

(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
2011年9月1日(01.09.2011)

PCT

(10) 国際公開番号
WO 2011/104823 A1

- (51) 国際特許分類:
G06F 9/52 (2006.01)
- (21) 国際出願番号: PCT/JP2010/052792
- (22) 国際出願日: 2010年2月23日(23.02.2010)
- (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): 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). 宮▲崎▼ 清志(MIYAZAKI, Kiyoshi) [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.
- (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, 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 (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, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

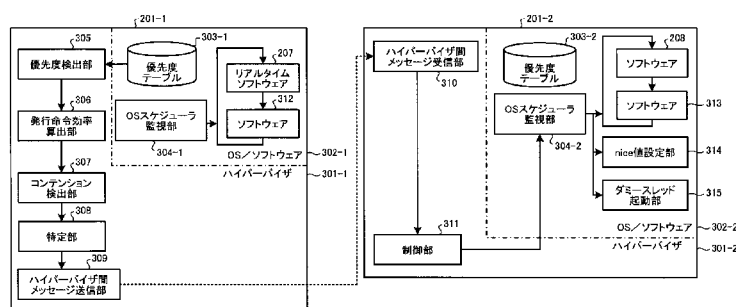
添付公開書類:

- 国際調査報告(条約第21条(3))

(54) Title: MULTI-CORE PROCESSOR SYSTEM, THREAD CONTROL METHOD, AND THREAD CONTROL PROGRAM

(54) 発明の名称: マルチコアプロセッサシステム、スレッド制御方法、およびスレッド制御プログラム

[図3]



- 207 Real-time software
301-1, 301-2 Hypervisor
302-1, 302-2 OS/software
303-1, 303-2 Priority table
304-1, 304-2 OS scheduler supervising unit
305 Priority detecting unit
306 Issued instruction efficiency calculating unit
307 Contention detecting unit
308 Specifying unit
309 Inter-hypervisor message transmitting unit
310 Inter-hypervisor message receiving unit
311 Control unit
208, 312, 313 Software
314 Nice value setting unit
315 Dummy thread starting unit

(57) Abstract: In CPU (201-1), a priority detecting unit (305) determines whether the CPU has the highest execution priority, an issued instruction efficiency calculating unit (306) calculates the efficiency of the issued instruction, and a contention detecting unit (307) detects contention. After said operations, CPU (201-1) is specified as the CPU for ensuring real-time processing by means of a specifying unit (308), and an inter-hypervisor message transmitting unit (309) transmits a message to all CPUs except the specified CPU. CPU (201-2) receives the message by means of the inter-hypervisor message receiving unit (310), and a control unit (311) controls a thread, that does not access a shared memory, in a manner such that said thread is executed for a predetermined time period among the time period in which contention occurred.

(57) 要約:

[続葉有]

WO 2011/104823 A1



CPU (201-1)は、優先度検出部 (305)によって実行優先度が最も高いCPUであるか検出し、発行命令効率算出部 (306)によって発行命令効率を算出し、コンテンション検出部 (307)によってコンテンションかを検出する。検出後、CPU (201-1)は、特定部 (308)にてリアルタイム処理を保証するCPUであることを特定し、ハイパーバイザ間メッセージ送信部 (309)によって、特定されたCPUを除いたすべてのCPUにメッセージを送信する。CPU (201-2)は、ハイパーバイザ間メッセージ受信部 (310)によってメッセージを受信し、制御部 (311)によって共有メモリにアクセスしないスレッドを、コンテンションが発生した期間のうち、所定の期間実行するように制御する。

明 細 書

発明の名称：

マルチコアプロセッサシステム、スレッド制御方法、およびスレッド制御プログラム

技術分野

[0001] 本発明は、スレッドを制御するマルチコアプロセッサシステム、スレッド制御方法、およびスレッド制御プログラムに関する。

背景技術

[0002] 従来から、1つのCPUに対して、複数のプログラムを動作させるマルチプログラミング技術が存在する。具体的には、OS（Operating System）は、CPU（Central Processing Unit）の処理時間を分割する機能を有し、分割された時間にプロセスやスレッドを割り当てることにより、CPUが同時に複数のプロセスやスレッドを動作する。ここで、プロセスはプログラムの実行単位である。

[0003] また、一定時間内に処理を終える必要があるプロセスに対して、OSがCPUに対するスレッドの割り当て時間を他のプロセスより多くすることで、優先的に処理を行い、一定時間内に処理を行う、という技術も開示されている。また、プロセスを効果的に切り替える技術として、OSがプロセスごとに実行命令数を取得し、実行命令数の多いプロセスを先に実行する技術が開示されている（たとえば、下記特許文献1を参照。）。前述の技術では、キャッシュメモリに一番多く保持されているプロセスが実行されることにより、全体のスループットを向上させることができる。

[0004] また、コンピュータシステムに複数のCPUを搭載するマルチコアプロセッサシステムの技術も開示されている。これにより、前述のマルチプログラミング技術において、OSは複数のプログラムを複数のプロセスに対して割り当てることができる。また、マルチコアプロセッサシステムの構成として、各CPUが専用のメモリを保持し、それ以外のデータが必要な場合には、

共有メモリにアクセスを行うことを特徴とする、分散システムの構造を持ったマルチコアプロセッサシステムが開示されている。また、各CPUがキャッシュメモリのみを保持し、必要なデータは共有メモリに格納することを特徴とする、集中共用システムの構造を持ったマルチコアプロセッサシステムも開示されている。

先行技術文献

特許文献

[0005] 特許文献1：特開平9-330237号公報

発明の概要

発明が解決しようとする課題

[0006] しかしながら、マルチコアプロセッサシステムでは、複数のCPUが共有メモリに同時にアクセスすることによってコンテンションが発生する。コンテンションが発生すると、CPUは、通常の処理時間以内に処理を終了できなくなり、一定時間内に処理を終わらせる必要があるリアルタイム処理を行えないという問題があった。リアルタイム処理とは、設計上あらかじめ決められた時刻に処理を終了しなければならない処理、および割り込み動作における、割り込みイベント発生から割り込み処理本体の開始時間までの許容されるインターバル時間が定められた処理をさす。

[0007] また、コンテンションはハードウェアを原因として引き起こされる。したがって、上述した従来技術において、マルチコアプロセッサシステムに特許文献1を適用しても、CPUはキャッシュメモリに一番多く保持されているプロセスによってコンテンションを起こす可能性もあり、コンテンションの解決に至らないという問題があった。

[0008] また、前述した分散システムを適用した場合、コンテンションが発生する頻度は少ないが、CPUごとにメモリを配置する必要があるため、コストと消費電力が大きくなるという問題があった。したがって、コストと消費電力に制限のある組み込み環境では、集中共用システムを適用したマルチコアプ

ロセッサシステムがよく適用される。しかし、集中共用システムを適用したマルチコアプロセッサシステムは、複数のCPUが共有メモリに同時にアクセスする機会が多く、コンテンションが発生する頻度が多いという問題があった。

[0009] 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、マルチコアプロセッサシステムにてリアルタイム処理を保証できるマルチコアプロセッサシステム、スレッド制御方法、およびスレッド制御プログラム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

[0010] 上述した課題を解決し、目的を達成するため、開示のマルチコアプロセッサシステムは複数のコアのうち、実行優先度が最も高い第1のコアを検出し、検出された第1のコアのうち、メモリに対してアクセス競合を発生させた第2のコアを特定し、複数のコアのうち、第1のコアと第2のコアとを除いたすべてのコアに対し、メモリにアクセスしないスレッドを、アクセス競合の期間のうち少なくとも一部の期間中に実行させるように制御することを備えることを要件とする。

発明の効果

[0011] 本マルチコアプロセッサシステム、スレッド制御方法、およびスレッド制御プログラムによれば、リアルタイム処理を行っており、コンテンション発生中のコアのリアルタイム処理を保証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

[0012] [図1]実施の形態にかかるマルチコアプロセッサシステムのハードウェア構成を示すブロック図である。

[図2]マルチコアプロセッサシステム100の各CPUのハードウェアの一部の構成とソフトウェア構成を示すブロック図である。

[図3]マルチコアプロセッサシステム100の機能的構成を示すブロック図である。

[図4] コンテンション状態を示す説明図である。

[図5] コンテンションが解消された状態を示す説明図である。

[図6] 本実施の形態を適用したマルチコアプロセッサシステム100の性能比を示す説明図である。

[図7] 優先度テーブル303-1の記憶内容の一例を示す説明図である。

[図8] ハイパーバイザによるメッセージ送信処理を示すフローチャートである。

[図9] ハイパーバイザによるメッセージ受信処理を示すフローチャートである。

発明を実施するための形態

[0013] 以下に添付図面を参照して、本発明にかかるマルチコアプロセッサシステム、スレッド制御方法、およびスレッド制御プログラムの好適な実施の形態を詳細に説明する。

[0014] (マルチコアプロセッサシステムのハードウェア構成)

図1は、実施の形態にかかるマルチコアプロセッサシステムのハードウェア構成を示すブロック図である。マルチコアプロセッサシステムとは、コアが複数搭載されたプロセッサを含むコンピュータのシステムである。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であるCPUが並列されているプロセッサ群を例にあげて説明する。

[0015] マルチコアプロセッサシステム100は、CPUを複数搭載するCPUs 101と、ROM (Read - Only Memory) 102と、RAM (Random Access Memory) 103と、フラッシュROM 104と、を備えている。また、マルチコアプロセッサシステム100は、ユーザやその他の機器との入出力装置として、ディスプレイ105と、I/F (Interface) 106と、を備えている。また、各構成部はバス108によってそれぞれ接続されている。本実施の形態にかかるハードウ

エアの構成は、集中共用システムを適用した構成となっている。

[0016] ここで、CPU 101は、マルチコアプロセッサシステムの全体の制御を司る。また、CPU 101は、シングルコアのプロセッサを並列して接続したすべてのCPUを指している。詳細は、図2にて後述する。ROM 102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 103は、CPU 101のワークエリアとして使用される。

[0017] フラッシュROM 104は、書き換えが可能であり、電源を切ってもデータが消えないという特徴を持つ不揮発性の半導体メモリである。フラッシュROM 104は、ソフトウェアプログラムや、データを記憶している。フラッシュROM 104の代わりに、磁気ディスクであるHDD（ハードディスクドライブ）にて記憶してもよいが、フラッシュROM 104を使用することにより、機械的に動作するHDDに比べて振動に強くすることができる。たとえば、マルチコアプロセッサシステム100で構成する装置に対し、強い振動があった場合でも、フラッシュROM 104であればデータが消える可能性を低くすることができる。

[0018] ディスプレイ105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。ディスプレイ105は、たとえば、TFT液晶ディスプレイなどを採用することができる。また、ディスプレイ105は、タッチパネル式で入力する形態でもよい。

[0019] I/F 106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107に接続され、このネットワーク107を介して他の装置に接続される。そして、I/F 106は、ネットワーク107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 106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。

[0020] 図2は、マルチコアプロセッサシステム100の各CPUのハードウェアの一部の構成とソフトウェア構成を示すブロック図である。マルチコアプロ

セッサシステム100のハードウェア構成は、CPU 101と、共有メモリ203で構成される。CPU 101は、複数のCPUとしてCPU 201-1、CPU 201-2、・・・、CPU 201-nで構成される。

[0021] CPU 201-1、CPU 201-2、・・・、CPU 201-nは、それぞれ、キャッシュメモリ202-1、キャッシュメモリ202-2、・・・、キャッシュメモリ202-nを保持している。各CPUと共有メモリ203は、バス108によってそれぞれ接続されている。以下の記述では、CPU 201-1、CPU 201-2について記述する。

[0022] また、マルチコアプロセッサシステム100のソフトウェア構成として、CPU 201-1は、ハイパーバイザ204-1、OS 205-1を実行する。CPU 201-1は、ハイパーバイザ204-1の制御により、監視ライブラリ206-1を実行する。同様に、CPU 201-1は、OS 205-1の制御により、リアルタイムソフトウェア207を実行する。同様にCPU 201-2も、ハイパーバイザ204-2、OS 205-2を実行する。CPU 201-2は、ハイパーバイザ204-2の制御により、監視ライブラリ206-2を実行する。同様に、CPU 201-2は、OS 205-2の制御により、ソフトウェア208を実行する。

[0023] CPU 201-1がリアルタイムソフトウェア207を実行する際には、データへのアクセス先が2通りあり、アクセス経路209とアクセス経路210である。同様に、CPU 201-2がソフトウェア208を実行する際にも、データへのアクセス先が2通りあり、アクセス経路211とアクセス経路212である。また、ハイパーバイザ204-1とハイパーバイザ204-2と、他のCPU上で動作しているハイパーバイザは、ハイパーバイザ間通信213を行う。

[0024] CPU 201-1、CPU 201-2、・・・、CPU 201-nは、マルチコアプロセッサシステム100の制御を司る。CPU 201-1、CPU 201-2、・・・、CPU 201-nは、対称的、均一的に処理が割り付けられたSMP (Symmetric

M u l t i - p r o c e s s i n g) でもよい。また、C P U 2 0 1 - 1、C P U 2 0 1 - 2、・・・、C P U 2 0 1 - n は、処理内容によって分担するC P Uを決めておくA S M P (A s y m m e t r i c M u l t i - P r o c e s s i n g) でもよい。A S M Pの例として、マルチコアプロセッサシステム100は、C P U 2 0 1 - 1にて優先して処理すべきであるリアルタイム処理を割り当ててもよい。

[0025] 共有メモリ203は、C P U 2 0 1 - 1、C P U 2 0 1 - 2、・・・、C P U 2 0 1 - nからアクセス可能な記憶領域である。記憶領域とは、具体的には、たとえば、ROM102、RAM103、フラッシュROM104である。また、たとえば、C P U 2 0 1 - 1が、ディスプレイ105に画像データの表示を要求する場合、RAM103に含まれるVRAM (V i d e o R A M) にアクセスし、画像データをVRAMに書き込むことになる。したがって、C P U 2 0 1 - 1が、ディスプレイ105へアクセスする場合も、共有メモリ203にアクセスすることに含める。

[0026] また、たとえば、C P U 2 0 1 - 1が、I / F 106にアクセスする場合も同様である。たとえば、I / F 106の具体例がLANアダプタとして、LANアダプタにあるバッファにアクセスする形式か、RAM103にアクセスし、その後LANアダプタに転送する形式かのどちらかになる。どちらの場合であっても、C P U 2 0 1 - 1、C P U 2 0 1 - 2からみると共有メモリにアクセスしていることとなるため、C P U 2 0 1 - 1、C P U 2 0 1 - 2が、I / F 106へアクセスする場合も、共有メモリ203にアクセスすることに含める。同様に、C P U 2 0 1 - 1が、I / F 106にアクセスする場合も、I / F 106を制御するデバイスドライバが用意する共有の記憶領域にアクセスすることになるため、結果、共有メモリ203にアクセスすることになる。

[0027] ハイパーバイザ204-1、ハイパーバイザ204-2は、それぞれC P U 2 0 1 - 1、C P U 2 0 1 - 2にて動作するプログラムである。ハイパーバイザの機能は、OSとC P Uの間にて位置し、OSの監視を行い、OSが

ハングアップした際にリセットをする他、OSが何もスレッドを実行していない場合に、省電力設定にする。また、CPU201-1、CPU201-2は、それぞれのハイパーバイザによって、本実施の形態での特徴である、コンテンションの監視をする監視ライブラリ206-1、監視ライブラリ206-2を実行する。

[0028] OS205-1、OS205-2は、それぞれ、CPU201-1、CPU201-2にて動作するプログラムであり、ハイパーバイザ204-1、ハイパーバイザ204-2上で動作する。たとえば、OS205-1は、リアルタイムソフトウェア207をCPU201-1に割り当てて実行させるスレッドのスケジューラを持つ。

[0029] 監視ライブラリ206-1、監視ライブラリ206-2は、それぞれ、ハイパーバイザ204-1、ハイパーバイザ204-2上で動作するプログラムであり、共有メモリ203に対してアクセス競合によるコンテンションが発生していないかを監視する。監視の結果、コンテンションが発生している場合、監視ライブラリは、ハイパーバイザ間で情報を送信し、コンテンションを起こしたことを、他のハイパーバイザに送信する。

[0030] リアルタイムソフトウェア207は、OS205-1によってCPU201-1に割り当てられたプログラムである。リアルタイムソフトウェアの具体例としては、通信パケット処理がある。通信パケット処理は、プロトコルの決められた時間以内に処理を行う必要があり、リアルタイム処理を要求される。ソフトウェア208は、OS205-2によってCPU201-2に割り当てられたプログラムである。ソフトウェア208は、リアルタイム処理を要求されていない。前述のように、本実施の形態では、CPU201-1にてリアルタイム処理の保証を必要とするソフトウェアを実行している状態を想定する。

[0031] アクセス経路209は、CPU201-1がキャッシュメモリ202-1にアクセスする経路である。アクセス経路210は、CPU201-1が共有メモリ203にアクセスする経路である。アクセス経路209とアクセス

経路 210 の違いとして、リアルタイムソフトウェア 207 がアクセスしたいデータがキャッシュメモリ 202-1 にあればアクセス経路 209、なければアクセス経路 210 となる。アクセス経路 211、アクセス経路 212 も同様で、アクセス経路 211 は、CPU 201-2 がキャッシュメモリ 202-2 にアクセスする経路である。アクセス経路 212 は、CPU 201-2 が共有メモリ 203 にアクセスする経路である。

[0032] ハイパーバイザ間通信 213 は、ハイパーバイザ間でメッセージの送受信をする通信である。具体的には、たとえば、CPU 201-1 にて、リアルタイムソフトウェア 207 を実行中にコンテンション状態になった場合に、ハイパーバイザ 204-1 からハイパーバイザ 204-2 を含むすべてのハイパーバイザにメッセージをブロードキャスト送信する。

[0033] (マルチコアプロセッサシステムの機能的構成)

次に、マルチコアプロセッサシステム 100 の機能的構成について説明する。図 3 は、マルチコアプロセッサシステム 100 の機能的構成を示すブロック図である。マルチコアプロセッサシステム 100 は、優先度検出部 305 と、発行命令効率算出部 306 と、コンテンション検出部 307 と、特定部 308 と、制御部 311 と、を含む構成である。この制御部となる機能（優先度検出部 305 ~ 制御部 311）は、具体的には、たとえば、図 1 に示した ROM 102、RAM 103、フラッシュ ROM 104 などの記憶装置に記憶されたプログラムを CPUs 101 が実行することにより、その機能を実現する。

[0034] また、CPU 201-1、CPU 201-2、・・・、CPU 201-n は、ハイパーバイザと OS / ソフトウェアを実行する。さらに、一点破線で分割された領域のうち、領域 301-1 に図示している優先度検出部 305 ~ ハイパーバイザ間メッセージ送信部 309 は、CPU 201-1 がハイパーバイザ 204-1 の機能の一部として実行することにより実現する。同様に、領域 301-2 内に図示しているハイパーバイザ間メッセージ受信部 310、制御部 311 は、CPU 201-2 が、ハイパーバイザ 204-2 の

機能の一部として実行することにより実現する。

[0035] また、図示していないが、CPU 201-1以外のコアが実行しているハイパーバイザにも、優先度検出部305～ハイパーバイザ間メッセージ送信部309の機能を有している。同様に、CPU 201-2以外のコアが実行しているハイパーバイザにも、ハイパーバイザ間メッセージ受信部310、制御部311の機能を有している。また、優先度検出部305～ハイパーバイザ間メッセージ送信部309は、監視ライブラリ206-1に相当する。同様に、ハイパーバイザ間メッセージ受信部310、制御部311は、監視ライブラリ206-2に相当する。

[0036] また、領域302-1に図示しているOSスケジューラ監視部304-1、リアルタイムソフトウェア207、ソフトウェア312は、CPU 201-1がOS 205-1の機能の一部として実行することにより実現する。優先度テーブル303-1は、ハイパーバイザ204-1またはOS 205-1からアクセスできるテーブルである。

[0037] また、領域302-2に図示しているOSスケジューラ監視部304-2、ソフトウェア208、ソフトウェア313、nice値設定部314、ダミースレッド起動部315は、CPU 201-2がOS 205-2の機能の一部として実行することにより実現する。優先度テーブル303-2は、ハイパーバイザ204-2またはOS 205-2からアクセスできるテーブルである。

[0038] 優先度テーブル303-1、優先度テーブル303-2は、マルチコアプロセッサシステム100で実行される処理と処理の優先度を関連付けて管理するテーブルである。優先度テーブル303-1の内容の詳細は、図7にて後述する。

[0039] OSスケジューラ監視部304-1、OSスケジューラ監視部304-2は、CPU 201-1、CPU 201-2に割り当てるソフトウェアを監視する機能を有する。具体的には、たとえば、リアルタイムソフトウェア207がCPU 201-1に割り当てられており、リアルタイムソフトウェア2

０７がＲＡＭ１０３やフラッシュＲＯＭ１０４上にある共有資源にアクセス要求する場合を想定する。

[0040] この時、既に他のソフトウェアが共有資源を使用すると宣言中であった場合に、ＯＳスケジューラ監視部３０４－１は、リアルタイムソフトウェア２０７の実行状態を待ち状態にする。続けて、ＯＳスケジューラ監視部３０４－１は、実行可能状態となっていた別のソフトウェア、たとえばソフトウェア３１２を実行状態にして、ＣＰＵ２０１－１に割り当てる。

[0041] また、別の具体例として、たとえば、リアルタイムソフトウェア２０７が所定期間以上ＣＰＵ２０１－１に割り当てられていた場合にも、ＯＳスケジューラ監視部３０４－１は、別のソフトウェアをＣＰＵ２０１－１に割り当てる。また、前述のように、ＣＰＵに割り当てられたソフトウェアを切り替えることを、ディスパッチという。

[0042] また、ＯＳスケジューラ監視部３０４－１は、新しくソフトウェアを起動する際に、ソフトウェアの実行単位となるスレッドとして起動する。各スレッドは、スタック領域、プログラムカウンタを含むレジスタ情報等を持つ。ＯＳスケジューラ監視部３０４－１は、ディスパッチを行うたびに、現在実行中のレジスタ情報等を共有メモリ２０３に退避し、次のソフトウェアのレジスタ情報等を共有メモリ２０３から取得し、ＣＰＵのレジスタ情報に設定する。

[0043] また、マルチコアプロセッサシステム１００は、スレッドの集合から、１つのプロセスを構成してもよい。スレッド間ではメモリ空間が共通であるが、プロセス間ではメモリ空間が独立しており、お互いのメモリ空間に直接アクセスできない。本実施の形態では、スレッドを用いて説明しているが、プロセスに置き換えてもよい。

[0044] 別のソフトウェアをＣＰＵ２０１－１に割り当てる際に、複数の割り当て候補となるソフトウェアが存在する場合、ＯＳスケジューラ監視部３０４－１は、優先度テーブル３０３－１に基づいて割り当てを行ってもよい。また、ＯＳスケジューラ監視部３０４－１は、それぞれのソフトウェアの割り当

て時刻に基づいて、割り当て時刻が一番古いソフトウェアを割り当ててもよい。

[0045] 優先度検出部 305 は、複数のコアのうち、実行優先度が最も高い第 1 のコアを検出する機能を有する。ここでのコアは、CPU 101 を構成している CPU 201-1、CPU 201-2、・・・、CPU 201-n に相当する。また、マルチコアプロセッサシステム 100 が ASMP であり、リアルタイム処理が割り当てられた CPU が存在するならば、CPU によって第 1 のコアを検出してもよい。

[0046] 具体的には、たとえば、CPU 201-1 は、現在割り当てられているソフトウェアの優先度を、優先度テーブル 303-1 から取得し、優先度が“リアルタイム”の場合に、実行優先度が最も高い第 1 のコアとする。なお、検出した第 1 のコアの情報は、キャッシュメモリ 202-1、CPU 201-1 の汎用レジスタなどの記憶領域に記憶される。

[0047] 発行命令効率算出部 306 は、コアごとに、コアが命令を発行した発行命令数とコアのサイクル数に基づいて、発行命令効率を算出する機能を有する。発行命令数は、所定時間内に CPU が命令を行った数である。発行命令数は CPU の特殊なレジスタである発行命令カウンタ I に格納されており、ハイパーバイザは、スーパバイザモードに移行することにより、発行命令カウンタ I の値を取得する。サイクル数は、所定時間内に CPU が入力されたクロックの数である。サイクル数は CPU のレジスタであるクロックカウンタ C に格納されている。発行命令効率は、1 命令にかかったクロック数であり、 C/I にて算出される。発行命令効率算出部 306 は、発行命令効率を I/C として算出し、後述する閾値 τ も逆数にして比較してもよい。

[0048] 具体的には、たとえば、ハイパーバイザ 204-1 が起動するたびに、CPU 201-1 は、 C/I を算出する。ハイパーバイザは、数十マイクロ秒～数ミリ秒に 1 回の間隔で実行されるため、その時間での発行命令カウンタ I とクロックカウンタ C を取得し、 C/I を算出する。なお、算出された発行命令効率は、キャッシュメモリ 202-1、CPU 201-1 の汎用レジ

スタなどの記憶領域に記憶される。

[0049] コンテンション検出部 307 は、発行命令効率算出部 306 によって算出された発行命令効率と所定の閾値とに基づいて、アクセス競合を検出する機能を有する。所定の閾値とは、仕様から設定できる値であり、 τ で表す。閾値 τ の具体的な設定方法は、図 4 にて後述する。具体的には、たとえば、発行命令数が閾値 τ より大きい場合、CPU 201-1 は、共有メモリ 203 に対するアクセス競合によるコンテンツションが発生していることを検出する。なお、検出された結果は、キャッシュメモリ 202-1、CPU 201-1 の汎用レジスタなどの記憶領域に記憶される。

[0050] 特定部 308 は、複数のコアのうち、優先度検出部 305 によって検出された第 1 のコアのうち、共有メモリ 203 に対してアクセス競合を発生させた第 2 のコアを特定する機能を有する。また、特定部 308 は、共有メモリ 203 に対してアクセス競合を発生させており第 2 のコアと競合する第 3 のコアを特定してもよい。アクセス競合によるコンテンツション発生を検出する際に CPU 201-1 は、コンテンツション検出部 307 によって検出してもよい。

[0051] 具体的には、たとえば、CPU 201-1 は、優先度検出部 305 によって“リアルタイム”である CPU から、コンテンツションが発生している CPU を検出して、第 2 のコアとして特定する。また、CPU 201-1 は、複数の CPU からコンテンツションが発生している CPU を検出して、第 3 のコアとして特定してもよい。なお、特定された第 2 のコア、または第 3 のコアの情報は、キャッシュメモリ 202-1、CPU 201-1 の汎用レジスタなどの記憶領域に記憶される。

[0052] ハイパーバイザ間メッセージ送信部 309 は、他のハイパーバイザにメッセージをブロードキャスト送信する機能を有する。具体的には、たとえば、リアルタイム処理を行っており、コンテンツション発生を検出したハイパーバイザ 204-1 は、バス 108 を通じてハイパーバイザ 204-2 や、その他のハイパーバイザに対して、メッセージをブロードキャスト送信する。な

お、送信したメッセージの内容は、キャッシュメモリ202-1、CPU201-1の汎用レジスタなどの記憶領域に記憶されてもよい。

[0053] ハイパーバイザ間メッセージ受信部310は、他のハイパーバイザが送信したメッセージを受信する機能を有する。具体的には、たとえば、ハイパーバイザ204-2は、リアルタイム処理を行っておりコンテンション発生を検出したハイパーバイザ204-1からメッセージを受信する。なお、受信したメッセージの内容は、キャッシュメモリ202-2、CPU201-2の汎用レジスタなどの記憶領域に記憶される。

[0054] 制御部311は、複数のコアのうち、第1のコアと特定部308によって特定された第2のコアを除いた第3のコアに対し、共有メモリ203にアクセスしないスレッドを実行するように制御する機能を有する。また、制御部311は、第2のコアを除いた第3のコアに対し、共有メモリ203にアクセスしないスレッドを実行するように制御してもよい。また、特定部308によって第3のコアが特定されている場合、制御部311は、特定された第3のコアに対し、共有メモリ203にアクセスしないスレッドを実行するように制御してもよい。

[0055] また、共有メモリ203にアクセスしないスレッドを実行する期間は、アクセス競合が発生した期間のうち、所定の期間である。所定の期間とは、OS205-2が保持しているタイムスライスの値となる。また、所定の期間は、アクセス競合が発生した期間のうち、第3のコアに割り当てられていたスレッドと、共有メモリ203にアクセスしないスレッドとで、時分割した期間としてもよい。

[0056] 具体的には、たとえば、特定された第2のコアは、ハイパーバイザ間通信を行うため、メッセージを受信したCPUが第3のコアとなり、共有メモリ203にアクセスしないスレッドを実行するようにOSスケジューラ監視部304-2を制御する。制御内容としては、制御部311は、nice値設定部314か、または、ダミースレッド起動部315を実行するようにOSスケジューラ監視部304-2を制御する。

- [0057] `nice`値設定部314は、現在実行されているソフトウェアの`nice`値を設定する機能を有する。`nice`値とは、POSIX (Portable Operating System Interface for UNIX (登録商標)) にて定義されている`nice`コマンドで設定する値である。`nice`コマンドによって設定値を変更することにより、OS205-2はソフトウェアの実行優先度を制御する。
- [0058] 具体的には、たとえば、リアルタイム処理を要求されないソフトウェアに対して、`nice`値を上昇させると、優先度が低くなる。`nice`コマンドの実装の一例としては、OS205-2は、ソフトウェアの割り当て終了時刻に`nice`値を加算した値を算出する。次に、OSスケジューラ監視部304-2は、加算した値に基づいて、一番小さい値を持つソフトウェアをディスパッチ対象に決める、という方法をとってもよい。
- [0059] 結果、対象のソフトウェアの`nice`値が大きいほど対象のソフトウェアの優先度が低くなる。したがって、もしOS205-2がPOSIX仕様に準拠していなく、`nice`コマンドが存在しない場合でも、前述の処理を加えることで`nice`値設定部314を実現してもよい。なお、設定された値は、RAM103、フラッシュROM104などの記憶領域に記憶される。
- [0060] ダミースレッド起動部315は、共有メモリ203にアクセスしないスレッドを生成する機能を有する。具体的には、たとえば、CPU201-2は、CPUに対して何も動作しないコードである`nop`を一定時間行うスレッドを起動する。また、`nice`値設定部314とダミースレッド起動部315は、OSスケジューラ監視部304-2によってアクセス競合によるコンテンション発生の際のうち少なくとも一部の期間中に実行する。
- [0061] 図4は、コンテンション状態を示す説明図である。初めに、CPU201-1は、ハイパーバイザ204-1とリアルタイムソフトウェア207を実行しており、CPU201-2は、ハイパーバイザ204-2とソフトウェア208を実行している。それぞれのCPUは、実行中のソフトウェアによってキャッシュメモリにアクセスするか、または、共有メモリ203にアク

セスする。

[0062] ハイパーバイザは周期的に起動しており、起動間隔は、数十マイクロ秒から、数ミリ秒である。図4では、どちらのメモリにアクセスしたかという動作に従って、示した時間を、時間401、時間402、時間403に分割して示している。時間401と時間403にて、CPU201-1とCPU201-2は、リアルタイムソフトウェア207と、ソフトウェア208によって同時に共有メモリ203にアクセスしていないため、コンテンション状態とはならない。

[0063] しかし、時間402にて、CPU201-1とCPU201-2は、同時に共有メモリ203にアクセスしているため、共有メモリ203に対してアクセス競合によるコンテンション状態となっている。コンテンション状態になると、CPU201-1は、メモリアクセスにかかる時間が数百サイクルとなり、リアルタイムソフトウェア207の処理遅延を発生させてしまう。結果、CPU201-1は、リアルタイムソフトウェア207に求められている時刻までに処理を終了することができない可能性があり、リアルタイム処理の保証を行えない状態となる。

[0064] 次に、リアルタイム処理の保証について説明する。リアルタイム処理は、決められた時間以内に応答を返す必要があり、その時間を、 Δ [秒] とする。ここで、リアルタイム処理を行うCPU201-1のクロックサイクルを clk [1/秒] とする。したがって、時間 Δ を消費する間に許されるCPU201-1のクロックのカウント数は $\Delta \cdot clk$ [個] となる。もしコンテンション状態になっており、CPU201-1が $\Delta \cdot clk$ [個] のカウント数で1命令を実行できない場合に、リアルタイム処理の保証を行えないことになる。

[0065] 1命令あたりのクロック数は、一定時間内での発行命令カウンタIと、クロックカウンタCに基づいて、 C/I を算出することで求められる。ここで、閾値 τ を、 $\tau = \Delta \cdot clk$ で示す。 C/I が閾値 τ 以下の場合、CPU201-1は、リアルタイム処理の保証を行える状態であり、 C/I が閾値 τ

より大きい場合、CPU 201-1は、リアルタイム処理の保証を行えない状態である。

[0066] Δ やclkは、仕様策定時に決定可能な値であるため、閾値 τ も仕様策定時に決定可能である。具体的には、たとえば、 $\Delta=2$ [マイクロ秒]、clk=500 [MHz]である場合、 $\tau=1000$ となる。通常、CPU 201-1は、共有メモリ203へのアクセスは数十カウント消費する。しかし、共有メモリ203へのアクセス競合によるコンテンション発生時には、数十～数百カウント消費し、CPU 201-1の動作効率は最大でピーク時の30%にまで低減することがある。

[0067] 図5は、コンテンションが解消された状態を示す説明図である。図5も図4と同様に、CPU 201-1は、ハイパーバイザ204-1とリアルタイムソフトウェア207を実行しており、CPU 201-2は、ハイパーバイザ204-2とソフトウェア208を実行している。図4では、時間402にて、CPU 201-1とCPU 201-2が同時に共有メモリ203にアクセスし、アクセス競合によるコンテンションが発生していた。

[0068] しかし、図5での時間402でのCPU 201-2は、ソフトウェア208とダミースレッドを交互に実行することにより、アクセス競合によるコンテンションを解消している。結果、CPU 201-1は、リアルタイムソフトウェア207に求められている時刻までに処理を終了することができ、リアルタイム処理を保証できる。

[0069] 図6は、本実施の形態を適用したマルチコアプロセッサシステム100の性能比を示す説明図である。図6の横軸は、バス108に設定したバッファ段数であり、縦軸は従来例のバッファ段数1を基準とした性能比である。従来例のバッファ段数1と等しい性能の場合、縦軸が1.00にプロットされることになる。本実施の形態にかかるマルチコアプロセッサシステム100に関して、バッファ段数ごとに、従来例との性能比をプロットして曲線で結んだ結果が曲線601である。また、 $\Delta=1$ [ミリ秒]、clk=600 [MHz]である。同様に、従来例にかかるマルチコアプロセッサシステムに

関して、バッファ段数ごとに、従来例との性能比をプロットして曲線で結んだ結果が曲線 602 である。

[0070] また、曲線 601、曲線 602 は、従来例のバッファ段数 1 を基準として、それより性能比がよい場合には領域 603 に位置し、悪い場合には領域 604 に位置する。領域 603 に位置する場合、マルチコアプロセッサシステム 100 はリアルタイム処理を保証でき、領域 604 に位置する場合、リアルタイム処理を保証できないことになる。マルチコアプロセッサシステムは、バスのバッファ段数が増えるとバス利用の効率をあげることができるが、リアルタイム処理を保証することが難しくなる。

[0071] 従来例にかかる曲線 602 は、バッファ段数が 5 段以上では領域 604 に位置している。したがって、従来例にかかるマルチコアプロセッサシステムは、バッファ段数が 5 段以上になると、リアルタイム処理保証を行えない。本実施の形態にかかる曲線 601 では、バッファ段数が 13 段となるまで領域 603 に位置している。したがって、本実施の形態にかかるマルチコアプロセッサシステム 100 は、バッファ段数が 13 段まで、リアルタイム処理を保証することができる。

[0072] 図 7 は、優先度テーブル 303-1 の記憶内容の一例を示す説明図である。優先度テーブル 303-1 は、処理名称フィールドと、実行優先度フィールドで構成する。なお、優先度テーブル 303-2 も同様のデータが設定されている。処理名称フィールドは、具体的な処理の内容を記述している。実際には、処理内容を記述したプログラムが ROM 102、RAM 103、フラッシュ ROM 104 のいずれかに存在し、CPU 201-1 は、プログラムをロードし、スレッドとして実行する。実行優先度フィールドは、対応する処理名称の実行する際の優先度を設定している。

[0073] たとえば、“通信パケット受信”処理は、一定時間内にパケット処理を行わないと、タイムアウトになるため、リアルタイム処理を保証する必要がある。したがって、実行優先度フィールドは、“リアルタイム”となる。続けて、“描画レンダリング”処理は、通常の処理であって、リアルタイム処理

を保証する必要がない。したがって、実行優先度フィールドは、“通常”となる。同様に、“UI入力”処理は、仕様によりユーザに対する応答時間を決められている場合、リアルタイム処理を保証する必要がある。“辞書の先読み検索”処理は、リアルタイム処理を保証する必要がない。

[0074] コンテンションが起こる状態として、たとえば、マルチコアプロセッサシステム100が、Webブラウジング処理を行っている状態を想定する。前述の状態でCPU201-1は、通信パケット受信処理を実行しており、CPU201-2は、描画レンダリング処理を実行している。描画レンダリング処理は、メモリアクセスが多く、通信パケット受信処理と共有メモリ203に対するアクセス競合を起こす可能性が高い。

[0075] 本実施の形態を適用した状態では、前述の状態のようにアクセス競合によるコンテンツが発生した際に、CPU201-2が、OS205-2によって描画レンダリング処理の*n i c e*値を上昇させる。OS205-2は、*n i c e*値を上昇させた描画レンダリング処理をよりまばらになるようにCPU201-2に割り当てる。結果、マルチコアプロセッサシステム100は、アクセス競合によるコンテンツを回避でき、描画レンダリング処理のリアルタイム処理を保証できる。

[0076] また別のコンテンツ状態として、たとえば、マルチコアプロセッサシステム100が、ユーザからの文字入力を受け付けている状態を想定する。前述の状態で、CPU201-1は、UI入力処理を実行しており、CPU201-2は、辞書の先読み検索処理を実行している。辞書の先読み検索処理は、I/Oアクセスが多く、通信パケット受信処理と共有メモリ203に対するアクセス競合を起こす可能性が高い。

[0077] 本実施の形態を適用した状態では、前述の状態のようにアクセス競合によるコンテンツが発生した際に、CPU201-2が、OS205-2によって辞書の先読み検索処理の*n i c e*値を上昇させる。OS205-2は、*n i c e*値を上昇させた辞書の先読み検索処理をよりまばらになるようにCPU201-2に割り当てる。結果、マルチコアプロセッサシステム100

0は、アクセス競合によるコンテンションを回避でき、UI入力処理のリアルタイム処理を保証できる。

[0078] 図8は、ハイパーバイザによるメッセージ送信処理を示すフローチャートである。メッセージ送信処理は、ハイパーバイザが起動するたびに行われる。CPU201-1は、リアルタイムソフトウェアが実行中かを確認する（ステップS801）。リアルタイムソフトウェアが実行中の場合（ステップS801: Yes）、CPU201-1は、発行命令カウンタIを取得する（ステップS802）。続けて、CPU201-1は、クロックカウンタCを取得する（ステップS803）。取得後、CPU201-1は、コンテンション中かの判断値となるC/I値を算出する（ステップS804）。算出後、CPU201-1は、C/I値と閾値 τ を比較する（ステップS805）。

[0079] C/I値が閾値 τ より大きい場合（ステップS805: Yes）、コンテンション中となり、CPU201-1は、nice値上昇メッセージを生成する（ステップS806）。このメッセージを受信したCPUは、現在動作中のソフトウェアのnice値を上昇させ、nice値が上昇したソフトウェアは優先度が下がるため、現在動作中のソフトウェアの実行をまばらにすることになる。

[0080] 生成後、CPU201-1は、ハイパーバイザ間にメッセージをブロードキャスト送信する（ステップS807）。送信後、CPU201-1は、通常のハイパーバイザ処理を実行し（ステップS810）、処理を終了する。C/I値が閾値 τ 以下の場合（ステップS805: No）、コンテンション中ではないことになり、CPU201-1は、ステップS810の処理を行い、処理を終了する。

[0081] リアルタイムソフトウェアが実行中でない場合（ステップS801: No）、CPU201-1は、続けて、実行中のソフトウェアのnice値が初期値かを確認する（ステップS808）。初期値でない場合（ステップS808: No）、CPU201-1は、実行中のソフトウェアのnice値を

初期値に設定し（ステップS809）、ステップS810の処理に移行する。n i c e値が初期値の場合（ステップS808：Y e s）、CPU201-1は、ステップS810の処理に移行する。

[0082] n i c e値が初期値でない場合、CPU201-1で実行していた処理がコンテンションの原因だったことを示しており、CPU201-1は、ステップS809の処理にて、コンテンション回避のために低下していた処理を元に戻すことができる。コンテンションを解決する場合、コンテンションの原因となっている処理を、OSのスケジューラが切り替え可能とする最小単位の時間で休止することで、コンテンションの解決を得られるケースが多い。もし、最小単位の時間で解決しないことが多い場合、CPU201-1は、ステップS808：N oの後にC/I値を算出し、閾値 τ と比較してコンテンションが解決したことを確認した後にステップS809の処理を実行してもよい。

[0083] 図9は、ハイパーバイザによるメッセージ受信処理を示すフローチャートである。CPU201-2は、ハイパーバイザ間のメッセージを受信する（ステップS901）。本実施の形態では、CPU201-1が送信したメッセージを受信する。次に、CPU201-2は、自身のCPUがメッセージをブロードキャストしたかを確認する（ステップS902）。

[0084] ブロードキャストしている場合（ステップS902：Y e s）、リアルタイム処理中でコンテンション中であり、スレッドの制御を行わないため、CPU201-2は、処理を終了する。ブロードキャストしていない場合（ステップS902：N o）、コンテンションの原因となるので、CPU201-2は、共有メモリ203にアクセスしない処理を実行する。

[0085] たとえば、CPU201-2は、現在動作中のソフトウェアのn i c e値を上昇させるようにOS205-2に指示する（ステップS903）。もし、n i c e値の機能を持っていないOSの場合、CPU201-2は、OS205-2に対してダミースレッドを起動させるよう指示してもよい。

[0086] また、CPU201-2は、ステップS902：N oにて、リアルタイム

ソフトが動作していない場合、 C/I 値を算出し、 C/I 値と閾値 τ を比較しコンテンション中であった場合に、スレッドの制御を行ってもよい。この場合、 C/I 値の比較の分だけ処理が増加するが、コンテンションが発生している CPU だけを対象にすることができる。

[0087] また、本実施の形態では、CPU 201-1 は、メッセージ送信処理にて優先度をリアルタイム実行か否かという 2 段階で分けたうえでコンテンションのチェックをしたが、優先度を 3 段階以上に分けてコンテンションのチェックをしてもよい。

[0088] その場合の処理を行う例として、CPU 201-1 は、優先度テーブル 303-1 の実行優先度フィールドの取りうる値を 3 段階以上にする。たとえば、“UI 入力” 処理の実行優先度が“リアルタイム”と“通常”の間である“高優先”であり、“辞書の先読み検索”処理の実行優先度が“通常”の下である“低優先”とする。さらに、メッセージ送信処理では、ステップ S801 の処理にて、「リアルタイムソフトウェアが実行中か？」を「低優先以外の優先度を持つソフトウェアが実行中か？」に置き換える。さらに、ステップ S806 の処理にて、メッセージの内容に、現在動作中のソフトウェアの優先度を付与する。

[0089] 続けて、メッセージ受信処理では、ステップ S902 : No の処理の後とステップ S903 の処理の間に新たな条件として、「受信したメッセージの優先度が現在動作中のソフトウェアの優先度より高いか？」を付け加える。条件が Yes の場合、実行主体となる CPU は、ステップ S903 の処理を行い、No の場合には、ステップ S903 の処理を行わず処理を終了する。

[0090] 前述した処理の状態にて、たとえば、図 7 にて前述した高優先である UI 入力処理を実行している CPU は、ステップ S801 にて Yes となり、ステップ S807 の処理にて他のコアにメッセージをブロードキャスト送信する。前述のメッセージを通常の優先度である描画レンダリングを実行している CPU が受信した場合、「受信したメッセージの優先度が現在動作中のソフトウェアの優先度より高いか？」が、Yes となり、ステップ S903 の

処理を行い、`n i c e`値の制御を行う。

[0091] もし、前述のメッセージをリアルタイム処理が要求される通信パケットを実行しているCPUが受信した場合、「受信したメッセージの優先度が現在動作中のソフトウェアの優先度より高いか？」が、`No`となるため、`n i c e`値の制御を行わない。このように、メッセージ送信処理を行うCPUは、優先度を3段階以上に分けてコンテンションのチェックを行い、メッセージ受信処理を行うCPUは、優先度の判断を加えることで、優先度の低い処理を行っているCPUの処理をまばらにする。これにより、マルチコアプロセッサシステム100は、優先度の高い処理を先に処理することができる。

[0092] また、前述の実行優先度が3段階以上の処理の場合でステップS903を実行する際に、受信したメッセージの優先度と現在動作中のソフトウェアの優先度に基づいて、`n i c e`値の上昇させる値を設定してもよい。たとえば、受信したメッセージの優先度がリアルタイムであり、現在動作中のソフトウェアの優先度が通常であった場合、優先度が2段階離れているため、`n i c e`値を2上昇させる、という処理を行ってもよい。このように、ハイパーバイザ204-2によって`n i c e`値を段階的に制御することで、OS205-2は、優先度の低い処理ほど、よりまばらに実行することになり、CPU201-1に割り当てられているリアルタイム処理を先に処理することができる。

[0093] また、実行優先度が2段階の状態においても、`n i c e`値の上昇させる値を2段階以上あげる処理を追加してもよい。具体的には、たとえば、CPU201-2が、メッセージ受信処理を受けて`n i c e`値を1上昇させた後に、`n i c e`値を初期値に戻す前に、メッセージ受信した場合である。この場合、`n i c e`値を上昇させたにもかかわらず、まだコンテンション状態であることを意味しているため、CPU201-2はさらに`n i c e`値を1上昇させるように設定することで、コンテンション状態がより解消しやすくなる。

[0094] 以上説明したように、マルチコアプロセッサシステム、スレッド制御方法

、およびスレッド制御プログラムによれば、リアルタイム処理でコンテンション中のCPUを特定する。そして、リアルタイム処理中のCPUと特定されたCPUを除いたすべてのCPUが、共有メモリにアクセスしないスレッドを実行するよう制御する。これにより、マルチコアプロセッサシステムは、リアルタイム処理を保証できる。

[0095] また、マルチコアプロセッサシステムは、特定されたCPUを除いたすべてのCPUに対し、共有メモリにアクセスしないスレッドを実行してもよい。これにより、特定されたCPUから、特定されたCPUを除いたすべてのCPUに対して制御依頼をする際に、競合している相手を探さず、自身のCPU以外のすべてのCPUに制御依頼を行うことで検索処理を行わないため、処理を簡略化できる。

[0096] また、マルチコアプロセッサシステムは、複数のCPUのうち、コンテンション中のCPUを特定し、特定されたCPUが、共有メモリにアクセスしないスレッドを実行するよう制御してもよい。これにより、マルチコアプロセッサシステムは、コンテンションを起こしたCPUのみスレッドを制御させ、コンテンションを起こしていないCPUに対しては、通常処理を続けさせることができる。

[0097] また、マルチコアプロセッサシステムは、コンテンションが発生した期間のうち、スレッドを制御するCPUに対して、制御するCPUに割り当てられていたスレッドの実行時間と、メモリにアクセスしないスレッドの時間を、時分割で分割して割り当ててもよい。これにより、マルチコアプロセッサシステムは、コンテンションを解消し、また、制御するCPUに割り当てられていたスレッドの処理も行うことができる。

[0098] また、マルチコアプロセッサシステムは、CPUごとに、CPUの命令を発行した発行命令数とCPUのサイクル数とに基づいて、発行命令効率を算出し、算出された発行命令効率と所定の閾値 τ に基づいて、コンテンションを検出してもよい。これにより、マルチコアプロセッサシステムは、アクセス競合によるコンテンションを検出することができ、リアルタイム処理を保

証できる。

[0099] また、マルチコアプロセッサシステムは、CPUに割り当てられたスレッドの実行優先度が最も高いコアを検出してもよい。これにより、マルチコアプロセッサシステムは、リアルタイム処理の保証を必要とするスレッドを決めておくことで、そのスレッドがどのCPUに割り当てられていても、コンテンションを解消しリアルタイム処理の保証を行うことができる。

[0100] なお、本実施の形態で説明したスレッド制御方法は、予め用意されたプログラムをコンピュータで実行することにより実現することができる。本スレッド制御プログラムは、ハードディスク、フレキシブルディスク、CD-ROM、MO、DVD等のコンピュータで読み取り可能な記録媒体に記録され、コンピュータによって記録媒体から読み出されることによって実行される。また本スレッド制御プログラムは、インターネット等のネットワークを介して配布してもよい。

符号の説明

- [0101] 201-1 CPU
 201-2 CPU
 301-1 領域
 301-2 領域
 302-1 領域
 302-2 領域
 303-1 優先度テーブル
 303-2 優先度テーブル
 304-1 OSスケジューラ監視部
 304-2 OSスケジューラ監視部
 305 優先度検出部
 306 発行命令効率算出部
 307 コンテンション検出部
 308 特定部

- 3 0 9 ハイパーバイザ間メッセージ送信部
- 3 1 0 ハイパーバイザ間メッセージ受信部
- 3 1 1 制御部
- 3 1 2 ソフトウェア
- 3 1 3 ソフトウェア
- 3 1 4 n i c e 値設定部
- 3 1 5 ダミースレッド起動部

請求の範囲

- [請求項1] 複数のコアと前記複数のコアからアクセス可能なメモリとを備えたマルチコアプロセッサシステムであって、
- 前記複数のコアのうち、実行優先度が最も高い第1のコアを検出する検出手段と、
- 前記検出手段によって検出された第1のコアのうち、前記メモリに対してアクセス競合を発生させた第2のコアを特定する特定手段と、
- 前記複数のコアのうち、前記第1のコアと前記特定手段によって特定された第2のコアとを除いた第3のコアに対し、前記メモリにアクセスしないスレッドを、前記アクセス競合が発生した期間のうち、所定の期間実行するように制御する制御手段と、
- を備えることを特徴とするマルチコアプロセッサシステム。
- [請求項2] 前記制御手段は、
- 前記複数のコアのうち、前記特定手段によって特定された第2のコアを除いた第3のコアに対し、前記メモリにアクセスしないスレッドを、前記アクセス競合が発生した期間のうち、所定の期間実行するように制御することを特徴とする請求項1に記載のマルチコアプロセッサシステム。
- [請求項3] 前記特定手段は、
- 前記複数のコアのうち、前記メモリに対してアクセス競合を発生させており前記第2のコアと競合する第3のコアを特定し、
- 前記制御手段は、
- 前記特定手段によって特定された第3のコアに対し、前記メモリにアクセスしないスレッドを、前記アクセス競合が発生した期間のうち、所定の期間実行するように制御することを特徴とする請求項1に記載のマルチコアプロセッサシステム。
- [請求項4] 前記所定の期間は、前記アクセス競合が発生した期間のうち、前記第3のコアに割り当てられていたスレッドと、前記メモリにアクセス

しないスレッドとで、時分割した期間であることを特徴とする請求項 1～3 のいずれか一つに記載のマルチコアプロセッサシステム。

[請求項5] 前記コアごとに、当該コアが命令を発行した発行命令数と当該コアのサイクル数とに基づいて、発行命令効率を算出する算出手段と、
前記算出手段によって算出された発行命令効率と所定の閾値とに基づいて、前記アクセス競合を検出する競合検出手段と、
をさらに備えることを特徴とする請求項 4 に記載のマルチコアプロセッサシステム。

[請求項6] 前記検出手段は、
前記複数のコアのうち、当該コアに割り当てられたスレッドの実行優先度が最も高い第 1 のコアを検出することを特徴とする請求項 5 に記載のマルチコアプロセッサシステム。

[請求項7] 複数のコアと前記複数のコアからアクセス可能なメモリと検出手段と、特定手段と、制御手段とを備えたマルチコアプロセッサシステムの前記コアが、
前記検出手段により、前記複数のコアのうち、実行優先度が最も高い第 1 のコアを検出する検出工程と、
前記特定工程により、前記検出工程によって検出された第 1 のコアのうち、前記メモリに対してアクセス競合を発生させた第 2 のコアを特定する特定工程と、
前記制御手段により、前記複数のコアのうち、前記第 1 のコアと前記特定工程によって特定された第 2 のコアとを除いた第 3 のコアに対し、前記メモリにアクセスしないスレッドを、前記アクセス競合が発生した期間のうち、所定の期間実行するように制御指示する制御指示工程と、
を実行することを特徴とするスレッド制御方法。

[請求項8] 複数のコアと前記複数のコアからアクセス可能なメモリとを備えたマルチコアプロセッサシステムの前記コアを、

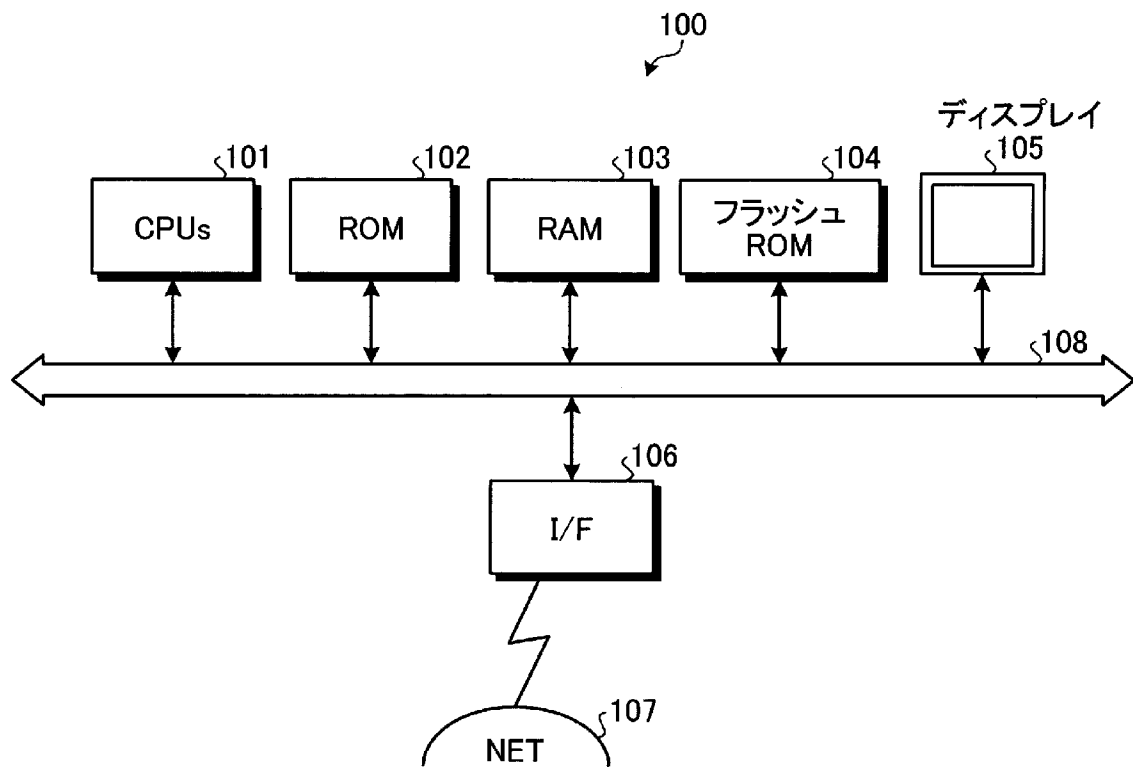
前記複数のコアのうち、実行優先度が最も高い第 1 のコアを検出する検出手段、

前記検出手段によって検出された第 1 のコアのうち、前記メモリに対してアクセス競合を発生させた第 2 のコアを特定する特定手段、

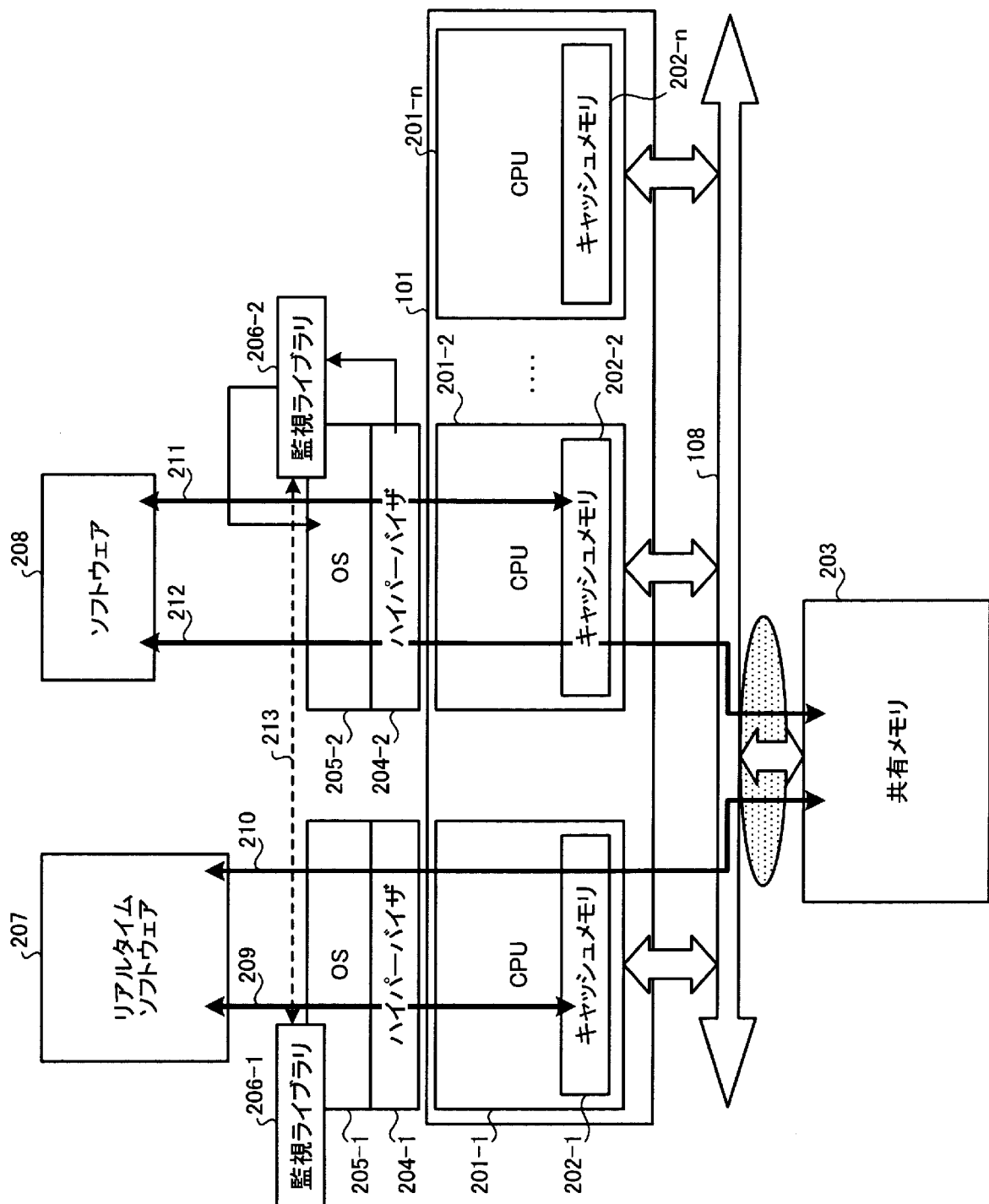
前記複数のコアのうち、前記第 1 のコアと前記特定手段によって特定された第 2 のコアとを除いた第 3 のコアに対し、前記メモリにアクセスしないスレッドを、前記アクセス競合が発生した期間のうち、所定の期間実行するように制御指示する制御指示手段、

として機能させることを特徴とするスレッド制御プログラム。

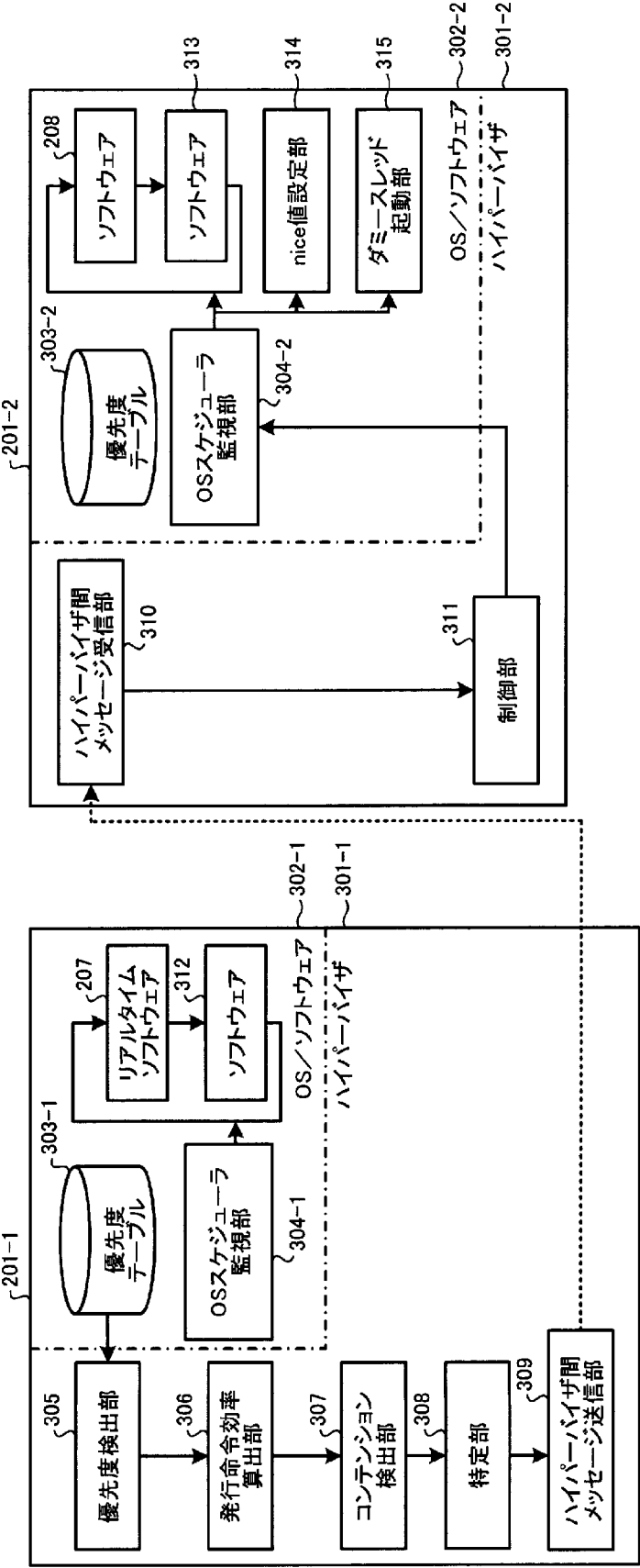
[図1]



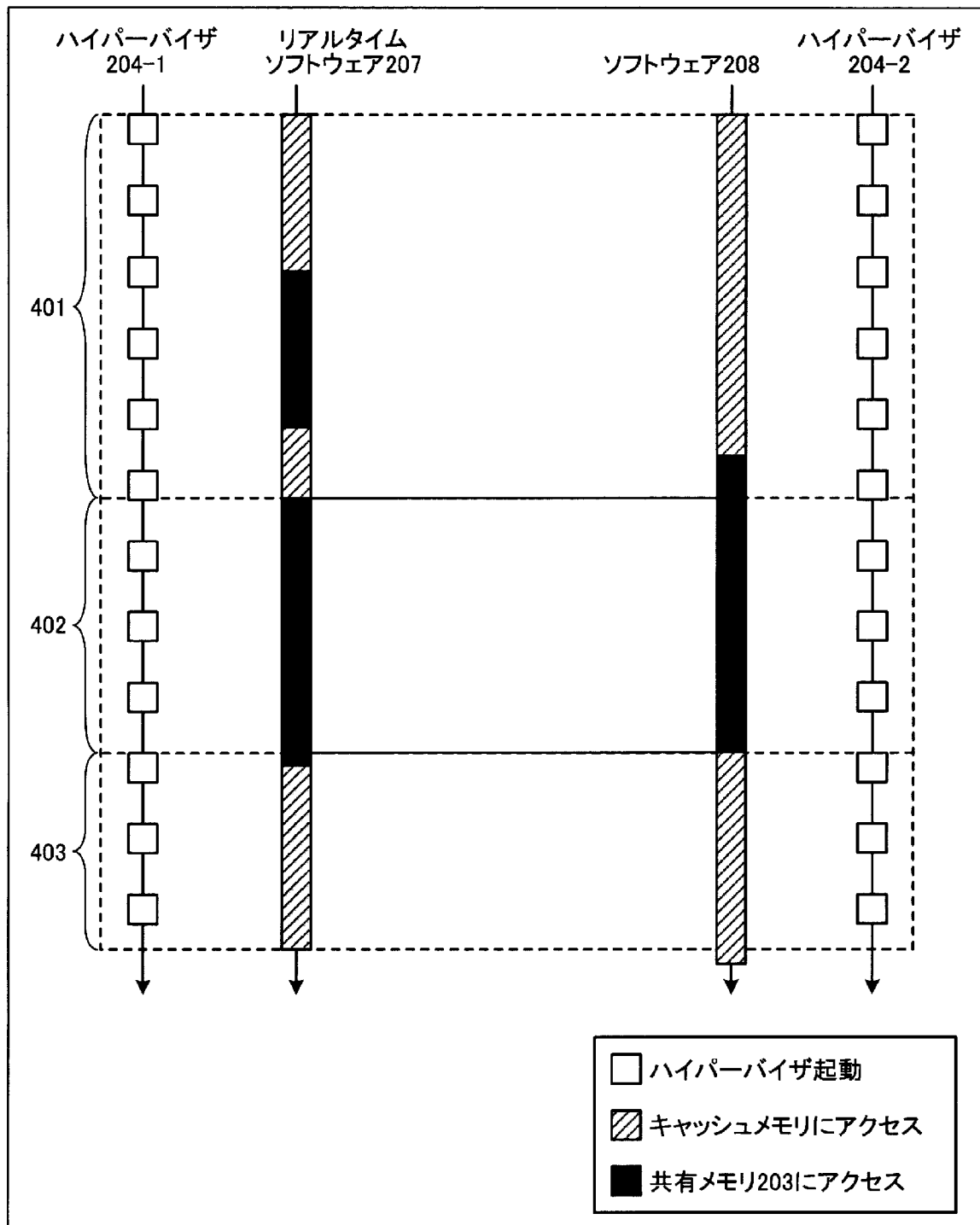
[図2]



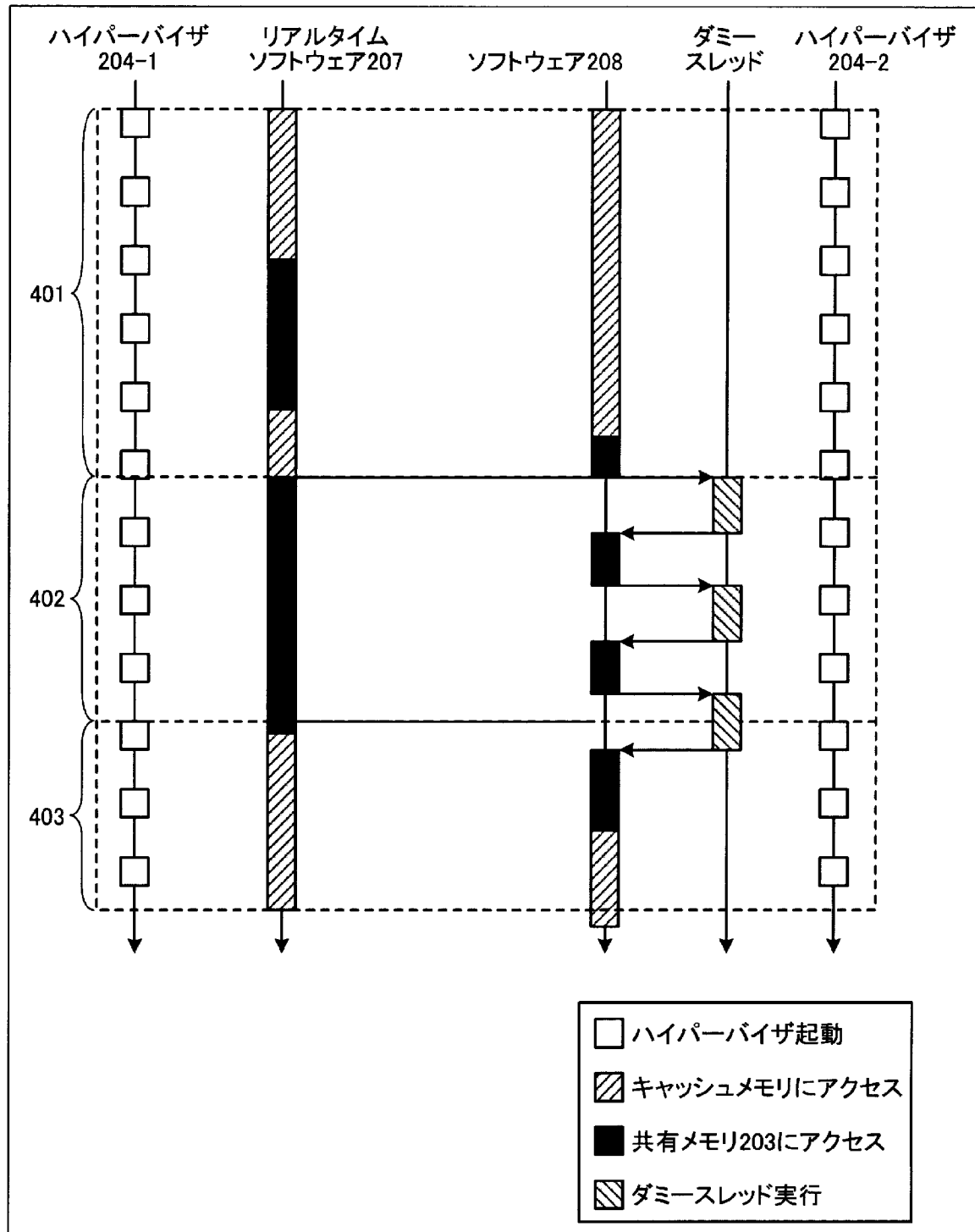
[図3]



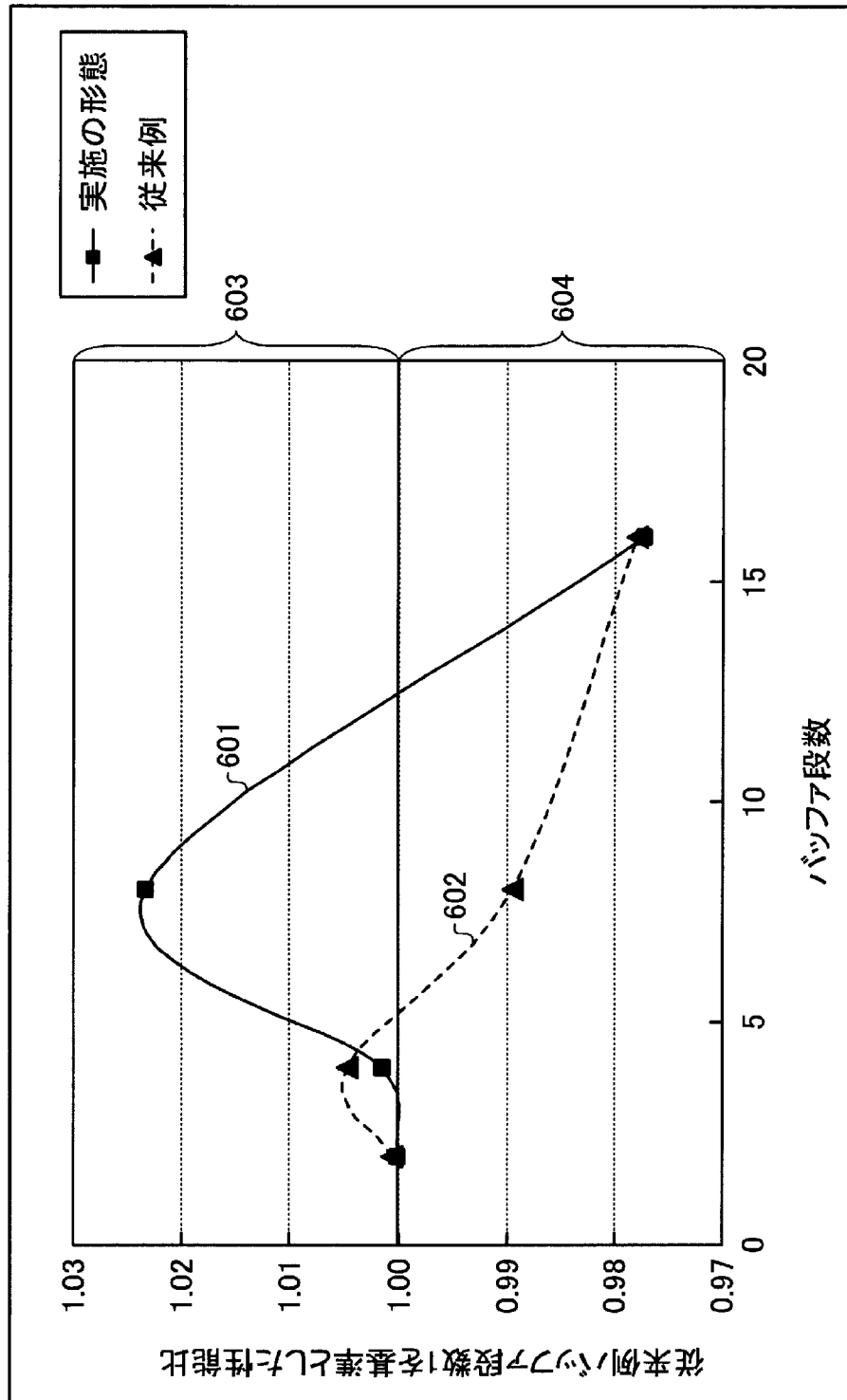
[図4]



[図5]



[図6]

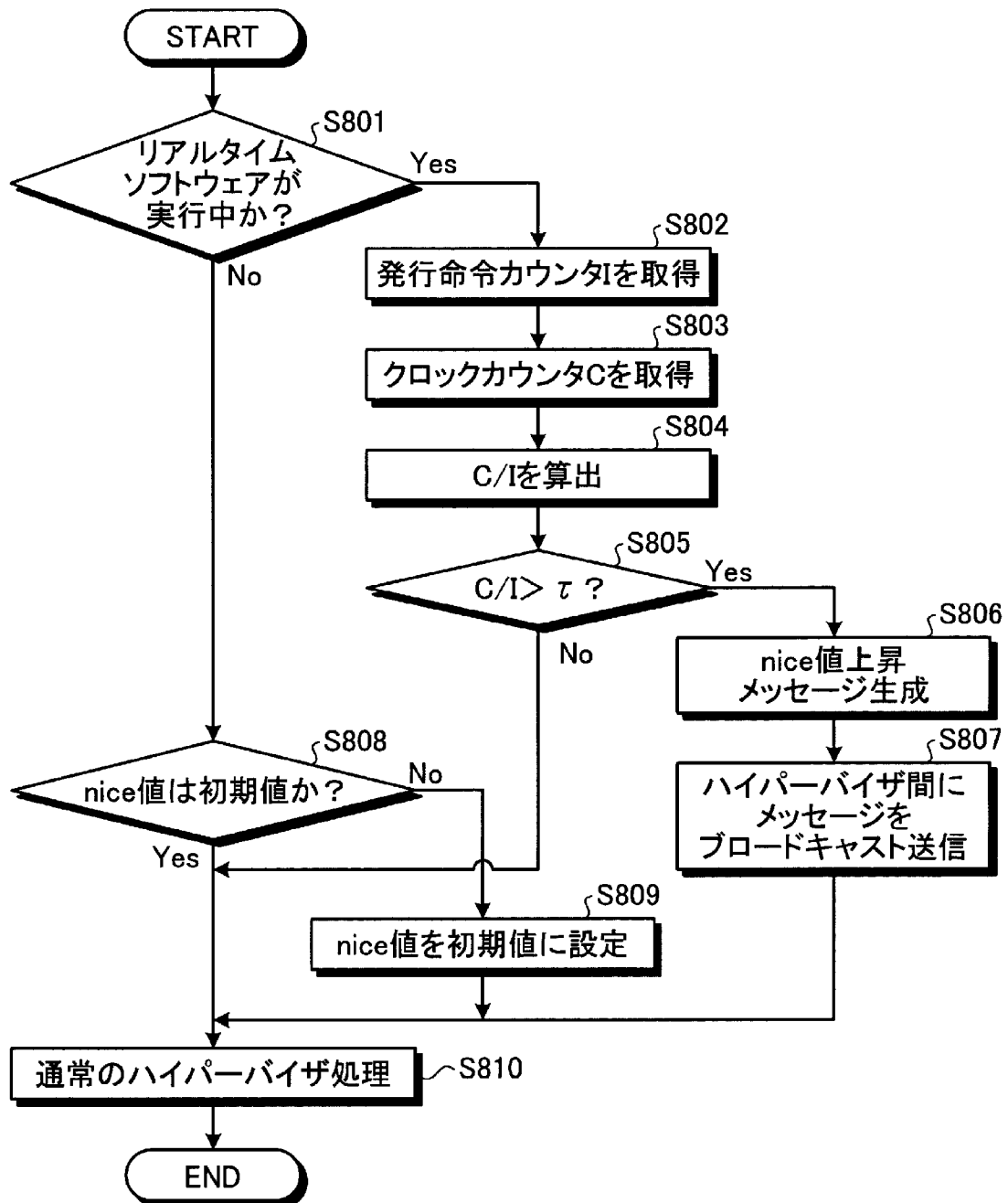


[図7]

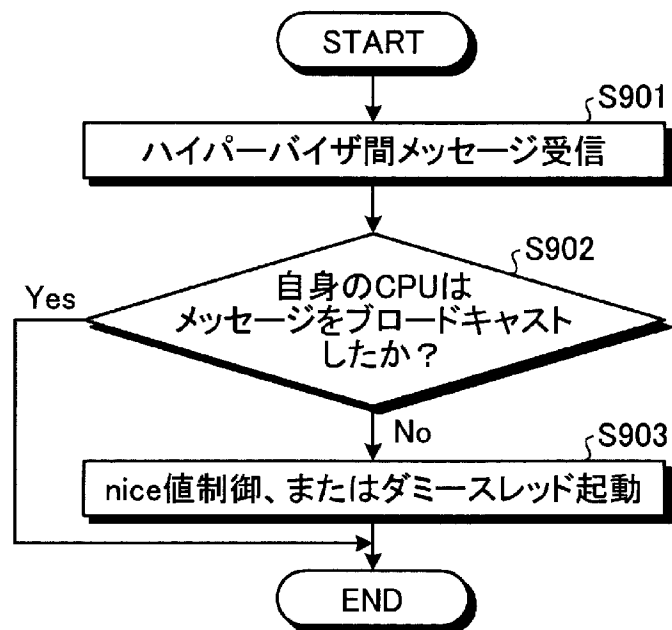
303-1

処理名称	実行優先度
通信パケット受信	リアルタイム
描画レンダリング	通常
UI入力	リアルタイム
辞書の先読み検索	通常

[図8]



[図9]



INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP2010/052792

A. CLASSIFICATION OF SUBJECT MATTER

G06F9/52(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)

G06F9/46-9/54, G06F15/16-15/177

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 1996-2010

Kokai Jitsuyo Shinan Koho 1971-2010 Toroku Jitsuyo Shinan Koho 1994-2010

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.
Y	JP 2003-30042 A (Fujitsu Ten Ltd.), 31 January 2003 (31.01.2003), paragraphs [0018] to [0022]; fig. 1, 2 (Family: none)	1-8
Y	JP 2001-84235 A (NEC Corp.), 30 March 2001 (30.03.2001), paragraphs [0002] to [0004], [0029] to [0032] (Family: none)	1-8
Y	JP 11-15793 A (International Business Machines Corp.), 22 January 1999 (22.01.1999), paragraph [0011] & US 5893157 A	1-8

☒ 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
11 March, 2010 (11.03.10)Date of mailing of the international search report
23 March, 2010 (23.03.10)Name and mailing address of the ISA/
Japanese Patent Office

Authorized officer

Facsimile No.

Telephone No.

INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP2010/052792

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	JP 5-225149 A (Toshiba Corp.), 03 September 1993 (03.09.1993), paragraphs [0002] to [0003] (Family: none)	1-8
Y	Hiroshi SASAKI et al, "CMP no Tokeiteki Modeling ni yoru Jikkoji Saitekika Shuho", Information Processing Society of Japan Kenkyu Hokoku, 29 July 2008 (29.07.2008), vol. 2008, no.75, pp.31-36	5, 6
Y	JP 2008-269578 A (NEC Corp.), 06 November 2008 (06.11.2008), paragraph [0076] & US 2008/0244583 A1	5, 6

A. 発明の属する分野の分類（国際特許分類（I P C））

Int.Cl. G06F9/52(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. G06F9/46-9/54, G06F15/16-15/177

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 0 年
日本国実用新案登録公報	1 9 9 6 - 2 0 1 0 年
日本国登録実用新案公報	1 9 9 4 - 2 0 1 0 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
Y	JP 2003-30042 A（富士通テン株式会社）2003.01.31, 段落[0018]－[0022], 第1,2図（ファミリーなし）	1-8
Y	JP 2001-84235 A（日本電気株式会社）2001.03.30, 段落[0002]－[0004], [0029]－[0032]（ファミリーなし）	1-8
Y	JP 11-15793 A（インターナショナル・ビジネス・マシーンズ・コー ポレイション）1999.01.22, 段落[0011] & US 5893157 A	1-8

☒ 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

1 1 . 0 3 . 2 0 1 0

国際調査報告の発送日

2 3 . 0 3 . 2 0 1 0

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）

大塚 良平

5 B

3 8 5 6

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
Y	JP 5-225149 A (株式会社東芝) 1993. 09. 03, 段落[0002]－[0003] (ファミリーなし)	1－8
Y	佐々木広、外 2 名, CMP の統計的モデリングによる実行時最適化 手法, 情報処理学会研究報告, 2008. 07. 29, 第 2008 巻, 第 75 号, pp. 31－36	5, 6
Y	JP 2008-269578 A (日本電気株式会社) 2008. 11. 06, 段落[0076] & US 2008/0244583 A1	5, 6