

(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



(10) 国際公開番号

WO 2011/114476 A1

(43) 国際公開日

2011 年 9 月 22 日(22.09.2011)

PCT

(51) 国際特許分類:

G06F 11/00 (2006.01)

(21) 国際出願番号:

PCT/JP2010/054606

(22) 国際出願日:

2010 年 3 月 17 日(17.03.2010)

(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.

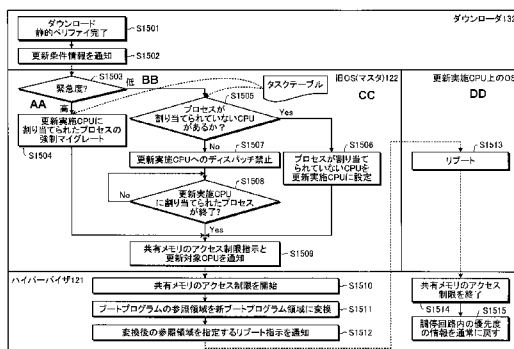
(84) 指定国 (表示のない限り、全ての種類の広域保護が可能): ARIPO (BW, GH, GM, KE, LS, MW, MZ,

[続葉有]

(54) Title: MULTICORE PROCESSOR SYSTEM, NOTIFICATION PROGRAM, AND NOTIFICATION METHOD

(54) 発明の名称: マルチコアプロセッサシステム、通知プログラム、および通知方法

[図16]



132. DOWNLOADING
S1501 DOWNLOADING AND STATIC VERIFICATION ARE COMPLETED
S1502 NOTIFICATION OF UPDATE CONDITION INFORMATION IS PROVIDED
122. OLD OS (MASTER)
S1503 DEGREE OF URGENCY?
AA. HIGH
S1504 PROCESS ALLOCATED TO UPDATE EXECUTION CPU IS FORCIBLY MIGRATED
BB. LOW
CC. TASK TABLE
S1505 CPU HAVING NO PROCESS ALLOCATED THERETO IS INCLUDED?
S1506 CPU HAVING NO PROCESS ALLOCATED THERETO IS SET TO BE UPDATE EXECUTION CPU
S1507 DISPATCH TO UPDATE EXECUTION CPU IS PROHIBITED
S1508 IS PROCESS ALLOCATED TO UPDATE EXECUTION CPU FINISHED?
S1509 NOTIFICATION OF INSTRUCTION TO RESTRICT ACCESS TO SHARED MEMORY AND OF CPU TO BE UPDATED IS PROVIDED
121. HYPERVISOR
S1510 RESTRICTION OF ACCESS TO SHARED MEMORY IS STARTED
S1511 REFERENCE AREA OF BOOT PROGRAM IS CONVERTED TO NEW BOOT PROGRAM AREA
S1512 NOTIFICATION OF REBOOT INSTRUCTION WHICH DESIGNATES CONVERTED REFERENCE AREA IS PROVIDED
DD. OS ON UPDATE EXECUTION CPU
S1513 REBOOT
S1514 RESTRICTION OF ACCESS TO SHARED MEMORY IS FINISHED
S1515 INFORMATION RELATING TO DEGREE OF PRIORITY WITHIN ARBITRATION CIRCUIT IS RETURNED TO NORMAL

(57) Abstract: When downloading and static verification of a new boot program are finished and the degree of urgency is low (step S1503: low), an old OS (master) (122) determines whether a core having no process allocated thereto is included in cores booted using an old boot program (step S1505). When the old OS (master) (122) determines that the core having no process allocated thereto is included (step S1505: Yes), a hypervisor (121) converts a reference area of the boot program from an old boot program area to a new boot program area (step S1511) and provides the core having no process allocated thereto with notification of a reboot instruction which designates the converted reference area (step S1512).

(57) 要約: 新ブートプログラムのダウンロードおよび静的ペリファイが終了し、緊急度が低い場合 (ステップ S1503: 低)、旧 OS (マスター) (122) が、旧ブートプログラムを用いてブートしたコアのうち、プロセスが割り当てられていないコアがあるか否かを判断する (ステップ S1505)。旧 OS (マスター) (122) が、プロセスが割り当てられていないコアがあると判断した場合 (ステップ S1505: Yes)、ハイパーバイザ (121) がブートプログラムの参照領域を旧ブートプログラム領域から新ブートプログラム領域に変換し (ステップ S1511)、変換後の参照領域を指定するリブート指示をプロセスが割り当てられていないコアへ通知する (ステップ S1512)。

WO 2011/114476 A1



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))

明 細 書

発明の名称：

マルチコアプロセッサシステム、通知プログラム、および通知方法

技術分野

[0001] 本発明は、オペレーティングシステムを更新するマルチコアプロセッサシステム、通知プログラム、および通知方法に関する。

背景技術

[0002] 従来の携帯型端末では、オペレーティングシステム（以下、「OS（Operating System）」と称する。）の更新に当たり、更新期間中にはアプリケーションの動作を停止させる必要があった。このため、更新期間中には、利用者はアプリケーションの動作（以下、「機能」と称する。）を利用することができないので、深夜時間帯など利用者の利用頻度が下がる時間帯に更新が行われていた。しかし、汎用計算機とは異なり、携帯型端末などの組み込みシステムは24時間365日動かし続けることを想定しており、機能を停止させなければならないことが問題となっていた。

[0003] サーバ分野においては、無停止でアップデートを行う2つの手法が提案されている。一つ目は、主システムの他に待機システムを用意し、主システムのアップデート中には待機システムで処理を行うもの（従来技術1）である（たとえば、下記特許文献1、2を参照。）。2つ目は、仮想マシンを運用し、アップデート期間中にOSを含むソフトウェアを仮想マシンで動作させること（従来技術2）である。

先行技術文献

特許文献

[0004] 特許文献1：特開昭54-106146号公報

特許文献2：特開2003-330744号公報

発明の概要

発明が解決しようとする課題

[0005] しかしながら、従来技術 1 および従来技術 2 の場合、組み込みシステムにおいては、システム規模や費用の増加を招くという問題があった。一方、待機システムや仮想マシンがない場合、アップデートを代行することができないため、アップデートの際に機能を停止しなければならないという問題があった。

[0006] このように、携帯型端末などの組み込みシステムは 24 時間 365 日動かし続けることを想定しており、従来では、更新期間中には、利用者は機能を利用することができず、また、機能を停止させなければならないことが問題となっていた。

[0007] 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、システム規模や費用の低減化を図りつつ、OS の無停止アップデートを実現することができるマルチコアプロセッサシステム、通知プログラム、および通知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

[0008] 本発明の一観点によれば、マルチコアプロセッサと、旧ブートプログラムおよび新ブートプログラムを記憶する記憶装置とを備えるマルチコアプロセッサシステムであって、前記マルチコアプロセッサの中で前記旧ブートプログラムを用いてブートしたコアのうち、プロセスが割り当てられていないコアを検出する検出手段と、前記検出手段により前記プロセスが割り当てられていないコアが検出されると、参照領域を前記旧ブートプログラムの記憶領域から前記新ブートプログラムの記憶領域に変換する変換手段と、前記変換手段による変換後の参照領域を指定するリブート指示を前記割り当てられていないコアへ通知する通知手段と、を備えるマルチコアプロセッサシステムを提供する。

発明の効果

[0009] 本マルチコアプロセッサシステム、通知プログラム、および通知方法によれば、システム規模や費用の低減化を図りつつ、OS の無停止アップデートを実現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

- [0010] [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の機能的構成を示すブロック図である。
- [図3]新ブートプログラムのダウンロード準備例とダウンロード例を示す説明図である。
- [図4]新ブートプログラム領域が設定される例を示す説明図である。
- [図5]CPU#Nがリブートされる例を示す説明図である。
- [図6]ハイパーバイザ121によるメモリアクセス制限例を示す説明図（その1）である。
- [図7]ハイパーバイザ121によるメモリアクセス制限例を示す説明図（その2）である。
- [図8]ハイパーバイザ121によるメモリアクセス制限例を示す説明図（その3）である。
- [図9]ハイパーバイザ121によるメモリアクセス制限例を示す説明図（その4）である。
- [図10]CPU#Nで新OSがブートされた例を示す説明図である。
- [図11]CPU#Mで新OSをブートさせる準備例を示す説明図である。
- [図12]CPU#Mに割り当てられているプロセスがすべて完了した例を示す説明図である。
- [図13]すべてのCPUで新OSがブートされた例を示す説明図である。
- [図14]マルチコアプロセッサシステム100によるOSの更新手順を示すフローチャート（その1）である。
- [図15]マルチコアプロセッサシステム100によるOSの更新手順を示すフローチャート（その2）である。
- [図16]ハイパーバイザ121によるメモリアクセス制限の制限処理手順を示すフローチャートである。

[図17] マルチコアプロセッサシステム100によるOSの更新手順を示すフローチャート（その3）である。

発明を実施するための形態

[0011] 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、通知プログラム、および通知方法の好適な実施の形態を詳細に説明する。

[0012] （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。図1において、マルチコアプロセッサシステム100は、CPU（Central Processing Unit）#Mと、CPU#Nと、調停回路102と、共有メモリ103と、ストレージ104と、I/F（Interface）105と、を備えている。また、各構成部は、バス101によってそれぞれ接続されている。

[0013] ここで、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CPU#MとCPU#Nの2つのプロセッサが並列に接続されているプロセッサ群を例に挙げて説明する。

[0014] CPU#MとCPU#Nは、独立して動作可能なコアとキャッシュとレジスタなどを備えている。ストレージ104に記憶されているプログラムは、CPU#MまたはCPU#Nによって共有メモリ103にマッピングされ、ロードされることで、コーディングされている処理をCPU#MまたはCPU#Nに実行させることとなる。

[0015] 調停回路102は、別々のCPUから同時に共有メモリ103へのアクセスが発生しても競合しないように調停する回路である。調停回路102内の優先度の情報は、CPUごとの優先度を示す情報であり、調停回路102は優先度の情報に基づいて共有メモリ103へのアクセスを調停する。本実施

の形態では、優先度は高いか否かの２段階とする。高優先は優先度が高いことを示し、低優先は優先度が低いことを示す。

[0016] 共有メモリ１０３は、CPU#Mローカル領域１１０やCPU#Nローカル領域１１１を有するメモリである。共有メモリ１０３は、たとえば、ROM (Read Only Memory)、RAM (Random Access Memory) などにより構成されている。たとえば、RAMは、CPU#MとCPU#Nのワークエリアとして使用される。

[0017] ストレージ１０４は、MBR１０８と、旧ブートプログラムの記憶領域（以下、「旧ブートプログラム領域１０９」）と、ユーザのファイルシステム領域を有している。ストレージ１０４は、たとえば、フラッシュROMやハードディスクドライブなどにより構成されている。

[0018] MBR (Master Boot Record) １０８とは、CPU#MやCPU#Nの起動時に最初に読み込まれるストレージ１０４上の先頭部分(セクタ)である。MBR１０８には、ストレージ１０４内に収められたどのOSをどのように起動するかなどの情報が記録され、具体的には、ブートストラップローダ、パーティションテーブル、ブートシグニチャがそれぞれ格納されている。

[0019] ブートストラップローダには、起動用のプログラムが格納されている。パーティションテーブルにはブートプログラムが格納されている開始アドレスおよび終了アドレスが示されている。ブートシグニチャには0xA A 5 5という値が識別子として格納され、該ブートシグニチャに0xA A 5 5が格納されていればMBR１０８が有効であることを示し、0xA A 5 5が格納されていなければMBR１０８が無効であることを示している。

[0020] ここで、ブートプログラムとは、OSの中核部分のプログラムである。たとえば、Linuxの場合、OSの中核部分はカーネルである。旧ブートプログラム領域１０９は、旧ブートプログラムが記憶されている領域であり、該領域の開始アドレスから終了アドレスはパーティションテーブルに記憶されている。本実施の形態で示すブートプログラムの新・旧は、先にストレージ

ジ 1 0 4 に記憶されているか否か、または版数が更新されているか否かを示している。

[0021] CPU#M（またはCPU#N）を起動するとまずMBR 1 0 8 が読み込まれ、ブートストラップローダに格納されている起動用のプログラムにコーディングされている処理がCPU#M（またはCPU#N）に実行される。ブートストラップローダはパーティションの位置や大きさなどを記録したパーティションテーブルを読み込み、OSの旧ブートプログラムの開始アドレスを読み込む。

[0022] ここで、CPU#Mは旧ブートプログラムを読み込み、読み込んだ旧ブートプログラムを共有メモリ 1 0 3 にマッピングする。CPU#Mがマッピングした旧ブートプログラムの共有メモリ 1 0 3 内の領域が、上述のCPU#Mローカル領域 1 1 0 である。そして、CPU#Mが、マッピングした旧ブートプログラムにコーディングされている処理を旧OS（マスタ） 1 2 2 として起動する。

[0023] 一方、CPU#Nは旧ブートプログラムを読み込み、読み込んだ旧ブートプログラムを共有メモリ 1 0 3 にマッピングする。CPU#Nがマッピングした旧ブートプログラムの共有メモリ 1 0 3 内の領域が、上述のCPU#Nローカル領域 1 1 1 である。そして、CPU#Nが、マッピングした旧ブートプログラムにコーディングされている処理をマスタOSの旧OS（スレーブ） 1 2 3 として起動する。

[0024] そして、I/F 1 0 5 は、ネットワーク 1 0 6 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 1 0 5 には、たとえばモデムなどを採用することができる。I/F 1 0 5 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 1 0 6 に接続され、このネットワーク 1 0 6 を介してOSの更新データ配信元 1 0 7 に接続される。

[0025] つぎに、ハイパーバイザ 1 2 1 は、CPU#MとCPU#Nなどのハード

ウェア上で直接動作するプログラムであり、該ハードウェア内のレジスタに記憶されているプログラムである。ハイパーバイザ１２１はＣＰＵごとに独立して動作するが、すべて同一処理を実施するため、図１では省略してハイパーバイザ１２１のみ記載している。

[0026] なお、たとえば、後述する旧ブートプログラムの削除処理であれば、複数のハイパーバイザ１２１が一つしか存在しない旧ブートプログラムを削除しているようであるが、最も早く削除処理まで処理がたどり着いたＣＰＵのハイパーバイザ１２１が削除を行う。よって、あとから他のＣＰＵのハイパーバイザ１２１が削除処理を行っても破棄する旧ブートプログラムはすでに存在しないだけである。

[0027] ハイパーバイザ１２１は、ＣＰＵ＃ＭやＣＰＵ＃Ｎ内のレジスタを直接参照したり、ＣＰＵ＃ＭやＣＰＵ＃Ｎ内のレジスタの情報を読み出したり、ＣＰＵ＃ＭやＣＰＵ＃Ｎ内のレジスタの情報を書き換えたりする特権命令を実施することができる。

[0028] さらに、ＣＰＵ＃ＭやＣＰＵ＃Ｎへのリブート指示もハイパーバイザ１２１の特権命令である。ハイパーバイザ１２１によるリブート指示とは、具体的には、パーティションテーブル内に記述されている旧ブートプログラム領域１０９の開始アドレスにジャンプするようにＣＰＵ＃ＭやＣＰＵ＃Ｎへ指示することである。本実施の形態では、新ブートプログラムの記憶領域（以下、「新ブートプログラム領域」と称する。）の開始アドレスへジャンプするようにＣＰＵ＃ＭやＣＰＵ＃Ｎへ指示することで、ＣＰＵ＃ＭやＣＰＵ＃Ｎは新ブートプログラムでＯＳを起動する。

[0029] 旧ＯＳ（マスタ）１２２および旧ＯＳ（スレーブ）１２３は、ハイパーバイザ１２１上で動作するソフトウェアである。ＯＳにおけるマスタ・スレーブとＣＰＵ間の接続関係におけるマスタ・スレーブとは意味が異なる。マスタＯＳ（本実施の形態では旧ＯＳ（マスタ）１２２）ではプロセスの割り当てを決定するスケジューラを動かし、スレーブＯＳ（本実施の形態では旧ＯＳ（スレーブ）１２３）はスケジューラにより割り当てられたプロセスを実

行する。

[0030] マスタOSは、プロセスの割り当て結果をタスクテーブルに登録する。ここで、タスクテーブルは、一般的なOSで管理されているプロセス情報のテーブルである。起動時間、メモリリソース、プロセス番号、起動したユーザ名、実行条件（リンクしているOSがサービスするライブラリの版数など）、実行形式ファイルのパスなどが記載されている。すなわち、CPU#Mは、旧OS（マスタ）122によりタスクテーブルから起動時間や実行条件を読み出すことで、各アプリケーション（プロセス）がどのOSに割り当てられているか、どのアプリケーション（プロセス）が実行終了したかを監視することができる。

[0031] また、更新検出アプリ131は、OS上で動作するアプリケーションプログラムであり、ブートプログラムの更新を検出する。更新検出アプリ131は、旧OS（マスタ）122上に割り当てられている。また、GUI（Graphic User Interface）のプロセスであるメインプロセス133が旧OS（マスタ）122に割り当てられている。

[0032] ダウンローダ132は、OS上で動作するアプリケーションプログラムであり、ブートプログラムやライブラリをダウンロードする。ダウンローダ132は、旧OS（マスタ）122と旧OS（スレーブ）123のうちいずれのOS上で動作しても良いが、ここでは、旧OS（スレーブ）123に割り当てられている。

[0033] ダウンローダ132は、I/F105を介してOSの更新データ配信元107から新ブートプログラムをダウンロードし、静的ベリファイを実施する。静的ベリファイとは、ブートプログラムのバイナリデータに誤りがないか、規格に従っているかなどを検査することである。動的ベリファイとは、新ブートプログラムで正常にブートされるかを検査することである。

[0034] （マルチコアプロセッサシステム100の機能的構成）

図2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で起動されているハイパ

ーバイザ１２１は、設定部２０１と、受付部２０２と、判断部２０３と、変換部２０４と、通知部２０５と、検出部２０６と、削除部２０７と、を含む構成である。上述のようにハイパーバイザ１２１は通知プログラムを備えるプログラムであるため、ストレージ１０４または共有メモリ１０３内のＲＯＭに記憶されたハイパーバイザ１２１をＣＰＵ＃ＭおよびＣＰＵ＃Ｎに実行させることにより、各機能（設定部２０１から削除部２０７）を実現する。

[0035] まず、設定部２０１は、新ブートプログラムの更新指示を受け付けると、旧ブートプログラム領域１０９と所定のマスク値に基づいて新ブートプログラム領域を設定する。

[0036] つぎに、受付部２０２は、設定部２０１により新ブートプログラム領域が設定された後、新ブートプログラムのダウンロードに関する旧ブートプログラム領域１０９へのアクセスの入力を受け付ける。

[0037] 判断部２０３は、受付部２０２により受け付けられたアクセスがライトアクセスであるかリードアクセスであるかを判断する。

[0038] 変換部２０４は、判断部２０３によりアクセスがライトアクセスであると判断された場合、ライトアクセスで指定されたアドレスを、前記所定のマスク値に基づいて新プログラム領域を指定するアドレスに変換する。

[0039] 通知部２０５は、アクセスがリードアクセスの場合、リードアクセスで指定されたアドレスへのアクセス指示をリードアクセスのアクセス元へ通知する。一方、通知部２０５は、アクセスがライトアクセスの場合、変換部２０４による変換後のアドレスへのアクセス指示をライトアクセスのアクセス元へ通知する。つぎに、新ブートプログラムがダウンロードされた後の処理について説明する。

[0040] 検出部２０６は、マルチコアプロセッサの中で旧ブートプログラムを用いてブートしたＣＰＵのうち、プロセスが割り当てられていないＣＰＵを検出する。本実施の形態では、後述するがＯＳはプロセスが割り当てられていないコアを検出し、ハイパーバイザ１２１へ検出したコアにプロセスが割り当てられていないことを通知する。そして、ハイパーバイザ１２１はこの通知

を受け付けることをプロセスが割り当てられていないコアを検出するとしている。

[0041] 変換部204は、検出部206によりプロセスが割り当てられていないCPUが検出されると、参照領域を旧ブートプログラム領域109から新ブートプログラムの記憶領域に変換する。

[0042] 通知部205は、変換部204による変換後の参照領域を指定するリブート指示を完了したCPUへ通知する通知する。

[0043] また、検出部206は、プロセスが割り当てられていないCPUが検出されない場合、割り当てられていたすべてのプロセスの動作が完了したCPUを検出する。

[0044] そして、変換部204は、検出部206により該完了したCPUが検出されると、参照領域を旧ブートプログラム領域109から新ブートプログラム領域に変換する。通知部205は、変換部204による変換後の参照領域を指定するリブート指示を完了したCPUへ通知する通知する。

[0045] 変換部204は、通知部205によりリブート指示が通知されると、記憶装置のマスターブートレコードに記憶されている旧ブートプログラム領域109を指定するアドレスを所定のマスク値に基づき新ブートプログラム領域を指定するアドレスに変換する。

[0046] 削除部207は、変換部204により記憶装置のマスターブートレコードに記憶されている旧ブートプログラム領域109が変更されると、旧ブートプログラム領域109に格納されている旧ブートプログラムを削除する。

[0047] 以上を踏まえて図を用いてマルチコアプロセッサシステム100の詳細な動作について説明する。

[0048] 図3は、新ブートプログラムのダウンロード準備例とダウンロード例を示す説明図である。まず、更新検出アプリ131がI/Fを介してブートプログラムへの更新を検出し、(1)旧OS(マスタ)122へブートプログラムの更新を通知する。つぎに、旧OS(マスタ)122がブートプログラムの更新を受け付けると、(2)ハイパーバイザ121へブートプログラムの

更新を通知する。そして、ハイパーバイザ 121 が新ブートプログラムの更新を受け付けると、(3) 新ブートプログラム領域 112 を所定のマスク値と旧ブートプログラムの領域に基づいて設定する。

[0049] 図 4 は、新ブートプログラム領域 112 が設定される例を示す説明図である。ここで、(3) 新ブートプログラムの領域の設定について詳細例を説明する。ハイパーバイザ 121 は MBR 108 のパーティションテーブル 401 を参照し、旧ブートプログラム領域 109 の開始アドレスおよび終了アドレスを取得する。ハイパーバイザ 121 は旧ブートプログラム領域 109 の開始アドレスとマスク値との論理和を取ることにより新ブートプログラム領域 112 の開始アドレスを算出する。

[0050] ハイパーバイザ 121 は旧ブートプログラム領域 109 の終了アドレスとマスク値との論理和を取ることにより新ブートプログラム領域 112 の終了アドレスを算出する。ハイパーバイザ 121 は、ストレージ 104 内の算出した新ブートプログラムの開始アドレスが示す領域から算出した新ブートプログラムの終了アドレスが示す領域までを新ブートプログラムの領域として確保する。

[0051] 図 3 に戻って、(4) ハイパーバイザ 121 が旧 OS (マスタ) 122 へダウンロード開始指示を通知し、新ブートプログラムのダウンローダ 132 および静的ベリファイに関する旧ブートプログラムへのアクセスを監視する。旧 OS (マスタ) 122 がダウンロード開始通知を受け付けると、(5) ダウンローダ 132 へダウンロード開始指示を通知する。そして、ダウンローダ 132 はダウンロード開始指示を受け付けると、I/F 105 を介して OS の更新データ配信元 107 から新ブートプログラムをダウンロードする。

[0052] そして、ダウンローダ 132 が新ブートプログラムをダウンロード中、旧ブートプログラムへのライトアクセスを行うために (6) 旧 OS (スレーブ) 123 へ旧ブートプログラムへのアクセスを入力する。旧 OS (スレーブ) 123 が、新ブートプログラムのダウンロードに関する旧ブートプログラ

ムへのアクセスを受け付けると、(7) ハイパーバイザ 1 2 1 へ該アクセスを入力する。

[0053] そして、ハイパーバイザ 1 2 1 が、新ブートプログラムのダウンロードに関する旧ブートプログラムへのアクセスを受け付けると、該アクセスがライトアクセスであるかリードアクセスであるかを判断する。ハイパーバイザ 1 2 1 がライトアクセスであると判断した場合、該ライトアクセスが示すアドレスを所定のマスク値に基づいて新ブートプログラム領域 1 1 2 を指定するアドレスに変換する。

[0054] そして、ハイパーバイザ 1 2 1 が、(8) 変換後のアドレスへのアクセス指示を旧 OS (スレーブ) 1 2 3 へ通知する。一方、ハイパーバイザ 1 2 1 がリードアクセスであると判断した場合、リードアクセスで指定されたアドレスへのアクセス指示を旧 OS (スレーブ) 1 2 3 へ通知する。

[0055] アクセスが指定するアドレスを 0×1110 とし、所定のマスク値を 0×1000 とする。たとえば、リードアクセスの場合、ハイパーバイザ 1 2 1 が 0×1110 へのアクセス指示を旧 OS (スレーブ) 1 2 3 へ通知する。たとえば、ライトアクセスの場合、ハイパーバイザ 1 2 1 が 0×1110 と 0×1000 の論理和を算出する(下記式(1)) ことで新ブートプログラム領域 1 1 2 を指定するアドレスに変換する。そして、変換後のアドレスへのアクセス指示を旧 OS (スレーブ) 1 2 3 へ通知する。

[0056] ブートプログラム領域を指定するアドレス $= 0 \times 1110 + 0 \times 1000$
 $= 0 \times 0110$

... (1)

[0057] ここで、「+」が論理和を示している。よって、ハイパーバイザ 1 2 1 が 0×0110 へのアクセス指示を旧 OS (スレーブ) 1 2 3 へ通知する。

[0058] 旧 OS (スレーブ) 1 2 3 は、ハイパーバイザ 1 2 1 からアクセス指示を受け付けると、(9) ダウンローダ 1 3 2 へアクセス指示を通知する。図 3 では、ライトアクセスを示しているため、ダウンローダ 1 3 2 は、(10) 変換後のアドレスへのアクセスを実行する。

- [0059] 図5は、CPU#Nがリブートされる例を示す説明図である。新ブートプログラム領域112への新ブートプログラムのダウンロード後、更新検出アプリ131が(11)更新条件情報を旧OS(マスタ)122へ通知する。更新条件情報は、実行制約と、対象版数と、更新実施CPU数と、緊急度の情報とを有している。緊急度の情報は緊急度が高いか否かに関する情報である。
- [0060] 旧OS(マスタ)122は、更新条件情報を受け付けると、受け付けた更新条件情報から緊急度の情報を読み出し、読み出した緊急度の情報が、緊急度が高いことを示す場合、タスクテーブルを用いて更新実施CPUに割り当てられているプロセスを特定し、(12)強制マイグレートする。ここでは、プロセス134が特定される。強制マイグレートにより更新実施CPUに割り当てられているプロセス134がCPU#Mに割り当てられている。
- [0061] ここで、更新実施CPUをCPU#Nとする。更新実施CPUとは、他のCPUに先だって新ブートプログラムでリブートするCPUである。更新実施CPUの決定方法については、少なくとも旧OSへのプロセスの割り当てが可能なマスタOSが旧OSの中で最後にリブートされればよいと、特に限定しない。
- [0062] たとえば、4つのCPUによりマルチコアプロセッサが構成されている場合、マスタOSの処理を実行するCPUを除く残余のCPUから2つのCPUを更新実施CPUとする。そして、マスタOSの処理を実行するCPUともう一つのCPUを更新実施CPUがリブート後にリブートするCPUとしてもよい。また、マスタOSの処理を実行するCPUを除く残余のCPUをすべて更新実施CPUとしてもよい。
- [0063] また、読み出した緊急度の情報が、緊急度が高くない(緊急度が低い)ことを示す場合、旧OS(マスタ)122が、マルチコアプロセッサのうちマスタOSを動作するCPUを除くCPUからいずれのプロセスも割り当てられていないCPUを検出する。CPU#Nには、プロセスが割り当てられているため、いずれのプロセスも割り当てられていないCPUは検出されない

。つぎに、旧OS（マスタ）122が、CPU#Nへのディスパッチを禁止し、CPU#Nに割り当てられているプロセスが終了するのを検出する。

[0064] そして、強制マイグレートまたはプロセスの終了によってCPU#Nに割り当てられていたプロセスが無くなると、(13)旧OS（マスタ）122がCPU#Nに割り当てられていたプロセスが完了したこと（または、更新実施CPUでもよい。）およびメモリアクセス制限指示をハイパーバイザ121へ通知する。ハイパーバイザ121が該通知を受け付けると、(14)共有メモリ103のアクセス制限を開始する。ハイパーバイザ121が共有メモリ103のアクセス制限を開始すると、たとえば、各OSやアプリケーションはハイパーバイザ121からのアクセス指示を受け付けるまで共有メモリ103へアクセスできないこととする。

[0065] そして、ハイパーバイザ121が、共有メモリ103のアクセス制限を開始した後、ブートプログラムの参照領域を旧ブートプログラム領域109から新ブートプログラム領域112へ変換し、(15)変換後の参照領域を指定するリブート指示を旧OS（スレーブ）123へ通知する。旧OS（スレーブ）123は、リブート指示を受け付けると、(16)新ブートプログラムを参照してリブートを開始する。また、更新対象CPUのうち一つのCPUが新OSをマスタOSとしてリブートする。ここでは、CPU#Nのみであるため、CPU#Nが新OSをマスタOSとしてリブートする。つぎに、ハイパーバイザ121によるメモリアクセス制限について図6から図9を用いて説明する。

[0066] 図6は、ハイパーバイザ121によるメモリアクセス制限例を示す説明図（その1）である。図では、(17)リブート中の旧OSから共有メモリ103内のCPU#Mローカル領域110へのアクセスが発生した場合について説明する。ハイパーバイザ121は、メモリアクセス制限を開始すると、アプリケーションから共有メモリ103へのアクセスおよびアクセスのアドレスをすべて検出する。

[0067] ハイパーバイザ121が(18)リブート中の旧OSから共有メモリ10

3内のCPU#Mローカル領域110へのアクセスとアクセスのアドレスを検出する。ハイパーバイザ121がCPU#Nからのアクセスか否かを判断し、ここでは、旧OS（スレーブ）123からのアクセスであるためCPU#Nからのアクセスであると判断される。つづいて、ハイパーバイザ121が他のCPUのローカル領域へのアクセスか否かを検出したアクセスのアドレスに基づいて判断する。

[0068] ここでは、検出されたアクセスがCPU#Mローカル領域110へのアクセスであると判断され、(19)ハイパーバイザ121が旧OS（スレーブ）123の更新失敗を出力、またはダウンロードした新ブートプログラムが不正であることを出力する。出力形式としては、たとえば、ディスプレイへの表示、I/F105による外部装置への送信がある。また、共有メモリ103内の各CPUのローカル領域を除く領域などに記憶することとしてもよい。

[0069] ブート中であるCPU#NからCPU#MのワークエリアであるCPU#Mローカル領域110へのアクセスは正常なブートプログラムであれば発生しない。したがって、(17)～(19)により異常であるブートプログラムで更新するのを防止することができる。

[0070] 図7は、ハイパーバイザ121によるメモリアccess制限例を示す説明図（その2）である。図7では、(20)リブート中の旧OSから共有メモリ103内のCPU#Nローカル領域111へのアクセスが発生した場合について説明する。

[0071] ハイパーバイザ121が(21)リブート中の旧OSから共有メモリ103内のCPU#Nローカル領域111へのアクセスを検出する。ハイパーバイザ121がCPU#Nからのアクセスか否かを判断し、ここでは、旧OS（スレーブ）123からのアクセスであるためCPU#Nからのアクセスであると判断される。つづいて、ハイパーバイザ121が他のCPUのローカル領域へのアクセスか否かを検出したアクセスが指定するアドレスに基づいて判断する。

- [0072] ここでは、検出されたアクセスがCPU#Nローカル領域111へのアクセスであると判断され、ハイパーバイザ121が(22)調停回路102内のCPU#Nに関する優先度の情報を低優先に設定する。そして、ハイパーバイザ121がアクセス指示を通知する。
- [0073] ブート中であるCPU#Nは、各種テーブルの初期化やシステムファイルの更新、システムファイルの一時退避・復元、該一時退避・復元にともなうキャッシュデータのフラッシュなどを行う。すなわち、ブート中であるCPU#Nは、通常のアプリケーションを動作中であるCPU#Mと比較して共有メモリ103へのアクセスが多い。
- [0074] 上述のように調停回路102内のCPU#Nに関する優先度の情報が低優先に設定されることで、メモリコンテンション（メモリアクセス権の争奪）を防止することができる。そして、ブート中でないCPU#Mで動作しているアプリケーションに影響を与えることなくOSを更新できる。
- [0075] 図8は、ハイパーバイザ121によるメモリアクセス制限例を示す説明図（その3）である。図8では、(23)旧OS（マスタ）122から共有メモリ103内のCPU#Mローカル領域110へのアクセスが発生した場合について説明する。
- [0076] ハイパーバイザ121が(24)旧OS（マスタ）122から共有メモリ103内のCPU#Mローカル領域110へのアクセスを検出する。ハイパーバイザ121が検出したアクセスがCPU#Nからのアクセスか否かを判断し、ここでは、旧OS（マスタ）122からのアクセスであるためCPU#Nからのアクセスでないと判断される。つづいて、ハイパーバイザ121が他のCPUのローカル領域へのアクセスか否かを検出したアクセスが示すアドレスに基づいて判断する。
- [0077] ここでは、検出されたアクセスがCPU#Mローカル領域110へのアクセスであると判断され、ハイパーバイザ121が(25)調停回路102内のCPU#Nに関する優先度の情報を低優先に設定する。そして、ハイパーバイザ121が旧OS（マスタ）122へアクセス指示を通知する。

- [0078] 図9は、ハイパーバイザ121によるメモリアクセス制限例を示す説明図（その4）である。図9では、（26）旧OS（マスタ）122から共有メモリ103内のCPU#Nローカル領域111へのアクセスが発生した場合について説明する。
- [0079] ハイパーバイザ121が（27）旧OS（マスタ）122から共有メモリ103内のCPU#Nローカル領域111へのアクセスを検出する。ハイパーバイザ121が検出したアクセスがCPU#Nからのアクセスか否かを判断し、ここでは、旧OS（マスタ）122からのアクセスであるためCPU#Nからのアクセスでないと判断される。つづいて、ハイパーバイザ121が共有メモリ103内の他のCPUのローカル領域へのアクセスか否かを検出したアクセスが示すアドレスに基づいて判断する。
- [0080] ここでは、検出されたアクセスがCPU#Nローカル領域111へのアクセスであると判断され、ハイパーバイザ121が（28）アクセスエラーを出力する。出力形式としては、たとえば、ディスプレイへの表示、I/F105による外部装置への送信がある。また、共有メモリ103内の各CPUのローカル領域を除く領域などに記憶することとしてもよい。
- [0081] 図10は、CPU#Nで新OSがブートされた例を示す説明図である。図10では、CPU#Nで新OS（マスタ）124がマスタOSとしてブートされている。ここで、新OSと旧OSごとにマスタOSが存在することとなる。新たに割り当てられる新規起動プロセス136は新OS（マスタ）124によりスケジュールされて実行されるため、マルチコアプロセッサシステム100において新旧のマスタOSが共存していても競合することはない。また、プロセス135は旧OS（スレーブ）123がブート中に旧OS（マスタ）122に割り当てられたプロセスである。
- [0082] そして、ハイパーバイザ121が、CPU#Nで新OS（マスタ）124がブートされると、（29）メモリアクセス制限を解除し、（30）調停回路102内の優先度の情報を元に戻す。調停回路102内の優先度の情報は、たとえば、メモリ制限開始直前に共有メモリ103内に複製して記憶して

おき、メモリ制限解除後に複製した優先度の情報を用いて調停回路 102 内の変更された優先度の情報を復元する。

[0083] 図 11 は、CPU#M で新 OS をブートさせる準備例を示す説明図である。旧 OS (マスタ) 122 が、旧 OS (マスタ) 122 に割り当てられているプロセスをプロセステーブルに基づいて特定する。特定したプロセスのうち旧ブートプログラムにダイナミックリンクもしくはオープンしているプロセスは、旧 OS (マスタ) 122 で処理を続行する。一方、(31) 旧 OS (マスタ) 122 は、特定したプロセスのうち旧ブートプログラムにダイナミックリンクおよびオープンしていないプロセス (ここでは、メインプロセス 133 と更新検出アプリ 131) を新 OS (マスタ) 124 へマイグレートする。

[0084] 図 12 は、CPU#M に割り当てられているプロセスがすべて完了した例を示す説明図である。図 12 では、旧 OS (マスタ) 122 に割り当てられていたプロセスの処理が終了した。すなわち、旧 OS (マスタ) 122 により旧 OS (マスタ) 122 に割り当てられているプロセスが特定されない場合である。そして、(32) 旧 OS (マスタ) 122 がハイパーバイザ 121 へ旧 OS (マスタ) 122 に割り当てられているプロセスがすべて完了したことを通知する。

[0085] (33) ハイパーバイザ 121 が該通知を受け付けると、MBR 108 に記憶されているパーティションテーブル 401 が示すブートプログラムの参照領域を新ブートプログラム領域 112 に変換する。そして、(34) ハイパーバイザ 121 が旧 OS (マスタ) 122 へリブート指示を通知する。すでにパーティションテーブル 401 が示すブートプログラムの参照領域は変換済みであるため、(35) 旧 OS (マスタ) 122 はリブート指示を受け付けると、新ブートプログラムでリブートを開始する。ここで、旧 OS (マスタ) 122 は新 OS (スレーブ) 125 としてブートされる。そして、(36) ハイパーバイザ 121 が旧ブートプログラムを廃棄する。

[0086] 図 13 は、すべての CPU で新 OS がブートされた例を示す説明図である

。図 1 3 では、旧 OS（マスタ） 1 2 2 が新 OS（スレーブ） 1 2 5 として起動されている。さらに、ストレージ 1 0 4 内での旧ブートプログラムの領域が空いている。

[0087] （マルチコアプロセッサシステムによる OS の更新手順）

図 1 4 は、マルチコアプロセッサシステム 1 0 0 による OS の更新手順を示すフローチャート（その 1）である。ここでは、ハイパーバイザ 1 2 1 と旧 OS（マスタ） 1 2 2 とダウンロード 1 3 2 と更新検出アプリ 1 3 1 によるダウンロード開始前とダウンロードおよび静的ベリファイ時の処理手順を説明する。まず、更新検出アプリ 1 3 1 がブートプログラムの更新を検出し（ステップ S 1 4 0 1）、更新検出アプリ 1 3 1 がブートプログラムの更新を検出した場合、旧 OS（マスタ） 1 2 2 へブートプログラムの更新を通知する（ステップ S 1 4 0 2）。旧 OS（マスタ） 1 2 2 が、ブートプログラムの更新を受け付けると、ハイパーバイザ 1 2 1 へブートプログラムの更新を通知する（ステップ S 1 4 0 3）。

[0088] ハイパーバイザ 1 2 1 が、ブートプログラムの更新を受け付けると、新ブートプログラム領域を旧ブートプログラムと所定のマスク値に基づいて設定し、旧 OS（マスタ） 1 2 2 へダウンロード開始指示を通知する（ステップ S 1 4 0 4）。そして、ハイパーバイザ 1 2 1 が、ブートプログラム領域へのブートプログラムに関するアクセスを監視する（ステップ S 1 4 0 5）。アクセスの監視は、ステップ S 1 4 0 9～ステップ S 1 4 1 1 までの処理を指す。

[0089] 旧 OS（マスタ） 1 2 2 が、ハイパーバイザ 1 2 1 からのダウンロード開始指示を受け付けると、ダウンロード 1 3 2 へダウンロード開始指示を通知する（ステップ S 1 4 0 6）。そして、ダウンロード 1 3 2 が、ダウンロード開始指示を受け付けると、ダウンロードと静的ベリファイを開始する（ステップ S 1 4 0 7）。ダウンロード 1 3 2 が、ダウンロード中や静的ベリファイ中にストレージへのアクセスを行う場合、旧 OS（マスタ） 1 2 2 へアクセスを入力し、ダウンロード 1 3 2 を動かす OS が、ストレージへのアク

セスを受け付けると、ハイパーバイザ 121 へアクセスを入力する（ステップ S 1408）。

[0090] そして、ハイパーバイザ 121 が、アクセスの入力を受け付けると、受け付けたアクセスがライトアクセスであるかリードアクセスであるかを判断する（ステップ S 1409）。ハイパーバイザ 121 が、該アクセスがライトアクセスであると判断した場合（ステップ S 1409：ライトアクセス）、ライトアクセスで指定されたアドレスを所定のマスク値に基づいて変換する（ステップ S 1410）。そして、ハイパーバイザ 121 が、変換後のアドレスへのアクセス指示を、ダウンローダ 132 を動かす OS へ通知する（ステップ S 1411）。

[0091] 一方、ハイパーバイザ 121 が、該アクセスがリードアクセスであると判断した場合（ステップ S 1409：リードアクセス）、リードアクセスで指定されたアドレスへのアクセス指示を通知する（ステップ S 1411）。ステップ S 1411 のつぎに、ダウンローダ 132 を動かす OS がアクセス指示を受け付けると、ダウンローダ 132 へアクセス指示を通知し（ステップ S 1412）、ダウンローダ 132 が、アクセス指示を受け付けると、ストレージへアクセスする（ステップ S 1413）。ステップ S 1407 とステップ S 1413 間の矢印は、アクセス結果に沿ってダウンロードと静的ベリファイが続行されることを示している。

[0092] 図 15 は、マルチコアプロセッサシステム 100 による OS の更新手順を示すフローチャート（その 2）である。ここでは、ハイパーバイザ 121 と旧 OS（マスタ）122 とダウンローダ 132 によるダウンロードおよび静的ベリファイ後から更新実施 CPU のリブートが完了するまでの処理手順を説明する。更新実施 CPU には、旧 OS（マスタ）122 が含まれていないこととする。まず、ダウンローダ 132 が、ダウンロードと静的ベリファイを完了させると（ステップ S 1501）、更新条件情報を通知する（ステップ S 1502）。

[0093] 旧 OS（マスタ）122 が、更新条件情報を受け付けると、更新条件情報

から緊急度を読み出し、緊急度を判断する（ステップS 1 5 0 3）。緊急度が高い場合（ステップS 1 5 0 3：高）、旧OS（マスタ）1 2 2が、更新実施CPUに割り当てられたプロセスを強制マイグレートし（ステップS 1 5 0 4）、ステップS 1 5 0 7へ移行する。

[0094] 一方、緊急度が低い場合（ステップS 1 5 0 3：低）、旧OS（マスタ）1 2 2が、プロセスが割り当てられていないCPUがあるか否かを判断する（ステップS 1 5 0 5）。そして、旧OS（マスタ）1 2 2が、プロセスが割り当てられていないCPUがあると判断した場合（ステップS 1 5 0 5：Y e s）、プロセスが割り当てられていないCPUを更新実施CPUに設定し（ステップS 1 5 0 6）、ステップS 1 5 0 9に移行する。ここでは、更新実施CPUは、プロセスが割り当てられていないCPUのみである。

[0095] そして、旧OS（マスタ）1 2 2が、プロセスが割り当てられていないCPUがないと判断した場合（ステップS 1 5 0 5：N o）、更新実施CPUへのディスパッチを禁止する（ステップS 1 5 0 7）。そして、旧OS（マスタ）1 2 2が、更新実施CPUに割り当てられたプロセスが終了したか否かを判断する（ステップS 1 5 0 8）。旧OS（マスタ）1 2 2が、更新実施CPUに割り当てられたプロセスが終了していないと判断した場合、（ステップS 1 5 0 8：N o）、ステップS 1 5 0 8へ戻る。

[0096] 一方、旧OS（マスタ）1 2 2が、更新実施CPUに割り当てられたプロセスが終了したと判断した場合（ステップS 1 5 0 8：Y e s）、共有メモリへのアクセス制限指示と更新対象CPUをハイパーバイザ1 2 1へ通知する（ステップS 1 5 0 9）。

[0097] ハイパーバイザ1 2 1が、共有メモリへのアクセス制限指示と更新対象CPUとを受け付けると、アクセス制限を開始する（ステップS 1 5 1 0）。アクセス制限のフローチャートについては後述する。そして、ハイパーバイザ1 2 1が、ブートプログラムの参照領域を新ブートプログラム領域に変換する（ステップS 1 5 1 1）。

[0098] つぎに、ハイパーバイザ1 2 1が、変換後の参照領域を指定するリポート

指示を更新実施CPU上のOSへ通知する（ステップS 1512）。更新実施CPU上のOSとは、更新実施CPUに処理を実行させているOSである。ここでは、更新実施CPUのうちいずれか一つのCPUへ新ブートプログラムで起動する際にマスタOSとなるように指示する。更新実施CPU上のOSは、リブート指示を受け付けると、リブートし、リブートが完了するとリブートの完了をハイパーバイザ121へ通知する（ステップS 1513）。そして、更新実施CPUの全CPUからリブートの完了を受け付けると、共有メモリのアクセス制限を終了し（ステップS 1514）、調停回路内の優先度の情報を通常に戻す（ステップS 1515）。調停回路内の優先度の情報は、アクセス制限により変更されるため、優先度の情報を通常に戻すとは、アクセス制限開始前の優先度の情報に戻すことを意味する。

[0099] 図16は、ハイパーバイザ121によるメモリアccess制限の制限処理手順を示すフローチャートである。まず、ハイパーバイザ121が、メモリアccess制限を開始すると、アプリケーションから共有メモリへのアクセスを検出する（ステップS 1601）。つぎに、ハイパーバイザ121が、更新対象CPUからのアクセスか否かを判断する（ステップS 1602）。ここで、更新対象CPUは、リブート中である。そして、ハイパーバイザ121が、更新対象CPUからのアクセスであると判断した場合（ステップS 1602：Yes）、他のCPUのローカル領域へのアクセスか否かを判断する（ステップS 1603）。

[0100] ハイパーバイザ121が、他のCPUのローカル領域へのアクセスであると判断した場合（ステップS 1603：Yes）、更新失敗を出力し（ステップS 1604）、復旧する（ステップS 1605）。一方、ステップS 1602において、ハイパーバイザ121が、更新対象CPUからのアクセスでないと判断した場合（ステップS 1602：No）、他のCPUのローカル領域へのアクセスか否かを判断する（ステップS 1606）。ハイパーバイザ121が、他のCPUのローカル領域へのアクセスであると判断した場合（ステップS 1606：Yes）、アクセスエラーを出力する（ステップS

1607)。

[0101] 一方、ハイパーバイザ121が、他のCPUのローカル領域へのアクセスでないと判断した場合（ステップS1603：No、またはステップS1606：No）、調停回路内の更新対象CPUに関する優先度の情報を低優先に設定する（ステップS1608）。そして、アクセス指示をアクセス元へ通知する（ステップS1609）。なお、ハイパーバイザ121が、メモリアクセス制限終了指示を受け付けるまで、処理を繰り返す。

[0102] 図17は、マルチコアプロセッサシステム100によるOSの更新手順を示すフローチャート（その3）である。まず、新OSが起動されると、新OS（マスタ）124は、新既プロセスを新OSに割り当てるように制御する（ステップS1701）。つぎに、旧OS（マスタ）122が、旧OSで既動作のプロセスをタスクテーブルに基づいて特定する（ステップS1702）。

[0103] そして、旧OS（マスタ）122が、旧OSで既動作のプロセスがあるか否かを判断する（ステップS1703）。まず、旧OS（マスタ）122が、旧OSで既動作のプロセスがあると判断した場合（ステップS1703：Yes）、未選択の既動作のプロセスがあるか否かを判断する（ステップS1704）。旧OS（マスタ）122が、未選択の既動作のプロセスがないと判断した場合（ステップS1704：No）、ステップS1702へ戻る。一方、旧OS（マスタ）122が、未選択の既動作のプロセスがあると判断した場合（ステップS1704：Yes）、未選択の既動作のプロセスから任意のプロセスを選択する（ステップS1705）。

[0104] そして、旧OS（マスタ）122が、選択したプロセスがシステムファイルにダイナミックリンクまたはオープンしているかを判断する（ステップS1706）。旧OS（マスタ）122が、システムファイルにダイナミックリンクまたはオープンしていると判断した場合（ステップS1706：Yes）、ステップS1704に戻る。一方、旧OS（マスタ）122が、システムファイルにダイナミックリンクおよびオープンしていないと判断した場

合（ステップS 1 7 0 6 : N o）、選択したプロセスを新OSへマイグレートし（ステップS 1 7 0 7）、ステップS 1 7 0 4へ戻る。

[0105] 一方、旧OS（マスタ）1 2 2が、旧OSで既動作のプロセスがないと判断した場合（ステップS 1 7 0 3 : N o）、ハイパーバイザ1 2 1へ既動作のプロセスがないことを通知する（ステップS 1 7 0 8）。つづいて、ハイパーバイザ1 2 1が、パーティションテーブル内の参照領域を新ブートプログラム領域に変換し（ステップS 1 7 0 9）、変換後の参照領域を指定するリブート指示を旧OSへ通知する（ステップS 1 7 1 0）。

[0106] そして、旧OS（マスタ）1 2 2を含む旧OSが、リブート指示を受け付けると、新ブートプログラムでリブートする（ステップS 1 7 1 2）。ステップS 1 7 1 0のつぎに、ハイパーバイザ1 2 1が、旧ブートプログラムを破棄する（ステップS 1 7 1 1）。

[0107] 以上説明したように、マルチコアプロセッサシステム、通知プログラム、および通知方法によれば、旧ブートプログラムでブートしたコアのうちプロセスが割り当てられていないコアから順に新ブートプログラムでリブートさせるリブート指示を通知する。したがって、リブート指示を通知されて受け付けたコアが新ブートプログラムでリブートする。これにより、空いているコアから直ぐにリブートするため、プロセスを停止させず、OSを無停止で更新することができる。

[0108] また、プロセスが割り当てられていないコアが無い場合、割り当てられているプロセスが完了したコアへ新ブートプログラムでリブートさせるリブート指示を通知することにより、動作中のプロセスを停止させることなくOSを更新できる。

[0109] また、新ブートプログラムの更新を受け付けてから、新ブートプログラムをダウンロードする新ブートプログラム領域を設定することで、ストレージ内の容量を効率的に使用できる。

[0110] また、リブート指示後にマスターブートレコードのパーティションテーブル内の旧ブートプログラム領域のアドレスを新ブートプログラム領域のアド

レスに変換する。これにより、新ブートプログラムでリブートしたCPUの電源が再投入されても、該CPUが新ブートプログラムでブートすることができる。

符号の説明

- [0111] 1 0 0 マルチコアプロセッサシステム
- 2 0 1 設定部
- 2 0 2 受付部
- 2 0 3 判断部
- 2 0 4 変換部
- 2 0 5 通知部
- 2 0 6 検出部

請求の範囲

- [請求項1] マルチコアプロセッサの中で旧ブートプログラムを用いてブートしたコアのうち、プロセスが割り当てられていないコアを検出する検出手段と、
- 前記検出手段により前記プロセスが割り当てられていないコアが検出されると、参照領域を前記旧ブートプログラムの記憶領域から新ブートプログラムの記憶領域に変換する変換手段と、
- 前記変換手段による変換後の参照領域を指定するリブート指示を前記割り当てられていないコアへ通知する通知手段と、
- 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] 前記検出手段は、
- 前記割り当てられていないコアが検出されない場合、割り当てられていたすべてのプロセスの動作が完了したコアを検出し、
- 前記変換手段は、
- 前記検出手段により完了したコアが検出されると、参照領域を前記旧ブートプログラムの記憶領域から前記新ブートプログラムの記憶領域に変換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] 前記新ブートプログラムの更新指示を受け付けると、前記旧ブートプログラムの記憶領域と所定のマスク値に基づいて前記新ブートプログラムの記憶領域を設定する設定手段と、
- 前記設定手段により前記新ブートプログラムの記憶領域が設定された後、前記新ブートプログラムのダウンロードに関する前記旧ブートプログラムの記憶領域へのアクセスの入力を受け付ける受付手段と、
- 前記受付手段により受け付けられたアクセスがライトアクセスであるかリードアクセスであるかを判断する判断手段と、を備え、
- 前記変換手段は、
- 前記判断手段により前記アクセスが前記ライトアクセスであると判

断された場合、前記ライトアクセスで指定されたアドレスを、前記所定のマスク値に基づいて前記新ブートプログラムの記憶領域を指定するアドレスに変換し、

前記通知手段は、

前記アクセスが前記リードアクセスの場合、前記リードアクセスで指定されたアドレスへのアクセス指示を、前記リードアクセスのアクセス元へ通知し、前記アクセスが前記ライトアクセスの場合、前記変換手段による変換後のアドレスへのアクセス指示を前記ライトアクセスのアクセス元へ通知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]

前記変換手段は、

前記通知手段によりリブート指示が通知されると、前記記憶装置のマスターブートレコードに記憶されている前記旧ブートプログラムの記憶領域を指定するアドレスを前記所定のマスク値に基づき前記新ブートプログラムの記憶領域を指定するアドレスに変換す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]

旧ブートプログラムおよび新ブートプログラムを記憶する記憶装置にアクセス可能なマルチコアプロセッサのうちの一のコアに、

前記マルチコアプロセッサの中で前記旧ブートプログラムを用いてブートしたコアのうち、プロセスが割り当てられていないコアを検出する検出工程と、

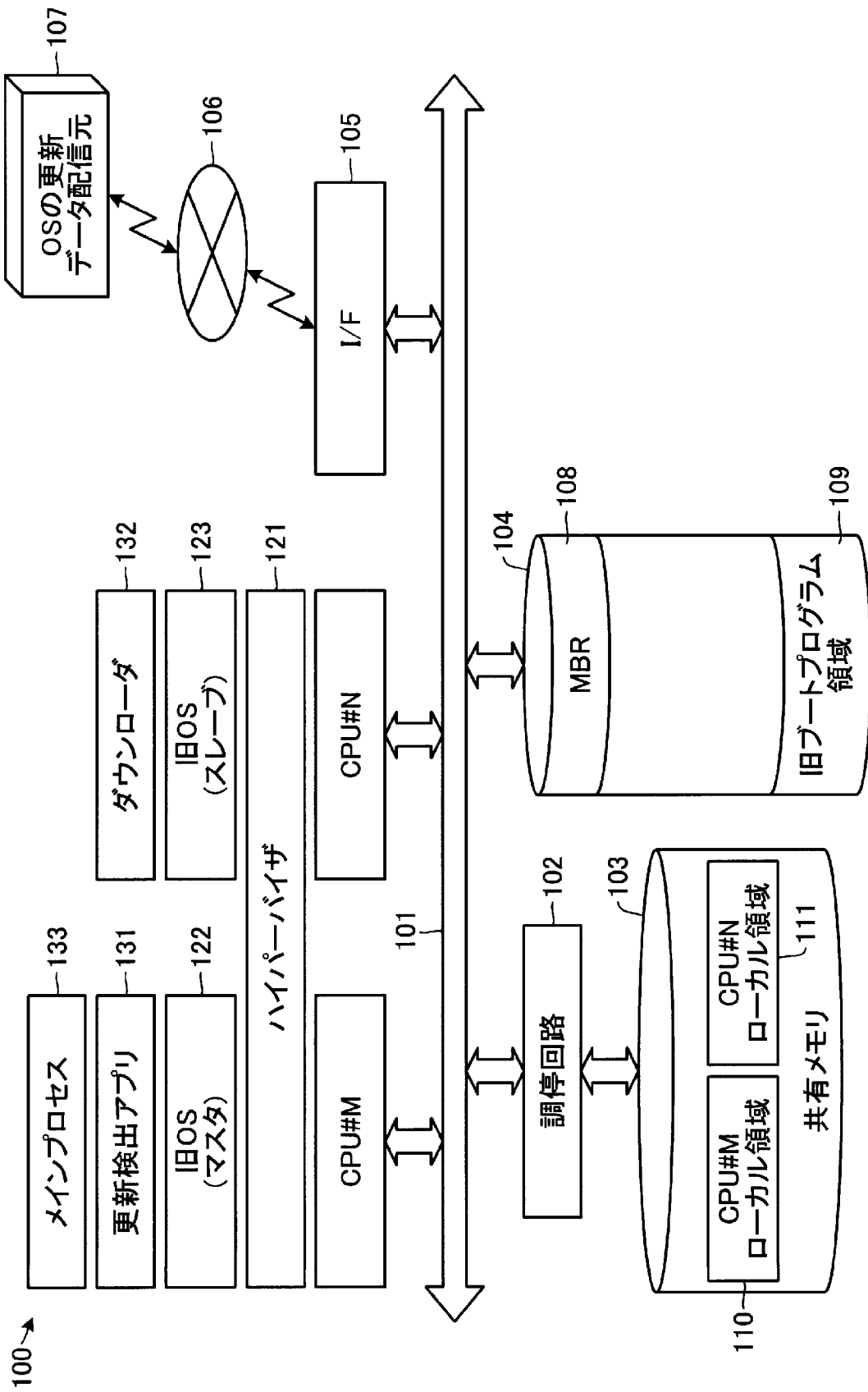
前記検出工程により前記プロセスが割り当てられていないコアが検出されると、参照領域を前記旧ブートプログラムの記憶領域から前記新ブートプログラムの記憶領域に変換する変換工程と、

前記変換工程による変換後の参照領域を指定するリブート指示を前記完了したコアへ通知する通知工程と、

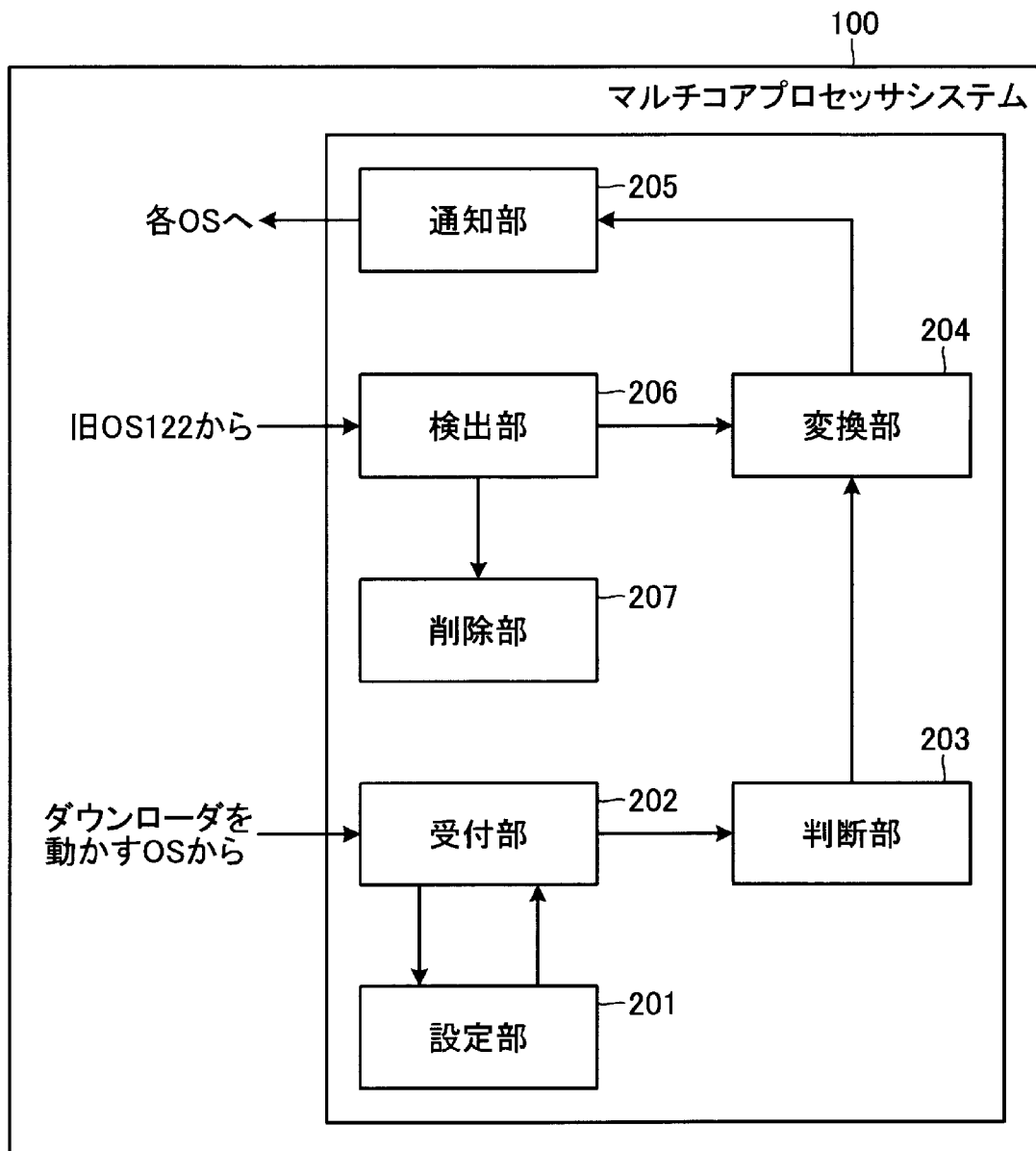
を実行させることを特徴とするプログラム。

- [請求項6] 旧ブートプログラムおよび新ブートプログラムを記憶する記憶装置にアクセス可能なマルチコアプロセッサのうちの一のコアが、
- 前記マルチコアプロセッサの中で前記旧ブートプログラムを用いてブートしたコアのうち、プロセスが割り当てられていないコアを検出する検出工程と、
- 前記検出工程により前記プロセスが割り当てられていないコアが検出されると、参照領域を前記旧ブートプログラムの記憶領域から前記新ブートプログラムの記憶領域に変換する変換工程と、
- 前記変換工程による変換後の参照領域を指定するリブート指示を前記完了したコアへ通知する通知工程と、
- を実行することを特徴とする通知方法。

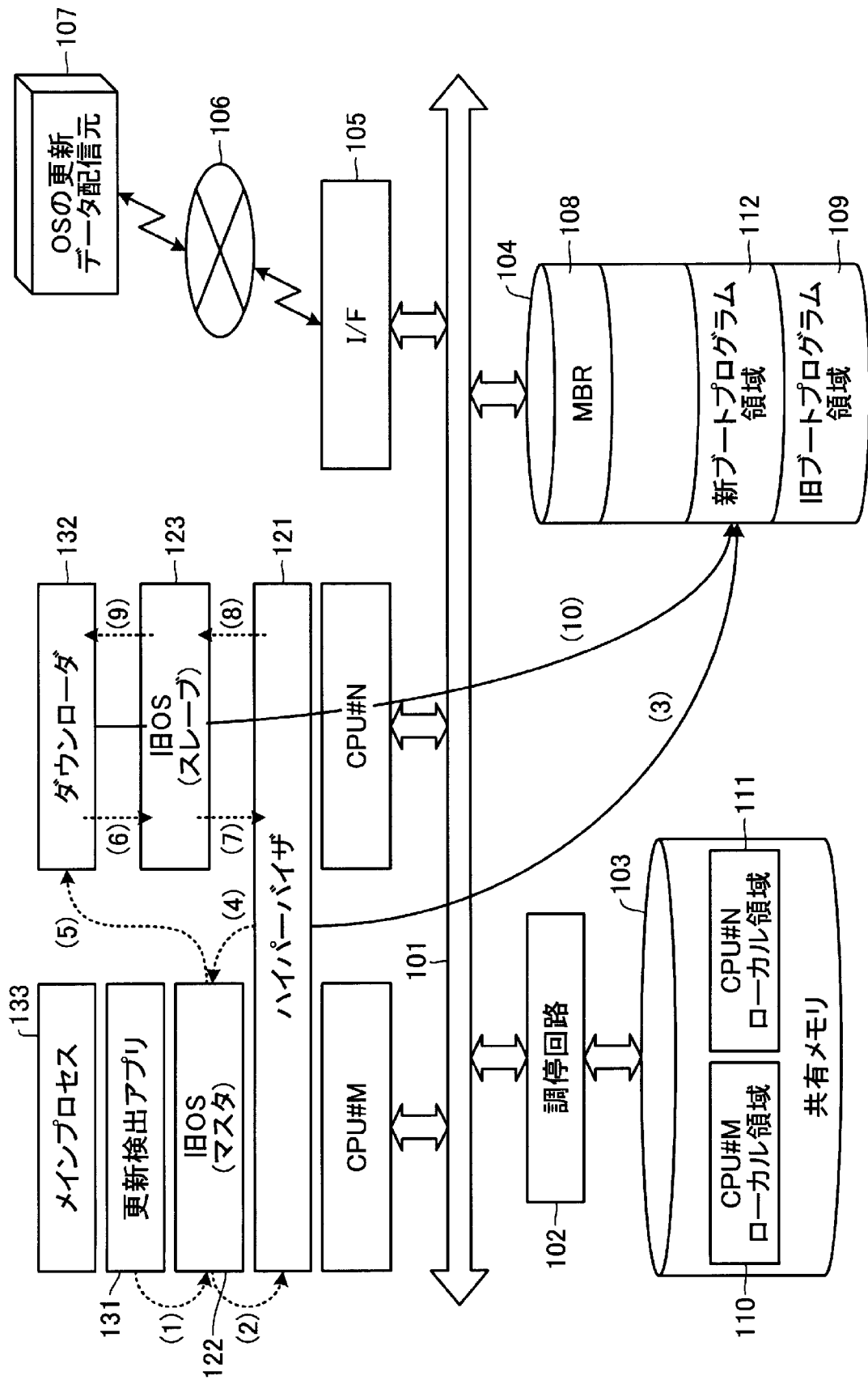
[図1]



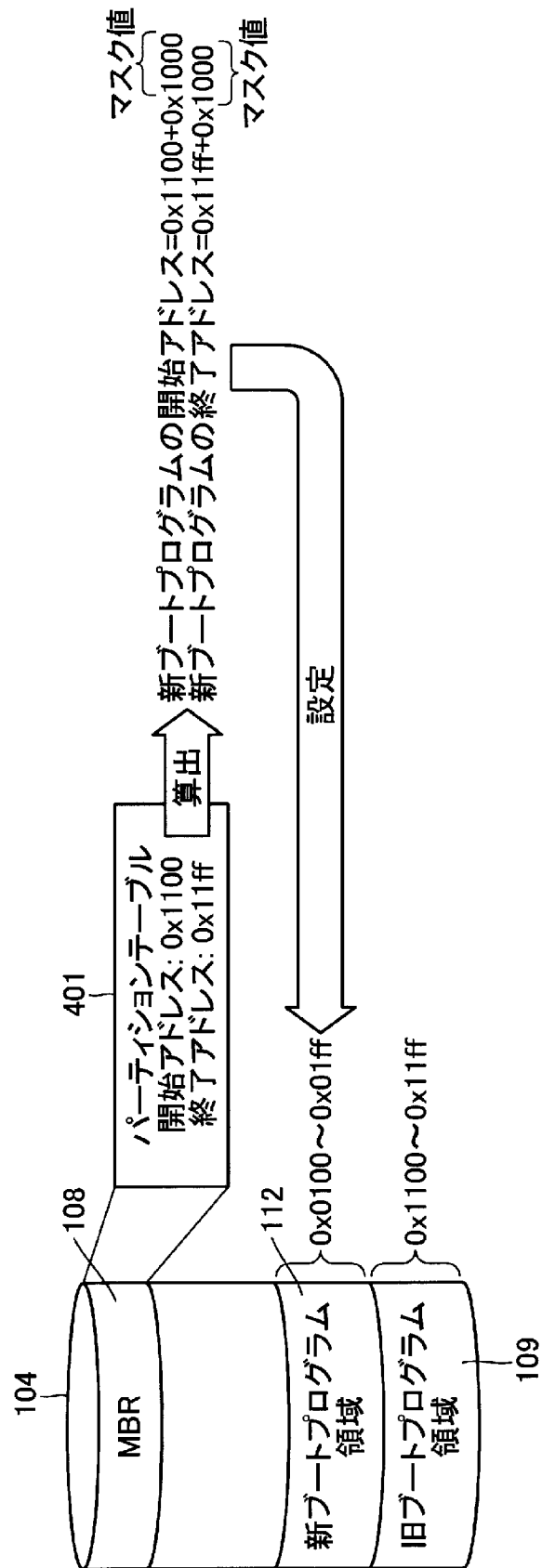
[図2]



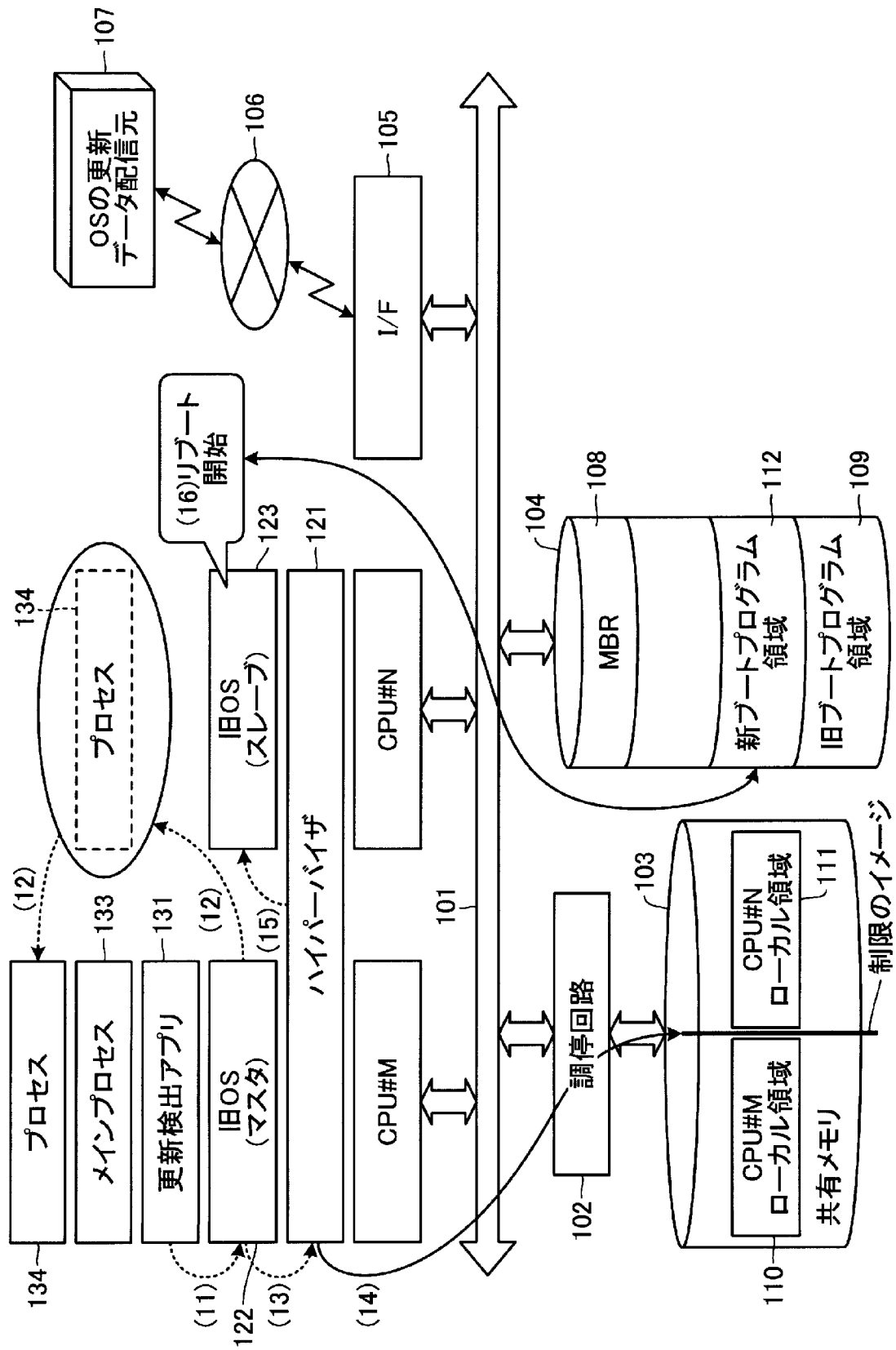
[図3]



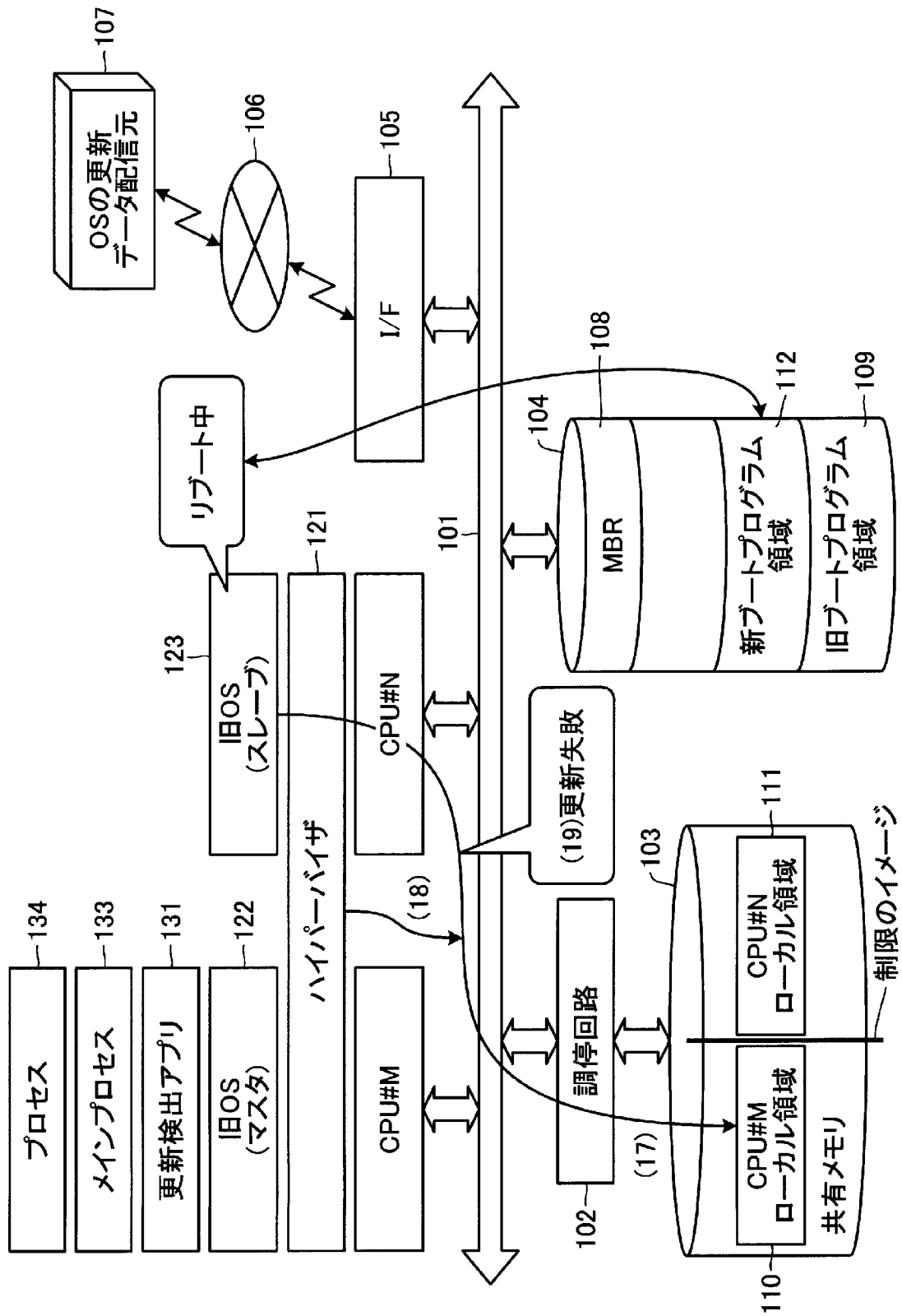
[図4]



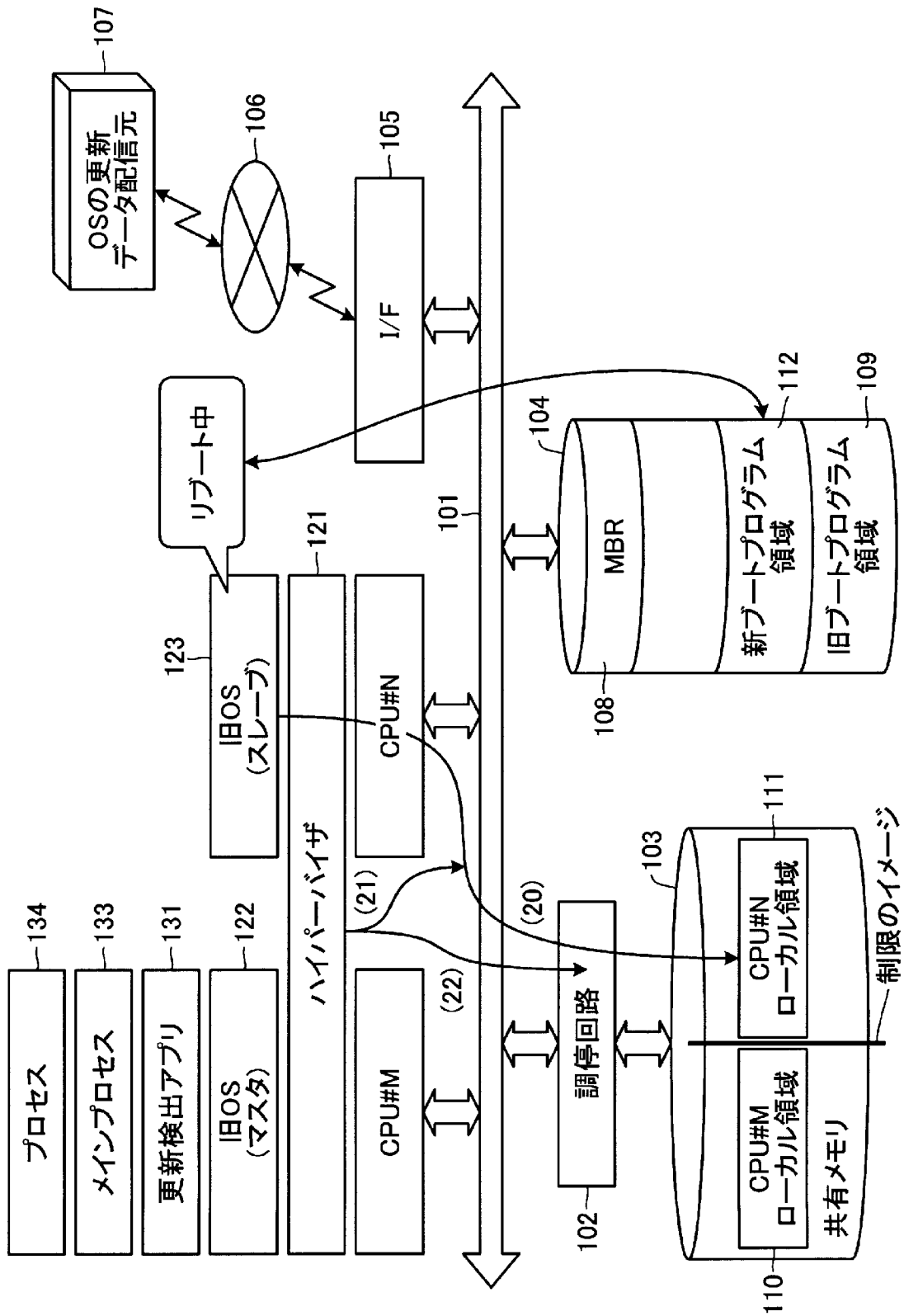
[図5]



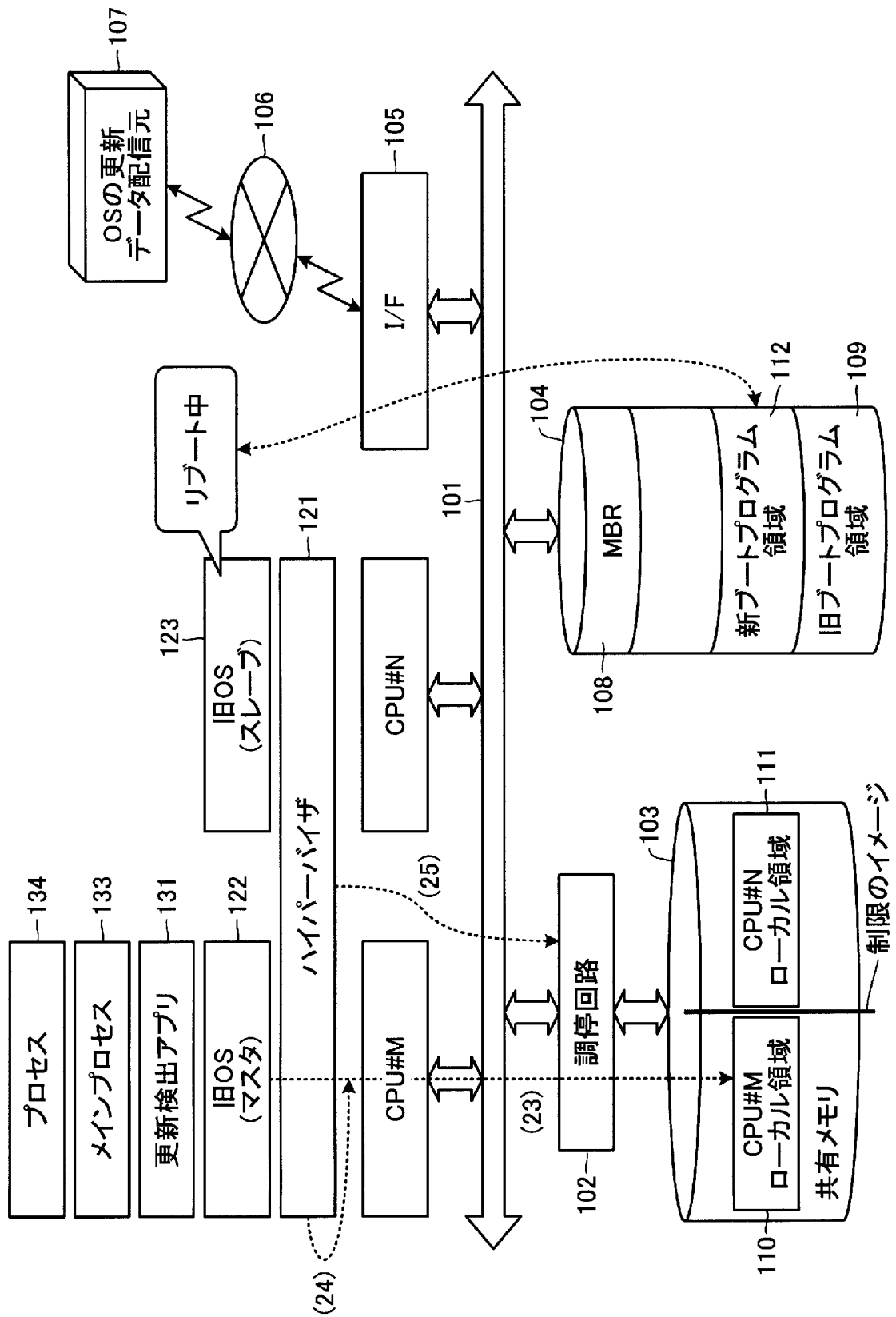
[図6]



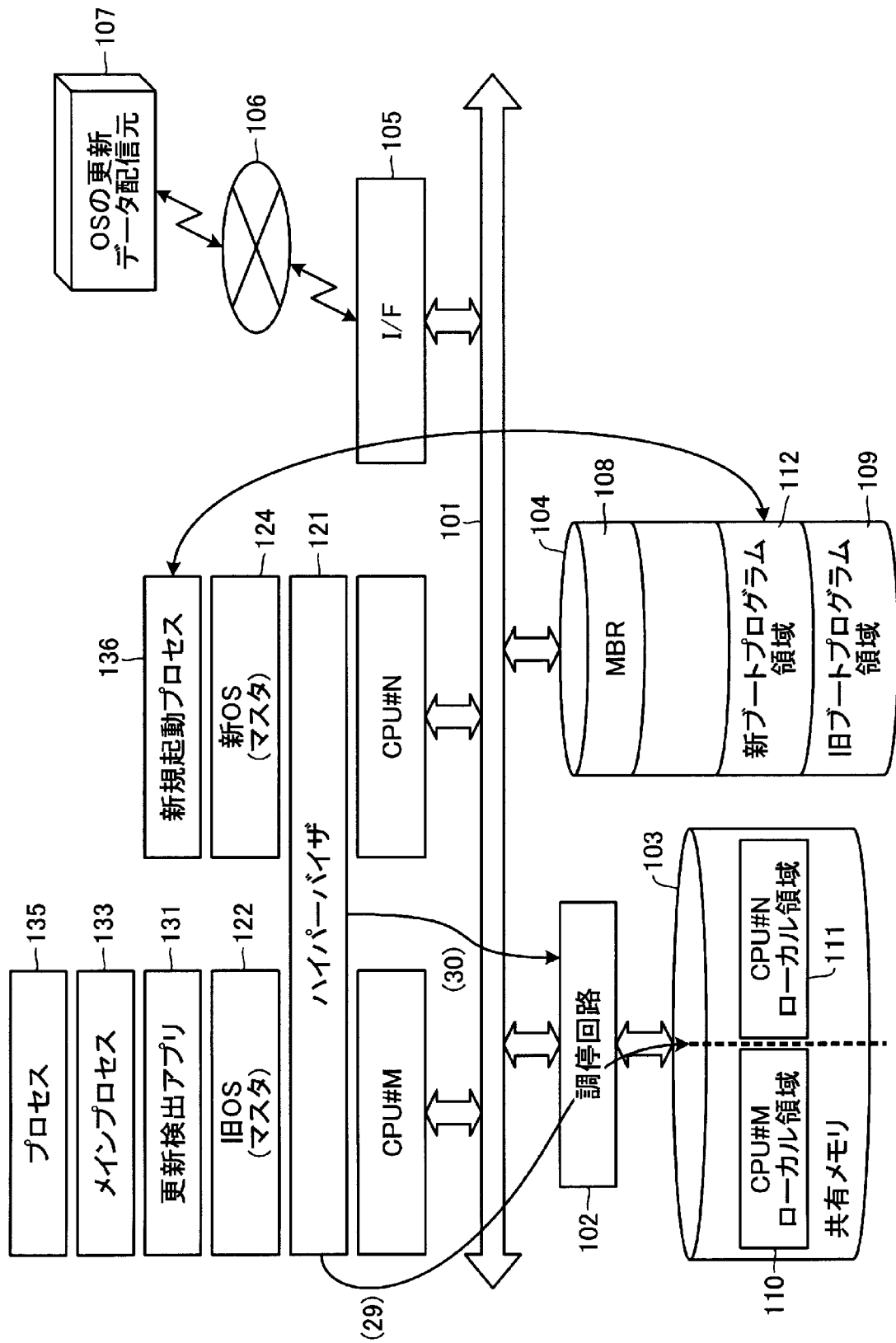
[図7]



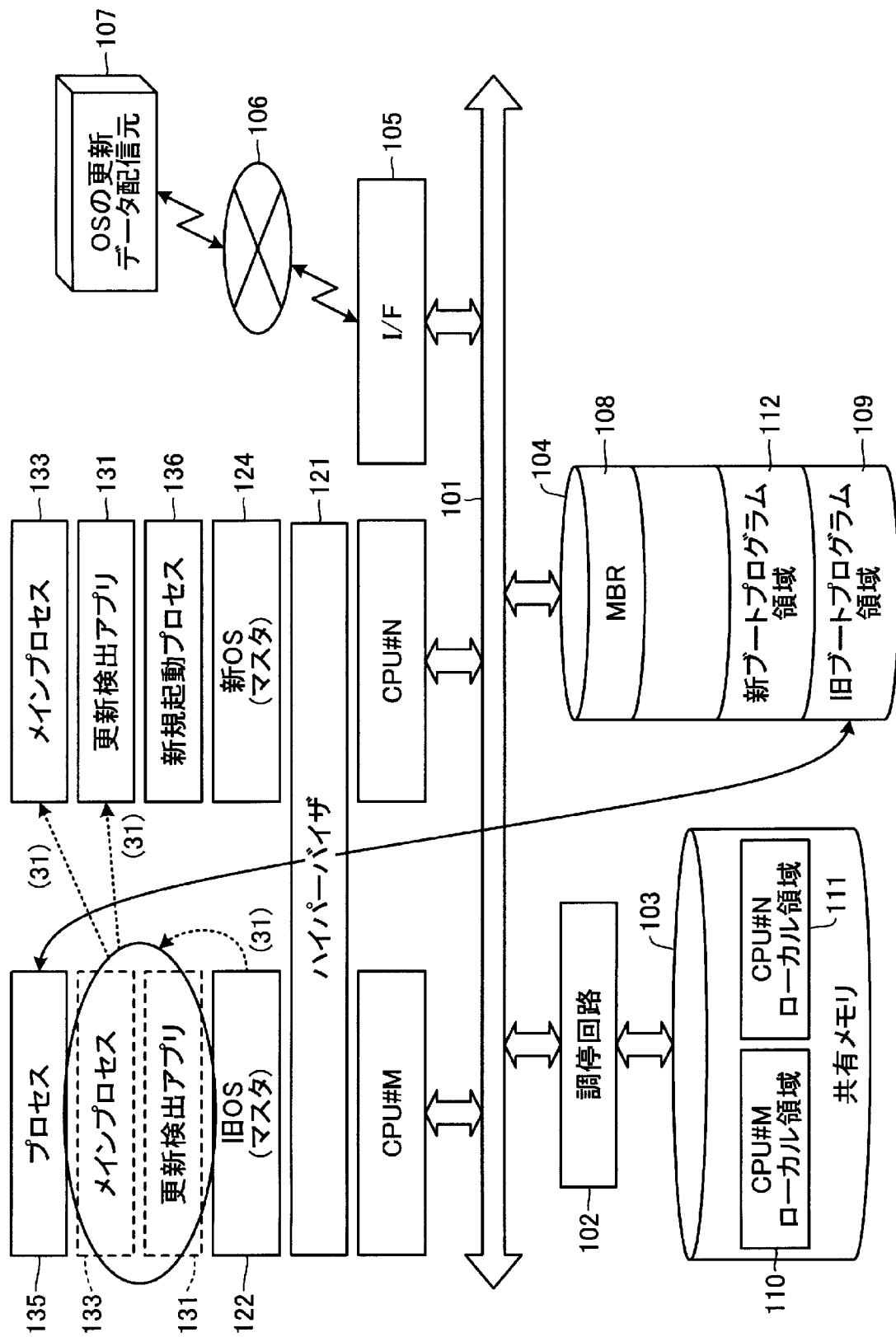
[図8]



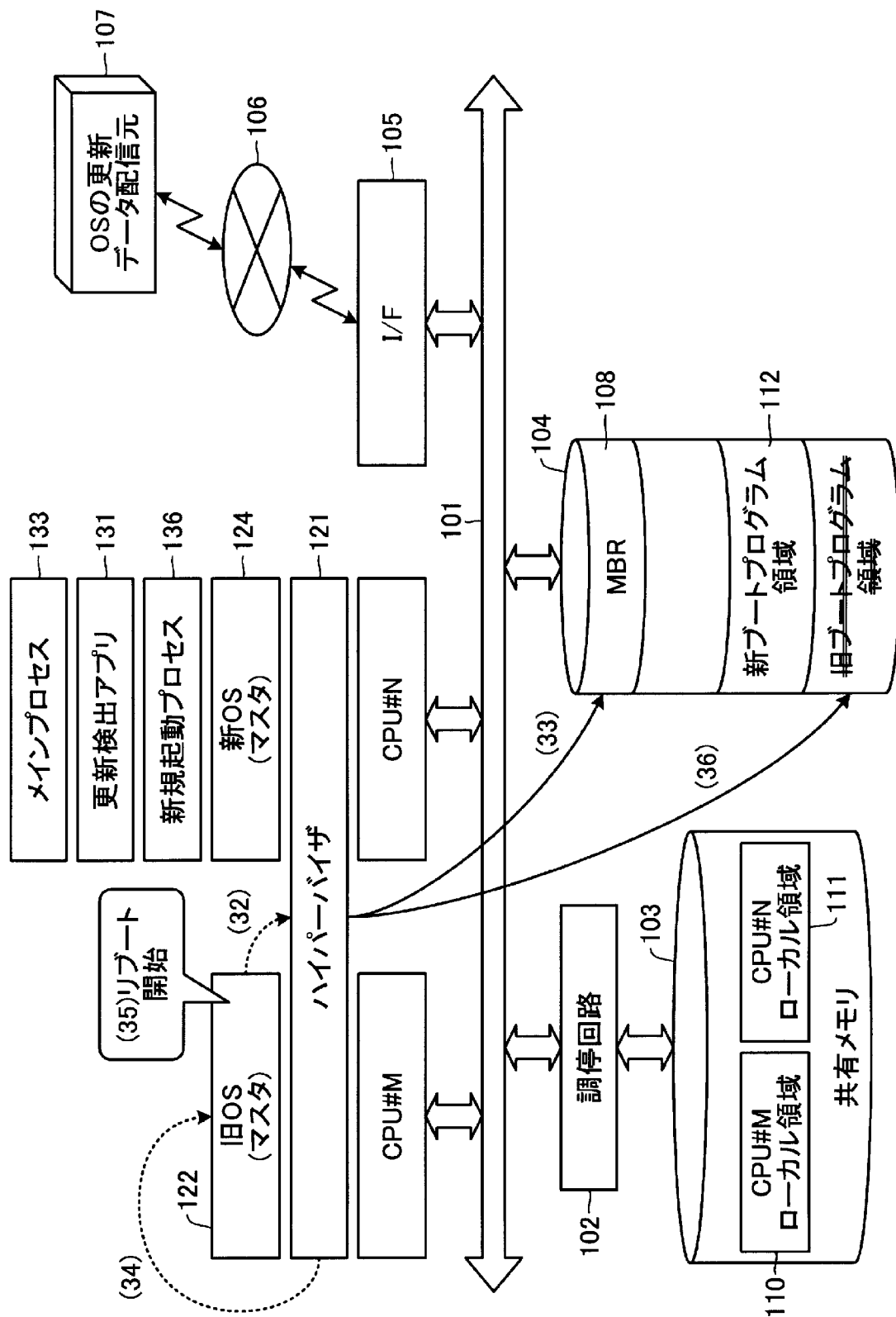
[図10]



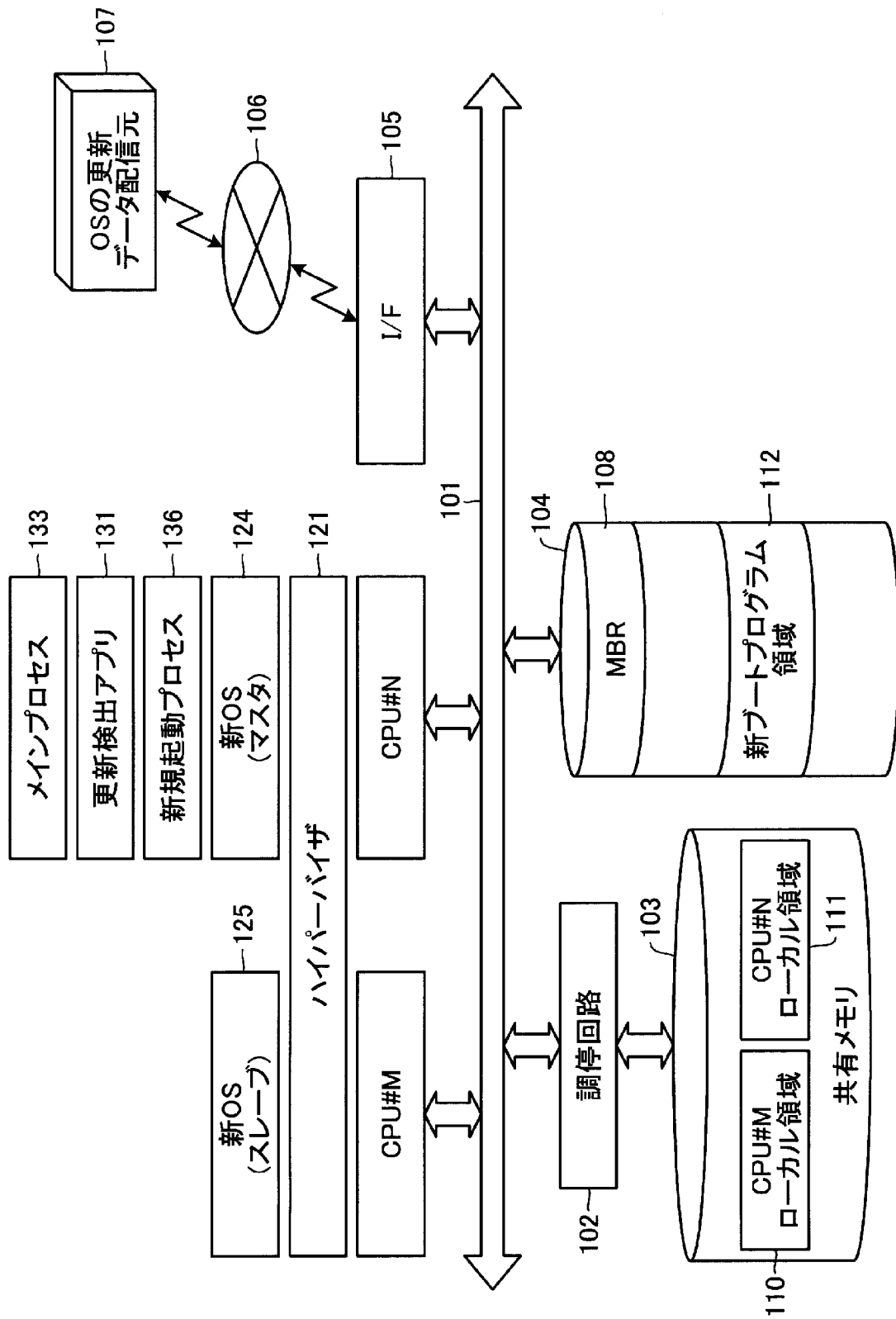
[図11]

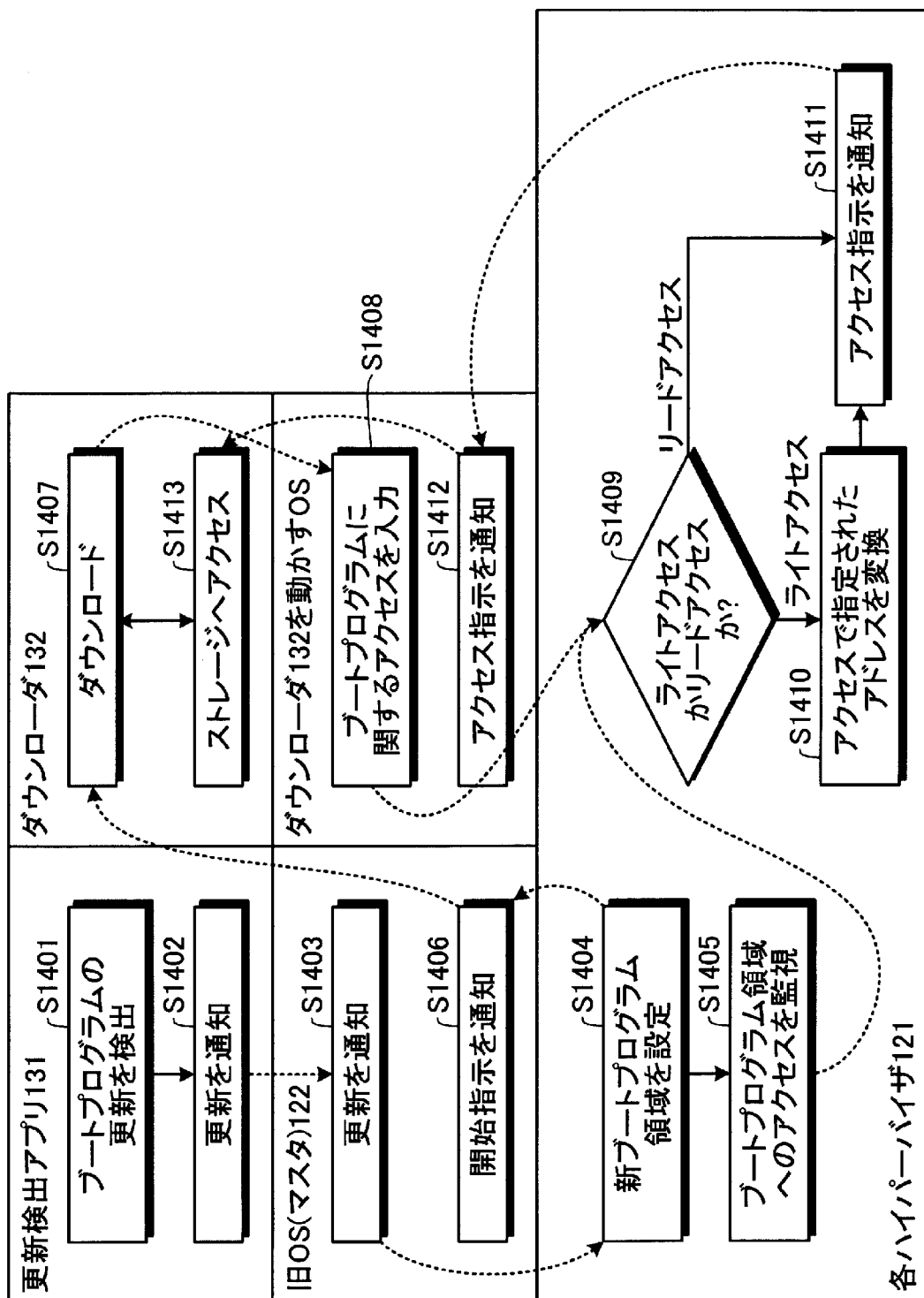


[図12]

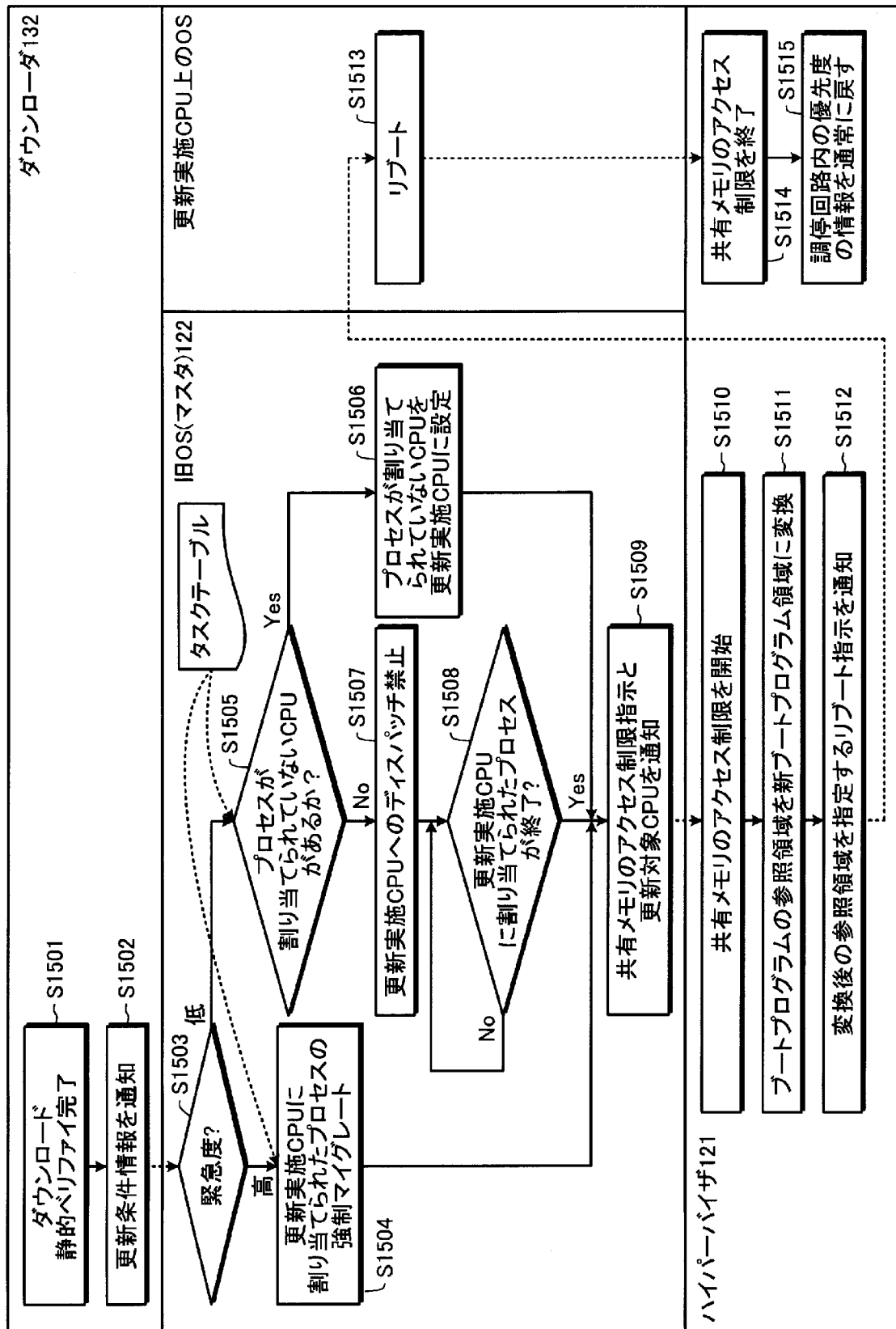


[図13]

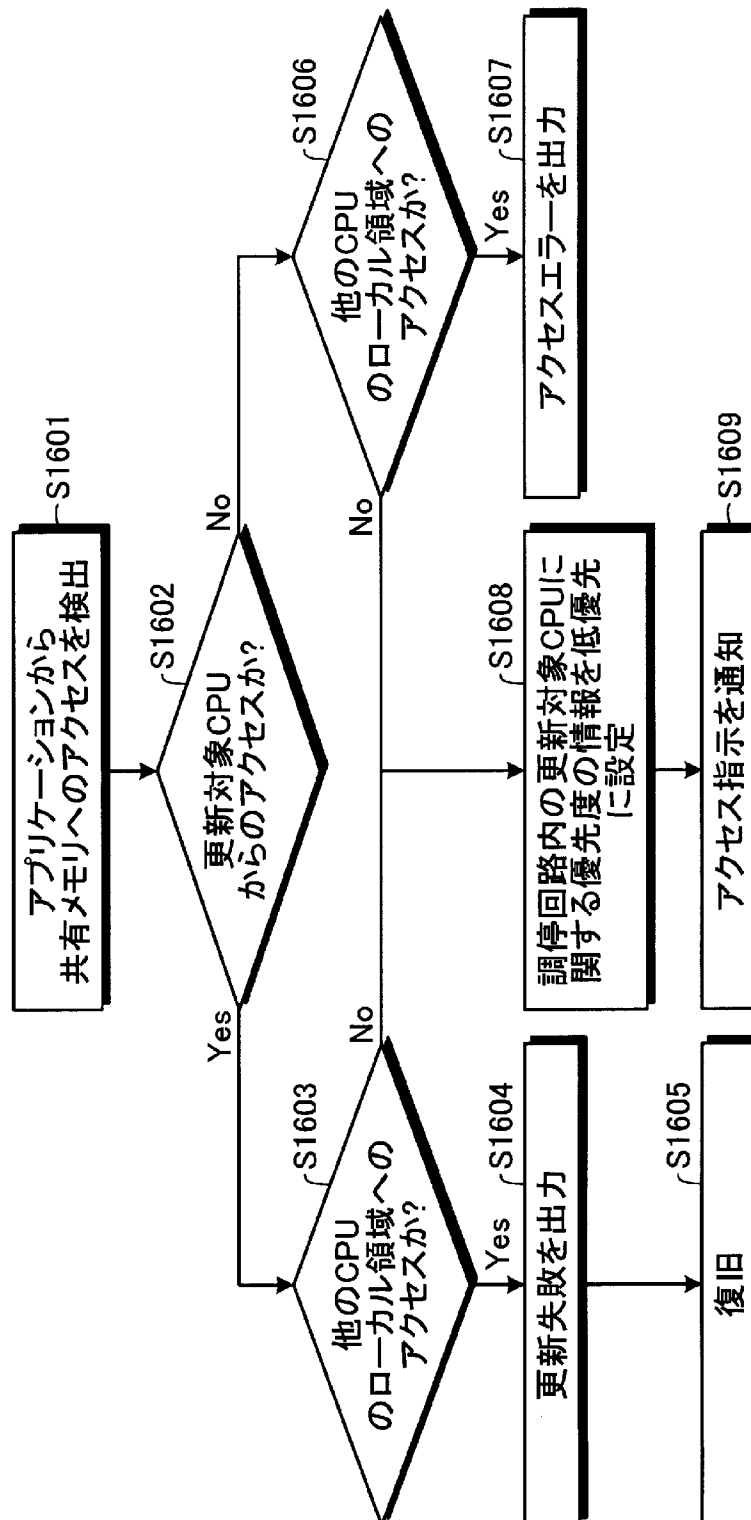




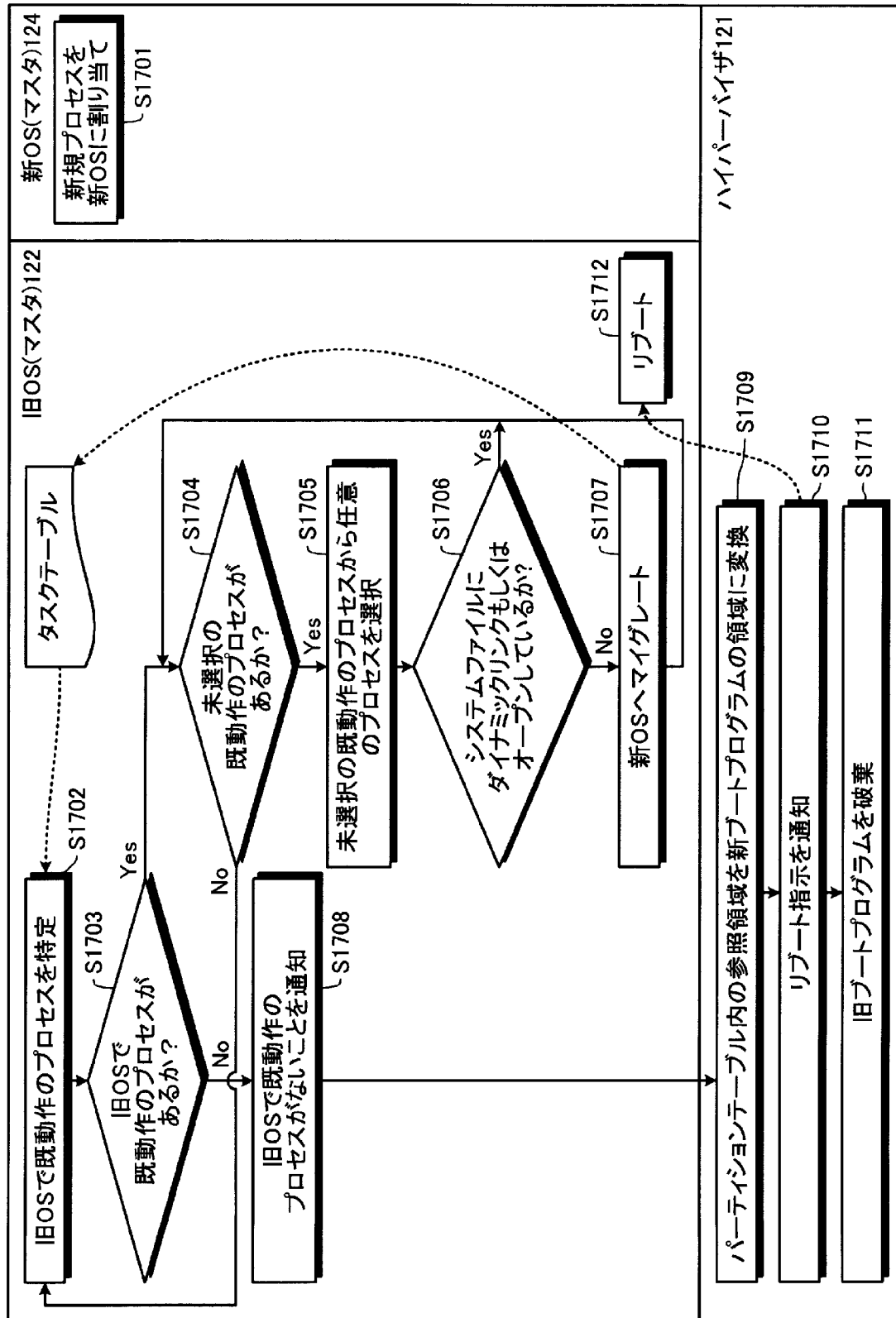
[図15]



[図16]



[図17]



INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP2010/054606

A. CLASSIFICATION OF SUBJECT MATTER

G06F11/00 (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)

G06F11/00

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 10-260845 A (Fujitsu Ltd.), 29 September 1998 (29.09.1998), entire text; all drawings (Family: none)	1, 2, 5, 6
A	same as the above	3, 4
Y	JP 2007-219696 A (Fujitsu Ltd.), 30 August 2007 (30.08.2007), entire text; all drawings (Family: none)	1, 2, 5, 6
A	same as the above	3, 4



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

02 April, 2010 (02.04.10)

Date of mailing of the international search report

13 April, 2010 (13.04.10)

Name and mailing address of the ISA/

Japanese Patent Office

Authorized officer

Facsimile No.

Telephone No.

INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP2010/054606

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 2007-193596 A (NEC Corp.), 02 August 2007 (02.08.2007), entire text; all drawings & US 2007/0169098 A1	1, 2, 5, 6
A	same as the above	3, 4
A	JP 7-311749 A (Toshiba Corp.), 28 November 1995 (28.11.1995), entire text; all drawings & US 5825649 A	1-6

A. 発明の属する分野の分類（国際特許分類（I P C）） Int.Cl. G06F11/00(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. G06F11/00			
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 1922-1996年 日本国公開実用新案公報 1971-2010年 日本国実用新案登録公報 1996-2010年 日本国登録実用新案公報 1994-2010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 10-260845 A（富士通株式会社）1998.09.29, 全文、全図 （ファミリーなし）	1、2、 5、6	
A	同上	3、4	
Y	JP 2007-219696 A（富士通株式会社）2007.08.30, 全文、全図 （ファミリーなし）	1、2、 5、6	
A	同上	3、4	
☒ 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 02.04.2010		国際調査報告の発送日 13.04.2010	
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100-8915 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） 林 毅 電話番号 03-3581-1101 内線 3545	5B 9193

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 2007-193596 A (日本電気株式会社) 2007.08.02, 全文、全図 & US 2007/0169098 A1	1、2、 5、6
A	同上	3、4
A	JP 7-311749 A (株式会社東芝) 1995.11.28, 全文、全図 & US 5825649 A	1－6