

(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
2017 年 6 月 8 日(08.06.2017)



(10) 国際公開番号
WO 2017/094132 A1

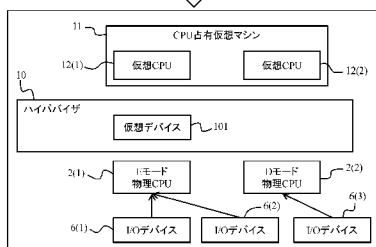
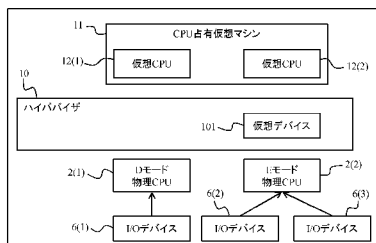
- (51) 国際特許分類:
G06F 9/48 (2006.01) **G06F 9/46** (2006.01)
- (21) 国際出願番号: PCT/JP2015/083846
- (22) 国際出願日: 2015 年 12 月 2 日(02.12.2015)
- (25) 国際出願の言語: 日本語
- (26) 国際公開の言語: 日本語
- (71) 出願人: 株式会社日立製作所(HITACHI, LTD.)
[JP/JP]; 〒1008280 東京都千代田区丸の内一丁目
6 番 6 号 Tokyo (JP).
- (72) 発明者: 田口 学豊(TAGUCHI, Gakuho); 〒
1008280 東京都千代田区丸の内一丁目 6 番 6 号
株式会社日立製作所内 Tokyo (JP).
- (74) 代理人: 特許業務法人ウィルフォート国際特許
事務所(WILLFORT INTERNATIONAL PATENT
FIRM); 〒1030016 東京都中央区日本橋小網町 1
9-7 日本橋TCビル 1階 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, BN, 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, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, 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, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), ユーラシア (AM, AZ, BY, KG, KZ, 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, KM, ML, MR, NE, SN, TD, TG).

[続葉有]

(54) Title: COMPUTER AND METHOD FOR CONTROLLING COMPUTER

(54) 発明の名称: 計算機および計算機の制御方法



11 CPU occupying virtual machine
12(1) Virtual CPU
12(2) Virtual CPU
10 Hypervisor
101 Virtual device
2(1) D mode physical CPU
2(2) E mode physical CPU
6(1) I/O device
6(2) I/O device
6(3) I/O device
11 CPU occupying virtual machine
12(1) Virtual CPU
12(2) Virtual CPU
10 Hypervisor
101 Virtual device
2(1) D mode physical CPU
2(2) E mode physical CPU
6(1) I/O device
6(2) I/O device
6(3) I/O device

(57) Abstract: The computer 1 according to the present invention includes a plurality of physical processors 2, a hypervisor 10, and a virtual machine 11 controlled by the hypervisor. The virtual machine includes a virtual processor 12 and a guest program 13. The hypervisor produces a first physical processor that processes an external interruption in a first execution mode and a second physical processor that processes an external interruption in a second execution mode. The first physical processor has all received external interruptions processed by the guest program on the virtual processor allocated to the first physical processor. The second physical processor has all received external interruptions subjected to emulation processing by the hypervisor. The hypervisor controls whether an external interruption is destined to the first physical processor or the second physical processor.

(57) 要約: 本発明の計算機 1 は、複数の物理プロセッサ 2 と、ハイパバイザ 10 と、ハイパバイザにより制御される仮想計算機 11 を備える。仮想計算機は、仮想プロセッサ 12 と、ゲストプログラム 13 とを含む。ハイパバイザは、第 1 実行モードで外部割込みを処理する第 1 物理プロセッサと、第 2 実行モードで外部割込みを処理する第 2 物理プロセッサとを生成する。第 1 物理プロセッサは、受け付けた全ての外部割込みを、第 1 物理プロセッサに割り当てられた仮想プロセッサ上のゲストプログラムにより処理させる。第 2 物理プロセッサは、受け付けた全ての外部割込みを、ハイパバイザでエミュレーション処理させる。ハイパバイザは、外部割込みの宛先を第 1 物理プロセッサまたは第 2 物理プロセッサのいずれにするかを制御する。



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： 計算機および計算機の制御方法

技術分野

[0001] 本発明は、計算機および計算機の制御方法に関する。

背景技術

[0002] サーバの運用コストおよび複雑さを低減するために、複数サーバを一台のサーバにまとめる仮想計算機技術が注目されている。仮想計算機技術とは、ハイパバイザと呼ばれるソフトウェアが、物理計算機資源を分割する処理を行うことで、一つの物理サーバ上で複数のOSを稼働させる技術である。

[0003] 分割された物理計算機資源が割り当てられることで実現される、複数の仮想的な計算機を仮想マシンと呼ぶ。仮想マシン上で動作するOSをゲストOSと呼ぶ。ゲストOSおよびゲストOS上で動作するアプリケーションプログラムを総称して、ゲストと呼ぶ。仮想マシンは、1つ以上の仮想CPUを有し、仮想CPUに対して物理CPUが割り当てられることで、ゲストは処理を実行することができる。

[0004] ゲストは、仮想マシン上においても、物理計算機環境にある場合と同じ動作をする。しかし、システム資源の構成を変えようとするイベントやシステム資源の構成に動作が依存しているイベントについては、ハイパバイザおよびハードウェアによる適切なエミュレーションが必要となる。このようなイベントをセンシティブなイベントと呼ぶ。センシティブなイベントを処理するためにエミュレーションを行う必要がある。このエミュレーションに伴うオーバーヘッドによって、仮想計算機環境では、非仮想計算機環境よりも性能が低下する。そのため、このオーバーヘッドをできるだけ低減することが求められる。

[0005] エミュレーションを支援するハードウェア機能は、従来、メインフレームのような高価な計算機のみに搭載されていた。しかし、近年では仮想化技術の普及に伴い、安価でより一般的なCPUにおいても、このような機能が搭

載されつつある。例えば、Intel（登録商標）はIntel VT（Virtualization Technology）という名称で、AMD（登録商標）はAMD-Vという名称で、CPUによる仮想化支援機能を提供している。

[0006] 仮想化支援機能は、CPUの動作モードとして、新たに2つの動作モードを用意している。1つは、ゲストを動作させる動作モードであり、以下ではゲストモードと呼ぶ。もう1つは、ハイパバイザを動作させる動作モードであり、以下ではホストモードと呼ぶ。ゲストモードで実行中にセンシティブなイベントが発生すると、CPUの動作モードは自動的にゲストモードからホストモードへ遷移し、ハイパバイザが適切なエミュレーションを行う。

[0007] 以下、Intel VTを例として、CPUによる仮想化支援機能について説明する。Intel VTでは、CPUの動作モードがゲストモードからホストモードに遷移することを”VM Exit”といい、逆にホストモードからゲストモードに遷移することを”VM Entry”という。仮想CPUは、それぞれVMCS（Virtual Machine Control Structure）というデータ構造を持つ。

[0008] ハイパバイザは、VMCSを設定することで、仮想CPUのVM Exit/VM Entryの挙動を制御することができる。VM Exit/VM Entryに伴うプロセス状態の情報の受け渡しも、VMCSを介して行う。

[0009] 仮想化支援機能の1つに、ゲストOSにI/Oデバイスの割込みを伝達するための、External Interrupt Exitingという機能がある。External Interrupt Exitingとは、物理Local APICに割込みが届いたことを契機にVM Exitする機能である。物理Local APICとは、物理CPUごとに1つ存在する、割込みを制御するハードウェアである。ゲストOSへのI/Oデバイス割込み伝達の従来技術では、External Interrupt Exiting機能を、以下のように用いている。

[0010] ハイパバイザは、全てのVMCSの”external Interrupt Exiting” VM-execution controlフィールドの値を1とすることで、全ての仮想CPUのExternal Interrupt Exiting機能を有効にする。これにより、ハイパバイザは、全てのI/Oデバイスからの割込みを処理することができる。割込みを受け

取ってホストモードに遷移したCPUでは、ハイパバイザがその割込みの発生源を特定し、適切な仮想CPUに仮想割込みを送る。

- [0011] 仮想割込みとは、ハイパバイザによるソフトウェアエミュレーションとCPUの仮想化支援機能との協働によって実現する、仮想CPUに対する仮想的な割込みである。この仮想割込みは、仮想CPUが有する仮想Local APICを介して、仮想CPUに伝達される。仮想割込みを受け取った仮想CPUでは、ゲストInterrupt Descriptor Table (IDT) を通して、割込みハンドラが実行される。
- [0012] ゲストIDTとは、仮想CPUごとに1つ存在し、その仮想CPUに届いた割込みにゲストOSの割込みハンドラを関連付けるためのテーブルである。ゲストOSが、割込みに対してベクタ値を設定し、そのベクタ値に対応するゲストIDTのエントリに割込みハンドラを設定することで、割込みと割込みハンドラとの関連付けが行われる。
- [0013] 以上の従来技術を用いた外部割込み処理では、VM Exit、ハイパバイザによるエミュレーション、VM Entryという仮想化特有の処理を行う必要があるため、非仮想化環境に比べて大きな性能低下が生じる。そこで、ハイパバイザの介入なしに、ゲストOSに外部割込みを直接処理させる方法が考えられている。
- [0014] ゲストOSに外部割込みを直接処理させる方法を仮想計算機環境で実現するためには、任意の外部割込みの処理方式をハイパバイザが制御できるようにする必要がある。ハイパバイザが、「ゲストOSが直接処理する方式」または「ハイパバイザがエミュレーションする方式」のいずれかを制御できることが必須要件となる。なぜなら、仮想計算機環境では、必ず、ハイパバイザが介入すべき外部割込みとゲストが直接実行できる外部割込みとが混在しており、かつ、それらの外部割込みをゲストOSおよびハードウェアは判別できないからである。
- [0015] 上記を実現する方法として、例えば、特許文献1の技術では、シャドウIDTを用いた方法が提案されている。シャドウIDTとは、ゲストIDTを

代替する形で、ハイパバイザがゲストOSのカーネル空間に用意した特別なIDTである。シャドウIDTの目的は、外部割込みの種類に応じて処理することにある。ハイパバイザの介入が必要ない割込みが届いた場合は、ゲストOSが設定した割込みハンドラに分岐させて、ゲストモードのままで、その外部割込みを処理する。これに対し、ハイパバイザの介入が必要な割込みが届いた場合は、ゲストモードからホストモードに遷移して、ハイパバイザのエミュレーションを介してその外部割込みを処理する。

[0016] シャドウIDTを用いることで、ハイパバイザの介入が必要な割込みが届いたときに、ホストモードに遷移してハイパバイザのエミュレーションを行う方法は、以下のように実現されている。

[0017] ハイパバイザは、シャドウIDTにおいて、ハイパバイザの介入が必要な割込みに対応したエントリだけPresentビットを0に設定する。物理CPUにハイパバイザの介入が必要な外部割込みが届き、割込みハンドラを実行するためにそのエントリにアクセスすると、物理CPUは例外を起こす。例外が発生すると強制的にホストモードに遷移する仮想化支援機能を利用することで、その割込みをハイパバイザが処理する。

先行技術文献

特許文献

[0018] 特許文献1: United States Patent 8892802

非特許文献

[0019] 非特許文献1: “Intel 64 and IA-32 Architectures Software Developer’s Manual combined volumes 3A, 3B, 3C: System programming guide”、[online]、2015年6月、Intel Corp、[平成27年8月1日検索]、インターネット<http://www.intel.com/content/dam/www/public/us/en/documents/manuals/64-ia-32-architectures-software-developer-system-programming-manual-325384.pdf>

発明の概要

発明が解決しようとする課題

- [0020] 特許文献 1 に記載の技術では、任意の外部割込みの処理方式をハイパバイザが制御するための、新たなデータ構造（シャドウ I D T）を用意している。しかし、ハイパバイザは、ゲスト O S によるゲスト I D T の書き換え時や、I P I（Inter-processor interrupt）の受信時などの契機に、シャドウ I D T をメンテナンスする必要がある。このメンテナンスにおいて、処理コストが発生したり、他 C P U との相互干渉が発生したりする。
- [0021] そこで、本発明では、シャドウ I D T を用いることなく、外部割込みを、仮想プロセッサ上のゲストプログラムで処理したり、あるいはハイパバイザでエミュレーション処理したりすることができる計算機および計算機の制御方法を提供することにある。

課題を解決するための手段

- [0022] 上記課題を解決すべく、本発明に従う計算機は、計算機であって、複数の物理プロセッサと、複数の物理プロセッサのうちの少なくとも一つの物理プロセッサにより実現されるハイパバイザと、ハイパバイザにより制御される少なくとも一つの仮想計算機であって、少なくとも一つの仮想プロセッサと、仮想プロセッサにより実行されるゲストプログラムとを含む仮想計算機と、を備え、ハイパバイザは、第 1 実行モードで外部割込みを処理する第 1 物理プロセッサと、第 2 実行モードで外部割込みを処理する第 2 物理プロセッサとを、複数の物理プロセッサから生成することができ、第 1 物理プロセッサは、受け付けた全ての外部割込みを、第 1 物理プロセッサに割り当てられた仮想プロセッサ上のゲストプログラムにより処理させ、第 2 物理プロセッサは、受け付けた全ての外部割込みを、ハイパバイザでエミュレーション処理させ、ハイパバイザは、外部割込みの宛先を第 1 物理プロセッサまたは第 2 物理プロセッサのいずれにするかを制御する。

発明の効果

- [0023] 本発明によれば、ハイパバイザは、第 1 実行モードで外部割込みを処理する第 1 物理プロセッサと、第 2 実行モードで外部割込みを処理する第 2 物理

プロセッサとを、複数の物理プロセッサから少なくとも1つずつ生成することができる。そして、第1物理プロセッサは、受け付けた全ての外部割込みを、第1物理プロセッサに割り当てられた仮想プロセッサ上のゲストプログラムにより処理させ、第2物理プロセッサは、受け付けた全ての外部割込みを、ハイパバイザでエミュレーション処理させることができる。

図面の簡単な説明

[0024] [図1]仮想計算機システムを動作させる物理計算機のブロック図。

[図2]物理計算機および仮想計算機システムの論理的構成を示すブロック図。

[図3]メモリ領域のデータ構造と、物理計算機および仮想計算機システムとの論理的な対応関係を示すブロック図。

[図4]仮想計算機システムの初期状態時から、仮想マシンを立ち上げて、外部割込み実行モードを各物理CPUに適用する例を示す説明図。

[図5]エミュレーションモード物理CPUにおける、ゲストへ外部割込みを直接伝達する様子を示すブロック図。

[図6]エミュレーションモード物理CPUにおける、ゲストへ外部割込みを伝達する処理を示すフローチャート。

[図7]直接実行モード物理CPUにおける、ゲストへ外部割込みを伝達する様子を示すブロック図。

[図8]直接実行モード物理CPUにおける、ゲストへ外部割込みを伝達する処理を示すフローチャート。

[図9]第2実施例に係り、仮想マシンの構成変更に伴って、外部割込み実行モードを動的に切り替える例を示す説明図。

[図10]物理CPUの縮退時に、ハイパバイザの介入が必要となる割込みの処理を負荷分散させるべく、外部割込み実行モードを動的に切り替える例を示す説明図。

[図11]ハイパバイザの介入が必要となる割込みの宛先を、複数の物理CPU間で定期的に切り替えて負荷分散する場合すべく、外部割込み実行モードを動的に切り替える例を示す説明図。

[図12]外部割込み実行モードを切り替えるための契機を通知する処理を示すフローチャート。

[図13]直接実行モードからエミュレーションモードへの切替を判定する処理を示すフローチャート。

[図14]図 1 3 中の外部割込み実行モード切替処理を示すフローチャート。

[図15]エミュレーションモードから直接実行モードへの切替を判定する処理を示すフローチャート。

[図16]図 1 5 中の外部割込み実行モード切替処理を示すフローチャート。

発明を実施するための形態

[0025] 以下、図面に基づいて、本発明の実施の形態を説明する。以下に述べるように、本実施形態は、仮想計算機環境において、外部割込みをゲストで直接実行することのできる技術を開示する。

[0026] 本実施形態では、物理CPU 2を、「ゲストが直接実行可能な外部割込みのみを受け付けて、受け付けた外部割込みの全てをゲストに直接実行させる物理CPU（第1物理プロセッサ）」と、「ゲストが直接実行可能な外部割込み以外の外部割込みも受け付け可能であり、受け付けた外部割込みの全てをハイパバイザにエミュレーションさせる物理CPU（第2物理プロセッサ）」との2つに大別する。

[0027] ハイパバイザ10は、任意の外部割込みの宛先を、第1物理プロセッサと第2物理プロセッサの間で切り替えることによって、その外部割込みをゲストに直接実行させるか、ハイパバイザにエミュレーションさせるかを、制御することができる。

[0028] さらに本実施形態では、ハイパバイザは、上記の2種類の物理CPU形態（第1物理プロセッサ、第2物理プロセッサ）を、動的に切り替えることができる。これにより、本形態では、外部割込み処理の負荷を分散したり、仮想計算機の構成を変更したりすることができる。

実施例 1

[0029] 図1～図8を用いて第1実施例を説明する。

[0030] <1. ハードウェア構成>

[0031] 図1は、仮想計算機システムを動作させる物理計算機1の構成を示す。物理計算機1は、後述のように、複数のCPU2（プロセッサと呼んでもよい）、ノースブリッジ3、メモリ4、サウスブリッジ5、I/Oデバイス6を備える。

[0032] CPU2は、仮想化支援機能をサポートしている。そのようなCPU2として、仮想化支援機能であるIntel VTまたはAMD-Vをサポートしているx86（もしくはIA-32）互換のCPUを挙げることができる。

[0033] 各CPU2は、それぞれ一つの物理Local APIC（Advanced Programmable Interrupt Controller）7を持ち、バス8Aを介してノースブリッジ3に接続される。ノースブリッジ3には、メモリバス8Bを介してメモリ4が接続されている。ノースブリッジ3には、バス8Cを介して、I/Oデバイス6（PCI Express（登録商標）デバイス）もしくはサウスブリッジ5が接続される。

[0034] ノースブリッジ3は、IOxAPIC（図示せず）を有する。サウスブリッジ5には、I/Oデバイス6がバス8Dを介して接続される。I/Oデバイス6としては、例えば、PCI（Peripheral Component Interconnect）デバイス、LPC（Low Pin Count）デバイス、レガシーデバイスなどがある。

[0035] <2. ソフトウェア構成>

[0036] 図2は、物理計算機1および仮想計算機システムの論理的な構成を示す図である。仮想計算機システムを採用した計算機は、仮想マシン11と、仮想マシン11を制御するハイパバイザ10とを有する。「仮想計算機」としての仮想マシン11は、仮想CPU12と、ゲスト13とを有する。ゲスト13は、ゲストOSおよびゲストOS上で動作するソフトウェアの総称であり、「ゲストプログラム」に該当する。

[0037] 仮想CPU12は、仮想Local APIC14を有する。仮想CPU12とは、ゲスト13が認識する仮想的なCPUであり、この仮想CPU12に対して物理CPU2が割り当てられることで、ゲスト13の処理が実行される。

- [0038] 仮想Local APIC 14は、仮想マシン11に割込みを認識させるために、ハイパバイザ10および物理CPU2の仮想化支援機能が仮想的に作り出したLocal APICである。仮想CPUに割り当てられる物理CPU2には、CPU占有仮想マシンの仮想CPUに割り当てられるものと、CPU共有仮想マシンの仮想CPUに割り当てられるものの2種類がある。
- [0039] CPU占有仮想マシンとは、その仮想マシンが有する一つ一つの仮想CPUに対して、特定の物理CPUが占有的に割り当てられるものである。CPU共有仮想マシンとは、その仮想マシンが有する仮想CPUに対して、複数の物理CPUが時分割で割り当てられるものである。
- [0040] 本実施例では、前者を占有割り当て物理CPU2と呼び、後者を共有割り当て物理CPU2と呼ぶ。また、ある物理CPU2に対して、その時点で実行させる仮想CPU12を選ぶことを、仮想CPU12のディスパッチという。
- [0041] なお、図中では、複数の、物理CPU2、仮想マシン11、仮想CPU12、ゲスト13などに対してカッコ付きの数字をそれぞれ添えているが、特に区別しない場合は、カッコ付きの数字を除いて説明する。
- [0042] 図3は、本実施例で使用するデータ構造と、物理計算機1および仮想計算機システムとの論理的な対応関係を示した説明図である。仮想CPU管理データD11は、ハイパバイザ10が仮想CPU12の挙動を制御するために設定するデータである。仮想CPU管理データD11は、IntelではVMCS (Virtual Machine Control Structure) と呼ばれており、AMDではVMCB (Virtual Machine Control Block) と呼ばれる。仮想CPU管理データD11は、仮想CPU12ごとに用意される。
- [0043] 仮想APICレジスタD12とは、ゲスト13による物理APICレジスタD15へのアクセスを仮想化するためのものである。仮想APICレジスタD15は、仮想割込みを管理するためのデータ構造である。仮想APICレジスタD12をハイパバイザ10およびハードウェアが適切に管理することで、仮想Local APIC 14が実現する。

- [0044] 外部割込み実行モード識別データD13は、対応する物理CPU2の外部割込み実行モードが何であることを識別するためのデータである。切替通知フラグD14とは、対応する物理CPU2に対して、外部割込み実行モードの切替指示が出ていることを通知するためのフラグである。
- [0045] 物理APICレジスタD15は、物理Local APIC7に届いた外部割込みを制御するためのレジスタ、もしくはIPIの生成を制御するためのレジスタである。物理APICレジスタD15は、EOI（End of Interrupt）レジスタ、ISR（In Service Register）などを含む。
- [0046] 割込みパラメータD16とは、I/Oデバイス6のそれぞれの割込みの、ベクタ値や宛先CPUなどのパラメータを決定するデータ構造である。ゲストOSやハイパバイザ10は、割込みパラメータD16を書き換えることによって、I/Oデバイス6の割込みを制御する。
- [0047] なお、外部割込み実行モード識別データD13と切替通知フラグD14は、第2実施例において使用されるデータ構造であるため、詳細は後述する。
- [0048] <3. 外部割込み処理方式の概要>
- [0049] 本実施例では、物理CPU2がゲストモードで実行中に、外部割込みが届いた場合の割込み処理方式を、外部割込み実行方式と呼ぶ。外部割込み実行方式として、直接実行方式とエミュレーション方式の2種類を定義する。ハイパバイザ10は、任意の外部割込みの実行方式を、上述の2つの外部割込み実行方式のいずれとするか制御できる。
- [0050] 直接実行方式では、物理Local APIC7に割込みが届いても、物理CPU2はゲストモードのままである。その割込み（外部割込み）は、物理CPU2上で実行中の仮想CPU12のゲストIDT130（図5参照）を通して、処理される。物理Local APIC7に対する外部割込み処理の終了通知（EOIの発行）も、ハイパバイザ10の介入なしにゲスト13が直接行う。つまり、直接実行方式では、物理CPU2に届いた外部割込みは、ハイパバイザ10を介さずに、ゲスト13によって処理される。従って、直接実行方式の場合、物理CPU2はゲストモードからホストモードへ移行する必要がなく、

比較的短時間で外部割込みを処理できる。

- [0051] エミュレーション方式を説明する。エミュレーション方式では、物理Local APIC 7 に外部割込みが届くと、物理CPU 2 はホストモードに遷移し、その割込みの処理がハイパバイザ 10 に渡る。ハイパバイザ 10 は、その割込みをエミュレーションし、適切な仮想CPU 12 へ割込みを伝達する。
- [0052] 仮想CPU 12 へ伝達された割込みは、その仮想CPU 12 がゲストモードの物理CPU 2 にディスパッチされた際に、ゲスト 13 によって処理される。ゲスト 13 によるEOI発行についてもハイパバイザ 10 が介入する。ハイパバイザ 10 は、物理Local APIC 7 に向けたEOIではなく、仮想Local APIC 14 へのEOI発行に変換する。このようにエミュレーション方式では、ハイパバイザ 10 によるソフトウェア処理が入るため、直接実行方式に比べて、外部割込みの処理性能が低下する。
- [0053] これら2つの外部割込み実行方式は、仮想CPU 12 単位で指定できる。物理CPU 2 に外部割込みが届いた場合、その外部割込みが直接実行方式またはエミュレーション方式のいずれで処理されるかは、その物理CPU 2 にディスパッチされている仮想CPU 12 の設定によって決まる。
- [0054] 直接実行方式では、物理CPU 2 に届いた外部割込みのベクタ値がそのままゲスト 13 に伝達される。従って、直接実行方式で処理可能な外部割込みとは、その外部割込みが届く物理CPU 2 にディスパッチされている仮想CPU 12 のゲストIDT 130 に、事前に割り当てられている外部割込みである。ゲスト 13 (詳しくはゲストOS) は、ゲストIDT 130 に外部割込みを予め設定することができる。
- [0055] 仮想CPU 12 の設定は、ハイパバイザ 10 が仮想CPU管理データD 11を書き換えることで行う。以下、仮想CPU 12 をどのように設定すれば、直接実行方式およびエミュレーション方式を実現できるかについて説明する。
- [0056] 仮想CPU 12 の外部割込み実行方式をエミュレーション方式とする場合は、仮想CPU 12 を以下のように設定する。

(E a) ゲストモード時に外部割込みが届くと、ホストモードへの遷移が起こる。

(E b) ゲストモード時に物理EOIレジスタへアクセスすると、ホストモードへの遷移が起こる。

(E c) 仮想割込みの扱いが有効である。

[0057] 仮想CPU12の外部割込み実行方式を直接実行方式とする場合は、仮想CPU12を以下のように設定する。

(D a) ゲストモード時に外部割込みが届いても、ホストモードへ遷移しない。

(D b) ゲストモード時に物理EOIレジスタ（図示せず）へアクセスしても、ホストモードへ遷移しない。

(D c) 仮想割込みの扱いが無効である。

[0058] 物理EOIレジスタとは、物理APICレジスタD15に含まれるレジスタである。物理EOIレジスタに書き込むと、該当する物理Local APIC7にEOIが発行される。仮想割込みとは、仮想Local APIC14を通して仮想CPU12に伝達される、仮想的な割込みである。仮想割込みは、ハイパバイザ10およびハードウェアの仮想化支援機能により生成される。

[0059] 上述した設定方法を、IntelのCPUを例にあげて説明する。ゲストモード時に外部割込みが届くとホストモードへの遷移が起こる機能を有効にするためには、VMCSの”external interrupt exiting” VM-execution controlというフィールドを1に設定する。逆にこの機能を無効にするためには、0を設定する。

[0060] 一般的には通常、物理APICレジスタD15へのアクセスはホストモードへの遷移を引き起こすように設定する。しかし、X2APICをサポートするCPUの場合は、VMCSの”use MSR bitmap” VM-execution controlフィールドを1に設定し、さらにMSR bitmapを適切に設定する。これにより、物理APICレジスタD15のうち物理EOIレジスタのみを、ホストモードへ遷移することなくアクセスできる。仮想割込みの扱いを有効にする場合はV

MCSの” Virtual-interrupt delivery” を1に設定し、無効にする場合は0に設定する。

[0061] <4. 物理CPUの外部割込み実行モード>

[0062] 外部割込み実行方式は、仮想CPU12ごとに設定可能である。本実施例では、物理CPU単位で、外部割込み実行方式を制御する。すなわち、一つの物理CPU2上で動作し得る全ての仮想CPU12の外部割込み実行方式は、直接実行方式かエミュレーション方式のどちらか一方に統一する。

[0063] 本実施例では、全ての外部割込みを直接実行方式で処理する物理CPU2を、直接実行モード物理CPUと呼ぶ。全ての外部割込みをエミュレーション方式で処理する物理CPUを、エミュレーションモード物理CPUと呼ぶ。またこれら2つの、物理CPU単位で設定する外部割込み実行方式を、外部割込み実行モードと呼ぶ。

[0064] ある物理CPU2において、その上で動作する仮想マシン11が存在しない場合（言い換えれば、その物理CPU2が割り当てられ得る仮想CPU12が無い場合）、その物理CPUの外部割込み実行モードはエミュレーションモード物理CPUとなる。なぜなら、そのような物理CPUはゲストモードで動作することはないため、その物理CPUに届く割り込みは、常にハイパバイザによって処理されるためである。このような物理CPUは、仮想計算機システムの初期状態のような、仮想マシンが一つも立ち上げられていない場合などに存在する。

[0065] 直接実行モード物理CPUは、「第1実行モードで外部割込みを処理する第1物理プロセッサ」に該当する。エミュレーションモード物理CPUとは、「第2実行モードで外部割込みを処理する第2物理プロセッサ」に該当する。

[0066] 外部割込みが上記2種類の外部割込み処理方式のうちいずれによって処理されるかは、その割込みの宛先となっている物理CPU2の外部割込み実行モードにより定まる。ハイパバイザ10は、任意の外部割込みの宛先CPUを制御できる。さらに、後述の実施例で述べるように、ハイパバイザ10は

、任意の物理CPU 2の外部割込み実行モードを動的に切り替えることができる。

[0067] ハイパバイザ10は、任意の外部割込みの宛先CPUを、直接実行モード物理CPUとエミュレーションモード物理CPUのいずれにするかを制御できる。複数の物理CPU 2のうち少なくともいずれか一つの物理CPU 2は、ホストモードで動作しているため、ハイパバイザ10は、外部割込みの宛先となる物理CPU 2を決定できる。すなわち、ハイパバイザ10は、外部割込みを処理する方式を制御することができる。

[0068] 本実施例では、各物理CPU 2がどちらの外部割込み実行モードで動作しているかを判別するために、外部割込み実行モード識別データD13を用意している。外部割込み実行モード識別データD13には、各物理CPU 2で現在実行中の外部割込み実行モードを識別するための情報が格納される。

[0069] まず、直接実行モード物理CPUについて説明する。直接実行モード物理CPUとは、その物理CPUが占有的に割り当てられている仮想CPUの外部割込み実行方式が直接実行方式である物理CPUである。

[0070] 直接実行モードが適用可能な物理CPUの条件は、以下の二つを満たすことである。

- ・占有割り当て物理CPUである。
- ・ハイパバイザの介入を必要とする外部割込みの宛先となっていない物理CPUである。

[0071] また、外部割込み実行モード物理CPUでは、その物理CPUを宛先とする外部割り込みに関して、ハイパバイザの特別な処理が必要となるため、以下で説明する。

[0072] 直接実行モード物理CPUでは、物理Local APIC 7に届いた外部割込みは全て、ハイパバイザの処理を介することなく、ゲストによって直接処理される。そのためゲストOSが、その直接実行モード物理CPUが割り当てられた仮想CPUに伝達されるよう設定した外部割込みは、全てその直接実行モード物理CPUを宛先としていなければならない。また、その直接実行モード物

理CPUを宛先とする外部割込みは、その直接実行モード物理CPUが割り当てられた仮想CPUに伝達されるよう、ゲストOSが設定した割り込みだけでなければならない。

[0073] そこで、ゲストOSが外部割込みを、直接実行モード物理CPUが割り当てられている仮想CPUに伝達されるように設定した場合、ハイパバイザはそれをトラップし、その外部割込みの宛先をその仮想CPUに割り当てられている直接実行モード物理CPUとするように、割り込みパラメータを変更する。

[0074] また、直接実行モード物理CPUに届いた外部割込みのベクタ値は、ハイパバイザ10のエミュレーションを介さずに、物理Local APIC7に届いたベクタ値がそのまま仮想マシンに伝達される。このため、物理Local APIC7に届く外部割込みのベクタ値と、ゲスト13のゲストOSがその割込みに割り当てたベクタ値とは、必ず一致していなければならない。

[0075] そこで、ゲストOSが、直接実行モード物理CPUが割り当てられている仮想CPUに伝達されるように設定されている外部割込みのベクタ値を設定した際は、ハイパバイザはそれをトラップし、そのベクタ値と同じベクタ値を、その外部割込みの割り込みパラメータに設定する

[0076] 次に、エミュレーションモード物理CPUについて説明する。エミュレーションモード物理CPUとは以下の二つを指す。

- ・割り当てられ得る全ての仮想CPUの外部割込み実行方式がエミュレーション方式である物理CPU。
- ・割り当てられ得る仮想CPUがない物理CPU。

エミュレーションモードが適用可能な物理CPUに特別な条件はない。

[0077] 図4は、仮想計算機システムの初期状態時から、仮想マシンを立ち上げて、外部割込み実行モードを各物理CPUに適用する例を表している。図中では、エミュレーションモードをEモードと、直接実行モードをDモードと、それぞれ略記する場合がある。

[0078] 図4(a)に示すように、CPU共有仮想マシンを立ち上げた際は、その

仮想マシンが動作する物理CPUは共有割り当て物理CPUとなる。図4（b）に示すように、共有物理CPUは直接実行モードを適用することはできないため、生成する仮想CPUの外部割込み処理方式はエミュレーション方式とし、その仮想CPUに共有される物理CPUにはエミュレーションモード物理CPUを適用する。

[0079] 図4（c）に示すように、CPU占有仮想マシンを立ち上げた際は、その仮想マシンが動作する物理CPUは占有割り当て物理CPUとなる。ハイパバイザの介入が必要な外部割込みを受け付けている物理CPUは直接実行モードを適用することはできないため、その物理CPUに割り当てるために生成する仮想CPUの外部割込み処理方式はエミュレーション方式とし、その物理CPUにはエミュレーションモード物理CPUを適用する。

[0080] ハイパバイザの介入が必要な外部割込みを受け付けていない物理CPUは、直接実行モードを適用することができるため、その物理CPUに割り当てるために生成する仮想CPUの外部割込み処理方式は直接実行方式とし、その物理CPUには直接実行モードを適用する。仮想マシンの立ち上げ後に、仮想マシンが動作したまま外部割込み実行モードを動的に切り替える方法については、後述の実施例で述べる。

[0081] 図5は、ゲストモードで実行中のエミュレーションモード物理CPU2に届いた外部割込みが、ハイパバイザ10を介してゲスト13へ伝達される様子を示す。図中では、エミュレーションモードをEモードと、直接実行モードをDモードと、それぞれ略記する場合がある。

[0082] I/Oデバイス6からエミュレーションモード物理CPUの物理Local API C7へ外部割込みが届くと、その外部割込みはいったんハイパバイザ10に受け取られる。ハイパバイザ10は、その外部割込みをエミュレーション処理して仮想割込みに変換し、その仮想割込みを仮想Local API C14に送る。このようにして、エミュレーションモード物理CPUへの外部割込みは、ハイパバイザ10を介して、ゲスト13に伝達される。

[0083] 図6は、図5で説明した処理の詳細を示すフローチャートである。図中で

は、割込みをINTと略記する場合がある。

- [0084] 物理Local APIC 7 に外部割込みが届くと (S 3 1)、エミュレーションモード物理CPU 2 はホストモードに遷移し (S 3 2)、ハイパバイザ 1 0 に処理を引き渡す。ハイパバイザ 1 0 が仮想Local APIC 1 4 に仮想割込みを伝達すると (S 3 3)、その仮想割込みは実行されずに一旦仮想Local APIC 1 4 に貯蔵される (S 3 4)。
- [0085] ハイパバイザ 1 0 は、物理APICレジスタD 1 5 のEOIレジスタに書き込んで、物理Local APIC 7 へEOIを発行する (S 3 5)。EOI発行により、物理Local APIC 7 は外部割込みの終了を検知する (S 3 6)。
- [0086] 物理CPU 2 はゲストモードに遷移し、ハイパバイザ 1 0 からゲスト 1 3 に処理が移る (S 3 7)。物理CPU 2 がゲストモードに遷移すると、貯蔵されていた仮想割込みが検知され (S 3 8)、ゲスト 1 3 の割込みハンドラの実行が開始される (S 3 9)。
- [0087] 割込みハンドラは、EOIを実行するために、物理APICレジスタD 1 5 のEOIレジスタにアクセスする。ここで、物理EOIレジスタへアクセスするとホストモードへ遷移するように、仮想CPU 1 2 は設定されている。従って、物理CPU 2 はゲストモードからホストモードへ遷移し、ハイパバイザ 1 0 に処理が移る (S 4 0)。
- [0088] ハイパバイザ 1 0 は、物理APICレジスタD 1 5 のEOIレジスタへの書き込みに代えて、仮想APICレジスタD 1 2 を操作する。このエミュレーション処理により、ハイパバイザ 1 0 は、物理Local APIC 7 へのEOI発行を、仮想Local APIC 1 4 へのEOI発行に変換する (S 4 1)。
- [0089] ハイパバイザ 1 0 によるEOI発行エミュレーションによって、仮想Local APIC 1 4 は仮想割込みの終了を検知する (S 4 2)。その後、物理CPU 2 はゲストモードに遷移し (S 4 3)、ハイパバイザ 1 0 からゲスト 1 3 に処理が移る。ゲスト 1 3 の処理が再開されると、ゲスト 1 3 は割込みハンドラを終了する (S 4 4)。ゲスト 1 3 は、外部割込みにより中断されていた処理を再開する (S 4 5)。

[0090] 上記で説明したように、エミュレーションモード物理CPU 2上で動作する仮想CPU 1 2へ向けて外部割込みをエミュレーションする場合は、仮想割込みを用いる。これに対し、直接実行モード物理CPU 2に占有的に割り当てられている仮想CPU 1 2に向けて外部割込みをエミュレーションする場合は、仮想割込みではなく、直接実行モード物理CPU 2の物理Local APIC 7に向けたIPIを用いる。これは、直接実行モード物理CPU 2上で動作する仮想CPU 1 2への外部割込みの伝達は、全て直接実行モード物理CPU 2の物理Local APIC 7を介した伝達とするためである。上記としない理由、直接実行モード物理CPU 2の説明にて後述する。

[0091] 図7は、直接実行モード物理CPU 2に外部割込みが届いたとき、ホストモードに遷移することなく、ゲストIDT 130を通して割込みハンドラを直接実行する様子を示した説明図である。

[0092] 図8は、図7におけるゲスト13、ハイパバイザ10、物理Local APIC 7、仮想Local APIC 14での各処理を示すフローチャートである。

[0093] 物理Local APIC 7に外部割込みが届くと（S51）、物理CPU 2はゲストモードのまま、ゲストIDT 130を通して、割込みハンドラの実行を開始する（S52）。割込みハンドラに渡されるベクタ値は、物理Local APIC 7に届いた外部割込みのベクタ値をそのまま使用する。物理CPU 2から物理Local APIC 7へのEOI発行も、ゲストモードのまま行われる（S53）。物理Local APIC 7は、EOIを受領すると、外部割込みが終了したことを検知する（S54）。ゲスト13はEOIを発行した後、割込みハンドラを終了し（S55）、割込み以前の処理を再開する（S56）。

[0094] 物理Local APIC 7のEOIの発行を、ゲストモードのまま行うことができる理由は次の通りである。直接実行モード物理CPU 2上で動作する仮想CPU 1 2を宛先とする割込みは、仮想割込みとして届くことはなく、全て物理Local APIC 7に向けた外部割込みとして届く。さらに、直接実行モード物理CPU 2の物理Local APIC 7に届く割込みは、全てその物理CPU 2上で動作するただ一つの仮想CPU 1 2に向けた割込みであることが保証されて

いる。すなわち、直接実行モード物理CPU 2の物理Local APIC 7に届く割込みは、全て直接モード物理CPU 2上で動作するただ一つのゲストOSに認識される割込みとなる。

[0095] また、直接実行モード物理CPU 2上で動作する仮想CPU 1 2は、仮想割込みを扱うことはない。従って、直接実行モード物理CPUでは、外部割込みの優先度を、仮想Local APIC 1 4と物理Local APIC 7とで調整する必要がない。

[0096] 以上の理由から、直接モード物理CPUでは、物理Local APIC 7へ向けて発行する全てのEOIの制御をゲスト1 3（ゲストOS）に委ねることができる。

[0097] 次に、直接実行モード物理CPU 2がホストモードで動作しているときに外部割込みが届いた場合について説明する。直接実行モード物理CPU 2においては、ホストモードで実行中は常にRFLAGS, IFおよびEFLAGS, IFがいずれも0となるようにハイパバイザ1 0が制御する。RFLAGSおよびEFLAGSは、実行されているプログラムの状態を示すレジスタである。IFとは、これらレジスタのInterrupt Enable Flagビットを指す。

[0098] RFLAGS, IFおよびEFLAGS, IFが1のときは、プログラムの状態が割込み可能であることを示し、0のときは、割込みが不可であることを示す。割込みが不可のときは、外部割込みが届いてもその処理は実行されず、Local APICに貯蔵される。そして、EFLAGS, IFおよびRFLAGS, IFが1となったとき、貯蔵されていた割込みの処理が実行される。

[0099] 直接実行モード物理CPUがホストモードで実行中の場合は、常にRFLAGS, IFおよびEFLAGS, IFを0に設定しておくことによって、ホストモードのときに届いた外部割込みを全て物理Local APIC 7に保留させることができる。そして、直接実行モード物理CPUがホストモードからゲストモードに遷移し、RFLAGS, IFおよびEFLAGS, IFが1になったときに、保留状態に置かれた割込みは、図8のフローチャートに従って処理される。

[0100] 以上のように設定する理由は、直接実行モード物理CPU 2に届いた外部

割込みは、ゲスト 13 に直接処理させる以外に、その外部割込みをゲスト 13 に認識させる方法がないためである。必ず全ての外部割込みをゲスト 13 に直接実行させるため、上記のように各レジスタを制御する。

[0101] このように構成される本実施例によれば、外部割込みを、仮想 CPU 上のゲスト 13 で処理したり、ハイパバイザ 10 でエミュレーション処理したりできる。

[0102] 本実施例では、各物理 CPU 2 を、直接実行モード物理 CPU とエミュレーションモード物理 CPU とに分ける。さらに、本実施例の直接実行モード物理 CPU は、受け付けた全ての外部割込みをゲスト 13 に直接処理させるため、全ての割込みをエミュレーション処理する場合に比べて、割込み処理に要する時間を短縮でき、計算機の性能を向上することができる。

[0103] 本実施例では、従来技術のシャドウ IDT を用いないため、ハイパバイザ 10 がシャドウ IDT をメンテナンスすることによる、コスト発生や他 CPU との相互干渉が生じるおそれがない。従って、本実施例の計算機は、従来技術に比べて高い信頼性を維持しつつ性能を向上することができる。

実施例 2

[0104] 図 9 ～ 図 16 を用いて第 2 実施例を説明する。本実施例を含む以下の各実施例は、第 1 実施例の変形例に該当するため、第 1 実施例との差異を中心に説明する。本実施例では、物理 CPU 2 の外部割込み実行モードを、ハイパバイザ 10 が任意のタイミングで動的に切り替える。

[0105] < 1. 仮想マシン動作中に動的に外部割込み実行モードを切り替える処理の概要 >

[0106] 本実施例では、外部割込み実行モードを仮想マシン動作中に動的に切り替える。外部割込み実行モードを動的に切り替えるとは、仮想マシン 11 の動作中に、物理 CPU 2 の外部割込み実行モードを、直接実行モードとエミュレーションモードの間で相互に切り替えることである。

[0107] 任意の物理 CPU 2 の外部割込み実行モードを、エミュレーションモードから直接実行モードへ動的に切り替えるための、物理 CPU 2 の状態に関する

る条件は、その物理CPU 2が占有割り当て物理CPU 2であることである。一方、直接実行モードからエミュレーションモードへの切り替えに関しては、物理CPU 2の状態に関する条件はない。なお、切替対象物理CPU 2を宛先とする外部割込みの調整は、外部割込み実行モード切替処理の中で行う。

[0108] 外部割込み実行モードの動的切替が必要となる場合は、以下のような場合である。第1の場合は、仮想マシン11やI/Oデバイス6の構成が動的に変更される場合である。仮想マシン11やI/Oデバイス6の構成に変更があると、外部割込みに対するハイパバイザ10の介入の必要性の有無も変更になることがあるため、それに伴って外部割込み実行モードも変更する必要がある。

[0109] 図9は、仮想マシン11の動的な物理CPU割り当て設定の変更に伴い、物理CPU 2の外部割込み実行モードが切り替わる例を示す。図9の上側では、3つの物理CPU 2(1)~2(3)の全てが共有割り当て物理CPU 2であり、エミュレーションモード物理CPU 2として動作している。これに対し、図9の下側では、3つの物理CPU 2のうち2つの物理CPU 2(1), 2(2)は、占有割り当て物理CPU 2であるため、直接実行モード物理CPU 2として動作している。図9の上側に示す仮想計算機環境と下側に示す仮想計算機環境とは、外部割込み実行モードを動的に切り替えることにより、相互に遷移可能である。

[0110] 外部割込み実行モードの動的切替が必要となる第2の場合を説明する。第2の場合は、ハイパバイザ10の介入が必要となる割込みを負荷分散する場合である。図10は、複数のエミュレーションモード物理CPU 2(3), 2(4)のうち、一部のエミュレーションモード物理CPU 2(3)が故障等で使用不可になった場合を示す。

[0111] 図10の上側に示すように、エミュレーションモード物理CPU 2(3), 2(4)は、仮想デバイス101に変換すべきI/Oデバイス6からの割込みを受け付けている。一方のエミュレーションモード物理CPU 2(3)

が、故障などの要因により使用不可となった場合に、その物理CPU 2 (3) で担当していた外部割込みを、他方のエミュレーションモード物理CPU 2 (4) に割り当てるとすると、他方のエミュレーションモード物理CPU 2 (4) の負荷が増大する。

[0112] そこで、図10の下側に示すように、直接実行モード物理CPU 2 (1) , 2 (2) のうち一方の物理CPU 2 (2) の外部割込み実行モードをエミュレーションモードに切り替えて、エミュレーションモード物理CPU 2 (2) とする。そして、故障等で使用できなくなったエミュレーションモード物理CPU 2 (3) の担当していた外部割込みの宛先を、新たなエミュレーションモード物理CPU 2 (2) に変更する。

[0113] 図10では、一つのエミュレーション物理CPU 2 (4) に、ハイパバイザ10の介入が必要となる割込みを全て集中させるのではなく、直接実行モード物理CPU 2 (2) をエミュレーションモード物理CPUに切り替えて、そのエミュレーションの必要な割込みを受け付ける。これにより、割込み処理の負荷分散を図っている。

[0114] 図11に示すように、仮想デバイス101に変換すべきI/Oデバイス6 (2) の割込みの宛先を、複数の物理CPU 2 (1) , 2 (2) 間で定期的に変更して負荷分散を図る場合も、外部割込み実行モードの動的切替を用いる。

[0115] 図11の上側では、一方の物理CPU 2 (1) が直接実行モード物理CPU、他方の物理CPU 2 (2) がエミュレーションモード物理CPUとなっている。I/Oデバイス6 (2) は仮想デバイス101に変換されるべきデバイスであり、そのI/Oデバイス6 (2) からの外部割込みは、エミュレーションモード物理CPUで処理される。

[0116] 図11の下側では、一方の物理CPU 2 (1) を直接実行モードからエミュレーションモードへ切り替えると共に、他方の物理CPU 2 (2) をエミュレーションモードから直接実行モードへ切り替える。ハイパバイザ10は、外部割込み実行モードの切替に伴い、I/Oデバイス6 (2) からの外部

割込みの宛先を変更する。

[0117] < 2. 外部割込み実行モード動的切替契機の通知処理 >

[0118] 外部割込み実行モードを動的に切り替えるためには、任意の物理CPU2を、任意のタイミングでホストモードに遷移させて、ハイパバイザ10に外部割込み実行モード切替処理を行わせる必要がある。本実施例では、以上の処理を外部割込み実行モード切替契機の通知処理と呼ぶ。

[0119] 一般的に、仮想計算機システムでは、IPIを任意の物理CPU2に発行してホストモードに遷移させることで、ハイパバイザ10に特定の処理の実行を通知する。しかし、直接実行モード物理CPU2では、IPIはホストモードへの遷移を引き起こさず、直接ゲスト13に伝達されてしまうため、IPI以外の方法で外部割込み実行モード切替契機を通知しなければならない。

[0120] そこで本実施例では、外部割込み実行モードの切替契機を通知するに際して、ゲストモードで一定時間Tpが経過したら定期的にホストモードに遷移するという、ハードウェア仮想化支援機能を用いる。この機能を以下、定期的なホストモード遷移機能と呼ぶ。

[0121] 例えばIntelのCPUでは、そのような機能がpreemption timer機能として提供されている。preemption timer機能とは、予めハイパバイザ10が指定したカウント数を、ゲストモードの間、一定間隔でデクリメントしていき、カウント値が0になった時点でホストモードに遷移するという機能である。この機能を用いると、物理CPU2を定期的にゲストモードからホストモードに遷移させることができる。

[0122] 任意の物理CPUを任意のタイミングでホストモードに遷移させる方法として、Non Maskable Interrupt (NMI) をその物理CPUに対して発行するなどの方法もある。しかし、本来のそれらの処理と、外部割込み実行モード切替処理とが競合することを防ぐために、本実施例では、定期的なホストモード遷移機能を用いる。

[0123] 定期的なホストモード遷移機能を用いることで、外部割込み実行モード切

替契機の通知は、ハイパバイザ10ではなく、ゲスト13の処理に割り込んで行われることを保証することができる。これにより、本実施例では、切替契機の通知が外部割込み実行モードに依存するハイパバイザ10の処理に割り込んで、外部割込み実行モードを切り替えてしまい、ハイパバイザ10の処理に不整合が起きてしまうことを防ぐことができる。なお、定期的にホストモードに遷移することによる性能低下を極力小さくするために、遷移の間隔Tpはできるだけ大きく設定するのが好ましい。

[0124] 以下、例えばpreemption timer機能のような機能を用いる場合を例にあげて、外部割込み実行モードを切り替える方法を説明する。

[0125] 最初に、直接実行モードからエミュレーションモードへの切替処理について説明する。前提として、エミュレーションモードへの切替処理は、ゲスト13での外部割込み処理が中断中でない場合のみ行う。

[0126] ゲスト13の外部割込み処理が中断中である場合とは、ゲスト13の外部割込み処理中にpreemption timerのカウントが0になり、ホストモードに遷移した場合である。理由は以下の通りである。ゲスト13は、直接実行モードでは物理Local APIC 7を介して外部割込みを制御するが、エミュレーションモードでは仮想Local APIC 14を介して外部割込みを制御する。従って、もしも割込みが中断されている状態でエミュレーションモードに切り替えると、中断されていた割込み処理の再開後に、物理Local APIC 7ではなく仮想Local APIC 14にEOIが発行されてしまう。これでは、物理Local APIC 7が検知した外部割込みが正常に終了しなくなってしまう。

[0127] 図12は、外部割込み実行モードの切替契機を通知する処理を示すフローチャートである。図12は、直接実行モードからエミュレーションモードへの切替契機の通知と、エミュレーションモードから直接実行モードへの切替契機の通知との両方に用いる。

[0128] 切替契機を通知する物理CPU 2は、外部割込み実行モードの切替契機が到来すると（S101：YES）、切替対象の物理CPU 2の外部割込み実行モードを切り替えさせるための切替契機通知フラグをセットする（S10

2)。以下、切替対象の物理CPU 2を切替対象CPU 2と呼ぶことがある。

[0129] <3. 直接実行モードからエミュレーションモードへの切替>

[0130] 図13は、切替対象CPU 2の実行する切替判定処理を示す。図13および図14は、切替対象物理CPU 2の外部割込み実行モードを直接実行モードからエミュレーションモードへ切り替える処理を示す。

[0131] 最初に、切替対象CPU 2では、カウンタTpの値を第1切替時間T1に設定されているものとする。カウンタTpは、例えばIntelのCPUでは、VMCSの“VMX-preemption timer value”フィールドに設定した値となる。第1切替時間T1は、比較的大きい値に設定される。ゲストモードからホストモードへの遷移の頻度を抑えて、処理性能の低下を防止するためである。ここで、Preemption timerのように構成されるカウンタTpは、ゲストモードでは設定することができないため、必ず最初はTp=T1となっている。

[0132] カウンタTpが0になると(S111: YES)、切替対象の物理CPU 2は、ゲストモードからホストモードへ遷移する(S112)。

[0133] ハイパバイザ10は、切替契機通知フラグがセットされているかどうかを調べる(S113)。ここでは、切替契機通知フラグの値が1の場合、フラグはセットされたものとして説明する。

[0134] 切替契機でない場合は切替契機通知フラグがセットされていないため(S113: NO)、カウンタをT1に再設定し(S114)、再度ゲストモードに遷移して(S115)、ゲスト13の処理に戻る。この流れの場合、カウンタTpの値は第1切替時間T1に戻される。一方、切替契機通知フラグがセットされている場合(S113: YES)、ハイパバイザ10は、切替対象CPU 2が外部割込み実行モードの切替契機であることを検知する。

[0135] ハイパバイザ10は、切替対象CPU 2の物理Local APIC 7のISR (In Service Register) が0であるかチェックする(S116)。切替対象CPU 2の仮想CPU 12で動作しているゲスト13の割込み処理が中断中でないかどうかを判断するためである。ここで、ISRとは、物理APICレジ

スタD 1 5 の1つで、現在処理中の割込みを示すレジスタである。

[0136] I S Rが0の場合（S 1 1 6 : Y E S）、割込み処理は中断中でないため、カウンタT pの値を第1切替時間T 1に戻し（S 1 1 7）、外部割込み実行モードを切り替える処理を行う（S 1 1 8）。ステップS 1 1 8の外部割込み実行モード切替処理は、図1 4で後述する。

[0137] 一方、I S Rの値が0でない場合（S 1 1 6 : N O）、何らかの割込み処理の中断中である。そこで、ハイパバイザ1 0は、カウンタT pを第2切替時間T 2に変更し（S 1 1 9）、ゲストモードへ遷移する（S 1 1 5）。第2切替時間T 2は、第1切替時間T 1よりも短く設定される（ $T 2 < T 1$ ）。ゲストモードへ遷移すると、中断していた割り込み処理が再開される。

[0138] カウンタT pの設定値を第1切替時間T 1から第2切替時間T 2に変更して、切替周期を短縮する理由を説明する。ハイパバイザ1 0は、外部割込み実行モードの切替契機であることを検知しているため、中断中の割込み処理が終了したら、なるべく早くホストモードに遷移して、外部割込み実行モード切替処理を実行するためである。

[0139] <4. 外部割込み実行モード切替処理>

[0140] 図1 4は、直接実行モードからエミュレーションモードへ外部割込み実行モードを切り替える処理のフローチャートである。本処理は、図1 3のステップS 1 1 8の詳細な一例を示す。

[0141] 外部割込み実行モードを直接実行モードからエミュレーションモードへ切り替える場合、ゲスト1 3（詳しくはゲストO S）の割込みハンドラが直接処理していた割込みは、ハイパバイザ1 0が介入してエミュレーションすることになる。そこで、ハイパバイザ1 0の割込みハンドラを再設定する（S 1 2 1）。

[0142] 次に、ハイパバイザ1 0は、切替対象C P U 2が占有的に割り当てられている仮想C P U 1 2の仮想C P U管理データD 1 1を操作し、仮想C P U 1 2の割込み処理方式をエミュレーション方式に変更する（S 1 2 2）。

[0143] 最後にハイパバイザ1 0は、切替通知フラグD 1 4をクリアし（S 1 2 3

）、ゲストモードに遷移する（S 1 2 4）。

[0144] <5. エミュレーションモードから直接実行モードへの切替>

[0145] 次に、エミュレーションモードから直接実行モードへの切替契機を通知する処理について説明する。

[0146] まず前提として、外部割込み実行モード切替処理のためにホストモードに遷移してから、外部割込み実行モードの切替処理を終了して再度ゲストモードに遷移するまでは、常にRFLAGS, IFおよびEFLAGS, IFは0にする。

[0147] 任意の物理CPU 2を任意のタイミングでホストモードに遷移させるための方法としては、直接実行モードからエミュレーションモードへの切り替えの際と同様に、preemption timer機能を用いることができる。

[0148] エミュレーションモード物理CPU 2は、外部割込みが届くとホストモードに遷移するため、IPIを用いて切替契機の通知を行うこともできる。しかし、本実施例では、あえて、preemption timerのようなカウンタTpを用いる。その理由は以下の通りである。

[0149] 外部割込み実行モード切替処理が始まってから、ゲストモードに戻るまでの間は、必ずRFLAGS, IFおよびEFLAGS, IFが0でなければならない。もしもハイパバイザ10での処理中に外部割込み実行モード切替処理が割り込むと、切替処理終了後にハイパバイザ10の処理が再開されてしまう。この場合、RFLAGS, IFおよびEFLAGS, IFが0であるということを保証するのが難しい。そこで、本実施例では、外部割込み実行モード切替処理は必ずゲストモードからホストモードに遷移したときに行われることを保証するために、カウンタTpを用いて切替契機を伝える。なお、直接実行モードからエミュレーションモードへの切替処理も、ゲスト13の外部割込み処理が中断中でないときのみ行う。

[0150] 図15は、エミュレーションモードから直接実行モードへ外部割込み実行モードを切り替えるための契機を通知する処理を示すフローチャートである。

[0151] 図15に示すエミュレーションモードから直接実行モードへの切替契機通

知処理と、図 13 で述べた直接実行モードからエミュレーションモードへの切替契機通知処理との違いは、ゲスト 13 の割込み処理中断中であるか否かのチェックを行うレジスタにある。図 13 の処理では物理 A P I C レジスタ D 1 5 の I S R をチェックしたが、図 1 5 の処理では仮想 A P I C レジスタ D 1 2 の I S R をチェックする。

[0152] つまり、図 1 5 に示すステップ S 1 3 1 ~ S 1 3 5, S 1 3 7 ~ S 1 3 9 は、図 1 3 に示すステップ S 1 1 1 ~ S 1 1 5, S 1 1 7 ~ S 1 1 9 と同一である。図 1 5 のステップ S 1 3 6 と図 1 3 のステップ S 1 1 6 とは、チェック対象のレジスタのみ異なる。従って、図 1 5 の処理についてのこれ以上の説明は割愛する。ステップ S 1 3 8 の外部割込み実行モード切替処理については後述する。

[0153] < 6. 外部割込み実行モード切替処理 >

[0154] 図 1 6 は、エミュレーションモードから直接実行モードへ外部割込み実行モードを切り替える処理のフローチャートである。本処理は、図 1 5 のステップ S 1 3 8 の詳細な一例を示す。

[0155] ハイパバイザ 1 0 は、切替対象 C P U 2 を宛先とする外部割込みのうち、その物理 C P U 2 が直接実行可能な外部割込み以外の外部割込みを、他のエミュレーションモード物理 C P U 2 を宛先となるように、割込みパラメータ D 1 6 を変更する (S 1 4 1) 。すなわち、切替対象 C P U 2 が受け付けていた外部割込みのうち、他のエミュレーションモード物理 C P U 2 で処理可能な外部割込みの宛先を変更する。

[0156] ハイパバイザ 1 0 は、他の物理 C P U 2 を宛先とする外部割込み (I / O デバイス 6 の外部割込み) のうち、切替対象 C P U 2 で直接実行可能な全ての割込みを、切替対象 C P U 2 を宛先とするように、割込みパラメータ D 1 6 を変更する (S 1 4 2) 。

[0157] 切替対象 C P U 2 で直接実行可能な割込みが仮想デバイス 1 0 1 からの割り込みである場合、その仮想デバイス 1 0 1 についてのエミュレーション処理は、切替対象 C P U 2 への I P I となるように、ハイパバイザ 1 0 の設定

を変更する。

[0158] ステップS 1 4 1, S 1 4 2 の処理により、切替対象C P U 2 で直接実行可能な外部割込みのみが、切替対象C P U 2 を宛先とする。さらに、切替対象C P U 2 で直接実行可能な割込みは全て、切替対象C P U 2 を宛先とする。

[0159] 切替対象C P U 2 で直接実行可能な外部割込みとは、その切替対象の物理C P U 2 が占有的に割り当てられている仮想C P U 1 2 のI D T にゲストO S が割り当てた外部割込みである。

[0160] ハイパバイザ1 0 は、切替対象C P U 2 を宛先とする外部割込みのベクタ値を、ゲストO S がそれぞれの割込みに設定したベクタ値と同一になるように、割込みパラメータD 1 6 を変更する（S 1 4 3）。

[0161] ハイパバイザ1 0 は、ステップS 1 4 1 で他の物理C P U 2 へ宛先を変更した割込みのうち、切替対象C P U 2 の物理Local APIC 7 に保留されているものがあるか調べる（S 1 4 4）。

[0162] 切替対象C P U 2 の物理Local APIC 7 で保留中の割込みは、切替対象C P U 2 を直接実行モードへ切り替えた後に、無効な割込みとして捨てられてしまう。そこで、ハイパバイザ1 0 は、宛先変更前のC P U 2 で保留中の割込みが処理されずに捨てられてしまうのを防止すべく、宛先変更後の物理C P U 2 に対して、変更後のベクタ値を持ったI P I を発行する（S 1 4 5）。

[0163] ハイパバイザ1 0 は、仮想Local APIC 1 4 で保留されている仮想割込みがあるかを調べる（S 1 4 6）。切替対象C P U 2 を直接実行モードへ切り替えた後は、仮想Local APIC 1 4 を用いた割込みの制御は行われなため、仮想Local APIC 1 4 に保留された割込みはゲスト1 3（ゲストO S）に認識されなくなる。

[0164] そこで、ハイパバイザ1 0 は、仮想Local APIC 1 4 に保留された割込みがある場合、保留中の割込みと同一のベクタ値を持ったSelf-IPIを再発行する（S 1 4 7）。I P I の再発行により、その割込みの保留先は、仮想Local APIC 1 4 から物理Local APIC 7 へ変換される。

- [0165] 外部割込み実行モード切替処理中は、RFLAGS. IFおよびEFLAGS. IFは0としているため、Self-IPIで受けた割込みはその時点では処理されず、ゲストモードに遷移するまで物理Local APIC 7にペンディングされたままとなる。
- [0166] ハイパバイザ10は、切替対象CPU 2が占有的に割り当てられている仮想CPU 12の仮想CPU管理データD11を操作し、割込み処理方式を直接実行方式に変更する(S148)。最後にハイパバイザ10は、外部割込みモード切替通知フラグD14をクリアし(S149)、ゲストモードに遷移する(S150)。
- [0167] このように構成される本実施例も第1実施例と同様の作用効果を奏する。さらに本実施例では、任意の物理CPU 2の外部割込み実行モードを、直接実行モードとエミュレーションモードの間で動的に変更することができる。従って、本実施例では、物理CPU 2に発生する障害に対応することができ、物理CPU 2の負荷を分散させることもできる。これにより、第1実施例に比べて使い勝手が向上する。
- [0168] なお、本発明は、上記各実施例に限定されず、様々な変形例を含む。例えば、上記各実施例は本発明を分かりやすく説明したものであり、本発明は実施例で説明した全ての構成を備える必要はない。実施例で述べた構成の少なくとも一部を、他の構成に変更したり、削除したりすることができる。さらに、実施例に新構成を追加することもできる。
- [0169] 実施例で述べた機能や処理などの一部または全部を、ハードウェア回路として実現してもよいし、ソフトウェアとして実現してもよい。コンピュータプログラムや各種データは、計算機内の記憶装置に限らず、計算機外部の記憶装置へ格納してもよい。

符号の説明

- [0170] 1：物理計算機、2：物理CPU、4：メモリ、6：I/Oデバイス、7：物理Local APIC、10：ハイパバイザ、11：仮想マシン、12：仮想CPU、13：ゲスト、14：仮想Local APIC

請求の範囲

[請求項1]

計算機であって、

複数の物理プロセッサと、

前記複数の物理プロセッサのうちの少なくとも一つの物理プロセッサにより実現されるハイパバイザと、

前記ハイパバイザにより制御される少なくとも一つの仮想計算機であって、少なくとも一つの仮想プロセッサと、前記仮想プロセッサにより実行されるゲストプログラムとを含む仮想計算機と、
を備え、

前記ハイパバイザは、第1実行モードで外部割込みを処理する第1物理プロセッサと、第2実行モードで外部割込みを処理する第2物理プロセッサとを、前記複数の物理プロセッサから生成することができ、

前記第1物理プロセッサは、受け付けた全ての外部割込みを、前記第1物理プロセッサに割り当てられた前記仮想プロセッサ上の前記ゲストプログラムにより処理させ、

前記第2物理プロセッサは、受け付けた全ての外部割込みを、前記ハイパバイザでエミュレーション処理させ、

前記ハイパバイザは、前記外部割込みの宛先を前記第1物理プロセッサまたは前記第2物理プロセッサのいずれにするかを制御する、
計算機。

[請求項2]

前記仮想プロセッサは、外部割込みを直接実行する直接実行方式、または、外部割込みを前記ハイパバイザによるエミュレーション処理を介して実行するエミュレーション方式のいずれかに設定することができ、

前記ハイパバイザは、

前記複数の物理プロセッサから選択する所定の物理プロセッサに割り当てられた全ての仮想プロセッサを前記直接実行方式に設定する

ことで、前記所定の物理プロセッサを前記第 1 実行モードで動作する前記第 1 物理プロセッサとし、

前記複数の物理プロセッサから選択する他の所定の物理プロセッサに割り当てられた全ての仮想プロセッサを前記エミュレーション方式に設定することで、前記他の所定の物理プロセッサを前記第 2 実行モードで動作する前記第 2 物理プロセッサとする、
請求項 1 に記載の計算機。

[請求項3] 前記ハイパバイザは、

前記複数の物理プロセッサのうち、特定の仮想プロセッサに占有されて割り当てられている物理プロセッサを、前記所定の物理プロセッサとして選択する、
請求項 2 に記載の計算機。

[請求項4] 前記ハイパバイザは、

前記複数の物理プロセッサのうち、複数の仮想プロセッサに共有して割り当てられている物理プロセッサと、前記エミュレーション処理を必要とする外部割込みを 1 つ以上受け付ける物理プロセッサとを、前記他の所定の物理プロセッサとして選択する、
請求項 3 に記載の計算機。

[請求項5] 前記複数の物理プロセッサは、前記ゲストプログラムを実行するための動作モードであるゲストモードと、前記ハイパバイザを実行するための動作モードであるホストモードとのいずれか一方の動作モードを選択することができ、

前記第 1 物理プロセッサは、外部割込みが届いた場合は、前記ゲストモードのままで、前記第 1 物理プロセッサに割り当てられた前記仮想プロセッサ上の前記ゲストプログラムにより処理させ、

前記第 2 物理プロセッサは、外部割込みが届いた場合は、前記ゲストモードから前記ホストモードへ遷移して、前記外部割込みを前記ハイパバイザに引き渡し、

前記ハイパバイザは、前記引き渡された外部割込みを前記エミュレーション処理し、各仮想プロセッサの中から選択した仮想プロセッサに前記外部割込みを伝達し、

前記外部割込みを伝達された仮想プロセッサは、前記複数の物理プロセッサのうち前記ゲストモードで動作する物理プロセッサが割り当てられた場合に、前記外部割込みを前記ゲストプログラムにより処理する、

請求項4に記載の計算機。

[請求項6] 前記ハイパバイザは、前記第1物理プロセッサを前記第2物理プロセッサへ切り替えることができ、かつ、前記第2物理プロセッサを前記第1物理プロセッサへ切り替えることができる、
請求項2～5のいずれか一項に記載の計算機。

[請求項7] 前記ハイパバイザは、前記第1物理プロセッサおよび前記第2物理プロセッサの中から選択した切替対象の物理プロセッサを指定した切替通知フラグをセットし、

前記切替対象の物理プロセッサは、前記ゲストプログラムを実行するための動作モードであるゲストモードから前記ハイパバイザを実行するための動作モードであるホストモードへ周期的に切り替えるための第1切替時間が経過すると、自プロセッサの動作モードを前記ゲストモードからホストモードへ切り替えてハイパバイザを実行し、

前記切替対象の物理プロセッサにより実行されるハイパバイザは、

前記切替通知フラグがセットされているか判定し、前記切替通知フラグがセットされていないと判定した場合は、前記動作モードを前記ホストモードから前記ゲストモードへ移行し、

前記切替通知フラグがセットされていると判定した場合は、前記切替対象の物理プロセッサに設けられた仮想プロセッサ上のゲストプログラムにより、外部割込みが処理中であることを判定し、

前記ゲストプログラムが外部割込みの処理中であると判定した場

合は、前記第 1 切替時間よりも短い第 2 切替時間へ変更し、前記動作モードを前記ホストモードから前記ゲストモードへ移行し、前記第 2 切替時間が経過するのを待ち、

前記ゲストプログラムが外部割込みを処理していないと判定した場合は、前記切替対象の物理プロセッサについて外部割込みを処理するモードを前記第 1 実行モードと前記第 2 実行モードとの間で切り替えるための実行モード切替処理を実行する、

請求項 6 に記載の計算機。

[請求項 8] 前記ハイパバイザは、前記実行モード切替処理において、前記第 1 実行モードから前記第 2 実行モードへ切り替える場合、前記切替対象の物理プロセッサに割り当てられている仮想プロセッサを前記エミュレーション方式に設定する、

請求項 7 に記載の計算機。

[請求項 9] 前記ハイパバイザは、前記実行モード切替処理において、前記第 2 実行モードから前記第 1 実行モードへ切り替える場合、

前記切替対象の物理プロセッサで直接実行可能な割込み以外の割込みの宛先を、前記第 2 物理プロセッサのいずれかに変更し、

前記第 2 物理プロセッサを宛先とする割込みのうち前記切替対象の物理プロセッサで直接実行可能な割込みの宛先を、前記切替対象の物理プロセッサへ変更し、

処理中の割込みがある場合、その処理中の割込みを再発行し、

前記切替対象の物理プロセッサに割り当てられている仮想プロセッサを前記直接実行方式に設定する、

請求項 7 に記載の計算機。

[請求項 10] 計算機の制御方法であって、

前記計算機は、複数の物理プロセッサと、前記複数の物理プロセッサのうちの少なくとも一つの物理プロセッサにより実現されるハイパバイザと、前記ハイパバイザにより制御される少なくとも一つの仮想

計算機であって、少なくとも一つの仮想プロセッサと、前記仮想プロセッサにより実行されるゲストプログラムとを含む仮想計算機と、を備え、

第1実行モードで外部割込みを処理する第1物理プロセッサと、第2実行モードで外部割込みを処理する第2物理プロセッサとを、前記複数の物理プロセッサから生成することができ、

前記第1物理プロセッサが受け付けた全ての外部割込みを、前記第1物理プロセッサに割り当てられた前記仮想プロセッサ上の前記ゲストプログラムにより処理させ、

前記第2物理プロセッサが受け付けた全ての外部割込みを、前記ハイパバイザでエミュレーション処理させる、
計算機の制御方法。

[請求項11]

前記複数の物理プロセッサから選択する所定の物理プロセッサに割り当てられた全ての仮想プロセッサを、外部割込みを直接実行する直接実行方式に設定することで、前記所定の物理プロセッサを前記第1実行モードで動作する前記第1物理プロセッサとし、

前記複数の物理プロセッサから選択する他の所定の物理プロセッサに割り当てられた全ての仮想プロセッサを、外部割込みを前記ハイパバイザによるエミュレーション処理を介して実行するエミュレーション方式に設定することで、前記他の所定の物理プロセッサを前記第2実行モードで動作する前記第2物理プロセッサとする、
請求項10に記載の計算機の制御方法。

[請求項12]

前記第1物理プロセッサは、外部割込みが届いた場合は、前記ゲストプログラムを実行するための動作モードであるゲストモードのまま、前記第1物理プロセッサに割り当てられた前記仮想プロセッサ上の前記ゲストプログラムにより処理させ、

前記第2物理プロセッサは、外部割込みが届いた場合は、前記ゲストモードから前記ハイパバイザを実行するための動作モードであるホ

ストモードへ遷移して、前記外部割込みを前記ハイパバイザに引き渡し、

前記ハイパバイザは、前記引き渡された外部割込みを前記エミュレーション処理し、各仮想プロセッサの中から選択した仮想プロセッサに前記外部割込みを伝達し、

前記外部割込みを伝達された仮想プロセッサは、前記複数の物理プロセッサのうち前記ゲストモードで動作する物理プロセッサが割り当てられた場合に、前記外部割込みを前記ゲストプログラムにより処理する、

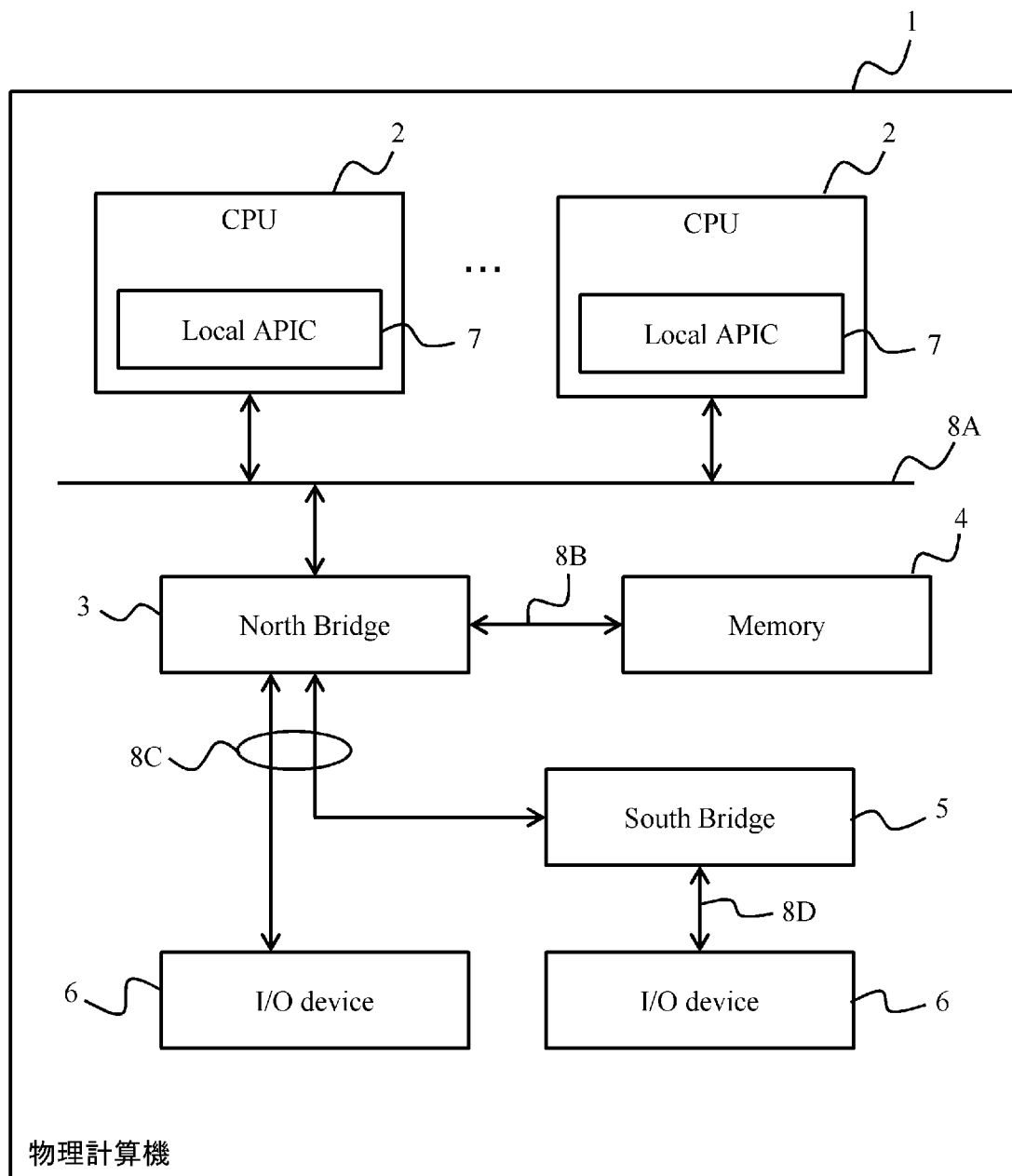
請求項 11 に記載の計算機の制御方法。

[請求項13]

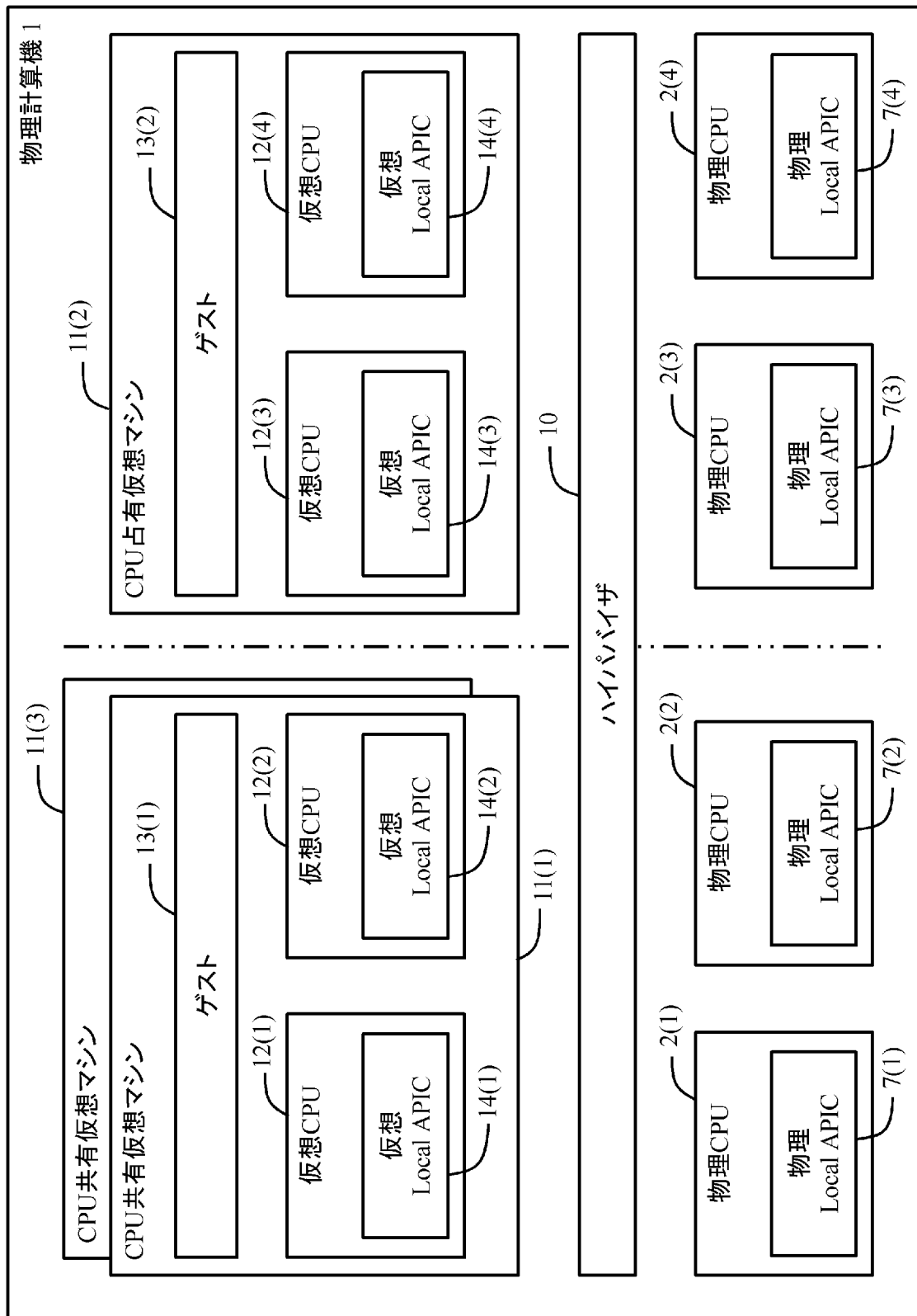
前記ハイパバイザは、前記第1物理プロセッサを前記第2物理プロセッサへ切り替えることができ、かつ、前記第2物理プロセッサを前記第1物理プロセッサへ切り替えることができる、

請求項 10～12 のいずれか一項に記載の計算機の制御方法。

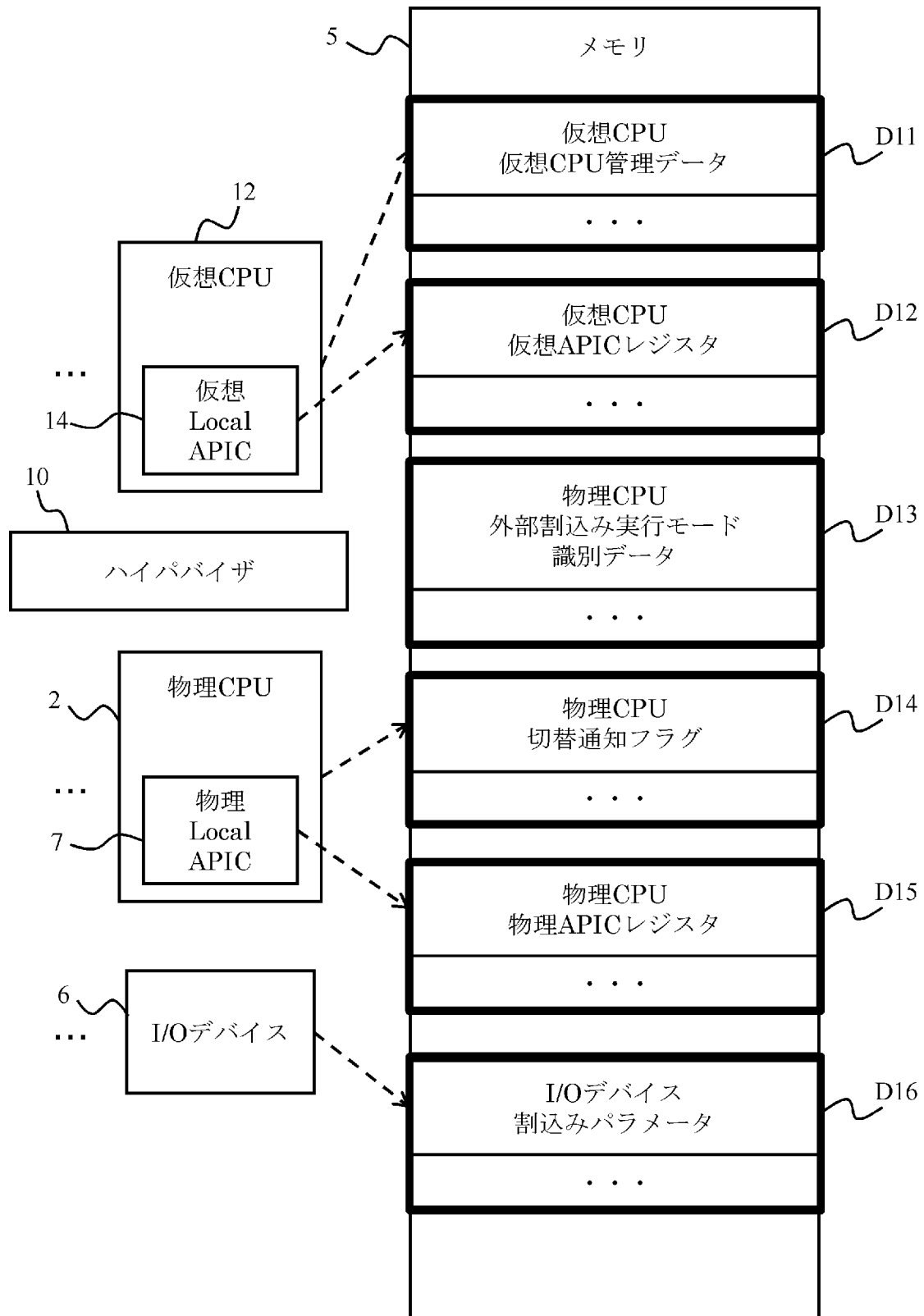
[図1]



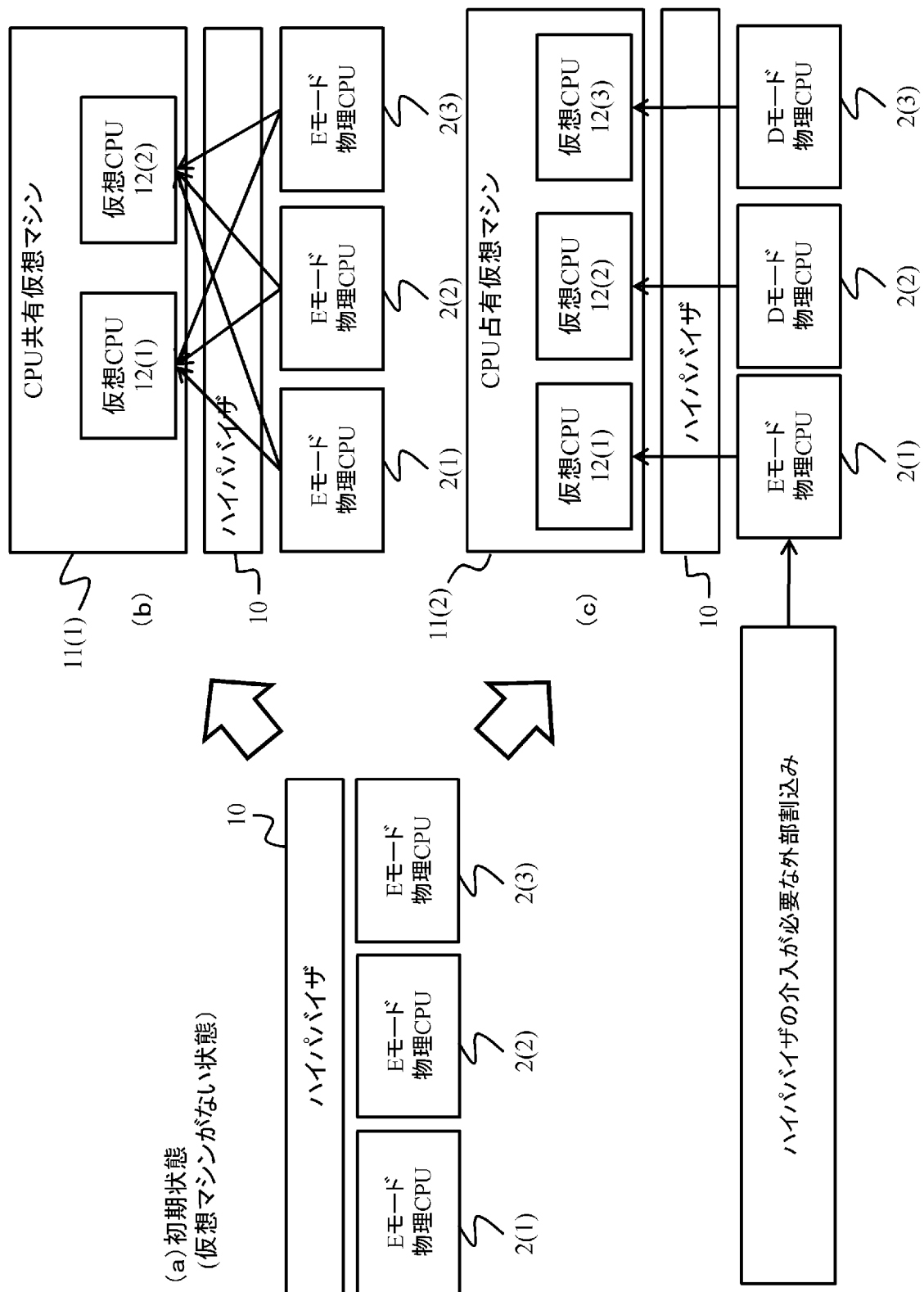
[図2]



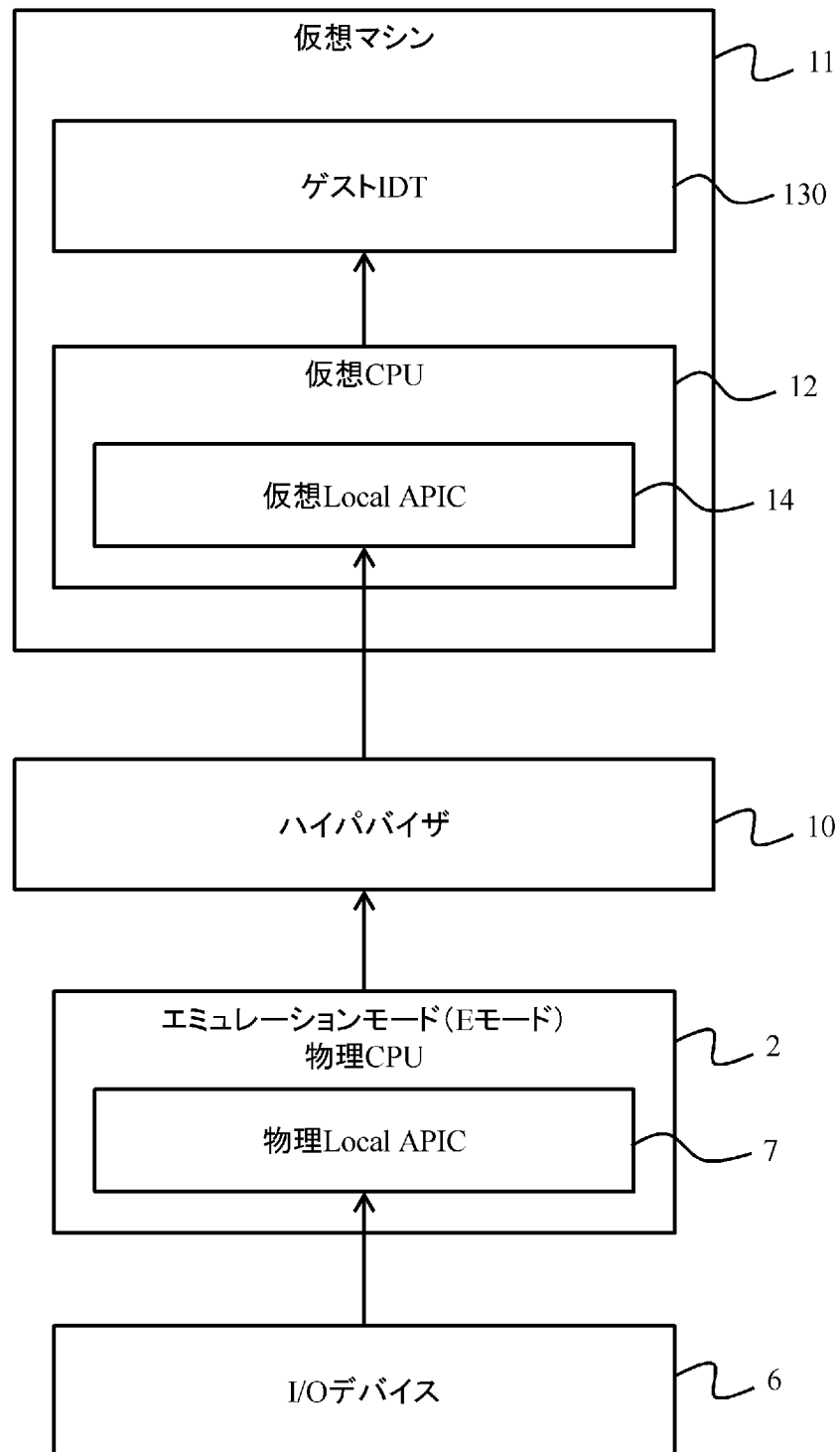
[図3]



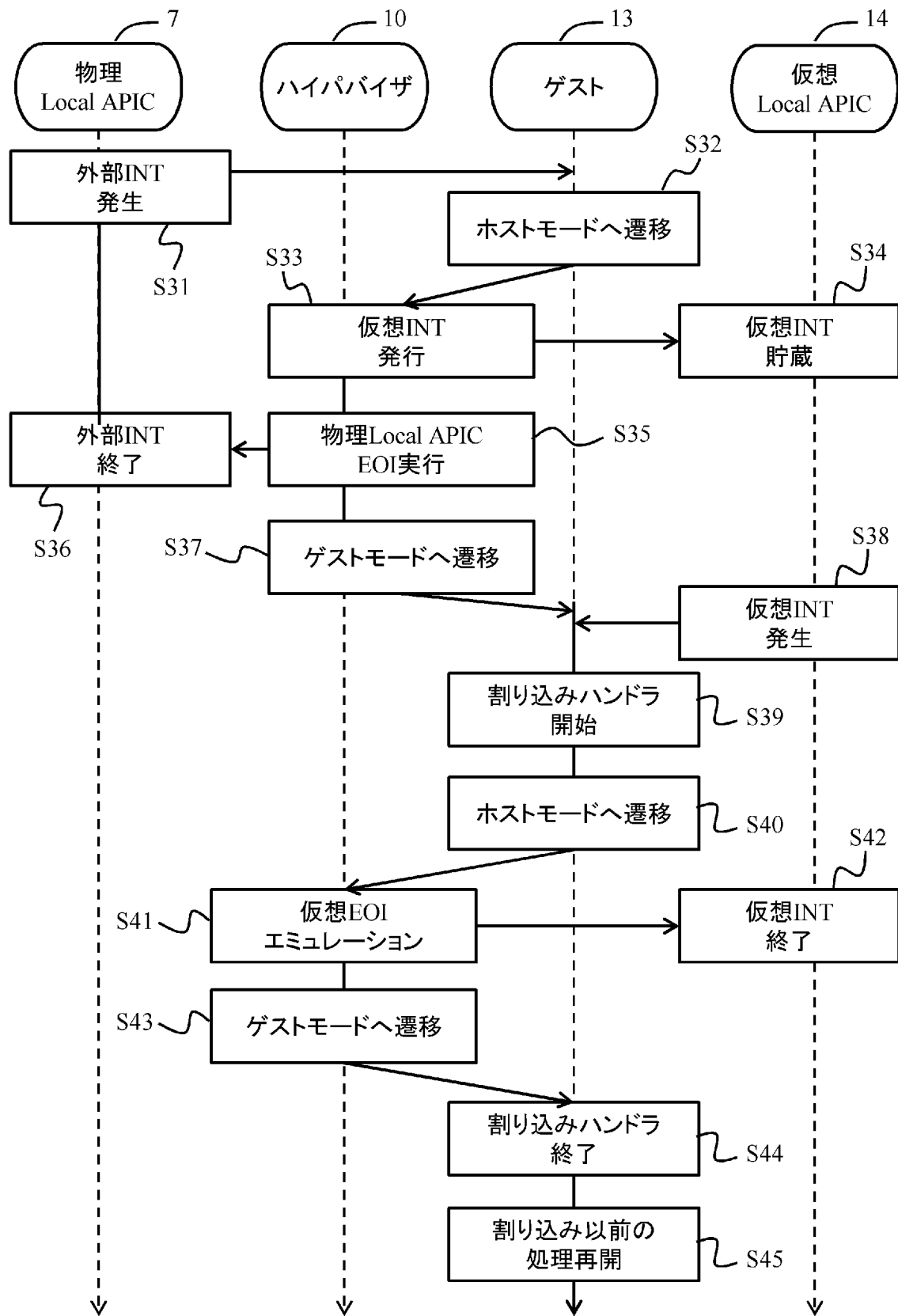
[図4]



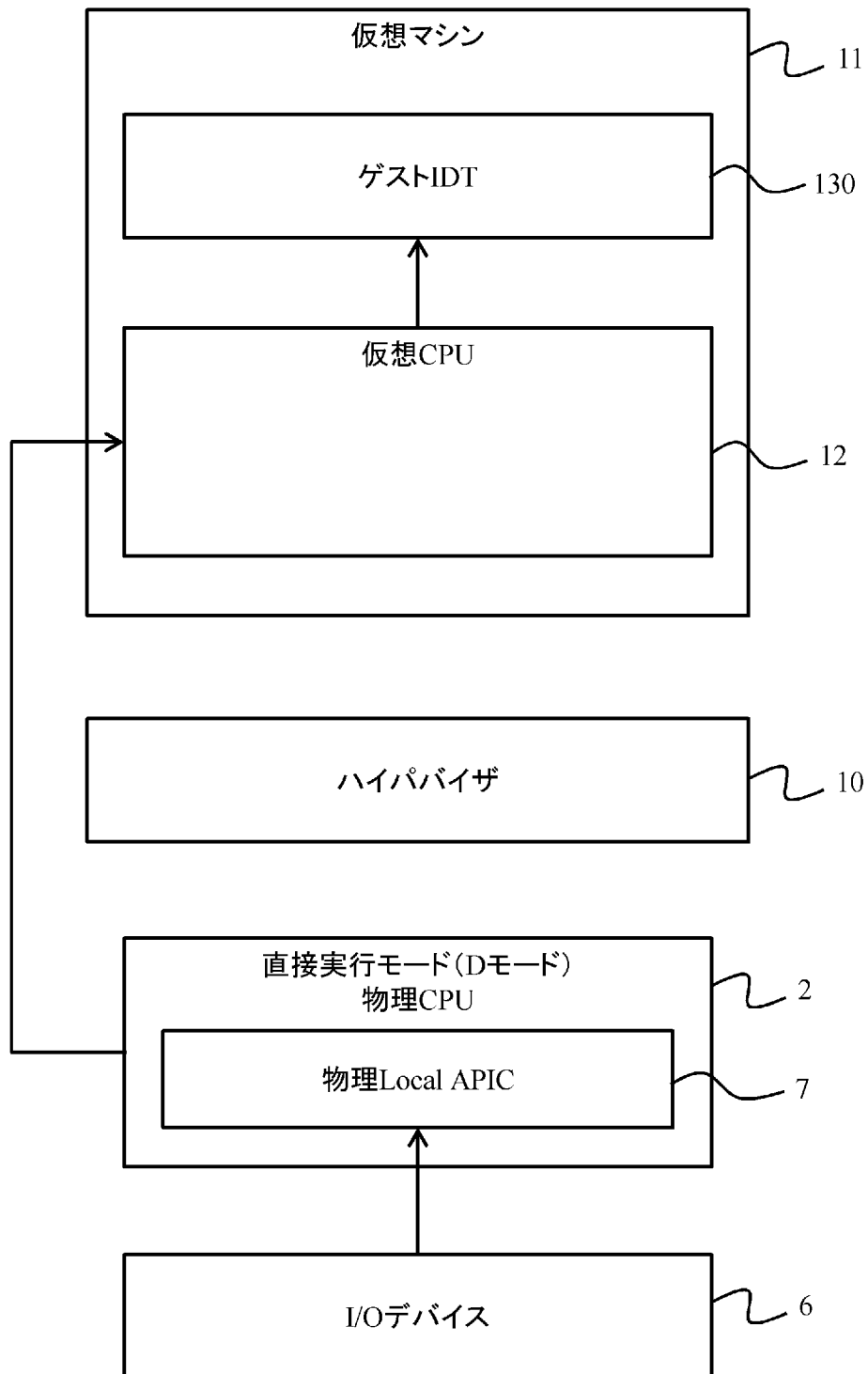
[図5]



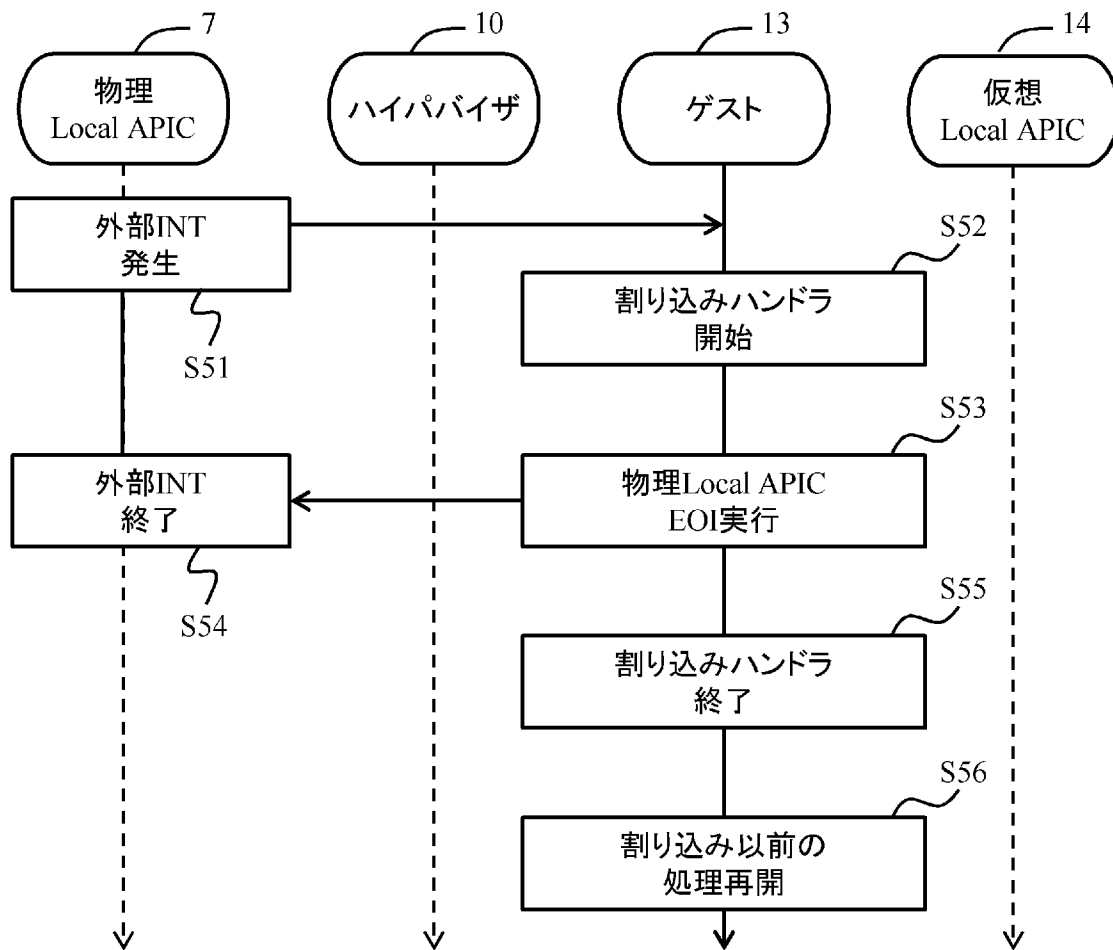
[図6]



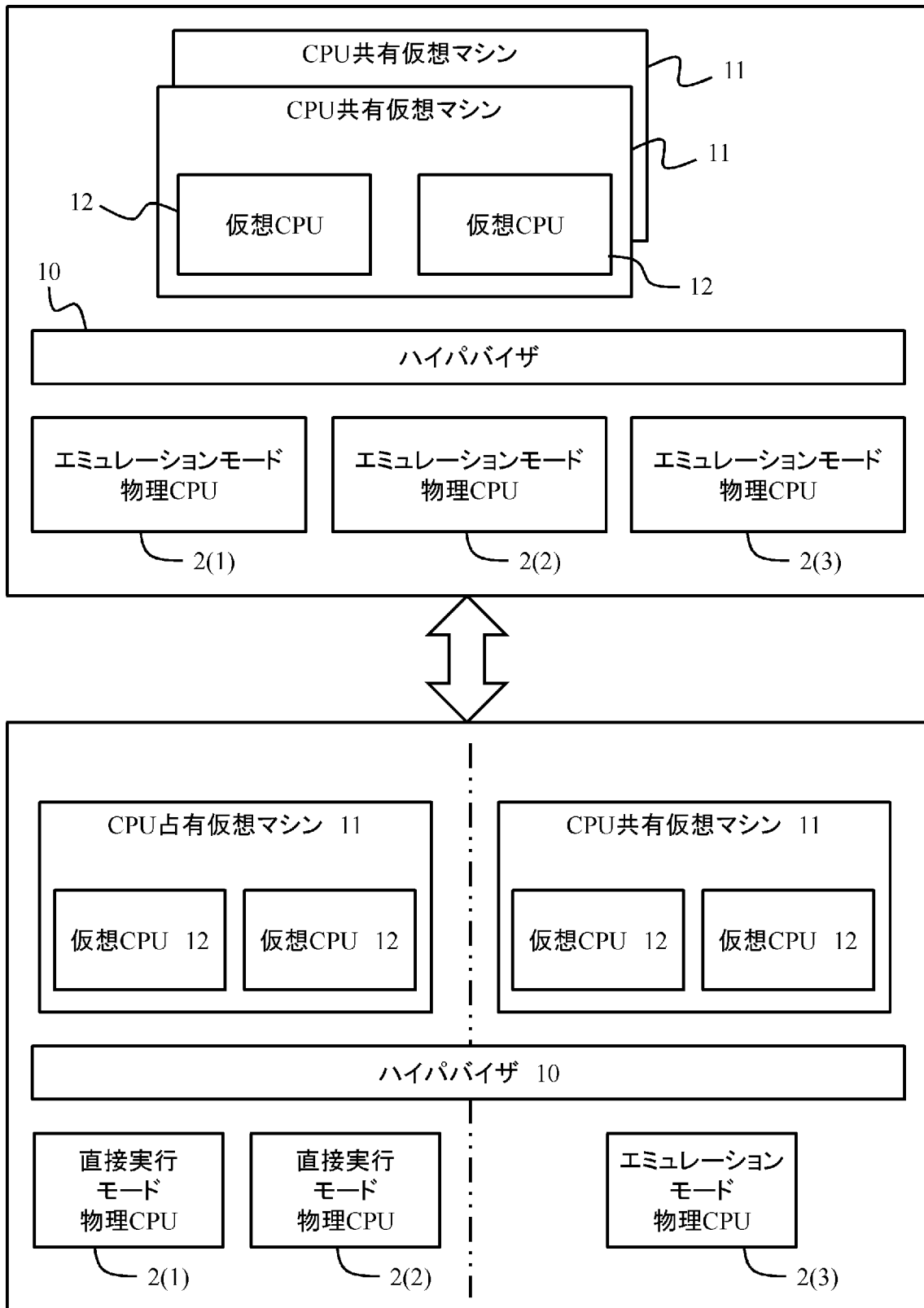
[図7]



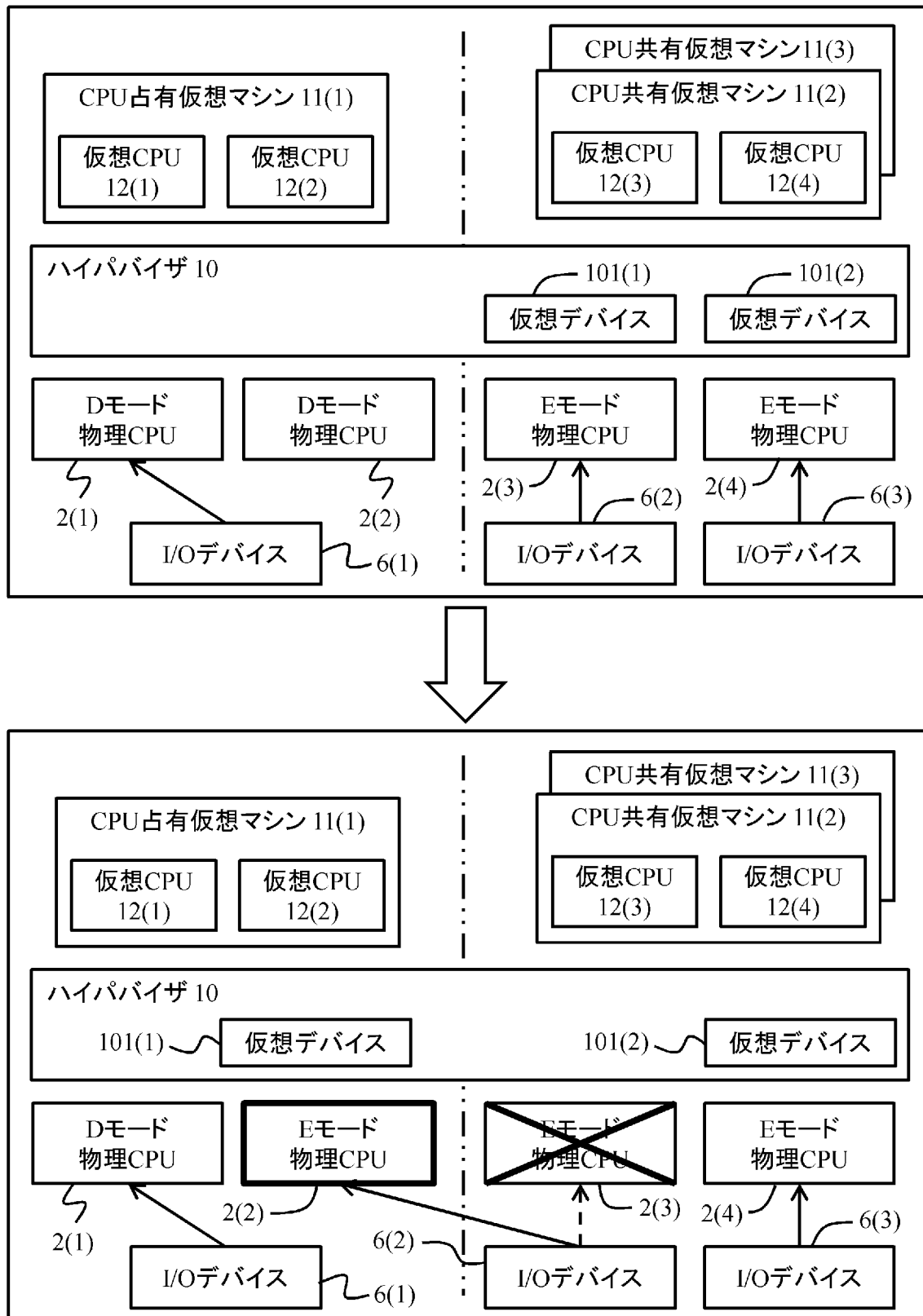
[図8]



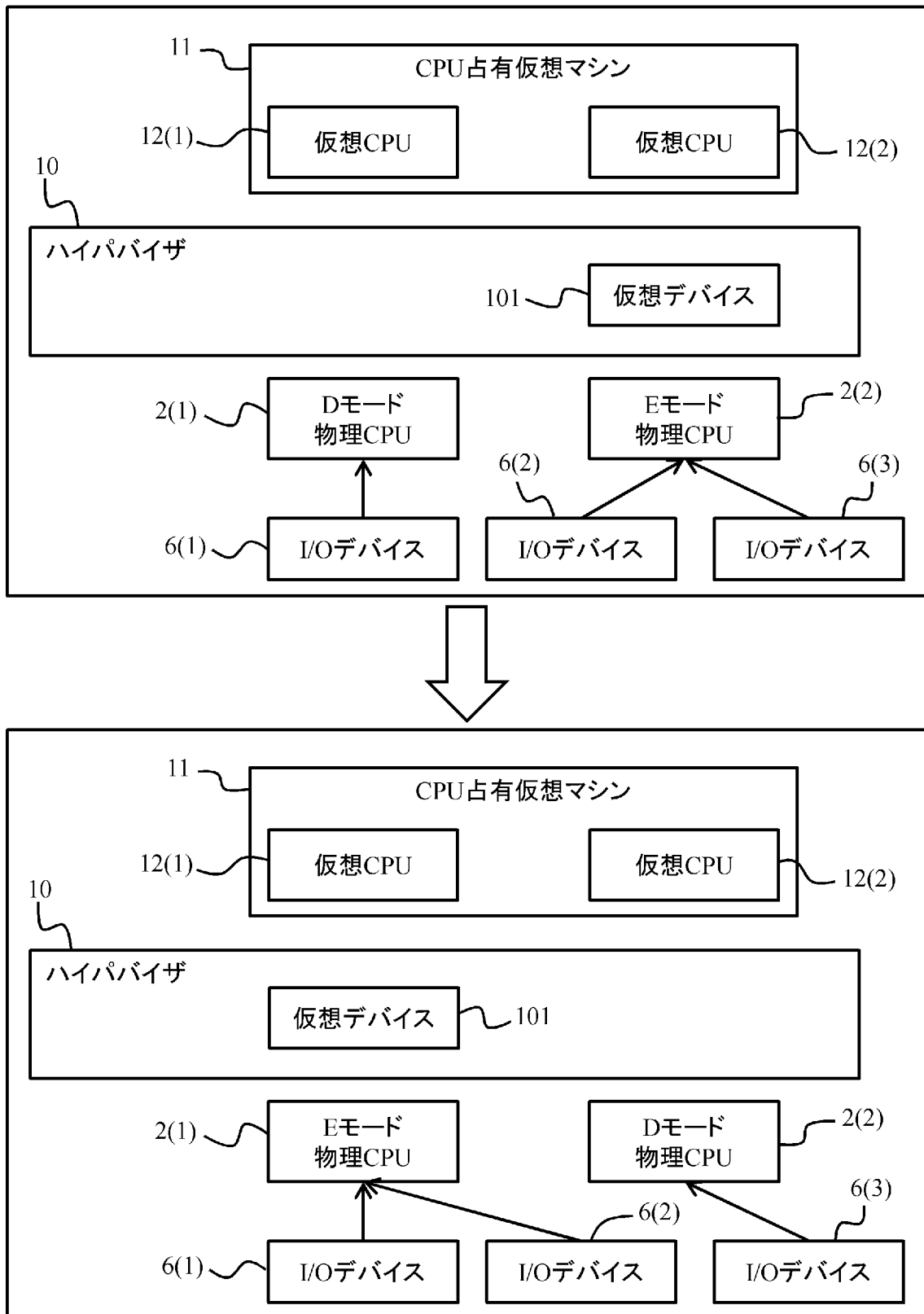
[図9]



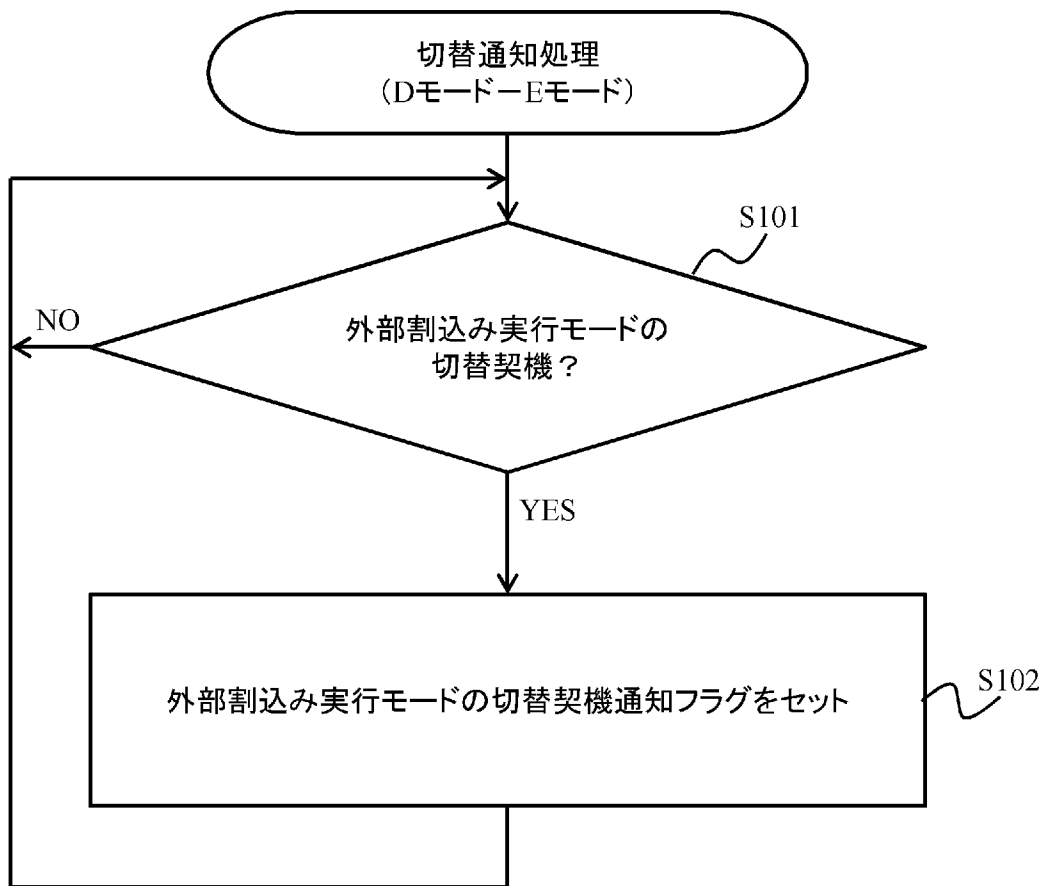
[図10]



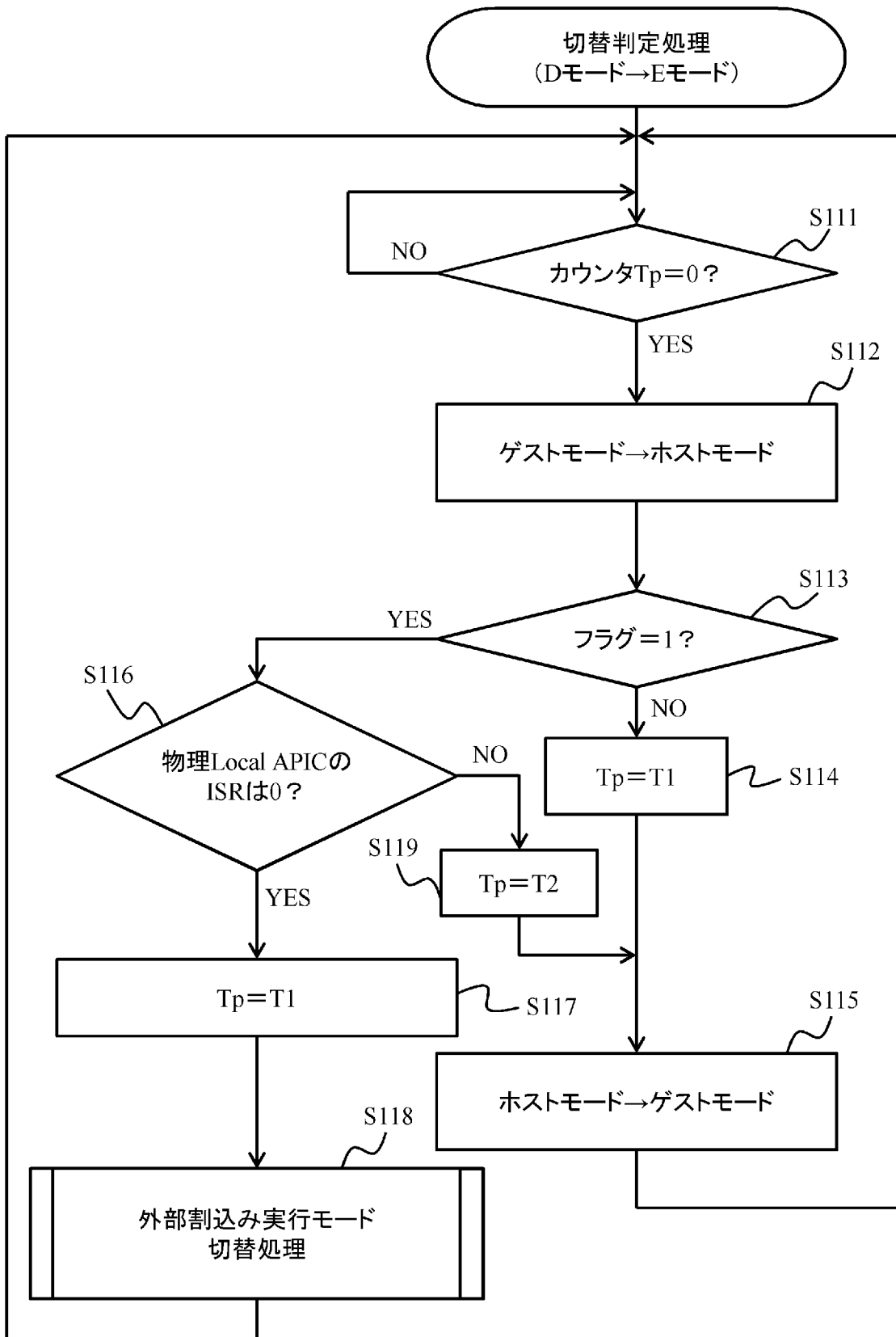
[図11]



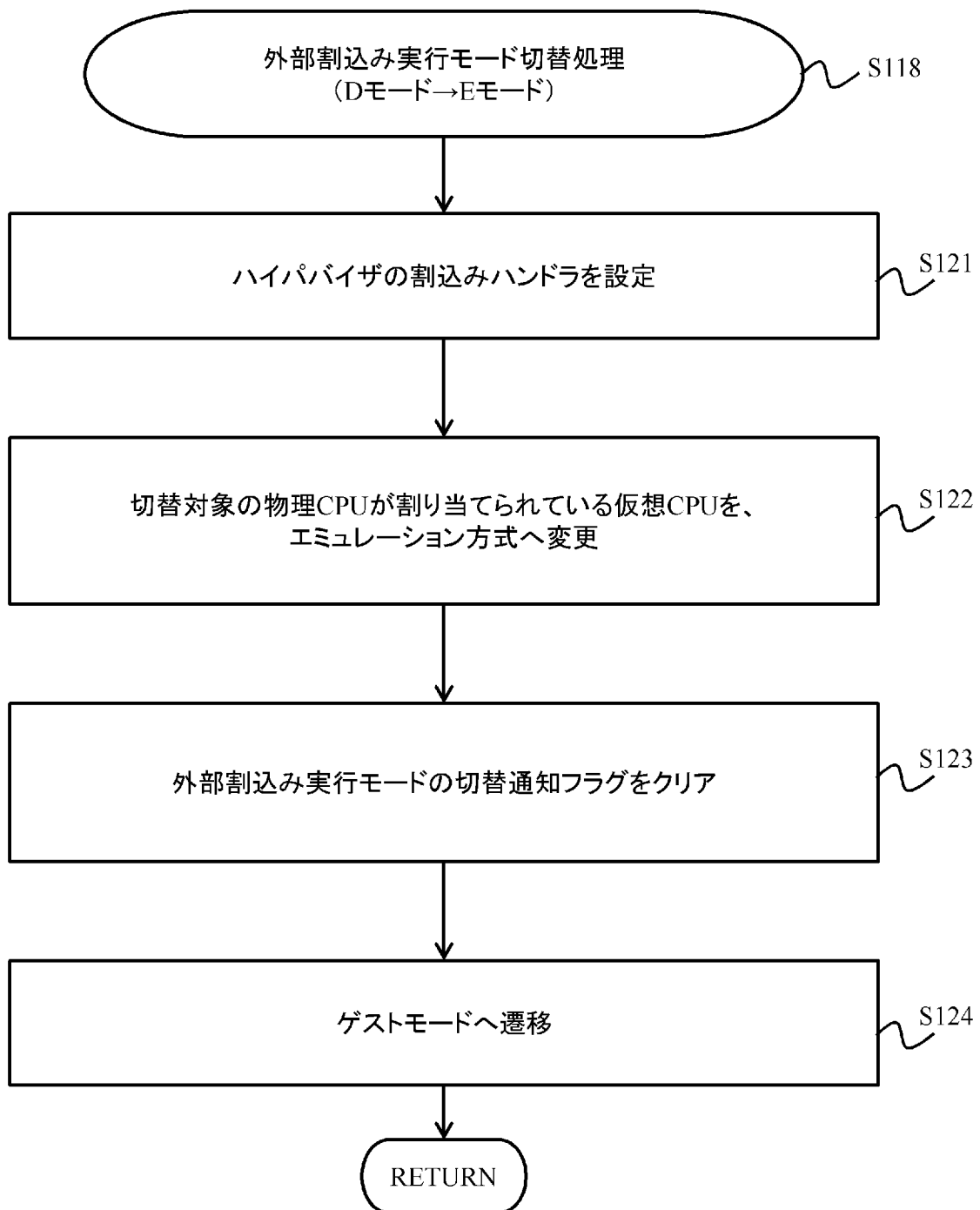
[図12]



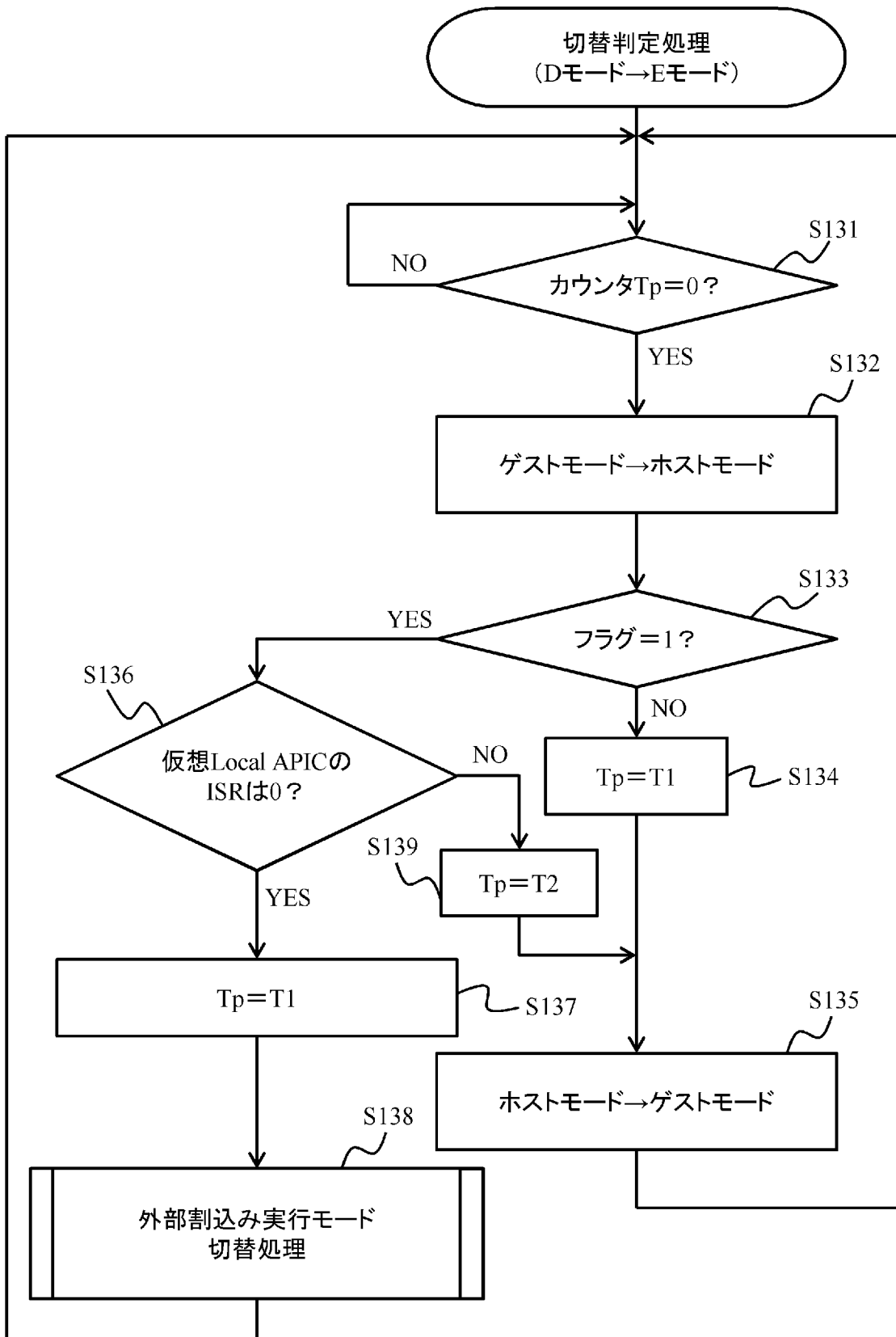
[図13]



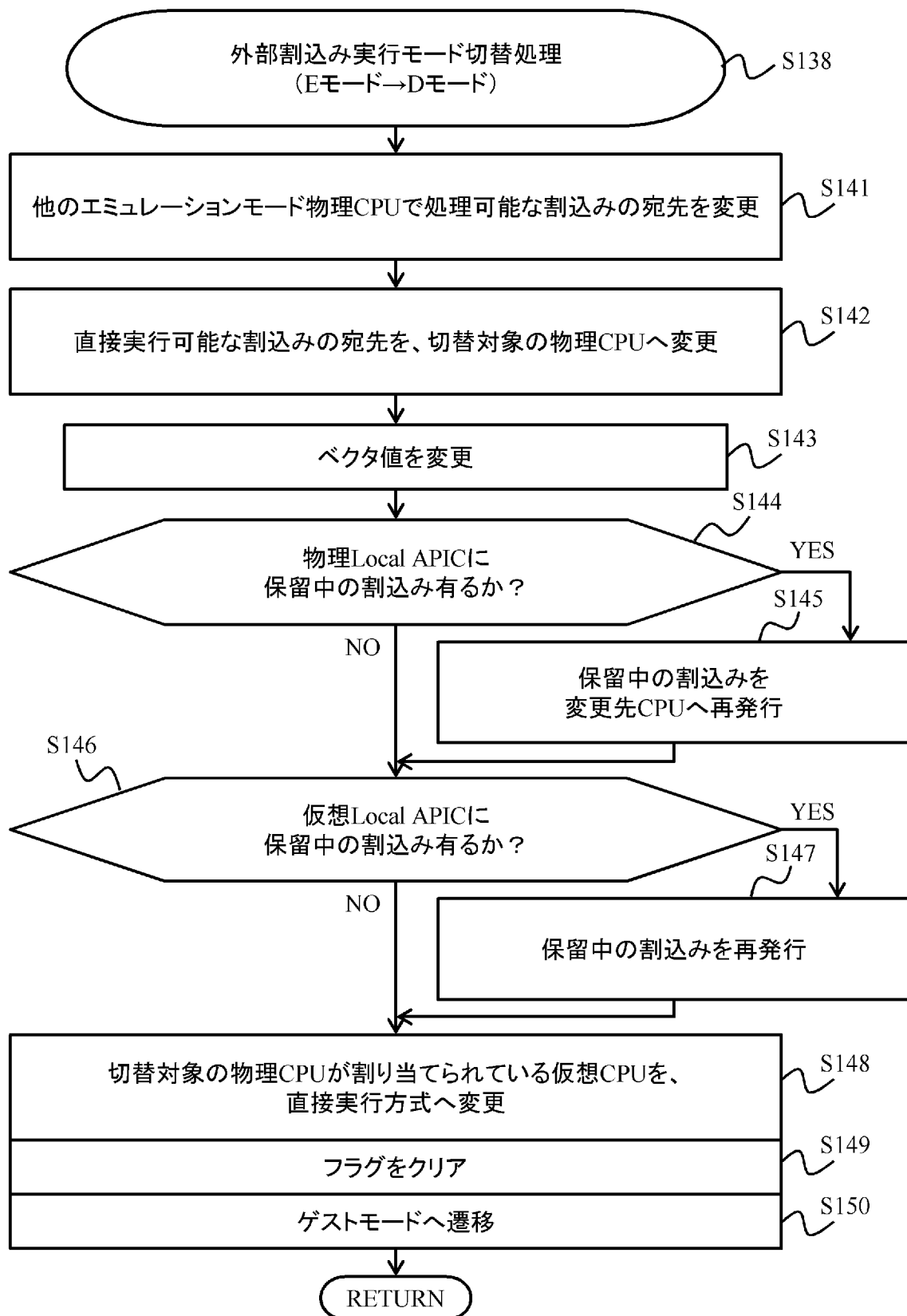
[図14]



[図15]



[図16]



INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP2015/083846

A. CLASSIFICATION OF SUBJECT MATTER

G06F9/48(2006.01)i, G06F9/46(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, G06F9/46

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	1996-2016
Kokai Jitsuyo Shinan Koho	1971-2016	Toroku Jitsuyo Shinan Koho	1994-2016

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 2006-209386 A (Hitachi, Ltd.), 10 August 2006 (10.08.2006), paragraphs [0015] to [0040]; fig. 1 to 5 (Family: none)	1, 10, 13 2-9, 11-12
A	JP 2012-515995 A (Advanced Micro Devices, Inc.), 12 July 2012 (12.07.2012), entire text & US 2010/0191885 A1 & WO 2010/085804 A1 & KR 10-2011-0113756 A & CN 102341783 A	1-13
A	US 2011/0088030 A1 (VMWARE, INC.), 14 April 2011 (14.04.2011), entire text & US 7853960 B1 & US 2013/0326518 A1	1-13

☐ 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
24 February 2016 (24.02.16)

Date of mailing of the international search report
08 March 2016 (08.03.16)

Name and mailing address of the ISA/
Japan Patent Office
3-4-3, Kasumigaseki, Chiyoda-ku,
Tokyo 100-8915, Japan

Authorized officer

Telephone No.

A. 発明の属する分野の分類（国際特許分類（IPC））

Int.Cl. G06F9/48(2006.01)i, G06F9/46(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（IPC））

Int.Cl. G06F9/48, G06F9/46

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	1971-2016年
日本国実用新案登録公報	1996-2016年
日本国登録実用新案公報	1994-2016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 2006-209386 A (株式会社日立製作所) 2006.08.10, 段落【0015】－【0040】, 図1－図5（ファミリーなし）	1, 10, 13 2-9, 11-12
A	JP 2012-515995 A (アドバンスト・マイクロ・ディバイシズ・インコーポレイ テッド) 2012.07.12, 全文 & US 2010/0191885 A1 & WO 2010/085804 A1 & KR 10-2011-0113756 A & CN 102341783 A	1-13
A	US 2011/0088030 A1 (VMWARE, INC.) 2011.04.14, 全文 & US 7853960 B1 & US 2013/0326518 A1	1-13

☐ 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

24.02.2016

国際調査報告の発送日

08.03.2016

国際調査機関の名称及びあて先

日本国特許庁（ISA/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）

井上 宏一

5 B

4177

電話番号 03-3581-1101 内線 3544