

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2014-134989

(P2014-134989A)

(43) 公開日 平成26年7月24日(2014.7.24)

(51) Int.Cl.	F I	テーマコード (参考)
G06F 1/14 (2006.01)	G06F 1/04 351B	5B089
G06F 9/46 (2006.01)	G06F 9/46 350	
G06F 13/00 (2006.01)	G06F 13/00 351C	

審査請求 未請求 請求項の数 10 O L (全 16 頁)

(21) 出願番号 特願2013-3058 (P2013-3058)  
 (22) 出願日 平成25年1月11日 (2013.1.11)

(71) 出願人 000005108  
 株式会社日立製作所  
 東京都千代田区丸の内一丁目6番6号  
 (74) 代理人 100100310  
 弁理士 井上 学  
 (74) 代理人 100098660  
 弁理士 戸田 裕二  
 (74) 代理人 100091720  
 弁理士 岩崎 重美  
 (72) 発明者 高田 有時  
 神奈川県横浜市戸塚区吉田町292番地  
 株式会社日立製作所横浜研究所内

最終頁に続く

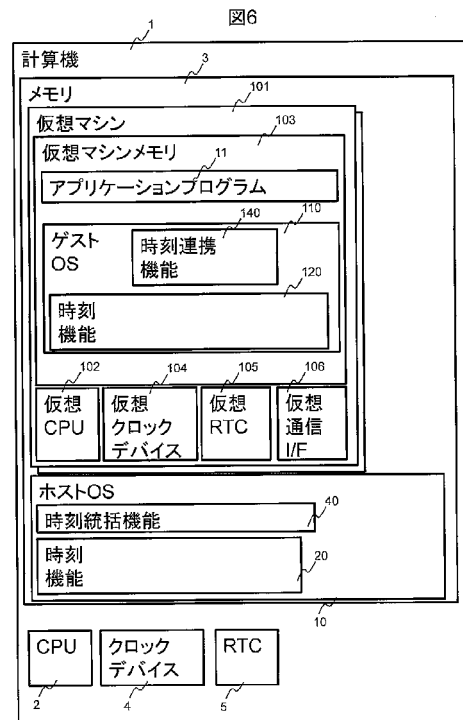
(54) 【発明の名称】 計算機システム及び計算機管理方法

(57) 【要約】

【課題】 ゲストOS内の時刻をホストOS内の時刻に反映させることが困難。

【解決手段】 計算機上で仮想計算機を構築・動作させる際、ある仮想計算機内部における時刻設定要求または時刻補正要求を、実計算機および他仮想計算機にも反映させる。また、それらの反映の可否および反映・被反映の関係を管理・制御する。仮想計算機内OS内に時刻設定要求、時刻補正要求・モード変更要求を受け付ける時刻連携機能を設け、一方実計算機OS内にそれらを統括管理する時刻統括機能を設け、両者を連携させる。

【選択図】 図6



**【特許請求の範囲】****【請求項 1】**

計算機システムは、ゲストOSとホストOSを備え、  
前記ゲストOSは、  
前記ゲストOSにおける時刻を管理する時刻機能部と、  
前記ゲストOSが時刻運用モードを受信し、ホストOSに送信する時刻連携部とを備え、  
前記ホストOSは、  
前記ホストOSにおける時刻を管理する時刻機能部と、  
前記ゲストOSから受信した時刻運用モードの変更要求に応じて時刻を補正する、又は前記ホストOS内で、時刻運用モードの変更要求に応じて時刻を補正する時刻連携部と、  
を備えることを特徴とする計算機システム。

10

**【請求項 2】**

請求項 1 に記載の計算機システムであって、  
前記ゲストOSは、運用モードの状態を記憶する全体モード情報記憶部を更に備え、  
前記ゲストOSは、前記時刻運用モードの変更を受信すると、前記全体モード情報記憶部に記憶された前記運用モードに基づいて前記変更の可否を判断することを特徴とする計算機システム。

**【請求項 3】**

請求項 2 に記載の計算機システムであって、  
前記時刻運用モードには、マスターモード、スレーブモード、独立モードを含み、  
前記ホストOSがマスターモードまたはスレーブモードである場合には、ホストOSの仮想計算機が持つ現在時刻を表す情報を更新する際に、前記ホストOSが持つ時刻の変化に同期する仮想デバイスが提供する情報を用い、  
独立モードである場合には、計算機ハードウェアが備える、仮想計算機外オペレーティングシステムに依存しないデバイスが提供する情報を用いることを特徴とする計算機システム。

20

**【請求項 4】**

請求項 3 に記載の計算機システムであって、  
前記ゲストOSのモードを変更する要求を受け付け、当該要求が独立モードから他モードへの変更、または、他モードから独立モードへの変更である場合には、前記時刻を表す情報を更新する際に用いるデバイスを切り替えることを特徴とする計算機システム。

30

**【請求項 5】**

請求項 3 に記載の計算機システムであって、  
前記ゲストOSのモードを変更する要求を受け付け、  
前記ゲストOSのモードをマスターに変更する要求である場合には、前記ゲストOSの現在時刻情報を仮想計算機外オペレーティングシステムおよびスレーブモードのホストOSに設定し、  
前記ゲストOSのモードをスレーブに変更する要求である場合には、ゲストOSの現在時刻情報を前記ホストOSに設定することを特徴とする計算機システム。

40

**【請求項 6】**

ゲストOSとホストOSを管理する計算機管理方法において、  
前記ゲストOSは、  
前記ゲストOSにおける時刻を管理し、  
前記ゲストOSが時刻運用モードを受信し、ホストOSに送信し、  
前記ホストOSは、  
前記ホストOSにおける時刻を管理し、  
前記ゲストOSから受信した時刻運用モードの変更要求に応じて時刻を補正する、又は前記ホストOS内で、時刻運用モードの変更要求に応じて時刻を補正することを特徴とする計算機管理方法。

50

**【請求項 7】**

請求項 6 に記載の計計算機管理方法であって、  
前記ゲストOSは、運用モードの状態を全体モード情報記憶部に記憶し、  
前記ゲストOSは、前記時刻運用モードの変更を受信すると、前記全体モード情報記憶部に記憶された前記運用モードに基づいて前記変更の可否を判断することを特徴とする計算機管理方法。

**【請求項 8】**

請求項 7 に記載の計算機管理方法であって、  
前記時刻運用モードには、マスターモード、スレーブモード、独立モードを含み、  
前記ホストOSがマスターモードまたはスレーブモードである場合には、ホストOSの仮想  
計算機が持つ現在時刻を表す情報を更新する際に、前記ホストOSが持つ時刻の変化に同期  
する仮想デバイスが提供する情報を用い、  
独立モードである場合には、計算機ハードウェアが備える、仮想計算機外オペレーティ  
ングシステムに依存しないデバイスが提供する情報を用いることを特徴とする計算機管理  
方法。

10

**【請求項 9】**

請求項 8 に記載の計算機管理方法であって、  
前記ゲストOSのモードを変更する要求を受け付け、当該要求が独立モードから他モー  
ドへの変更、または、他モードから独立モードへの変更である場合には、前記時刻を表す  
情報を更新する際に用いるデバイスを切り替えることを特徴とする計算機管理方法。

20

**【請求項 10】**

請求項 8 に記載の計算機管理方法であって、  
前記ゲストOSのモードを変更する要求を受け付け、  
前記ゲストOSのモードをマスターに変更する要求である場合には、前記ゲストOSの  
現在時刻情報を仮想計算機外オペレーティングシステムおよびスレーブモードのホストOS  
に設定し、  
前記ゲストOSのモードをスレーブに変更する要求である場合には、ゲストOSの現在  
時刻情報を前記ホストOSに設定する  
ことを特徴とする計算機管理方法。

**【発明の詳細な説明】**

30

**【技術分野】****【0001】**

本発明は、計算機上の仮想的な計算機システム及び計算機管理方法に関する。

**【背景技術】****【0002】**

計算機を運用するにあたって、計算機を構成するハードウェア、および、計算機内のC  
PUが実行する基本ソフト（オペレーティングシステム、以降OSと表記）が、現在時刻  
を表す情報（以降、時刻情報）を保有・管理する方法（以降、時刻管理と表記）が知られ  
ている。この時刻情報は、ある処理の実行要否の判断や、逆に、ある処理が実行されたタ  
イミングの把握や、処理に要した所要時間の算出などの目的で用いられる。

40

**【0003】**

また、近年、計算機の運用における利便性の向上などの目的にて、ある計算機（以降、  
ホストマシンと表記）上で仮想的な計算機ハードウェア（以降、ゲストマシンと表記）を  
1台または複数台再現し、各々のゲストマシン上で独立にソフトウェアを実行する「計算  
機仮想化」技術が広く用いられている。計算機仮想化技術には、ゲストマシンを実現する  
機構を備えるハードウェアまたはソフトウェアの位置付けにより様々なものがあるが、近  
年はホストマシン上で単一のOS（ホストOS）を実行し、このホストOSが備える機構  
を用いてゲストマシンを構築し、ゲストマシン上でOS（ゲストOS）およびアプリケー  
ションを実行する方法がある。

**【0004】**

50

このような計算機仮想化技術を適用したシステムにおいて、ホストマシン上で適切な時刻管理が行われていることを前提に、ホスト上の時刻情報を用いてゲストの時刻管理を行う方法が用いられている。

【0005】

例えば、非特許文献1には、仮想マシンの中に、仮想クロックデバイス(KVM: k v m c l o c k 又は P V c l o c k )を備え、ホストOSの時刻機能の進み方と同一になるよう仮想マシンが仮想クロックデバイスを制御する技術が記載されている。

【先行技術文献】

【非特許文献】

【0006】

【非特許文献1】" K V M "、[online]、[平成24年7月25日検索]、インターネット(URL : [http://www.linux-kvm.org/page/Main\\_Page](http://www.linux-kvm.org/page/Main_Page))

【発明の概要】

【発明が解決しようとする課題】

【0007】

計算機仮想化技術を、例えば制御システム等の、実世界におけるタイミング制御を行うようなシステムに適用する場合、仮想マシン内で動作するアプリケーションプログラムの制御に基づくホストOS内の時刻管理が必要とされることがある。これは、制御システムの運用上の制約により、制御システムの動作に支障が発生しない方法・タイミングで時刻情報の変更・補正を行う必要があることによる。しかしながら特許文献1に記載の技術では、仮想マシン内で動作するアプリケーションプログラムの制御に基づくホストOS内の時刻管理を行うことができない。

【0008】

さらに、例えば、ある任意の仮想マシンについて、テスト等の必要により、一時的に実時刻と異なる時刻情報を設定したい場合がある。しかしながら非特許文献1に記載の技術では、時刻管理に関するホストマシン・ゲストマシン間の連携について、必要に応じて動的にホスト・ゲストの関係を変更することが困難である。

【課題を解決するための手段】

【0009】

上記課題を解決するため、本発明では、ゲストOSは、ゲストOSにおける時刻を管理する時刻機能部と、ゲストOSが時刻運用モードを受信し、ホストOSに送信する時刻連携部とを備え、ホストOSは、ホストOSにおける時刻を管理する時刻機能部と、ゲストOSから受信した時刻運用モードの変更要求に応じて時刻を補正する、又は前記ホストOS内で、時刻運用モードの変更要求に応じて時刻を補正する時刻連携部と、を備える。

【発明の効果】

【0010】

本発明によれば、ゲストOS内の時刻をホストOS内の時刻に反映させることができ、更に、時刻管理に関するゲスト・ホスト間の関係を動的に変更することができる。例えば、特定のゲストマシンにおける時刻管理に、ホストマシンや他ゲストマシンを同期させることや、特定のゲストマシンを一時的に同期から切り離すことができる。

【図面の簡単な説明】

【0011】

【図1】計算機システムの一例を示す図である。

【図2】時刻機能の一例を示す図である。

【図3】時刻設定要求の一例を示す図である。

【図4】時刻補正要求の一例を示す図である。

【図5】仮想化技術を適用した計算機システムの一例を示す図である。

【図6】時刻連携機能及び時刻統括機能を導入した計算機システムの一例を示す図である。

【図7】時刻連携機能の一例を示す図である。

10

20

30

40

50

- 【図 8】時刻統括機能の一例を示す図である。
- 【図 9】全体モード情報の一例を示す図である。
- 【図 10】時刻設定要求発生時のフローの一例の前半を示す図である。
- 【図 11】時刻設定要求発生時のフローの一例の後半を示す図である。
- 【図 12】時刻設定情報の一例を示す図である。
- 【図 13】時刻補正要求発生時のフローの一例を示す図である。
- 【図 14】時刻補正情報の一例を示す図である。
- 【図 15】モード変更要求発生時のフローの一例を示す図である。
- 【図 16】モード変更処理のフローの一例を示す図である。
- 【図 17】モード変更要求の一例を示す図である。
- 【図 18】モード変更情報の一例を示す図である。
- 【図 19】モード変更結果情報の一例を示す図である。
- 【図 20】時刻統括機能の別の一例を示す図である。
- 【図 21】ホスト内モード変更要求の一例を示す図である。
- 【図 22】時刻設定情報の別の一例を示す図である。
- 【図 23】モード変更結果情報の別の一例を示す図である。
- 【発明を実施するための形態】

#### 【0012】

本発明の一実施形態の説明に先立ち、実施形態が前提とする時刻管理の仕組み、および、計算機仮想化環境について述べる。

図 1 から図 5 は公知技術の説明図である。

#### 【0013】

まず、計算機における既存の時刻管理の仕組みとして、一例を図 1 に示す。計算機 1 は、ハードウェアとして少なくとも CPU 2 とメモリ 3 を備えている。尚、CPU 2 は複数個存在してもよい。メモリ 3 上には少なくともオペレーティングシステム (OS) 10、および、アプリケーションプログラム 11 が展開されているものとする。また、計算機はクロックデバイス 4 およびリアルタイムクロック (RTC) 5 を備える。ここで、クロックデバイスとは、タイマ割込を CPU 2 に通知すること、または、CPU 2 から参照可能なレジスタに時間経過に応じて増減するカウンタ値を設定すること、などにより、CPU 2 に時間経過を表す情報を提供する任意のデバイスである。クロックデバイス 4 は、例えば Programmable Interval Timer (PIT) や、High Precision Event Timer (HPET) や Timestamp Counter (TSC) である。このような計算機において、CPU 2 は OS 10 内時刻機能 20 の働きにより、時刻管理を行う。尚、本発明にて時刻情報とは実時刻 (現在時刻) または特定のタイミングを表す情報であり、日付やタイムゾーン等の情報を含んでよい。

#### 【0014】

図 2 は、時刻機能 20 の構成の一例を示したものである。時刻機能 20 は、OS10 の時刻を管理する機能である。また時刻機能 20 は、時刻情報を保持するための領域である OS 時刻情報領域 21、時間の経過に応じて OS 時刻情報領域 21 を更新する機能である時刻更新機能 22 を備える。さらに、計算機の起動または停止の際に、計算機の動作状態によらずに時刻管理を行うデバイスである RTC 5 から時刻情報を復元、または逆に RTC 5 へ時刻情報を退避する機能である時刻退避・復元機能 23 を備える。さらに、アプリケーションプログラム 11 やその他プログラムの実行により発生する要求に応じた処理を行う時刻提供機能 24、時刻設定機能 25、時刻補正機能 26 を備える。ここで時刻提供機能 24 とはアプリケーションプログラム 11 やその他プログラムの実行に必要な時刻情報を提供する機能であり、時刻設定機能 25 とは、時刻情報と実時刻のずれを瞬時に解消するなどの目的で新たな時刻情報を OS 時刻情報領域 21 に設定する機能であり、時刻補正機能 26 とは、時刻情報が不連続に変化しないよう徐々に時刻情報と実時刻のずれを解消するなどの目的で時刻更新機能 22 による更新幅を調整する機能である。これらの機能は、例えば Linux (登録商標) における date コマンドや settimeofday

10

20

30

40

50

関数、adjtime関数として提供される。従って、これらの機能はアプリケーションプログラム11やユーザから呼び出される。時刻設定機能25は、例えば図3に示すような時刻設定要求27をアプリケーションプログラム11またはユーザから受け付ける。時刻設定要求27には、例えば新たに設定されるべき時刻である新時刻28が含まれる。また、時刻補正機能26は、例えば図4に示すような時刻補正要求29を受け付ける。時刻補正要求29には、例えば時刻の進み方に対する補正内容を表す補正值30が含まれる。

#### 【0015】

既存の時刻管理の仕組みを図5に示す。仮想化技術を適用した計算機には、1台以上の仮想マシン101が存在する。仮想マシン101は、プログラムおよびデータとしてメモリ3内に展開される。仮想マシン101は、計算機1におけるCPU2、メモリ3、クロックデバイス4、RTC5に相当する、仮想CPU102、仮想マシンメモリ103、仮想クロックデバイス104、仮想RTC105を備える。また、実計算機上のOS10またはその上で動作するプログラムと情報の伝達を行うための仮想通信インタフェース(仮想通信I/F)106を備えていてもよい。

10

#### 【0016】

仮想マシンメモリ103内には、計算機1上のOS10と同様に、ゲストOS110が存在し、さらにゲストOS110は時刻機能120を備える。この時刻機能120は、連携対象が仮想クロックデバイス104や仮想RTC105であることを除き、計算機1上の時刻機能20と同様である。

#### 【0017】

ここで、仮想クロックデバイス104は、仮想マシン101の働きによりつくられ、ホストOS10内時刻機能20が保持する時刻情報に基づく動作を行う擬似デバイスか、あるいはパススルー方式により透過的に参照可能となった計算機1内クロックデバイス4であってよい。また、仮想クロックデバイス104は一つの仮想マシン101内に複数種類存在してよく、それらには擬似デバイスとパススルー方式により参照されるクロックデバイス4の両方が含まれてよい。また、仮想RTC105は、例えばホストOS10内時刻機能20が保持する時刻情報に基づき時刻情報を提供する擬似デバイスであってよい。

20

#### 【0018】

以上で述べた既存の時刻管理方式、および、計算機仮想化環境に基づく本発明の実施形態について以降で述べる。尚、以降にて任意の仮想マシン101のことを、仮想マシン101外の機能・事象と区別する目的で、各々ゲスト・ホストと表記することがある。

30

#### 【0019】

図6は、本発明の実施形態の一例を示す図である。本実施形態では、図5に示した仮想化環境に加え、ゲストOS110内に時刻連携機能140、ホストOS10内に時刻統括機能40を新たに設ける。これらの時刻統括機能40および時刻連携機能140が仮想通信I/F106を介して互いに連携することで、柔軟なホスト・ゲスト間時刻管理連携を実現する。

#### 【0020】

本実施形態における時刻連携機能140の構成を図7に示す。時刻連携機能140には、モード変更受付機能141、時刻設定受付機能142、時刻補正受付機能143、ホスト要求受付機能144、ゲストモード情報145が含まれる。

40

#### 【0021】

ここで、時刻設定受付機能142、時刻補正受付機能143は、時刻機能120内時刻設定機能25および時刻補正機能26と同様または類似の呼出インタフェースを備え、時刻設定要求27および時刻補正要求29を受け付けるものとする。本実施形態では、アプリケーションプログラム11の書き換え、ゲストOS110の書き換え等の方法により、アプリケーションプログラム11が発行した時刻設定要求27および時刻補正要求29は、時刻設定機能27や時刻補正機能26ではなく時刻設定受付機能142および時刻補正受付機能143により処理されるものとする。

#### 【0022】

50

また、ホスト要求受付機能 144 は、後述するモード変更情報や時刻設定情報、時刻補正情報を、時刻統括機能 40 から仮想通信 I / F 106 経由で受け付け処理する機能である。

#### 【0023】

モード変更受付機能 141 及びゲストモード情報 145 に関する説明に先立ち、本実施形態における仮想マシン 101 のモード管理について概要を述べる。本実施形態では、動作中の各仮想マシン 101 について各々マスター・スレーブ・独立のいずれかのモードをとるものとする。ここで、マスターとは当該仮想マシン 101 内での時刻設定・補正がホスト OS 10 及び他ゲスト OS に伝達・適用される状態であり、スレーブとは逆にマスターまたはホスト OS 10 における時刻設定・補正が当該仮想マシン 101 にも自動的に適用される状態である。独立とは、ホスト OS 10 や他仮想マシン 101 と無関係に時刻設定・補正が実行・適用される状態である。

以上で述べたモード管理を実現するため、時刻連携機能 140 はモード変更受付機能 141 及びゲストモード情報 145 を備える。モード変更受付機能 141 はアプリケーションプログラム 11 またはユーザから当該ゲストのモード変更要求を受け付け、適切な処理を行う機能であり、ゲストモード情報 145 は当該ゲストのモードを記憶する領域である。すなわち、ゲストモード情報 145 として、先に述べたゲストが取りうるモードである、マスター、スレーブ、独立いずれかを示す識別子が記載される。

#### 【0024】

次に、本実施形態における時刻統括機能 40 の構成を図 8 に示す。時刻統括機能 40 は少なくとも、マスターモードの仮想マシン 101 内時刻連携機能 140 からの要求を受け付けるゲスト要求受付機能 41 と、各仮想マシン 101 のモードを保持・管理するデータ領域である全体モード情報 44 を備える。また、ホスト内における時刻設定要求 27 および時刻補正要求 29 を処理するため、時刻設定受付機能 42、時刻補正受付機能 43 を備えていてもよい。全体モード情報 44 は、図 9 に示す通り、各仮想マシン 101 について、ゲスト識別子 45 およびモード 46 を保持する。

#### 【0025】

以上で述べた構成を踏まえ、本実施形態における時刻管理に関する処理フローを説明する。尚、処理フローはメモリ上のプログラムとして実現された機能により実現されるため、以降では、仮想 CPU 102 または CPU 2 がある機能 A 内の処理 B を行うことを、単に機能 A が処理 B を行う、と表記する。実行の主体は、機能 A が仮想マシン 101 内である場合は仮想マシン 101 内の任意の仮想 CPU 102 であり、仮想マシン 101 外である場合は任意の CPU 2 である。

#### 【0026】

まず、ある仮想マシン 101 上にて、アプリケーションプログラム 11 が時刻設定要求 27 を発行した場合のフローについて、時刻設定情報 150 を送受信するまでのフローを示した図 10、時刻設定情報 150 を送受信した後のフローを示した図 11 を用いて説明する。最初に、時刻設定受付機能 142 が時刻設定要求 27 をアプリケーションプログラム 11 またはユーザから受け付ける (S101)。次に、時刻設定受付機能 142 は、当該ゲストがスレーブモードであるかどうか、ゲストモード情報 145 を参照することで判定する (S102)。判定の結果、スレーブモードであった場合は時刻設定受付機能 142 の実行を終了する (S103)。このとき、時刻設定要求 27 を発行したアプリケーションプログラム 11 に対して、単に設定の成功を示す結果コードを返してもよいし、エラーを示す結果コードを返してもよい。スレーブモードでなかった場合は、当該ゲスト内の時刻設定機能 25 を呼び出し、時刻設定要求 27 と同様の要求を時刻設定機能 25 に対して行う (S104)。次に、当該ゲストが独立モードであるか否か、ゲストモード情報 145 を参照することで判定する (S105)。独立モードであった場合は、時刻設定受付機能 142 の実行を終了する (S106)。

#### 【0027】

当該ゲストが独立モードでもなかった場合、すなわち、マスターモードであった場合、

時刻設定情報 150 を、仮想通信 I/F 106 を介して、ホスト内、ゲスト要求受付機能 41 に対して送信する。ここで、時刻設定情報 150 とは、図 12 に示すような、時刻設定要求 27 と同じ新時刻 28 を備えるデータである。また、時刻設定要求 27 が発生した仮想マシン 101 を示すゲスト識別子 45 を備えていてもよい。ホスト側、時刻統括機能 40 内ゲスト要求受付機能 41 は、時刻設定情報 150 を受信することで処理を開始する (S108)。

#### 【0028】

時刻設定情報 150 を受信したゲスト要求受付機能 41 は、まず時刻機能 20 内時刻設定機能 25 を呼び出し、時刻設定情報 150 内新時刻 28 を含む時刻設定要求 27 を行う (S109)。次に、全体モード情報 44 を参照し、スレーブモードである各仮想マシン 101 内ホスト要求受付機能 144 に対して、仮想通信 I/F 106 を介して時刻設定情報 150 を送信する。ここで、時刻設定情報 150 内新時刻 28 は、S108 にて受信した時刻設定情報 150 内新時刻 28 と同じものとする。時刻設定情報を受信した (S111)、スレーブモードである仮想マシン 101 内のホスト要求受付機能 144 は、各仮想マシン 101 内の時刻設定機能 25 を呼び出し、時刻設定要求 27 を発行する (S112)。この時刻設定要求 27 内新時刻 28 は、時刻設定情報 150 内新時刻 28 と同じものとする。以上をもって、アプリケーションプログラム 11 が時刻設定要求 27 を発行した場合のフローを終了する (S113)。

#### 【0029】

次に、ある仮想マシン 101 上にて、アプリケーションプログラム 11 が時刻補正要求 29 を発行した場合のフローについて、図 13 を用いて説明する。最初に、時刻補正受付機能 143 が時刻補正要求 29 を受け付ける (S201)。次に、時刻補正受付機能 143 は、当該ゲストがスレーブモードであるかどうか、ゲストモード情報 145 を参照することで判定する (S202)。判定の結果、スレーブモードであった場合は時刻補正受付機能 143 の実行を終了する (S203)。このとき、時刻補正要求 29 を発行したアプリケーションプログラム 11 に対して、単に補正の成功を示す結果コードを返してもよいし、エラーを示す結果コードを返してもよい。スレーブモードでなかった場合は、当該ゲストが独立モードであるかどうか同様に判定する (S204)。独立モードであった場合は、当該ゲスト内の時刻補正機能 26 を呼び出し、時刻補正要求 29 と同様の要求を時刻補正機能 26 に対して行い (S205)、時刻補正受付機能 143 の実行を終了する (S206)。

#### 【0030】

当該ゲストが独立モードでもなかった場合、すなわち、マスターモードであった場合、時刻補正情報 160 を、仮想通信 I/F 106 を介して、ホスト内、ゲスト要求受付機能 41 に対して送信する (S207)。ここで、時刻補正情報 160 とは、図 14 に示すような、時刻補正要求 29 と同じ補正值 30 を備えるデータである。時刻設定情報 150 同様、時刻補正情報 160 はゲスト識別子 45 を含んでもよい。

#### 【0031】

ホスト側、時刻統括機能 40 内ゲスト要求受付機能 41 は、時刻補正情報 160 を受信することで処理を開始する (S208)。ゲスト要求受付機能 41 は、時刻機能 20 内時刻補正機能 26 を呼び出し、時刻補正情報 160 内補正值 30 を含む時刻補正要求 29 を行い (S209)、実行を終了する (S210)。尚、時刻設定の場合と異なり、マスターモードおよびスレーブモードの仮想マシン 101 内の時刻の進み方は、後述するモード変更の際のクロックソース設定により、自動的にホストの時刻機能 20 が提供する時刻の進み方に同期するため、各仮想マシン 101 内時刻補正機能 26 を呼び出す必要はない。

#### 【0032】

最後に、ある仮想マシン 101 上にて、アプリケーションプログラム 11 がモード変更要求 170 を発行した場合のフローについて、全体のフローを示した図 15、モード変更処理部分のフローを示した図 16 を用いて説明する。尚、モード変更要求 170 は、図 17 に示すように、当該仮想マシン 101 の遷移先モードである新モード 171 を含む。



## 【 0 0 3 3 】

まず、モード変更受付機能 1 4 1 がアプリケーションプログラム 1 1 またはユーザからモード変更要求 1 7 0 を受け付ける ( S 3 0 0 )。次に、モード受付機能 1 4 1 は、モード変更情報 1 8 0 を、仮想通信 I / F 1 0 6 を介して、ホスト内、ゲスト要求受付機能 4 1 に対して送信する ( S 3 0 1 )。ここでモード変更情報 1 8 0 とは、図 1 8 に示すような、モード変更要求 1 7 0 と同じ新モード 1 7 1 を備えるデータである。

## 【 0 0 3 4 】

ホスト側、時刻統括機能 4 0 内ゲスト要求受付機能 4 1 は、モード変更情報 1 8 0 を受信することで処理を開始する ( S 3 0 3 )。ゲスト要求受付機能 4 1 は、まず、全体モード情報 4 4 を参照し、モード変更情報 1 8 0 に記されたモード変更により矛盾が発生しないか判定する ( S 3 0 3 )。ここで、矛盾が発生するとは、既にマスターモードである仮想計算機 1 0 1 が存在する状態で、新たに別の仮想計算機 1 0 1 がマスターモードに遷移し、マスターモードの仮想計算機 1 0 1 が複数になることを意味する。矛盾が発生しない場合のみ、ゲスト要求受付機能 1 4 1 は全体モード情報 4 4 の、モード変更情報 1 8 0 内ゲスト識別子 4 5 に対応する箇所のモード 4 6 を、新モード 1 7 1 記載の内容に変更する ( S 3 0 4 )。矛盾が発生した場合は、S 3 0 5 にすすむ。次に、仮想通信 I / F 1 0 6 を介して、モード変更結果情報 1 9 0 をモード変更情報 1 8 0 の送信元であるモード変更受付機能 1 4 1 に送信する ( S 3 0 5 )。ここで、モード変更結果情報 1 9 0 とは、図 1 9 に示す通り、少なくとも変更可否 1 9 1 を含む。また、先のモード変更情報 1 8 0 における新モードがスレーブである場合は、スレーブモードへの変更時に設定すべき時刻を示す新時刻 1 9 2 を含んでもよい。

10

20

## 【 0 0 3 5 】

モード変更受付機能 1 4 1 は、モード変更結果情報 1 9 0 を受信し処理を再開する ( S 3 0 6 )。モード変更結果情報 1 9 0 にて、モード変更可否 1 9 1 が可となっていた場合のみ、後述するモード変更処理を実行し ( S 3 0 8 )、処理を終了する ( S 3 0 9 )。

## 【 0 0 3 6 】

モード変更処理 ( S 3 0 8 ) の内容について図 1 6 を用いて述べる。モード変更受付機能 1 4 1 はモード変更処理の実行開始後 ( S 3 5 1 )、まずモード変更が独立モードと他のモードの間の変更であるか否かを判定する。ここで、独立モードと他のモードの間の変更であるとは、マスターモードまたはスレーブモードから独立モードへの変更であるか、逆に独立モードからマスターモードまたはスレーブモードへの変更であることを意味する。本条件に合致する場合は、時刻機能 1 2 0 内時刻更新機能 2 2 に対して、クロックソースの切替を要求する ( S 3 5 3 )。クロックソースとは、時刻更新機能 2 2 が時刻更新の基準とするクロックデバイス 4、仮想マシン 1 0 1 内では仮想クロックデバイス 1 0 4 のことである。独立モードから他モードへの遷移である場合は、クロックソースとしてホスト OS 1 0 内時刻機能 2 0 が保持する時刻情報に基づき動作する擬似デバイス方式を採用している仮想クロックデバイス 1 0 4 を、逆の場合はパススルー方式により透過的に参照可能なクロックデバイス 4 を切替先として選択する。これにより、独立モードの場合はホスト OS 1 0 とは直接無関係な仮想クロックデバイス 1 0 4 を、他のモードである場合はホスト OS 1 0 と進み方が一致する仮想クロックデバイス 1 0 4 をクロックソースとして

30

40

## 【 0 0 3 7 】

次に、遷移元・遷移先モードが各々独立モード・スレーブモードであるか判定する ( S 3 5 4 )。独立モードからスレーブモードへの遷移の場合は、ホスト側の時刻に同期するため、スレーブモード当該仮想マシン 1 0 1 内時刻設定機能 2 5 に対して、時刻設定要求 2 7 を発行し ( S 3 5 5 )、実行を終了する ( S 3 5 6 )。ここで、時刻設定要求 2 7 内新時刻 2 8 としては、モード変更結果情報 1 9 0 内新時刻 1 9 2 記載のものを用いる。

## 【 0 0 3 8 】

独立モードからスレーブモードへの遷移でなかった場合は、独立モードからマスターモードへの遷移であるか判定する ( S 3 5 7 )。マスターモードへの遷移の場合のみ、ホス

50

トおよびスレーブモードのゲストの時刻を同期させるため、仮想通信 I / F 106 を介して、ホスト内、ゲスト要求受付機能 41 に対して時刻設定情報 150 を送信する (S358)。ここで、時刻設定情報 150 内新時刻 28 としては、送信時点にて当該ゲスト内時刻提供機能 24 から取得した時刻情報を用いる。尚、ゲスト要求受付機能 41 が時刻設定情報 150 を受信してからのフローは、図 11 内 S108 以降のものと同一であるため省略する。以上をもって、モード変更処理 S308 を終了する (S359)。

#### 【0039】

以上で述べた実施形態の一例の通り、本発明を用いることで時刻管理に関するゲスト・ホスト間の連携を柔軟に設定・制御することが可能となる。例えば、特定のゲストマシンにおける時刻管理に、ホストマシンや他ゲストマシンを同期させることや、特定のゲストマシンを一時的に同期から切り離すことができる。

10

#### 【0040】

ただし、実施形態は本例に限らない。例えば、ゲスト・ホスト間で仮想通信 I / F を介した情報転送の可否をゲストモード情報 145 または全体モード情報 44 に基づき判断する箇所については、送信元で可否を判断する代わりに受信側にてゲストモード情報 145 または全体モード情報 44 に基づき採用可否を判断してもよい。

#### 【0041】

また、本例ではモード変更受付機能 141 を仮想計算機 101 内時刻連携機能 140 内に設けることでゲスト側でのモード変更操作を可能としているが、時刻統括機能 40 内に同様の機能を設け、ホスト側にて任意のゲストのモード変更を可能としてもよい。この場合、図 20 に示す通り、時刻統括機能 40 はモード変更受付機能 47 を備え、図 21 に示すホスト内モード変更要求 200 を受け付ける。モード変更受付機能 47 の動作は、ゲスト要求受付機能 141 が、先に述べたホスト内モード変更要求 200 内ゲスト識別子 45、新モード 171 を含むモード変更情報 180 を受信した場合 (S302) 以降の処理に準ずる。これにより、ホスト内アプリケーションプログラムまたはユーザからモード変更が可能となる。

20

#### 【0042】

さらに、ホスト・ゲストにて時刻設定機能 25 を呼び出すタイミングがずれることにより時刻にずれが発生することを防ぐため、時刻設定情報 150 およびモード変更結果情報 190 について、図 22 および図 23 に示す通り、新時刻 28、新時刻 192 に加え、基準カウンタ値 48、カウンタ倍率 49 を含んでもよい。ここで、基準カウンタ値 48 とは、時刻設定機能呼出 (S104) や、モード変更結果情報 190 送信 (S305)、マスターへの遷移に伴う時刻設定情報 150 送信 (S358) 時点におけるパススルー方式のクロックデバイス 4 または仮想クロックデバイス 104 が備えるカウンタの値であり、カウンタ倍率とは、例えば時刻更新機能 22 が OS 時刻情報領域 21 の更新の際の増分算出に用いている、前記パススルー方式のクロックデバイス 4 または仮想クロックデバイス 104 のカウンタ値の変動量と実時間の比率を示す値、例えば一秒間にカウンタ値が増加する量である。

30

#### 【0043】

このとき、ゲスト要求受付機能 41 やホスト要求受付機能 144 が時刻設定情報 150 に基づき時刻設定機能 25 を呼び出す際 (S109、S112)、あるいは、モード変更結果情報 190 に基づき時刻設定機能 25 を呼び出す際 (S355)、単に時刻設定情報 150 内新時刻 28 またはモード変更結果情報 190 内新時刻 192 を時刻設定要求 27 内新時刻 28 に設定するのではなく、改めて前記パススルー方式のクロックデバイス 4 または仮想クロックデバイス 104 のカウンタ値を取得し、基準カウンタ値 48 との差分を算出した後、カウンタ倍率 49 を用いて実時間に変換した値を時刻設定情報 150 内新時刻 28 またはモード変更結果情報 190 内新時刻 192 の値に加算し、加算後の値を時刻設定要求 27 内新時刻 28 に設定する。ここで、カウンタ倍率 49 を用いて実時間に変換するとは、例えばカウンタ倍率 49 が一秒間にカウンタ値が増加する量である場合は、カウンタ値の差分をカウンタ倍率 49 で除した値を算出することである。これにより、時刻管理

40

50

機能140および時刻統括機能40の処理フローおよび相互通信に遅延が発生する場合でも、時刻の同期が可能となる。

【符号の説明】

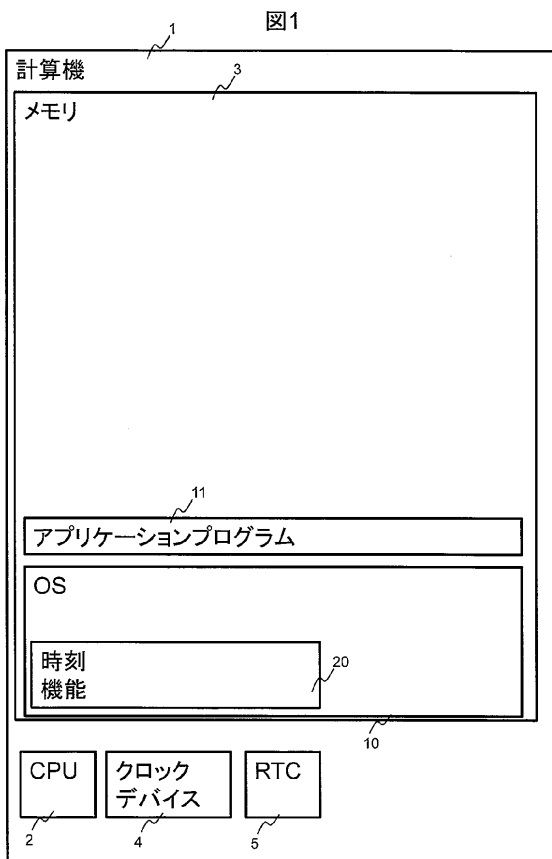
【0044】

- 1 計算機
- 2 CPU
- 3 メモリ
- 4 クロックデバイス
- 5 RTC
- 10 OS (ホストOS)
- 11 アプリケーションプログラム
- 20 時刻機能
- 40 時刻統括機能
- 101 仮想マシン
- 102 仮想CPU
- 103 仮想マシンメモリ
- 104 仮想クロックデバイス
- 105 仮想RTC
- 106 仮想通信I/F
- 110 ゲストOS
- 140 時刻連携機能

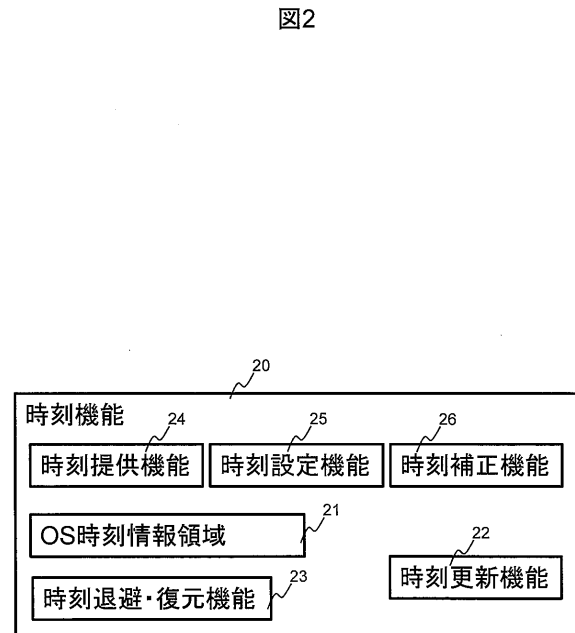
10

20

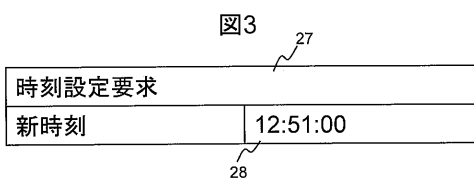
【図1】



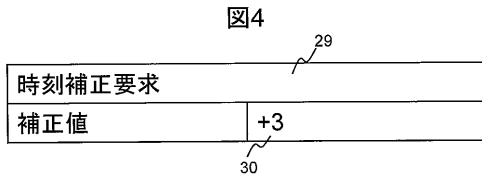
【図2】



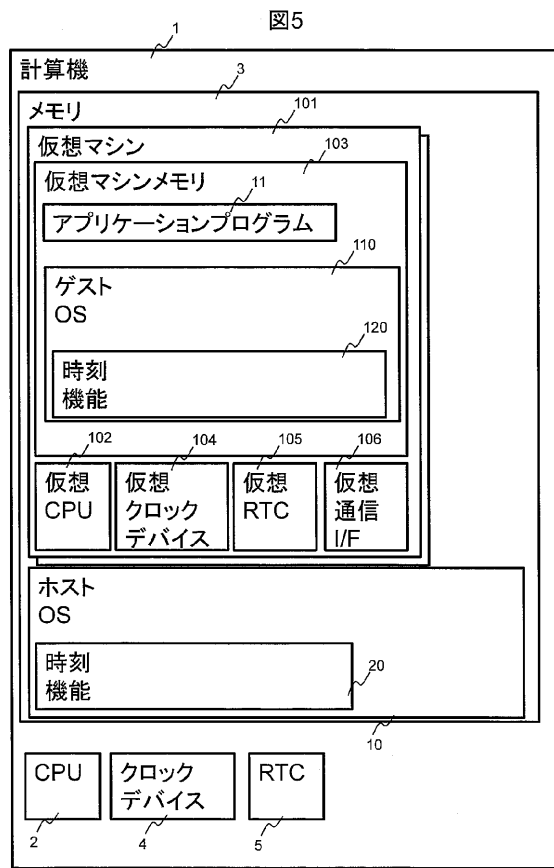
【図3】



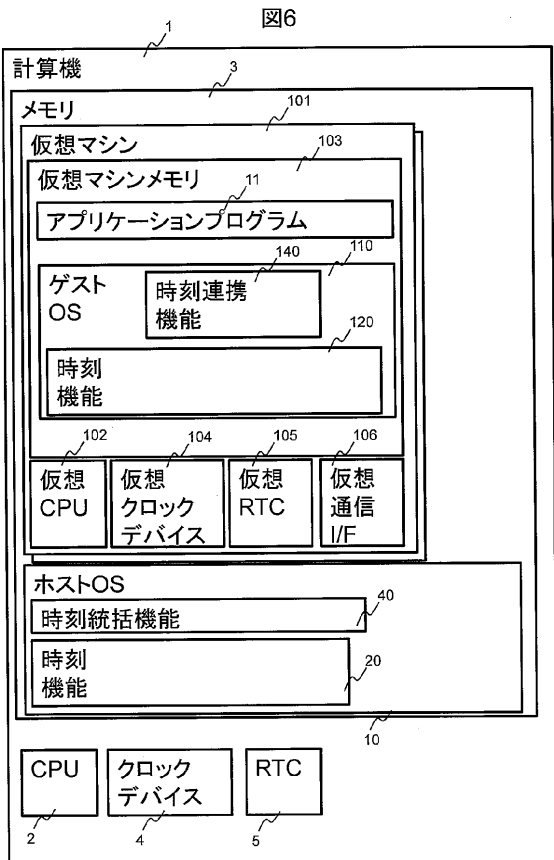
【 図 4 】



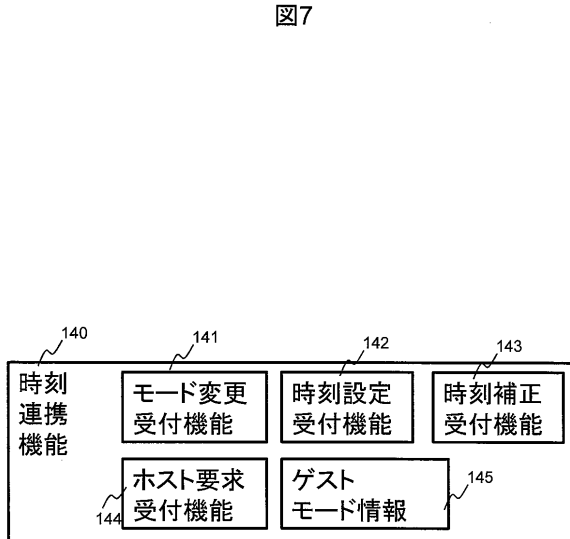
【 図 5 】



【 図 6 】

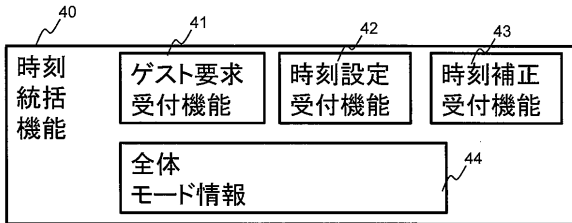


【 図 7 】



【 図 8 】

図8



【 図 9 】

図9

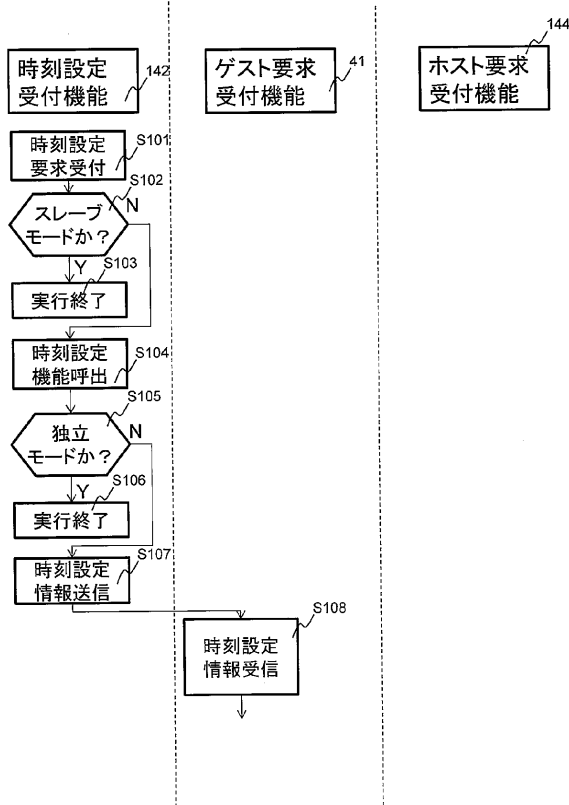
全体モード情報 (44)

ゲスト識別子	モード
1	マスター
2	独立
3	スレーブ
4	スレーブ

45, 46

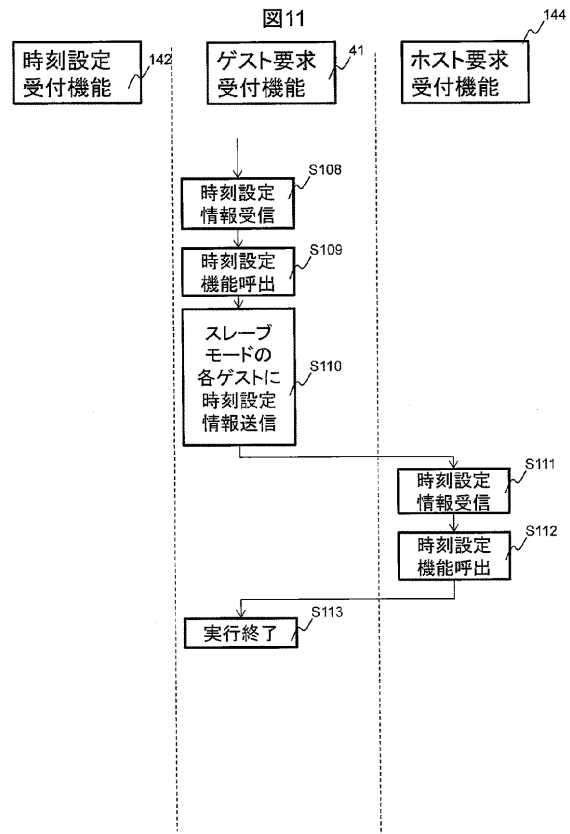
【 図 1 0 】

図10



【 図 1 1 】

図11



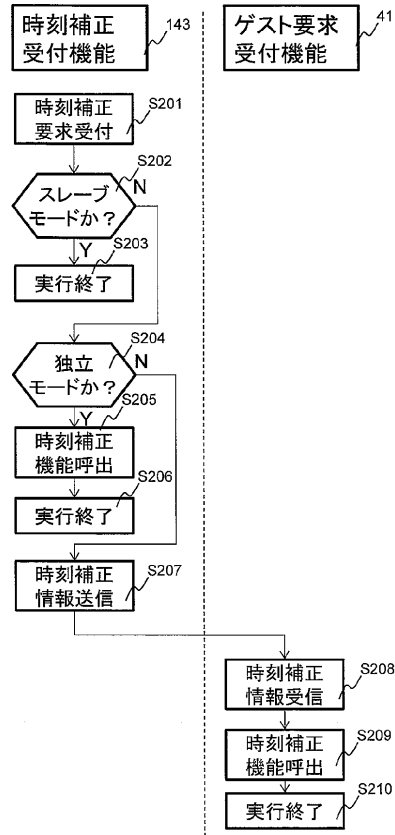
【図12】

図12

時刻設定情報		45	
ゲスト識別子	1		28
新時刻	12:51:00		

【図13】

図13



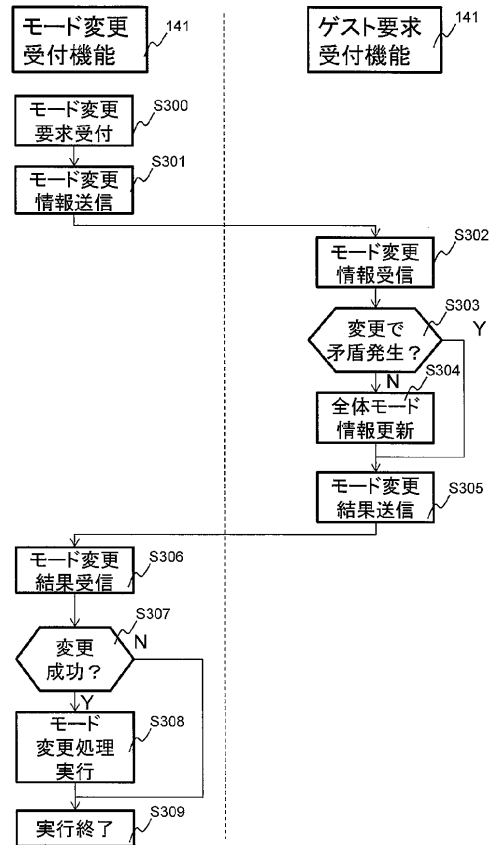
【図14】

図14

時刻補正情報		45	
ゲスト識別子	1		28
補正值	+3		

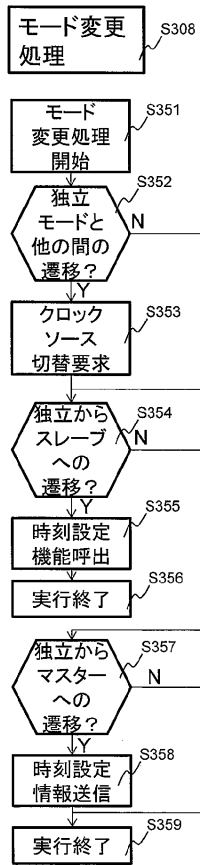
【図15】

図15



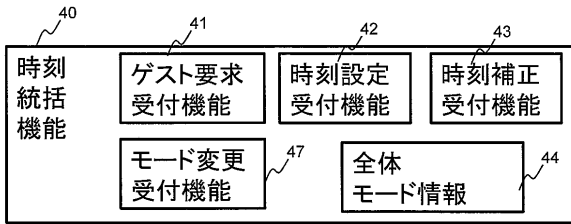
【 図 1 6 】

図16



【 図 2 0 】

図20



【 図 2 1 】

図21

ホスト内モード変更要求	
ゲスト識別子	1
新モード	マスター

【 図 1 7 】

図17

モード変更要求	
新モード	マスター

【 図 1 8 】

図18

モード変更情報	
ゲスト識別子	1
新モード	マスター

【 図 1 9 】

図19

モード変更結果情報	
ゲスト識別子	1
変更可否	可
新時刻	12:51:05

【 図 2 2 】

図22

時刻設定情報	
ゲスト識別子	1
新時刻	12:51:00
基準カウンタ値	1464078533
カウンタ倍率	1,000,000,000

【 図 2 3 】

図23

モード変更結果情報	
ゲスト識別子	1
変更可否	可
新時刻	12:51:05
基準カウンタ値	1464078533
カウンタ倍率	1,000,000,000

フロントページの続き

(72)発明者 稲葉 大祐

茨城県日立市大みか町五丁目2番1号 株式会社日立製作所情報制御システム事業部内

Fターム(参考) 5B089 JA35 KA13 KB04