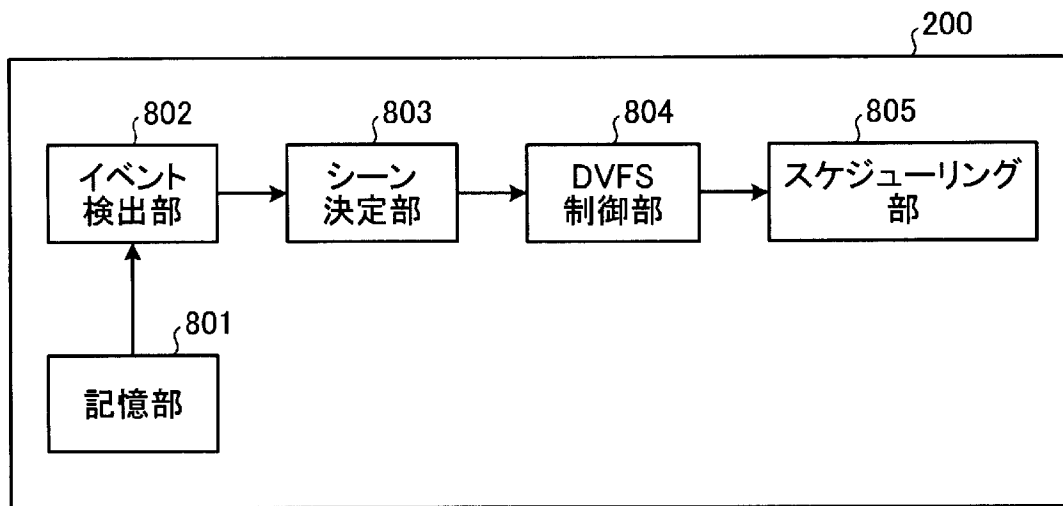
[図6]

601 { イベントの項目		602 { 稼働CPU数の項目		603 { 周波数の項目		604 { 電源電圧の項目		600 {
メーラの起動		2		300[MHz]		1.2[V]		
動画再生ソフトウェアの起動		3		300[MHz]		1.2[V]		
ブラウザの起動		4		400[MHz]		1.4[V]		
端末の閉		3		100[MHz]		1.1[V]		
キーYの押下		1		500[MHz]		1.6[V]		
...		...		...		...		

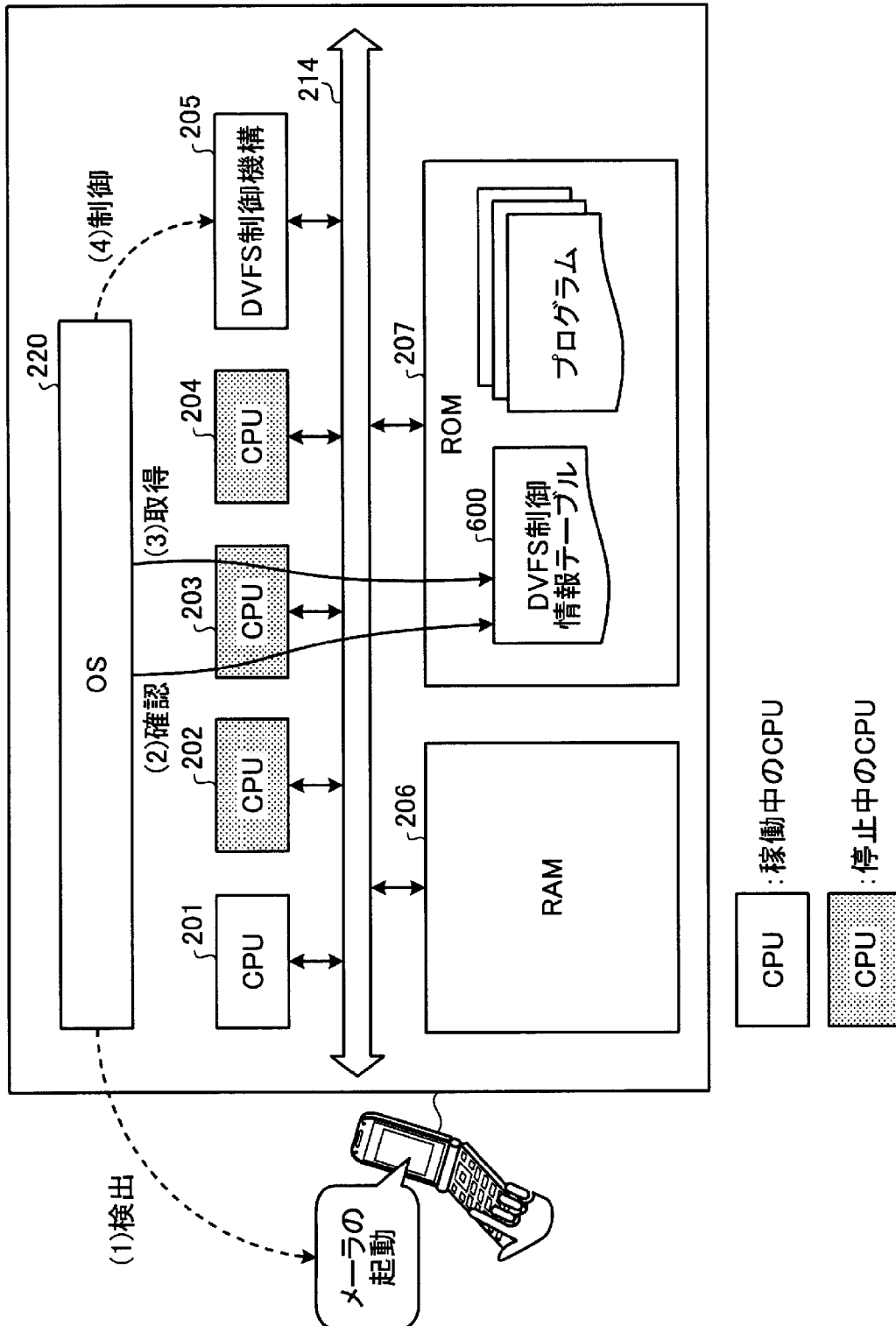
[図7]



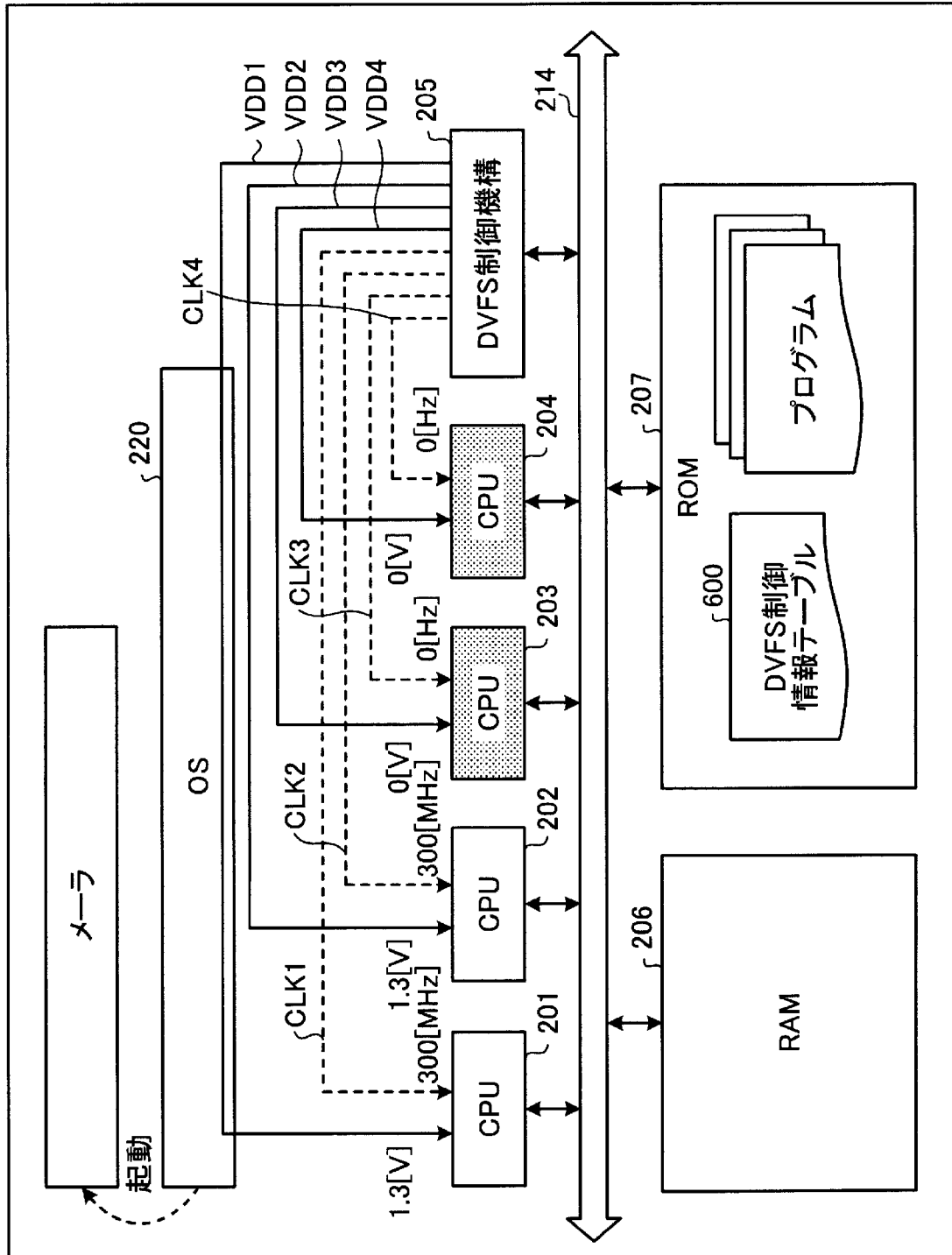
[図8]



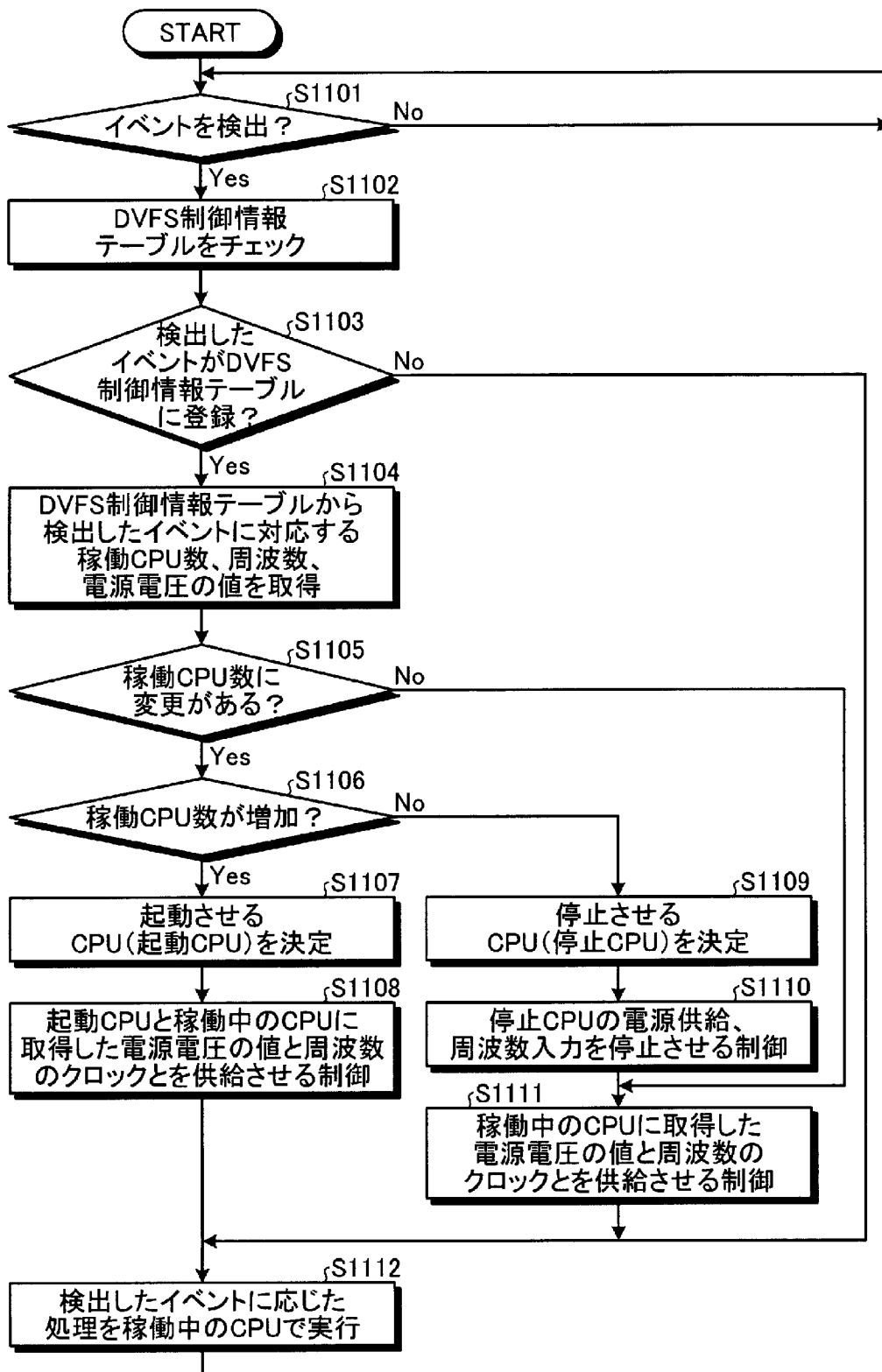
[図9]



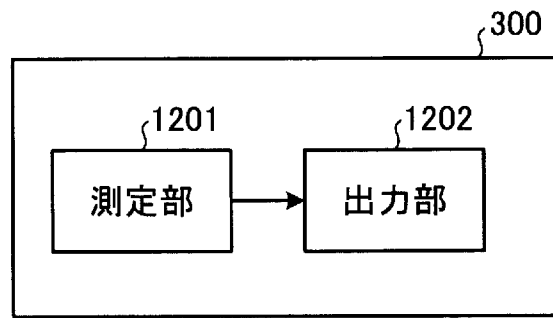
[図10]



[図11]



[図12]



[図13]

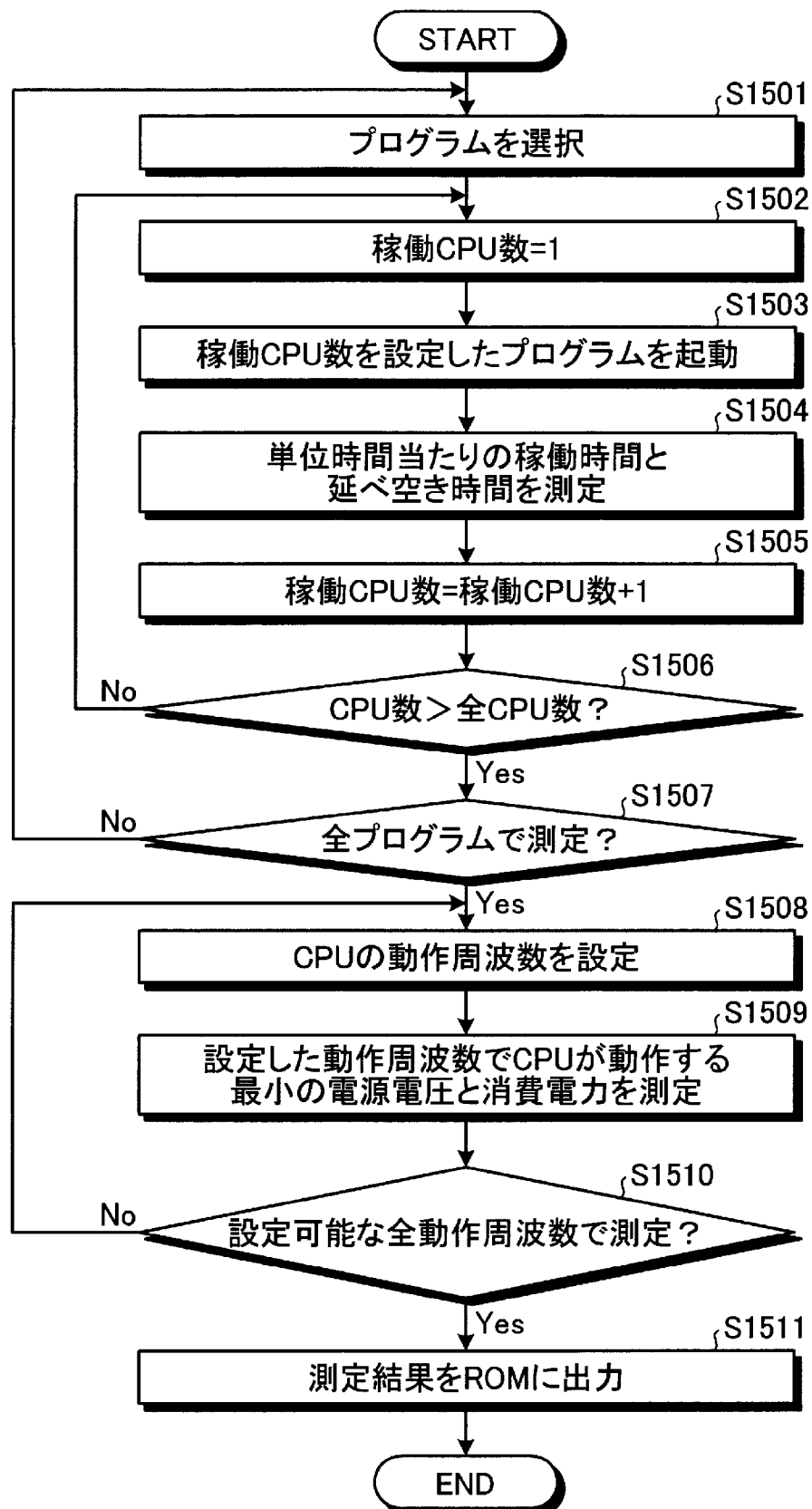
1301 CPU数の項目	1302 稼働時間の項目 (単位:[ms])	1303 空き時間の項目 (単位:[ms])	1300
1	75	0	
2	65	15	
3	60	30	
4	50	45	

1311 CPU数の項目	1312 稼働時間の項目 (単位:[ms])	1313 空き時間の項目 (単位:[ms])	1310
1	90	0	
2	60	20	
3	40	25	
4	25	30	

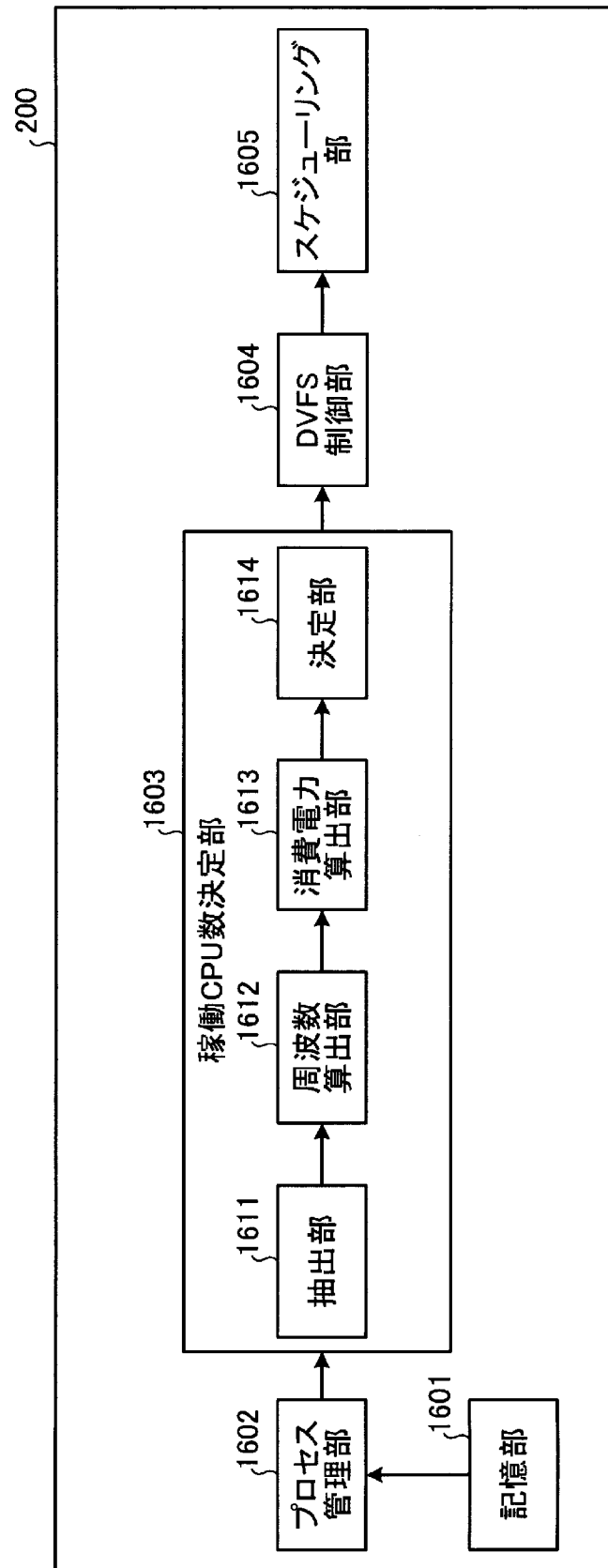
[図14]

1401 周波数の項目 (単位:[MHz])	1402 電源電圧の項目 (単位:[V])	1403 1 CPU当たりの消費電力の項目 (単位:[mW])	1400
100	0.9	15	
200	1.1	40	
300	1.2	60	
400	1.4	85	
500	1.6	180	

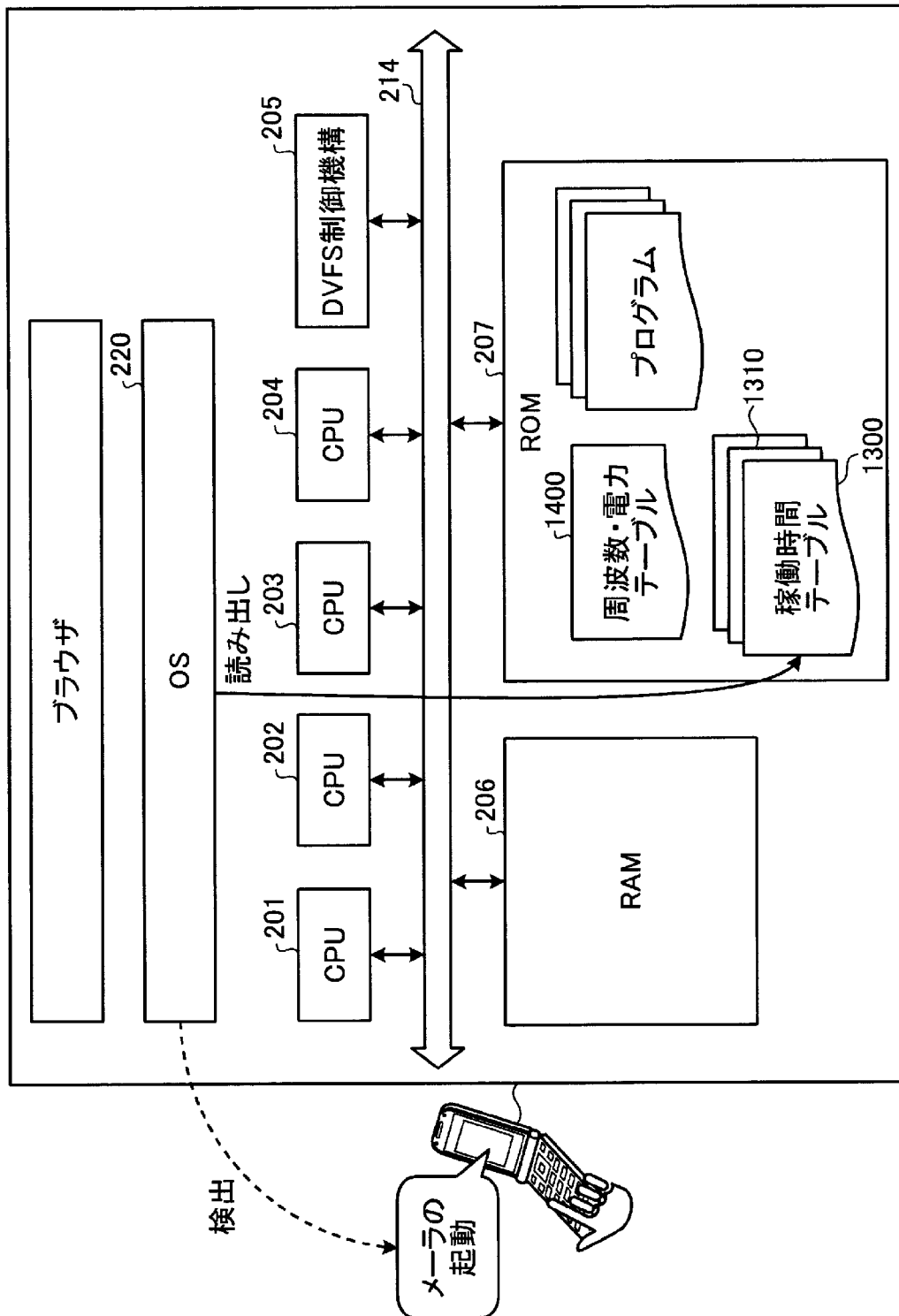
[図15]



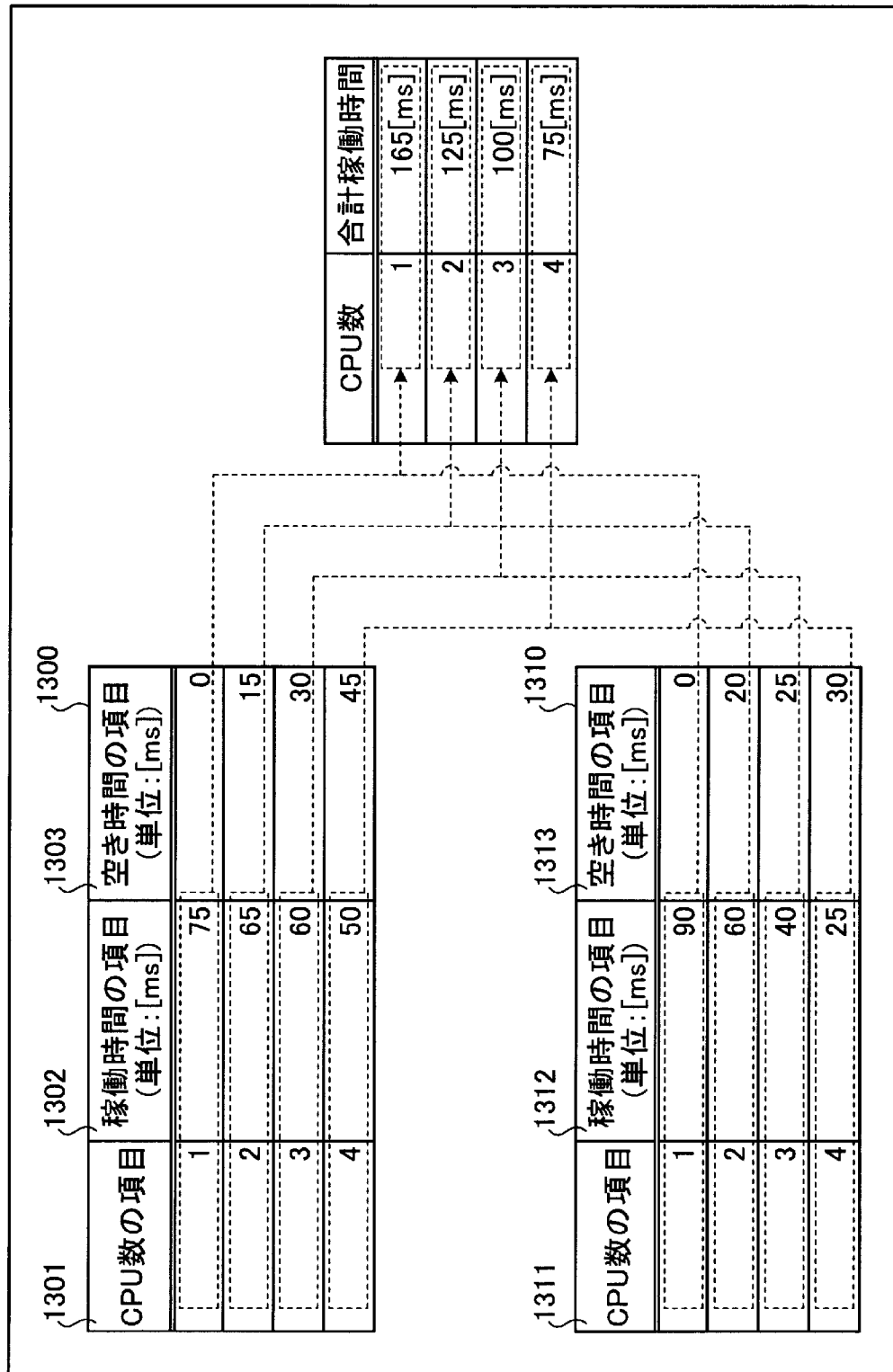
[図16]



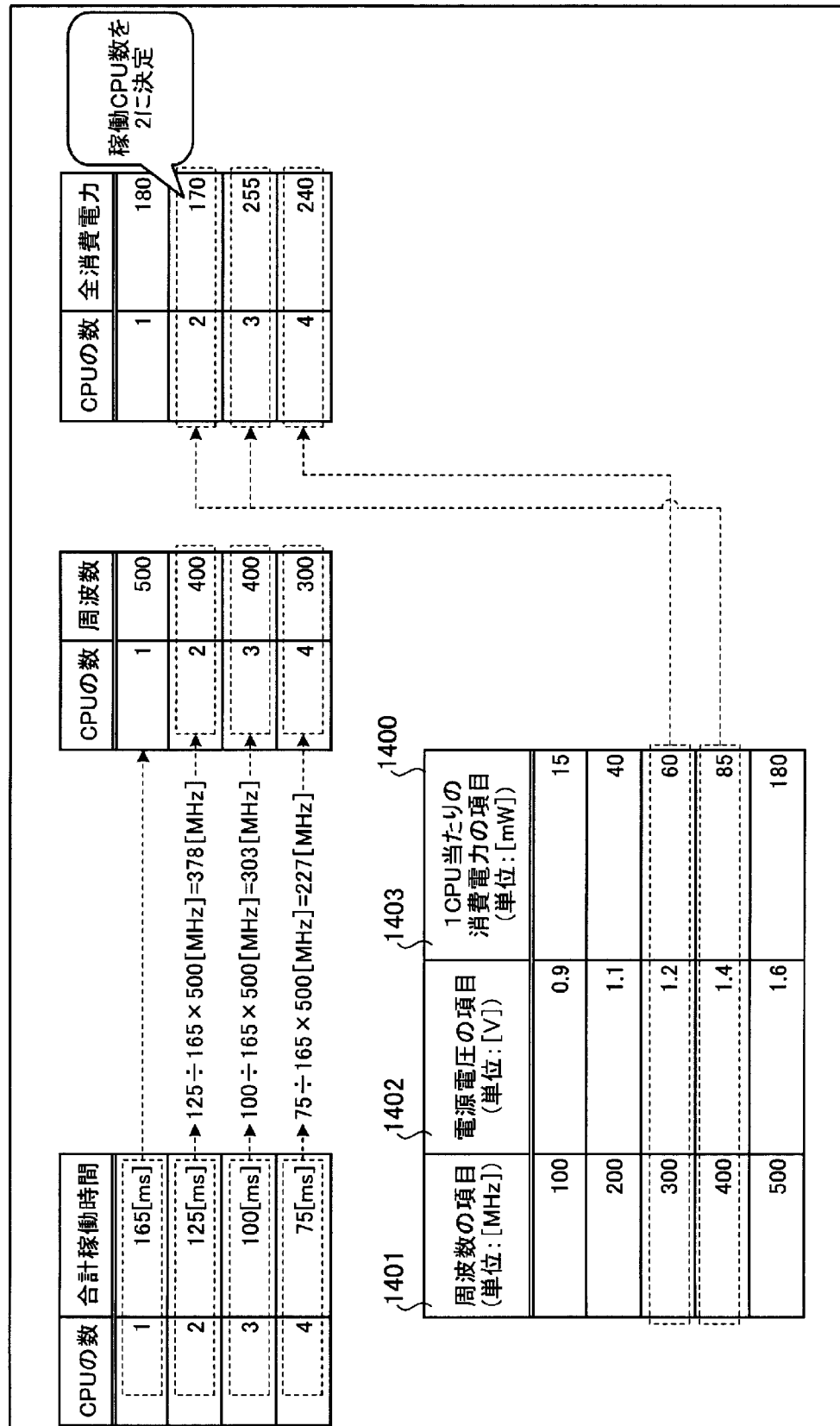
[図17]



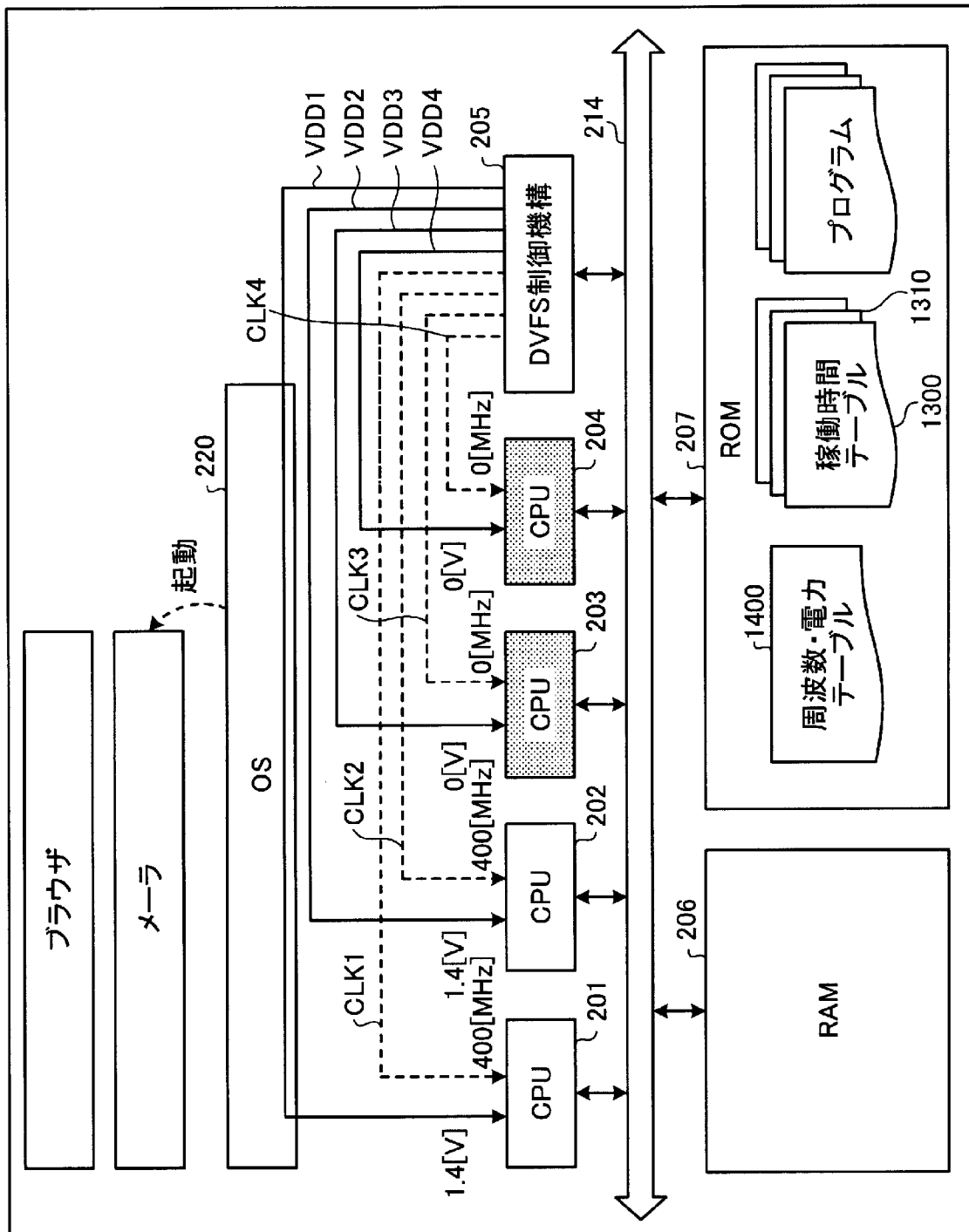
[図18]



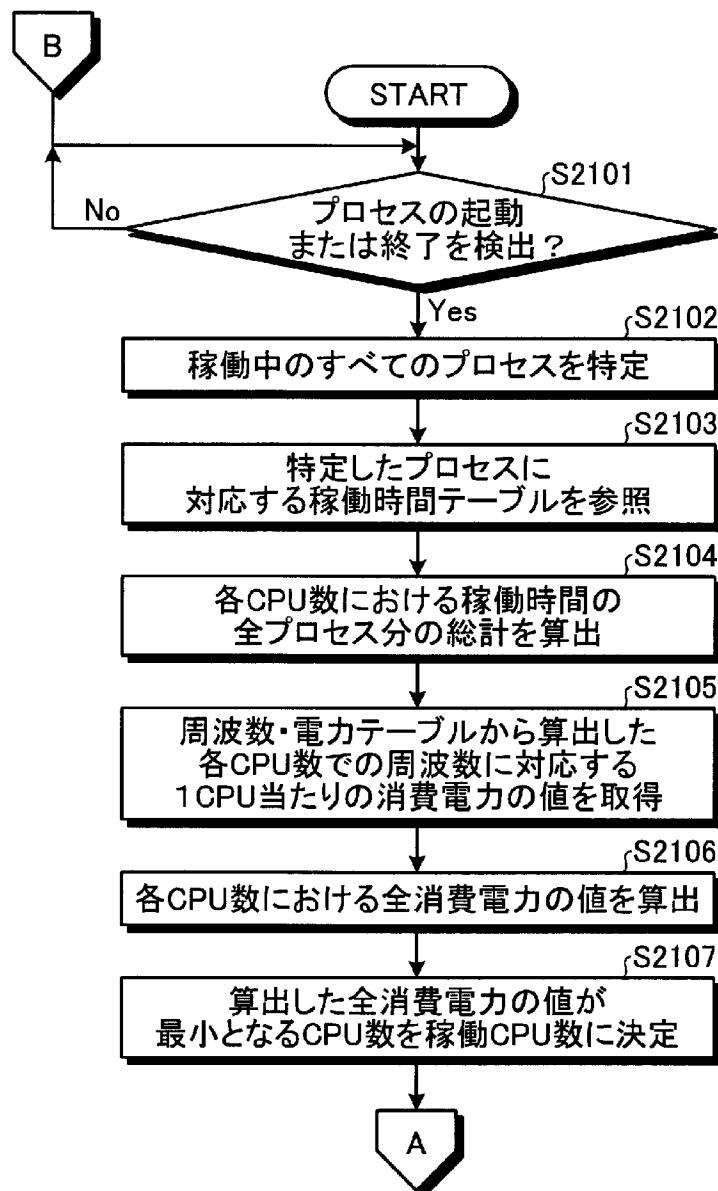
[図19]



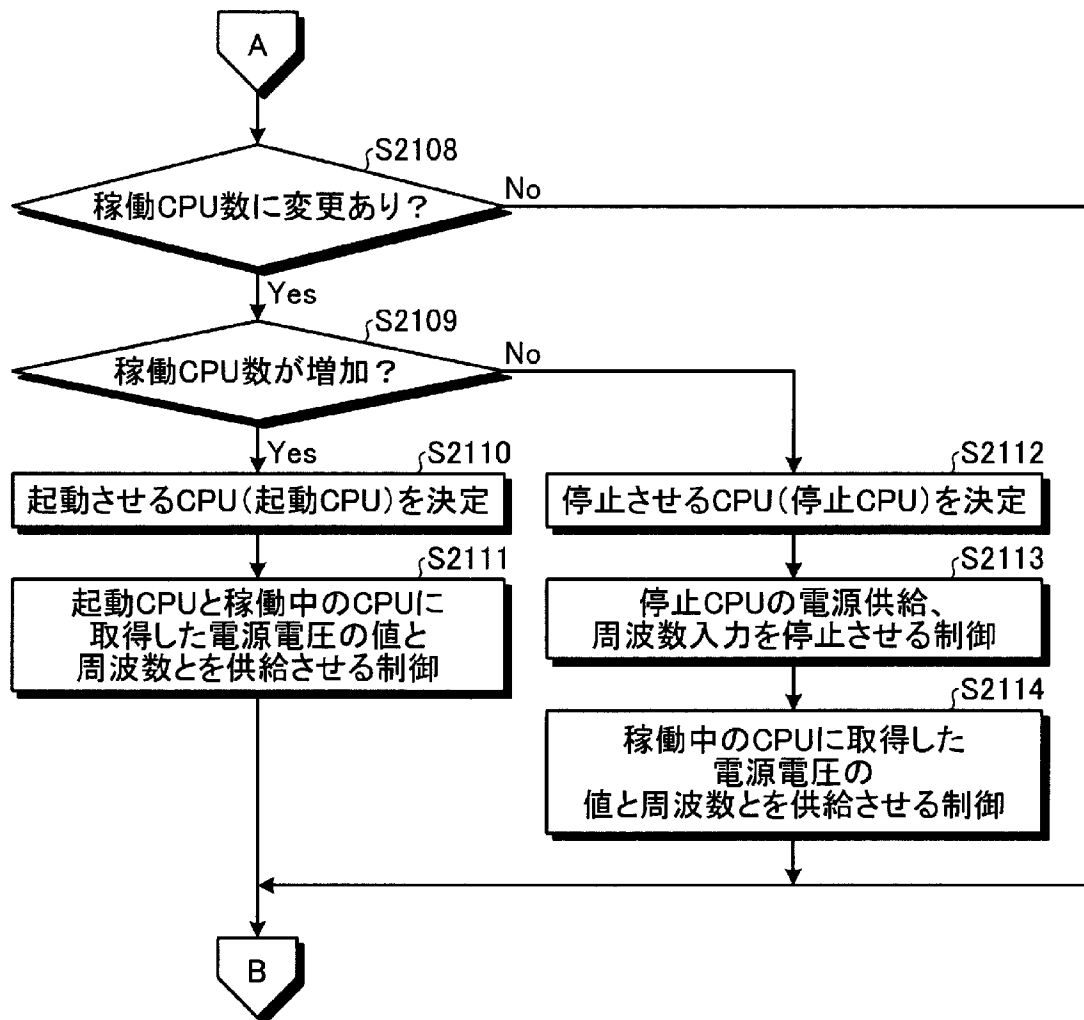
[図20]



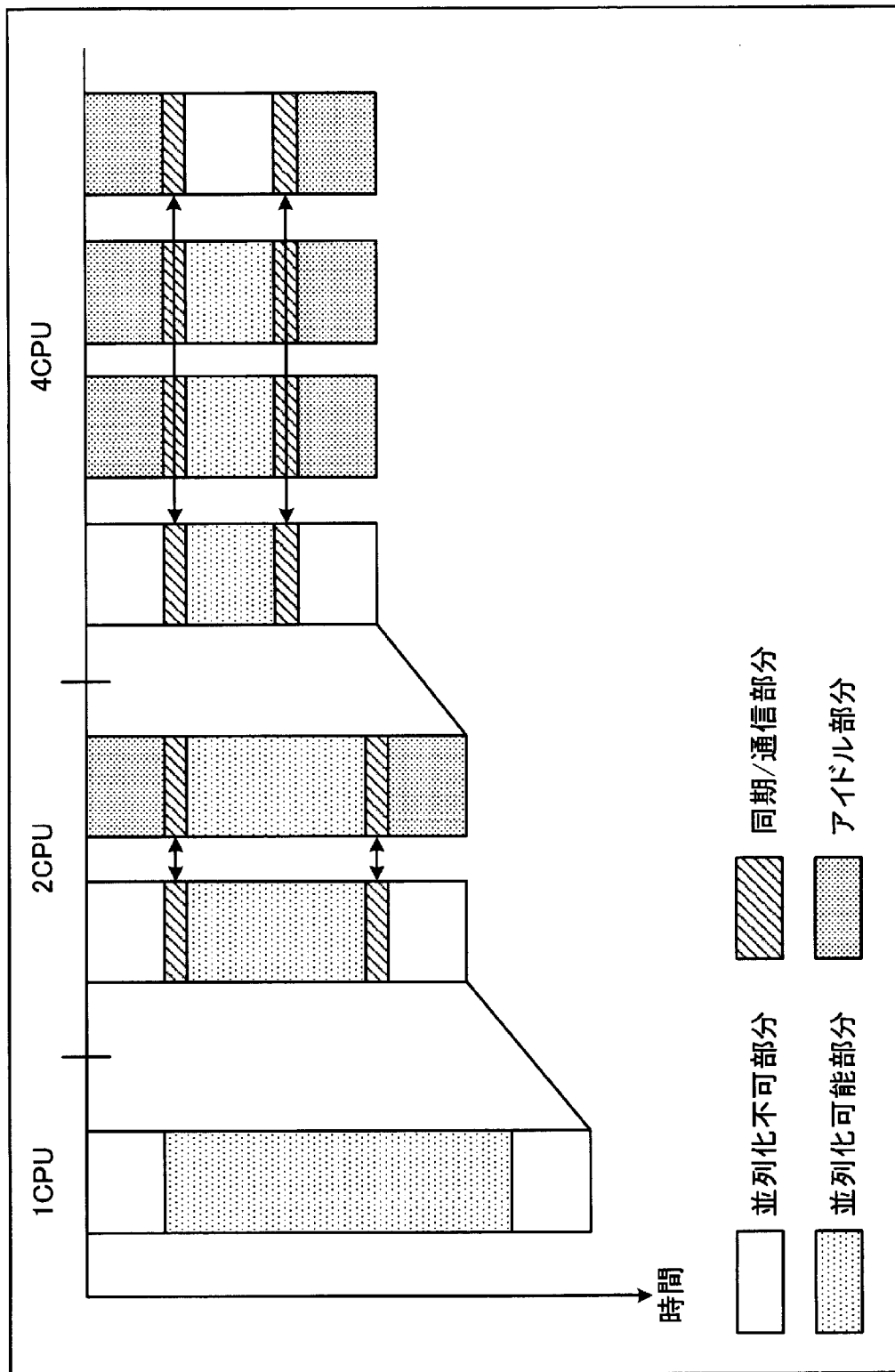
[図21]



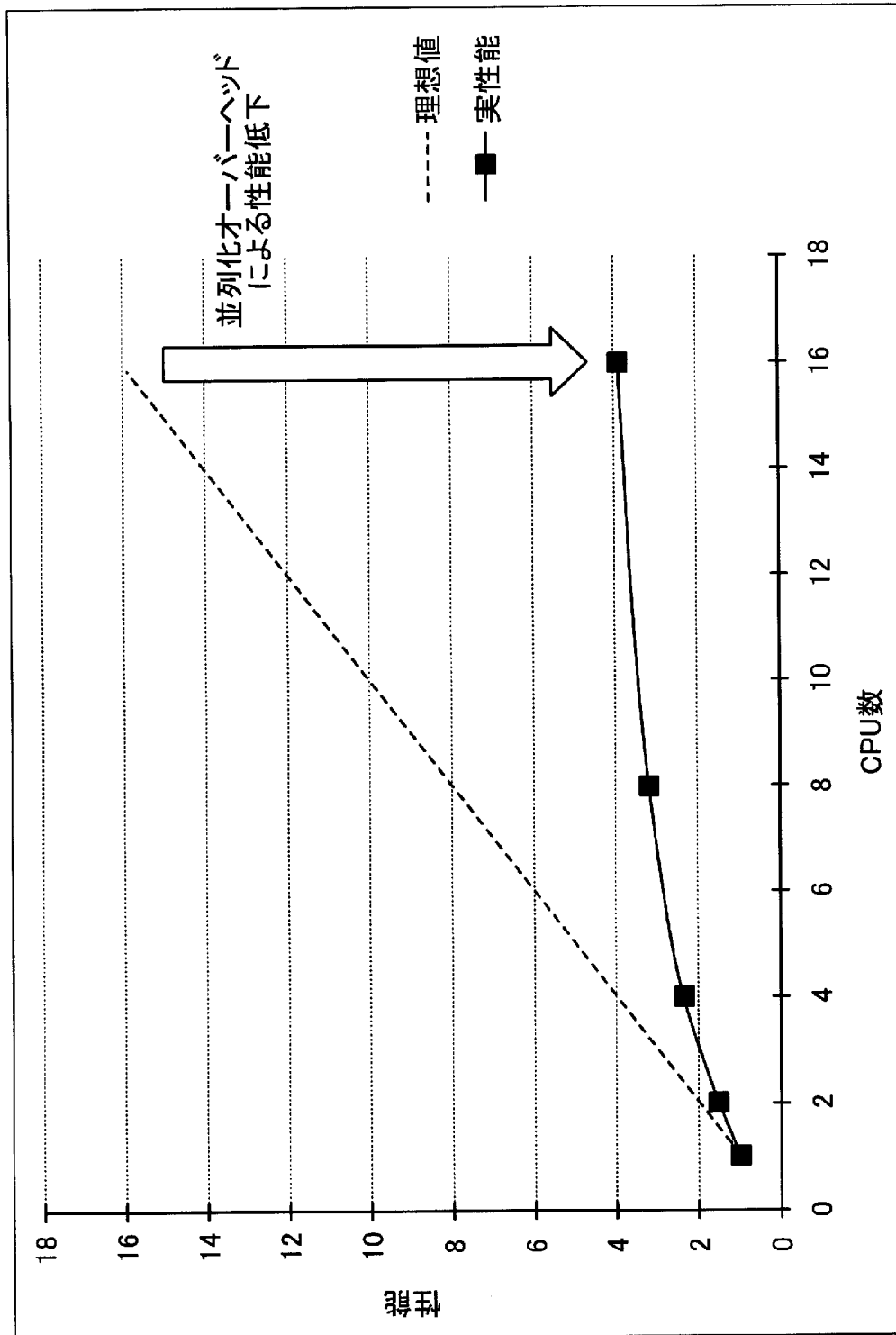
[図22]



[図23]



[図24]



INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP2011/052953

A. CLASSIFICATION OF SUBJECT MATTER

G06F9/48(2006.01) i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06F9/48

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Jitsuyo Shinan Koho 1922-1996 Jitsuyo Shinan Toroku Koho 1996-2011

Kokai Jitsuyo Shinan Koho 1971-2011 Toroku Jitsuyo Shinan Koho 1994-2011

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X A	JP 9-138716 A (Toshiba Corp.), 27 May 1997 (27.05.1997), paragraphs [0015] to [0026]; fig. 1, 2 & US 5913068 A1	1-4, 7-9 5, 6
X A	JP 2006-344162 A (Mitsubishi Electric Corp.), 21 December 2006 (21.12.2006), paragraphs [0019] to [0038]; fig. 1 to 3 (Family: none)	1, 5-7 2-4, 8, 9
A	JP 2006-11548 A (Sony Computer Entertainment Inc.), 12 January 2006 (12.01.2006), entire text; all drawings & US 2007/0143763 A1 & EP 1783608 A1 & WO 2005/124550 A1 & CN 1860447 A	1-9

☒ Further documents are listed in the continuation of Box C.☐ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search
07 April, 2011 (07.04.11)Date of mailing of the international search report
19 April, 2011 (19.04.11)Name and mailing address of the ISA/
Japanese Patent Office

Authorized officer

Facsimile No.

Telephone No.

INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP2011/052953

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	JP 2005-85164 A (Sharp Corp.) , 31 March 2005 (31.03.2005) , entire text; all drawings (Family: none)	1-9

A. 発明の属する分野の分類 (国際特許分類 (I P C))

Int.Cl. G06F9/48(2006.01) i

B. 調査を行った分野

調査を行った最小限資料 (国際特許分類 (I P C))

Int.Cl. G06F9/48

最小限資料以外の資料で調査を行った分野に含まれるもの

日本国実用新案公報	1 9 2 2 - 1 9 9 6 年
日本国公開実用新案公報	1 9 7 1 - 2 0 1 1 年
日本国実用新案登録公報	1 9 9 6 - 2 0 1 1 年
日本国登録実用新案公報	1 9 9 4 - 2 0 1 1 年

国際調査で使用した電子データベース (データベースの名称、調査に使用した用語)

C. 関連すると認められる文献

引用文献の カテゴリー*	引用文献名 及び一部の箇所が関連するときは、その関連する箇所の表示	関連する 請求項の番号
X A	JP 9-138716 A (株式会社東芝) 1997.05.27, 段落 (0 0 1 5) - (0 0 2 6), 第1図, 第2図 & US 5913068 A1	1-4, 7-9 5, 6
X A	JP 2006-344162 A (三菱電機株式会社) 2006.12.21, 段落 (0 0 1 9) - (0 0 3 8), 第1図-第3図 (ファミリーなし)	1, 5-7 2-4, 8, 9
A	JP 2006-11548 A (株式会社ソニー・コンピュータエンタテインメント) 2006.01.12, 全文, 全図 & US 2007/0143763 A1 & EP 1783608 A1 & WO 2005/124550 A1 & CN 1860447 A	1-9

☒ C欄の続きにも文献が列挙されている。☐ パテントファミリーに関する別紙を参照。

* 引用文献のカテゴリー

「A」特に関連のある文献ではなく、一般的技術水準を示すもの
「E」国際出願日前の出願または特許であるが、国際出願日以後に公表されたもの
「L」優先権主張に疑義を提起する文献又は他の文献の発行日若しくは他の特別な理由を確立するために引用する文献 (理由を付す)
「O」口頭による開示、使用、展示等に言及する文献
「P」国際出願日前で、かつ優先権の主張の基礎となる出願

の日の後に公表された文献

「T」国際出願日又は優先日後に公表された文献であって出願と矛盾するものではなく、発明の原理又は理論の理解のために引用するもの
「X」特に関連のある文献であって、当該文献のみで発明の新規性又は進歩性がないと考えられるもの
「Y」特に関連のある文献であって、当該文献と他の1以上の文献との、当業者にとって自明である組合せによって進歩性がないと考えられるもの
「&」同一パテントファミリー文献

国際調査を完了した日

0 7 . 0 4 . 2 0 1 1

国際調査報告の発送日

1 9 . 0 4 . 2 0 1 1

国際調査機関の名称及びあて先

日本国特許庁 (I S A / J P)

郵便番号 1 0 0 - 8 9 1 5

東京都千代田区霞が関三丁目4番3号

特許庁審査官 (権限のある職員)

鈴木 修治

5 B

3 5 6 0

電話番号 0 3 - 3 5 8 1 - 1 1 0 1 内線 3 5 4 5

C (続き) . 関連すると認められる文献		
引用文献の カテゴリー*	引用文献名 及び一部の箇所が関連するときは、その関連する箇所の表示	関連する 請求項の番号
A	JP 2005-85164 A (シャープ株式会社) 2005.03.31, 全文, 全図 (ファミリーなし)	1-9