

(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 2 月 2 日(02.02.2017)



(10) 国際公開番号
WO 2017/017707 A1

- (51) 国際特許分類:
G06F 11/30 (2006.01)
- (21) 国際出願番号: PCT/JP2015/003734
- (22) 国際出願日: 2015 年 7 月 24 日(24.07.2015)
- (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: 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: 福島 広隆 (FUKUSHIMA, Hirotaka); 〒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: 下田 憲次 (SHIMODA, Kenji); 〒2118588 神奈川県川崎市中原区上小田中 4 丁目 1 番 1 号 Kanagawa (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.

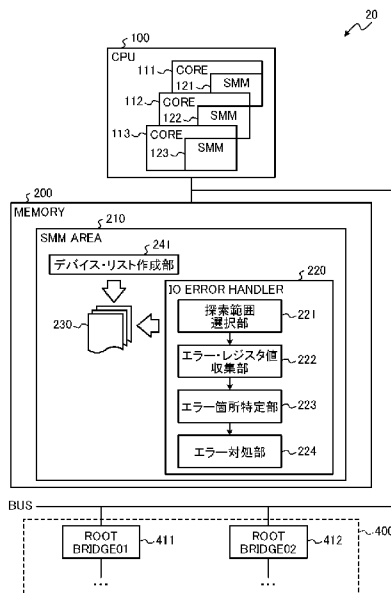
- (84) 指定国 (表示のない限り、全ての種類の広域保護が可能): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), ユーロパ (AM, AZ, BY, KG, KZ, RU, TJ, TM), ユーロッパ (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

添付公開書類:

— 国際調査報告 (条約第 21 条(3))

(54) Title: INFORMATION PROCESSING DEVICE, ERROR PROCESSING METHOD, AND ERROR PROCESSING PROGRAM

(54) 発明の名称: 情報処理装置、エラー処理方法およびエラー処理プログラム



111, 112, 113 Core
200 Memory
210 SMM area
220 IO error handler
221 Search range selecting unit
222 Error register value collecting unit
223 Error location specifying unit
224 Error handling unit
241 Device list creating unit
300 Bus
411, 412 Root bridge

(57) Abstract: [Problem] To specify failure of a peripheral device in a computer within a short time. [Solution] The present invention includes a first peripheral device to be accessed by a processing device via a bus, and is provided with: a plurality of peripheral devices which are connected in a tree structure with the first peripheral device set at a starting point and which is accessed by the processing device via the first peripheral device; and a storage device that stores therein configuration information indicating the connection configuration of the plurality of peripheral devices. The processing device refers to the configuration information stored in the storage device in response to an interruption due to an error occurring in any of the plurality of peripheral devices, specifies a plurality of terminal devices that are a plurality of peripheral devices located at the terminal side opposite to the first peripheral device, among the plurality of peripheral devices connected in the tree structure, and preferentially reads information on the error from the plurality of specified terminal devices to determine the presence/absence of the error, thereby specifying the peripheral device in which the error has occurred.

(57) 要約: 【課題】 コンピュータに含まれる周辺装置の故障を短時間で特定すること。 【解決手段】 バスを介して処理装置からアクセスされる第 1 周辺装置を含み、該第 1 周辺装置を起点にツリー状に接続されるとともに該第 1 周辺装置を介して前記処理装置からアクセスされる複数の周辺装置と、前記複数の周辺装置の接続構成を示す構成情報を記憶する記憶装置とを備える。前記処理装置は、前記複数の周辺装置のいずれかで発生したエラーに起因する割り込みに応じて、前記記憶装置に記憶された構成情報を参照して前記ツリー状に接続された複数の周辺装置のうち前記第 1 周辺装置とは逆側の末端に位置する複数の周辺装置である複数の末端装置を特定し、特定した前記複数の末端装置から優先的に、前記エラー情報を読み出してエラーの有無を判定することで、前記エラーを発生させた周辺装置を特定する。

明 細 書

発明の名称：

情報処理装置、エラー処理方法およびエラー処理プログラム

技術分野

[0001] 本発明は、情報処理装置、エラー処理方法およびエラー処理プログラムに関する。

背景技術

[0002] コンピュータのハードウェアに故障が発生すると、故障が発生した部品を特定し、特定した故障部品の保守や交換が行われる。コンピュータの故障部品の保守や交換に関連する従来技術として、次のような技術が知られている。

[0003] 第1の従来技術では、ハードウェア障害発生時に、短時間で障害箇所及び被疑部品を特定するために、ハードウェアを制御するデバイスの構成情報を取得して、デバイス間の経路と、対応するデバイスと、交換被疑部品とを対応付けた経路テーブルとを作成する。デバイス間の経路の異常を監視し、異常経路の発生を検出すると、経路情報テーブルを参照し、異常経路に対応する交換被疑部品を特定する。

[0004] また、第2の従来技術では、PCI-Express (Peripheral Component Interconnect-Express、登録商標) バスのリンク切れ等によってバスに接続された装置のレジスタの読み出しができない状態が発生する場合、バスのリンク切れに関する障害情報を記録する。そして、記録した障害情報に基づいて、CPU (Central Processing Unit) がPCI-Expressとは異なる他のバスを介してレジスタの値を読み出すことにより、障害原因箇所を特定する。

先行技術文献

特許文献

[0005] 特許文献1：特開2013-41438号公報

特許文献2：WO2012/029147号公報

発明の概要

発明が解決しようとする課題

[0006] コンピュータの仮想化環境においては、コンピュータに含まれる複数のCPUやメモリ、複数の周辺装置等の物理資源を用いて、ホストOS (Operating System) 上に複数のゲストOSを稼働させる。ここで、1つの周辺装置において回復不能な故障 (UE : Uncorrectable Error) が発生すると、CPUに対して強制的な割り込みが入り、ホストOSの動作を一時停止させることから、その間、全てのゲストOSの動作も一時的に停止する。故障した周辺装置を使用するゲストOSは動作を継続できないものの、故障した周辺装置を使用しないゲストOSは動作させることができるため、できる限り短時間で故障した周辺装置を特定し、動作が可能なゲストOSを再開させることが望ましい。

[0007] 本発明の1つの側面では、コンピュータに含まれる周辺装置の故障を短時間で特定することを目的とする。

課題を解決するための手段

[0008] 発明の一観点によれば、バスを介して処理装置からアクセスされる第1周辺装置を含み、該第1周辺装置を起点にツリー状に接続されるとともに該第1周辺装置を介して前記処理装置からアクセスされる複数の周辺装置と、前記複数の周辺装置の接続構成を示す構成情報を記憶する記憶装置とを備え、前記処理装置が、前記複数の周辺装置のいずれかで発生したエラーに起因する割り込みに応じて、前記記憶装置に記憶された構成情報を参照して前記ツリー状に接続された複数の周辺装置のうち前記第1周辺装置とは逆側の末端に位置する複数の周辺装置である複数の末端装置を特定し、特定した前記複数の末端装置から優先的に、前記エラー情報を読み出してエラーの有無を判定することで、前記エラーを発生させた周辺装置を特定する、情報処理装置が提供される。

発明の効果

[0009] 一実施態様によれば、コンピュータに含まれる周辺装置の故障を短時間で特定することができる。

図面の簡単な説明

[0010] [図1]図1は、一般的なインテル・アーキテクチャ（IA）のコンピュータの構成例を示す図である。

[図2]図2は、図1のコンピュータ上で2つの仮想マシンを稼働させた例を示す図である。

[図3]図3は、実施例1に係るコンピュータのシステム構成例を示す図である。

[図4]図4は、末端装置とブリッジ装置にエラーが発生した例を示す図である。

[図5]図5は、デバイス・リストの例を示す図である。

[図6]図6は、デバイス・リスト・エントリの例を示す図である。

[図7]図7は、ローカル・エラー・レジスタ及びグローバル・エラー・レジスタの例を示す図である。

[図8]図8は、IOエラーハンドラ処理のフローの例を示す図である。

[図9]図9は、探索範囲選択処理のフローの例を示す図である。

[図10]図10は、エラー・レジスタ値収集処理の例を示す図である。

[図11]図11は、実施例2に係るコンピュータのシステム構成例を示す図である。

[図12]図12は、複数のCPUコアを用いてエラー・レジスタ値収集処理を行う処理の全体フローの例を示す図である。

[図13]図13は、エラー・レジスタ値収集処理の終了待ち処理のフローの例を示す図である。

[図14]図14は、デバイス・リストの初期値の例を示す図である。

[図15]図15は、デバイス・リストの遷移例1を示す図である。

[図16]図16は、デバイス・リストの遷移例2を示す図である。

[図17]図17は、デバイス変更情報インタフェースの例を示す図である。

[図18]図18は、デバイス変更情報インタフェースの「ACTION」フィールドの値の例を示す図である。

[図19]図19は、デバイス削除後のデバイス・リストの例を示す図である。

[図20]図20は、デバイス追加後のデバイス・リストの例を示す図である。

[図21]図21は、実施例3に係るシステム構成図の例を示す図である。

発明を実施するための形態

[0011] 以下、本願の開示する情報処理装置およびエラー処理方法の実施例を図面に基づいて説明する。なお、以下の実施形態の構成は例示であり、本発明は実施形態の構成に限定されるものではない。

[0012] 図1は、一般的なインテル・アーキテクチャ（IA）のコンピュータ10の構成例を示す図である。一般的なインテル・アーキテクチャ（IA）のコンピュータ10では、CPU100や、メモリ200、の他に、PCIデバイスが実装される。図1において、PCIデバイスは、PCI周辺ブロック400に含まれ、バス300を介してCPU100からアクセスされる。

[0013] CPU100は、CPUコア111、112、113を有する。CPUコア111～113のそれぞれは、メモリ200に格納されたプログラムを実行することにより、オペレーティングシステム（OS）や、OS上で動作するアプリケーションプログラムの処理を実行する。インテル・アーキテクチャ（IA）のCPUは、アプリケーションプログラムの処理等を行う動作モードである通常動作モードの他に、System Management Mode（SMM）と呼ばれる通常動作とは別の動作モードを有する。図1では、CPUコア111～113のそれぞれが、対応するSMMの処理部121、122、123を有することを模式的に示している。

[0014] 各CPUコア111～113がSMMの動作モードへ遷移するためには、System Management Interrupt（SMI）と呼ばれる特別な割り込みが使用される。SMIは全CPUコア111～113にブロードキャストされるため、全CPUコア111～113が一斉にSMMの動作モードに遷移する。各CPUコア111～113がSMMの動作モードに遷移している間は、各C

PUコア111～113において通常動作は一時停止され、各CPUコア111～113において、SMIに対する特別な割り込み処理プログラムによる処理が行われる。

[0015] メモリ200は、SMM領域210を含む。SMM領域210は、各CPUコア111～113がSMMのモードで動作するためのプログラムを格納する専用のメモリ領域であり、IOエラーハンドラ220を含む。各CPUコア111～113は、SMMのモードに遷移した場合に、SMM領域210に格納されたIOエラーハンドラ220等に含まれるプログラムを実行することにより、SMMのモードでの処理を行う。

[0016] PCI周辺ブロック400では、ルートブリッジ装置411、412を始点としてツリー状に、ブリッジ装置421、422や、ツリーの末端に位置する末端装置（End Device）431～434がPCIバスで接続される。末端装置431～434には、例えば、LAN（Local Area Network）カード装置や、RAID（Redundant Arrays of Independent Disks）カード等のインタフェース装置等が含まれる。ルートブリッジ装置を始点にしたPCIバスで接続された木構造は、PCIバス・ツリーとも呼ばれる。

[0017] PCIバス・ツリーに含まれる各PCIデバイスは、それぞれ「ローカル・エラー・レジスタ」を有する。各ローカル・エラー・レジスタは、各PCIデバイスにおいてエラーが発生した場合に、発生したエラーに関する情報を記憶するレジスタである。ローカル・エラー・レジスタの例については、図7（A）を用いて後述する。

[0018] ルートブリッジ装置411、412配下のブリッジ装置や421、422や末端装置431～434でエラーが発生すると、PCIバス・ツリーの始点となるルートブリッジ装置421、422にエラーが発生したことが通知される。ルートブリッジ装置421、422に通知されたエラー情報は、ルートブリッジ装置421、422が有する特定のレジスタ（以下、「グローバル・エラー・レジスタ」と言う）に記憶される。グローバル・エラー・レジスタは、ルートブリッジ装置配下のブリッジ装置や末端装置のいずれかで

エラーが発生したか否かを示す情報を保持する。グローバル・エラー・レジスタに格納された情報は、CPUコア111～113がどのPCIバス・ツリー上でエラーが発生したかを調べる際に参照される。グローバル・エラー・レジスタの例については、図7（B）を用いて後述する。

[0019] OSの稼働中に、PCIデバイスの1つでエラーが発生した場合、ルートブリッジ装置421、422によってSMIの割り込みを発生させるシステム構成とすることができる。SMIの割り込みを受信したCPU100の各CPUコア111～113の動作モードはSMMに切り替わり、各CPUコア111～113上で動作するOSの動作は一時的に中断する。

[0020] SMMのモードにおいてIOエラーハンドラ220等の特別な処理ルーチンを実行することにより、エラーが発生したPCIデバイスの特定と、エラー情報の出力が行われる。その際、場合によってはリカバリ処理も実行される。IOエラーハンドラ220の処理が終了し、各CPUコア111～113がSMMのモードから抜けて通常動作モードに戻ることで、停止していたOSの動作が再開される。

[0021] 例えば、コンピュータ10のシステム上で1つのOSが稼働している動作環境において、コンピュータ10に含まれるデバイスに、致命的なエラーであるUncorrectable Error（UE）が発生した場合を想定する。この場合、コンピュータ10に含まれるデバイスで発生したUEによって各CPUコア111～113がSMMのモードに遷移した後、デバイスのエラーが解消されない限り運用を行うのが難しいことが多い。すなわち、デバイスのエラーの原因が解消されない限り、OSは正常動作できない。そのため、デバイスのエラーが解消されない状態で直ちにSMMのモードから通常動作モードに戻す必要はない。

[0022] 一方、コンピュータ10上で複数の仮想マシンを実行させ、各仮想マシン上でゲストOSが稼働している動作環境において、コンピュータ10に含まれるデバイスにUEが発生した場合、1つのOSだけが稼働している動作環境とは状況が異なってくる。特定のデバイスにUEが発生すると、コンピュ

ータ１０に含まれる全ＣＰＵコア１１１～１１３に対してＳＭＩ割り込みが入力されるため、全てのゲストＯＳの動作が停止する。ここで、エラーが発生したデバイスを使用するゲストＯＳは動作を継続できないが、エラーが発生したデバイスを使用しないゲストＯＳは運用を継続させることが可能である。そのため、ＳＭＭのモードでのエラー箇所を特定する処理の時間を短くし、再開可能なゲストＯＳの停止時間を短くすることが望ましい。

[0023] 図２は、図１のコンピュータ１０上で２つの仮想マシンを稼働させた例を示す図である。図２の例では、コンピュータ１０のハードウェア資源（ＣＰＵコア、メモリ、ＰＣＩデバイス等）を用いてパーティション５０１（PARTITION 01）及びパーティション５０２（PARTITION 02）が設けられている。パーティション５０１、５０２は、例えば論理パーティションとすることができる。

[0024] 図２において、パーティション５０１は、ＣＰＵコア１１１、１１２とメモリ２００の一部、ルートブリッジ装置４１１を始点とするＰＣＩデバイス・ツリー、および、ルートブリッジ装置４１２、ブリッジ４２２を介して接続される末端装置４３３を含む。また、パーティション５０２は、ＣＰＵコア１１３とメモリ２００の一部、および、ルートブリッジ装置４１２、ブリッジ４２２を介して接続される末端装置４３４を含む。

[0025] パーティション５０１では、パーティション５０１に含まれる物理資源によって不図示の仮想マシンが稼働し、その仮想マシン上でゲストＯＳ５１１（GUEST OS 01）が実行される。同様に、パーティション５０２では、パーティション５０２に含まれる物理資源（ＣＰＵコア１１３、末端装置４３４等）を用いて不図示の仮想マシンが稼働し、その仮想マシン上でゲストＯＳ５１２（GUEST OS 02）が実行される。

[0026] 図２に示される動作環境において、例えば末端装置４３３（DEVICE 21）でＵＥが発生し、ＳＭＩが各ＣＰＵコア１１１～１１３に入力されると、各ＣＰＵコア１１１～１１３はＳＭＭに遷移し、ゲストＯＳ５１１、５１２の動作は一時停止する。ゲストＯＳ５１１については、パーティション５０１に

含まれる末端装置４３３でエラーが発生しているため、エラーが解消しない限り運用の継続は難しい。しかし、ゲストOS５１２については、パーティション５０２に含まれるハードウェア資源でエラーは発生していないため、運用を継続させることが可能である。この場合、各CPUコア１１１～１１３がSMMのモードに遷移している間、ゲストOS５１２の動作が一時停止していることによる影響が懸念される。

[0027] ここで、PCIデバイスでエラーが発生した場合の従前の方法としては、例えば、SMMのモードに遷移したCPUコア１１１～１１３のうちの１つのCPUコア（例えば、CPUコア１１１）が、エラーが発生したPCIデバイスを特定する処理を行う。

[0028] 具体的には、例えばCPUコア１１１が、PCIバス・ツリーに含まれるPCIデバイスを、例えばルートブリッジ装置４１１、４１２から探索して、接続されている全てのPCIデバイスを特定する。PCIデバイスの探索は、例えば、ルートブリッジ装置４１１、４１２に割り振られたバス番号から開始して、順にバス番号、デバイス番号、ファンクション番号を増加させていきながら、それらの番号に合致するPCIデバイスが存在するかを確認する。PCIデバイスの存在確認は、例えば、実際にそれらの番号に対応するPCIデバイスのレジスタにアクセスを試み、アクセスの応答結果から判断するというを行う。

[0029] 次に、CPUコア１１１は、全てのPCIデバイスに含まれるローカル・エラー・レジスタの値を読み出し、読み出した情報に基づいてエラーを発生させたPCIデバイスを特定する。CPUコア１１１は、特定したPCIデバイスに対して可能であればリカバリ処理を行うが、エラーがハードウェア故障に起因するエラーであればリカバリ方法がないことがほとんどであるため、システム外部の管理端末にエラー発生の通知を行う。

[0030] 図１や図２の構成例では、説明を簡単にするためにPCI周辺ブロック４００に含まれるPCIデバイスの数は少ない例となっているが、コンピュータ１０のシステム構成によっては、数十から数百の数のPCIデバイスがP

ＣＩ周辺ブロック４００に含まれる場合がある。前述のＰＣＩデバイスの探索方法は、任意のＰＣＩバス・ツリーの構成に対応できる反面、探索対象となるＰＣＩデバイスの数が増えると多くの処理時間を要する。

[0031] また、リカバリ方法がないエラーが検出された場合、ＣＰＵコア１１１とシステム外部の管理端末との通信等においても処理時間を要する。そして、１つのＣＰＵコア１１１がＰＣＩデバイスのエラーを特定する処理を行っている間、他のＣＰＵコア１１２、１１３は待たされることとなる。すなわち、ＣＰＵコア１１１がＰＣＩデバイスのエラーを特定する処理を行っている間、全てのＣＰＵコア１１１～１１３上で動作していた全てのＯＳの動作が一時停止することとなる。動作を再開可能なゲストＯＳがあった場合でも、長時間動作を一時停止させると、タイムアウト等の理由によって動作を再開できなくなる場合も生じ得る。

[0032] そのため、図２のように、故障した末端装置４３３の影響を受けずに運用を継続可能なゲストＯＳ５１２がある場合には、できる限り短時間でＵＥを発生させたＰＣＩデバイスを特定し、ゲストＯＳ５１２の動作を再開させることが望ましい。

実施例 １

[0033] 以下、図３～図１０を用いて実施例１を説明する。

[0034] 図３は、実施例１に係るコンピュータ２０のシステム構成例を示す図である。

[0035] コンピュータ２０は、ＣＰＵ１００、メモリ２００、ＰＣＩ周辺ブロック４００を有し、バス３００を介してＣＰＵ１００がメモリ２００やＰＣＩ周辺ブロック４００へのアクセスを行う。ＣＰＵ１００及びＰＣＩ周辺ブロック４００に含まれる各ＰＣＩデバイス等の同じ符号が付された構成要素については図１で説明した内容と同じであるため、説明を省略する。

[0036] 実施例１では、ＰＣＩ周辺ブロック４００に含まれるＰＣＩデバイスでエラーが発生した場合に、ＰＣＩデバイスの接続構成に基づいてエラーを発生させたＰＣＩデバイスを特定する。具体的には、まず、ルートブリッジ装置

ごとにP C Iバス・ツリーに存在するP C Iデバイスの接続構成を示す構成情報、すなわち、P C Iバス・ツリー上でのP C Iデバイスの位置情報を示すデバイス・リスト330を作成しておく。作成したデバイス・リスト330に基づいて、探索対象となるP C Iバス・ツリーに含まれる末端装置から、その末端装置のローカル・エラー・レジスタの情報を収集する。

[0037] ここで、経験的に、P C Iデバイス・ツリーに含まれるP C Iデバイスのうち、ツリー状の末端の位置にある末端装置（End Device）においてエラーが発生しやすい。これは、L A Nカード装置やR A I Dカード装置等を含む末端装置431～434の方が、ブリッジ装置421、422やルートブリッジ装置411、412等に比べて機能が複雑である分、ハードウェアの故障も発生しやすいことによるものである。

[0038] そのため、実施例1のSMMにおけるエラー処理方法では、P C Iデバイス・ツリーに含まれるP C Iデバイスのうち、エラーが発生しやすい末端装置（End Device）から優先的にエラー発生の有無を確認する。末端装置でエラーを特定できた場合には、残りのブリッジ装置等のローカル・エラー・レジスタの収集を省略できるためである。全ての末端装置のローカル・エラー・レジスタの情報を参照してもエラー発生箇所を特定できない場合、各末端装置よりもツリーの上位側に接続されたブリッジ装置等の残りのP C Iデバイスのローカル・エラー・レジスタの情報を収集し、エラーの特定を行う。

[0039] 図3におけるSMMにおけるエラー処理方法では、末端装置から優先的にエラー発生の有無を確認するために、SMM領域210に、I Oエラーハンドラ220、デバイス・リスト230、デバイス・リスト作成部241を有する。

[0040] デバイス・リスト作成部241は、SMM領域210上にデバイス・リスト330を作成する。デバイス・リスト作成部241は、例えば、コンピュータ20のシステム起動時にB I O S（Basic Input/Output System）により呼び出される。デバイス・リスト作成部241は、例えば、従前からのP C Iデバイスの探索方法によって各P C Iデバイスを探索し、P C Iデバイ

スを見つけていく。例えば、ルートブリッジ装置 4 1 1、4 1 2 に割り振られたバス番号から開始して、順にバス番号、デバイス番号、ファンクション番号を増加させていきながら、それらの番号に合致する P C I デバイスが存在するかを確認する。

[0041] デバイス・リスト作成部 2 4 1 は、P C I デバイスを見つけるごとに、見つけたデバイスに対応するエントリ（以下、「デバイス・リスト・エントリ」と言う）を作成する。そして、作成したデバイス・リスト・エントリをデバイス・リスト 2 3 0 に追加する。そして、デバイス・リスト作成部 2 4 1 は、作成したデバイス・リスト 2 3 0 を、例えばメモリ 2 0 0 中の SMM 領域 2 1 0 に格納する。デバイス・リスト 2 3 0 の説明は図 5 を用いて後述する。

[0042] P C I デバイスにエラーが発生した場合に SMM のモードで呼び出される I O エラーハンドラ 2 2 0 は、探索範囲選択部 2 2 1 と、エラー・レジスタ値収集部 2 2 2 と、エラー箇所特定部 2 2 3 と、エラー対処部 2 2 4 とを有する。探索範囲選択部 2 2 1 は、デバイス・リスト 2 3 0 の初期化と、探索する P C I ルートブリッジ装置の選択を行う。エラー・レジスタ値収集部 2 2 2 は、デバイス・リスト 2 3 0 に基づいて P C I デバイスからエラー・レジスタの情報を収集する。エラー・レジスタ値収集部 2 2 2 によるエラー・レジスタ値の収集は、後述の複数の C P U コアを用いたエラー・レジスタの情報収集でも実行される。

[0043] エラー箇所特定部 2 2 3 は、読みだしたエラー・レジスタの情報に基づいてエラーが発生したデバイスを特定する。エラー対処部 2 2 4 は、エラーを発生させたデバイスが特定された場合に、エラーを発生させたデバイスに対して、リカバリが可能であればリカバリ処理を行う。エラーを発生させたデバイスのリカバリ処理を行えない場合も多く、この場合には、外部の管理装置等に対してエラーの情報の通知のみを行う。

[0044] SMM 領域 2 1 0 に含まれるデバイス・リスト作成部 2 4 1、I O エラーハンドラ 2 2 0 に含まれる各機能ブロックは、それぞれ C P U 1 0 0 に含ま

れるCPUコア111～113によって実行されることによって、各機能を実現する。

[0045] PCIデバイスにエラーが発生した場合、各CPUコア111～113がSMMのモードに遷移し、IOエラーハンドラ220に含まれるプログラムが実行される。具体的には、IOエラーハンドラ220に含まれる探索範囲選択部221、エラー・レジスタ値収集部222、エラー箇所特定部223によってデバイスのエラーが特定される。その際、予め作成しておいたデバイス・リスト230を用いてPCIバス・ツリーに含まれる末端装置から優先的にエラーの情報を収集する。PCIデバイス・ツリーに含まれる末端装置（End Point）から優先的にエラー発生の有無を特定することで、より効率的にエラーが発生したPCIデバイスを特定でき、コンピュータに含まれる周辺装置の故障を短時間で特定することができる。

[0046] 図4は、末端装置とブリッジ装置にエラーが発生した例を示す図である。図4では、ブリッジ装置424と末端装置435の2か所にエラーが発生した場合の例である。ブリッジ装置424にエラーが発生した場合、そのエラーの度合いによって影響範囲が異なってくる。エラーの度合いが大きい場合にはブリッジ装置424配下のデバイスである末端装置437にアクセスできなくなる。そのため、本実施例では、レジスタアクセスを行うことができない末端装置を発見した場合には、末端装置のエラー情報収集の後に、アクセスできなかった末端装置が接続されているブリッジ装置のエラー情報も併せて収集する。そのようにすることで、ブリッジ装置でのエラーも見逃さずに収集できる。

[0047] 一方、ブリッジ装置424のエラーが、ブリッジ装置424配下のデバイスにアクセスできる程度の軽度のエラーである場合、先に末端装置435のエラーを発見することによってブリッジ装置424のエラーを一度は見過ごすことになる。しかし、末端装置435のエラー処理を行った後もブリッジ装置424のエラーの状態はクリアされていないため、SMMのモードを抜けた後にもう一度SMIによる割り込みが上がり、SMMでのエラー特定処

理が行われることによって検出される。この場合、一度見過ごすエラーはエラーの症状が軽症であることと、同時に2か所でエラーが発生すること自体が非常に稀であること、一度見過ごしても再度検出可能であることから、一時的に見過ごしたブリッジ装置424でのエラーの影響は小さい。

[0048] 図5は、デバイス・リスト230の例を示す図である。ルートブリッジ装置を先頭に、ルートブリッジ装置配下に存在するPC Iデバイスに対応する「デバイス・リスト・エントリ」がリスト構造で接続された図で示されている。デバイス・リスト230の内容の例については、図14を用いて後述する。

[0049] 図5において各エントリに表示されている「ROOT BRIDGE01」等のラベルは、図1や図2中に表示されたラベルに対応する。例えば、「ROOT BRIDGE01」のラベルが付されたエントリは、図1等で示されるルートブリッジ装置411に対応する。ルートブリッジ装置配下の各デバイスには「Next device Link」の矢印で接続され、「ROOT BRIDGE01」から「ROOT BRIDGE02」へは、「Next root Link」の矢印で接続されている。

[0050] 図6は、デバイス・リスト230に含まれるデバイス・リスト・エントリの例を示す図である。各PC Iデバイスのデバイス・リスト・エントリには、図6の左側の枠内に示された各エントリの情報が設定される。デバイス・リスト・エントリの各要素を一例を以下に説明する。

[0051] 「ROOT FLAG」は、エントリに対応するデバイスがルートブリッジ装置である場合に値1が設定される。エントリに対応するデバイスがルートブリッジ装置以外の場合には、値0が設定される。「ROOT FLAG」は、デバイス・リスト・エントリの作成又は更新時に値が設定される。

[0052] 「ROOT DATA POINTER」は、ルートブリッジ装置用の拡張データである「Root Data」へのポインタである。「Root Data」については後述する。

[0053] 「NEXT DEVICE LINK」は、同じルートブリッジ装置の配下にある次の「デバイス・リスト・エントリ」を指す。「NEXT DEVICE LINK」は、デバイス・リスト・エントリの作成又は更新時に値が設定される。

- [0054] 「LOCATION」は、デバイスの位置情報を示す。デバイスの位置情報としては、例えば、セグメント番号、バス番号、デバイス番号、ファンクション番号がある。「LOCATION」は、デバイス・リスト・エントリの作成又は更新時に値が設定される。
- [0055] 「END DEVICE FLAG」は、エントリに対応するデバイスが末端装置 (End Device) である場合に値 1 が設定される。エントリに対応するデバイスが末端装置でない場合には、値 0 が設定される。「END DEVICE FLAG」は、デバイス・リスト・エントリの作成又は更新時に値が設定される。
- [0056] 「DISABLE FLAG」は、エラーの有無の探索対象としないことを示す情報であり、「DISABLE FLAG」に値 1 が設定されたデバイスは、エラーの有無の探索対象とされない。
- [0057] 例えば、エラー対処部 224 で、エラーが発生したデバイスを次回から使用しないように対処した場合、この「DISABLE FLAG」に値 1 を設定することで、次のエラー探索時に探索の対象から除外される。「DISABLE FLAG」は、デバイス・リスト・エントリの作成又は更新時に値 0 で初期化される。
- [0058] 「READ-FAIL FLAG」は、P C I デバイスのレジスタを読み出せない場合に、値 1 に設定される。例えば、ブリッジ装置にエラーが発生して、その配下のデバイスのレジスタが読めない場合にも、レジスタの読み出しを行えなかったデバイスのエントリの「READ-FAIL FLAG」に値 1 が設定される。「READ-FAIL FLAG」は、探索範囲選択部 221 により値 0 に初期化される。
- [0059] 「COMPLETE FLAG」は、デバイスからのエラー・レジスタの読み出しが完了すると、値 1 が設定される。デバイスからのエラー・レジスタの読み出しが完了するまでの間は値 0 となる。「COMPLETE FLAG」は、探索範囲選択部 221 により値 0 に初期化され、エラー・レジスタ値収集部 222 により値 1 が設定される。
- [0060] 「ERROR FLAG」は、エントリに対応するデバイスにエラーがある場合に値 1 が設定される。「ERROR FLAG」は、探索範囲選択部 221 により、P C I デバイスのエラーの探索を行う度に値 0 に初期化される。

- [0061] 「UPDATING LOCK」は、エントリに対応するデバイスへのアクセスを排他的に行うためのスピンロック変数である。「UPDATING LOCK」は、エラー・レジスタ値収集部222が、デバイスのエラー・レジスタの読み出しを行う前にロックを取得し、完了後にロックを開放する際に使用される。スピンロック変数は、例えば、値1が設定されている場合に、エントリに対応するデバイスをアクセスする可能性のあるCPUコア111～113のいずれかがロックを取得していることを示す。あるCPUコアが「UPDATING LOCK」の値を読み出した際に値0であれば、他のCPUコアがロックを取得していないことになるので、ロックの取得に成功し、値1を設定する。一方、あるCPUコアが「UPDATING LOCK」の値を読み出した際に値1であれば、他のCPUコアがロックを取得していることになるので、ロックの取得に失敗する。この場合、一定期間経過後に、再度ロックの取得を試みることになる。
- [0062] 「ERROR REGISTERS DUMP」は、読み出されたエラー・レジスタの値を格納する領域である。読み出し対象となるエラー・レジスタの数は、システム構成に依存するため、「ERROR REGISTERS DUMP」の領域のサイズは任意のサイズとする。
- [0063] 「ERROR REGISTERS DUMP」は、探索範囲選択部221によって値0等の初期値に初期化され、エラー・レジスタ値収集部222によって値が設定される。
- [0064] 次に「Root Data」の各エントリを説明する。「Root Data」は、「NEXT ROOT LINK」と、「GLOBAL ERROR FLAG」と、「DETECTED ERROR FLAG」と、「CHECK ALL FLAG」とを含む。
- [0065] 「NEXT ROOT LINK」は、次のルートブリッジ装置のデバイス・リスト・エントリを指す。次のルートブリッジ装置のエントリがない場合には、値0が設定される。「NEXT ROOT LINK」は、デバイス・リスト・エントリの作成又は更新時に値が設定される。
- [0066] 「GLOBAL ERROR FLAG」は、ルートブリッジ装置の配下のデバイスにエラーが発生した場合に値1が設定される。「GLOBAL ERROR FLAG」は、探索範囲選

択部 2 2 1 により、P C I ルートブリッジ装置の選択を行うたびに設定される。

[0067] 「ERROR DETECTED COUNT」は、ルートブリッジ装置の配下のデバイスにエラーが発見されるたびに、値が 1 ずつ増加される。「ERROR DETECTED COUNT」は、探索範囲選択部 2 2 1 によって、P C I デバイスのエラーの探索を行う度に値 0 に初期化される。

[0068] 「CHECK ALL FLAG」は、末端装置 (End Device) だけでなく、ブリッジ装置も含めたエラー箇所の特定が必要な場合に、値 1 が設定される。「CHECK ALL FLAG」は、探索範囲選択部 2 2 1 により、P C I デバイスのエラーの探索を行う度に値 0 に初期化される。

[0069] 図 7 は、ローカル・エラー・レジスタ及びグローバル・エラー・レジスタの例を示す図である。図 7 (A) は、ローカル・エラー・レジスタ 6 1 0 の例を示す。ローカル・エラー・レジスタ 6 1 0 は、各 P C I デバイ스에設けられ、各 P C I デバイスにおいてエラーが発生した場合に、発生したエラーに関する情報を記憶するレジスタである。図 7 (A) において、ローカル・エラー・レジスタ 6 1 0 は、複数の U E ステータス・レジスタ 6 1 1 - 1 ~ 6 1 1 - M (M は自然数) と、複数の C E ステータス・レジスタ 6 1 2 - 1 ~ 6 1 2 - L (L は自然数) とを有する。

[0070] U E ステータス・レジスタ 6 1 1 - 1 ~ 6 1 1 - M は訂正不可能なエラー (U E : Uncorrectable Error) の発生の有無を示し、N (N は自然数) 個の各ビットがエラーの種別 (Type) ごとのエラーの発生の有無を表す。同様に、C E ステータス・レジスタ 6 1 2 - 1 ~ 6 1 2 - L は訂正可能なエラー (C E : Correctable Error) の発生の有無を示す。C E ステータス・レジスタ 6 1 2 - 1 ~ 6 1 2 - L も、例えば、N 個の C E の発生の有無を示すビットを含むように実装することができる。ローカル・エラー・レジスタ 6 1 0 に記憶された情報は、エラー・レジスタ値収集部 2 2 2 によって読み出される。

[0071] 図 7 (A) において、U E 及び C E の各エラーの発生の有無を示すレジス

タを複数設けているが、各レジスタの数であるM、Lは、P C Iデバイスにおいて発生するエラーの発生原因や種類に応じて任意の数とすることができる。

[0072] なお、図7（A）では、ローカル・エラー・レジスタは各種エラー情報を保持する例を示しているが、デバイスの他のステータス情報も含めて保持するステータス・レジスタの一部の情報としてエラー情報を保持させることもできる。その場合、エラー・レジスタ値収集部222は、各P C Iデバイスのステータス・レジスタからエラーに関連する情報の読み出しを行う。

[0073] 図7（B）は、グローバル・エラー・レジスタ620の例を示す図である。グローバル・エラー・レジスタ620は、各ルートブリッジ装置に設けられるレジスタであり、ルートブリッジ装置配下のブリッジ装置や末端装置のいずれかでエラーが発生したか否かを示す情報を保持するレジスタである。

[0074] 図7（B）において、グローバル・エラー・レジスタ620は、複数のグローバルU Eステータス・レジスタ621-1～621-Y（Yは自然数）と、複数のグローバルC Eステータス・レジスタ622-1～622-Z（Zは自然数）とを有する。グローバルU Eステータス・レジスタ621-1～621-Y及びグローバルC Eステータス・レジスタ622-1～622-Zでも、ローカル・エラー・レジスタ610の場合と同様、N種類エラーの種別ごとのエラーの発生の有無を表すビットを含むように実装できる。

[0075] 例えば、ルートブリッジ装置配下のどこかの末端装置で、U E Type(1)のエラーが起これば、ルートブリッジ装置のグローバルU Eステータス・レジスタ621-1～621-YのいずれかのレジスタのU E Type(1)に対応するビットが設定される。なお、図7（B）の例においても、P C Iデバイスの実装に応じて、Y、Zの数は任意の数とすることができる。

[0076] 図8は、実施例1に係るI Oエラーハンドラ処理のフローの例を示す図である。まず、S101の処理において、エラーのあるデバイスを含むP C Iバス・ツリーのルートブリッジ装置を特定する。具体的には、探索範囲選択部221が、デバイス・リスト230の初期化と、エラーが発生したデバイ

スを含むP C Iバス・ツリーのルートブリッジ装置を特定する。S 1 0 1の処理の詳細については、図9を用いて後述する。

[0077] S 1 0 2では、P C Iデバイスのエラー箇所を特定する処理を継続するかどうかの判定を行う。デバイス・リスト2 3 0によって特定した全てのルートブリッジ装置のエントリの「GLOBAL ERROR FLAG」の値が「0」の場合（S 1 0 2でN O）、P C Iデバイスのエラーが存在しないため、S 1 0 8に進む。一方、いずれかのルートブリッジ装置のエントリの「GLOBAL ERROR FLAG」の値が「1」の場合（S 1 0 2でY E S）、S 1 0 3に進む。

[0078] S 1 0 3では、エラー・レジスタ値収集部2 2 2が、「GLOBAL ERROR FLAG」の値が「1」のルートブリッジ装置の配下のデバイスのローカル・エラー・レジスタの値を収集する。デバイスのローカル・エラー・レジスタの値を収集する際、エラー・レジスタ値収集部2 2 2は、末端装置のローカル・エラー・レジスタから優先的にエラー情報の収集を行う。ここで、予めデバイス・リスト作成部2 4 1によって作成したデバイス・リスト2 3 0を用いて末端装置（End Device）を特定し、末端装置から優先的にエラー情報の収集を行う。

[0079] S 1 0 3におけるエラー・レジスタ値収集部2 2 2による処理は、後述の図1 0又は図1 2の処理フローを用いて説明する。後述の図1 2は複数のC P Uコアでエラー・レジスタ値収集部2 2 2を実行できるようにするための処理フローを図示したものだが、1つのC P Uコアで実施する場合は、S 4 0 1の処理を省略すればよい。なお、S 1 0 3又は図1 2のS 4 0 2から図1 0の処理フローの呼び出しを行う際には、引数として「モード＝1」を指定して呼び出しを行う。詳細については後述するが、図8の全体フローから図1 2のフローが呼び出される場合、図8のS 1 0 3又はS 1 0 7から図1 2の処理が呼び出される。S 1 0 3とS 1 0 7の違いは、「モード」に指定する値が異なる点にある。図1 2のS 4 0 2とS 4 0 5から図1 0の処理が呼び出される際には、図8のS 1 0 3又はS 1 0 7で図1 2の処理を呼び出す際に指定された「モード」の値が引き継がれる。

- [0080] S 1 0 4 の処理では、エラー箇所特定部 2 2 3 が S 1 0 3 の処理で収集した末端装置 (End Device) のローカル・エラー・レジスタの情報に基づいてエラーを発生させた P C I デバイスの特定を行う。エラーを発生させたデバイスに対しては、そのデバイスに対応するデバイス・リスト・エントリの「ERROR FLAG」に値 1 を設定すると共に、ルートブリッジ装置のデバイス・リスト・エントリの「ERROR DETECTED COUNT」の値を 1 つ増加させる。
- [0081] S 1 0 5 では、デバイスのエラー箇所の特定処理を継続するかどうかを判定する。具体的には、デバイス・リスト 2 3 0 内の「GLOBAL ERROR FLAG」の値が「1」のルートブリッジ装置について、「ERROR DETECTED COUNT」および「CHECK ALL FLAG」の値に基づいてエラー箇所の特定を継続するかどうかを判定する。「ERROR DETECTED COUNT」のカウント値から末端装置のいずれかのデバイスにおいてエラーが発生したことを特定できた場合 (S 1 0 5 で Y E S)、エラー情報の収集処理を終了して S 1 0 8 のエラー対処処理に進む。ただし、「CHECK ALL FLAG」の値が 1 の場合には、ブリッジ装置にもエラーがある可能性があるため、S 1 0 6 に進む。
- [0082] 後で説明する「GLOBAL ERROR FLAG」の値が「1」のルートブリッジ装置において「ERROR DETECTED COUNT」の値が「0」の場合、エラー箇所を特定できていないことになるので (S 1 0 5 で N O)、S 1 0 6 に進む。また、「CHECK ALL FLAG」に「1」が設定されている場合にもブリッジ装置のエラーを確認する必要があるため (S 1 0 5 で N O)、S 1 0 6 に進む。
- [0083] S 1 0 6 では、図 1 0 又は図 1 2 のエラー・レジスタ値収集部 2 2 2 による処理を、「モード＝2」の引数を指定して呼び出す。「モード＝2」を指定して図 1 0 又は図 1 2 の処理フローを呼び出すことで、ローカル・エラー・レジスタの値を収集していないデバイスを対象にローカル・エラー・レジスタの情報収集が行われる。図 1 0 又は図 1 2 の処理の詳細については後述する。
- [0084] S 1 0 7 の処理では、エラー箇所特定部 2 2 3 が S 1 0 6 の処理で収集したブリッジ装置等のローカル・エラー・レジスタの情報に基づいてエラーを

発生させたP C Iデバイスの特定を行う。最終的にS 1 0 7の処理で、ブリッジ装置のエラーも含めてエラー箇所を特定する。本実施例では、デバイス・リスト2 3 0を利用することで、末端装置を優先して段階的にエラー・レジスタの情報収集を行うことが可能となる。

[0085] S 1 0 1～S 1 0 7の処理によってエラー箇所が特定できると、S 1 0 8においてエラーの対処を行う。S 1 0 8では、エラー対処部2 2 4により見つかったエラーに応じた処理を行う。

[0086] 図9は、探索範囲選択処理のフローの例を示す図である。探索範囲選択部2 2 1は、まずS 2 0 1においてデバイス・リスト2 3 0の初期化を行う。デバイス・リスト2 3 0の初期化処理では、全てのデバイス・リスト・エントリに対して、「READ-FAIL FLAG」、「COMPLETE FLAG」、「ERROR FLAG」及び「ERROR REGISTERS DUMP」の領域を値0で初期化する。また、「Root Data」については、「GLOBAL ERROR FLAG」、「ERROR DETECTED COUNT」、「CHECK ALL FLAG」を値0で初期化する。

[0087] 次に、探索範囲選択部2 2 1は、S 2 0 2において、デバイス・リスト2 3 0からルートブリッジ装置のデバイス・リスト・エントリを取り出す。ルートブリッジ装置のデバイス・リスト・エントリは、先頭のルートブリッジ装置のエントリから「NEXT ROOT LINK」をたどることによって取り出していく。ルートブリッジ装置のデバイス・リスト・エントリの中で、エラーの有無の確認を行っていないルートブリッジ装置がなければ（S 2 0 3でNO）図9の処理を終了し、ルートブリッジ装置のデバイス・リスト・エントリがあれば、S 2 0 4で「DISABLE FLAG」の設定の有無を判定する。

[0088] 取り出したルートブリッジ装置の「DISABLE FLAG」が値0でなければ（S 2 0 4でNO）エラーの有無の探索対象とされないため、S 2 0 2の処理へ戻る。取り出したルートブリッジ装置の「DISABLE FLAG」が値0であれば、S 2 0 5の処理に進む。

[0089] S 2 0 5では、「LOCATION」の情報に基づいてルートブリッジ装置のグローバル・エラー・レジスタとローカル・エラー・レジスタを読み出し、読み

出した結果を「ERROR REGISTERS DUMP」の領域に記憶し、「COMPLETE FLAG」を値1で設定する（S206）。S205でルートブリッジ装置のグローバル・エラー・レジスタ及びローカル・エラー・レジスタの情報を正常に読み出せなかった場合（S207でNO）、「READ-FAIL FLAG」に値1を設定して（S208）、S213の処理に進む。

[0090] S205でルートブリッジ装置のグローバル・エラー・レジスタ及びローカル・エラー・レジスタの情報を正常に読み出した場合（S207でYES）、S209の処理に進む。グローバル・エラー・レジスタから読み出した情報にエラーがあったことを示す情報が含まれていない場合（S209でNO）、処理は、S202に戻って次のルートブリッジ装置のエラー判定処理に進む。

[0091] 一方、グローバル・エラー・レジスタから読み出した情報にエラーがあったことを示す情報が含まれている場合（S209でYES）、「GLOBAL ERROR FLAG」に値1を設定する（S210）。そして、ルートブリッジ装置自体にエラーがあるかどうかを読み出したローカル・エラー・レジスタの情報に基づいて判定する（S211）。

[0092] ルートブリッジ装置自体にエラーがなければ（S212でNO）、S202の処理に戻り、次のルートブリッジ装置について探索を行う。ルートブリッジ装置のローカル・エラー・レジスタの情報からルートブリッジ装置自体にエラーがあると判定した場合（S212でYES）、ルートブリッジ装置自体のデバイス・リスト・エントリの「ERROR FLAG」を値1に設定する（S213）。そして、「ERROR DETECTED COUNT」の値を1つ増やす（S214）。そして、S202の処理に戻り、次のルートブリッジ装置について探索を行う。

[0093] 以上の図9の処理によってデバイス・リスト230の初期化と、エラー情報を有するルートブリッジ装置の特定を行う。

[0094] 図10は、エラー・レジスタ値収集処理の例を示す図である。図10の各処理は、エラー・レジスタ値収集部222によって実行される処理であり、

デバイス・リスト230に基づいてP C Iデバイスからエラー・レジスタの情報を収集する。図10の処理の呼び出しを行う際には引数によってモード(MODE)の変数が指定される。引数によって指定されるモードの値が1の場合、末端装置(End Device)のみをエラー・レジスタの情報の収集対象とし、モードの値が2の場合は全P C Iデバイスを収集対象とする。

[0095] S301の処理では、デバイス・リスト230から、ルートブリッジ装置のデバイス・リスト・エントリを取り出す。S301の処理において、エラー・レジスタ値収集部222は、図10の処理フローを呼び出された直後は先頭のルートブリッジ装置のエントリを取り出す。S303、S304、S306の処理から戻って再度S301の処理を行う際には、「NEXT ROOT LINK」をたどることによってルートブリッジ装置のエントリを取り出す。

[0096] エラー・レジスタ値収集処理を行っていないルートブリッジ装置のデバイス・リスト・エントリがない場合や、ルートブリッジ装置のデバイス・リスト・エントリが取り出せない場合には(S302でNO)、処理を終了する。エラー・レジスタ値収集処理を行っていないルートブリッジ装置のデバイス・リスト・エントリがある場合(S302でYES)、「DISABLE FLAG」の値が0であるかどうかを確認する(S303)。

[0097] 「DISABLE FLAG」の値が0でない場合(S303でNO)、処理はS301に戻り、次のルートブリッジ装置のデバイス・リスト・エントリの取り出し処理を行う。「DISABLE FLAG」の値が0の場合(S303でYES)、S304の処理に進む。

[0098] S304の処理は、ルートブリッジ装置のデバイス・リスト・エントリで「GLOBAL ERROR FLAG」の値が1であるかどうかを確認することにより、ルートブリッジ装置の配下のデバイスにエラーがあるかどうかを判定する。「GLOBAL ERROR FLAG」の値が1でない場合には(S304でNO)、ルートブリッジ装置の配下のデバイスにエラーが見つからなかったことになるので、処理はS301に戻り、次のルートブリッジ装置のデバイス・リスト・エントリの取り出し処理に進む。

- [0099] 「GLOBAL ERROR FLAG」の値が1の場合（S304でYES）、S301でエントリを取り出したルートブリッジ装置を始点とするPCIバス・ツリー上のいずれかのデバイスにエラーがあることになる。そのため、ルートブリッジ装置の配下のデバイスのデバイス・リスト・エントリを取り出す（S305）。ルートブリッジ装置の配下のデバイスのデバイス・リスト・エントリがない場合（S306でNO）、処理はS301に戻り、次のルートブリッジ装置のデバイス・リスト・エントリの取り出し処理に進む。
- [0100] ルートブリッジ装置の配下のデバイスのデバイス・リスト・エントリがある場合（S306でYES）、対象デバイスの「DISABLE FLAG」の設定の有無を確認する（S307）。「DISABLE FLAG」の値が0でなければ（S307でNO）、エラー・レジスタ値の収集の対象外とし、S305の処理に戻り、ルートブリッジ装置の配下の次のデバイスのデバイス・リスト・エントリを取り出す。
- [0101] 対象デバイスの「DISABLE FLAG」の値が0の場合（S307でYES）、モードの設定が1（末端装置のみが対象）でかつ対象デバイスの「END DEVICE FLAG」が1であるか、あるいは、モードの設定が2（全てのデバイスが対象）であるかを確認する（S308）。S308のいずれの場合にも該当しない場合（S308でNO）、エラー・レジスタ値の収集の対象外となるので、S305の処理に戻る。
- [0102] S308のいずれかの場合に該当する場合（S308でYES）、対象デバイスの「COMPLETE FLAG」の値が0であるかどうかを確認する（S309）。「COMPLETE FLAG」の値が0でなければ（S309でNO）既に対象デバイスのエラー・レジスタの値を収集しているため、S305の処理に戻る。
- [0103] 「COMPLETE FLAG」の値が0であれば（S309でYES）、排他処理をするための「UPDATING LOCK」の取得ができたかどうかを確認する（S310）。「UPDATING LOCK」の取得ができない場合（S310でNO）、S305の処理に戻り、「UPDATING LOCK」の取得ができた場合（S310でYES）、S311の処理に進む。なお、「COMPLETE FLAG」の値の確認及び「UPDATING

LOCK」を取得できたかどうかの確認は、特に後述の実施例 2 において意味を有するので、実施例 2 において具体的に述べる。

[0104] S 3 1 1 の処理は、対象デバイスのエラー・レジスタ値、すなわち、ローカル・エラー・レジスタの情報を読み出し、対象デバイスの「ERROR REGISTERS DUMP」領域に記憶する。ローカル・エラー・レジスタの情報を正常に読み出せなかった場合（S 3 1 2 で NO）、対象デバイスの「READ-FAIL FLAG」を値 1 に設定する（S 3 1 3）。

[0105] なお、先にも述べたとおり、1つのCPUコアでエラー・レジスタ値収集部 2 2 2 の処理を実行する場合でも図 1 2 の S 4 0 2、S 4 0 3 の処理を呼び出すことができ、その場合、後述の S 4 0 3 の処理で「CHECK ALL FLAG」が更新される。しかし、1つのCPUコアでエラー・レジスタ値収集部 2 2 2 の処理を実行する場合には、後述の S 4 0 3 の処理を省略して、図 1 0 の S 3 1 3 において「READ-FAIL FLAG」を値 1 に設定するとともに、「CHECK ALL FLAG」も値 1 に設定するように実装することができる。

[0106] S 3 1 2 で YES の場合、又は、S 3 1 3 の処理をした後、対象デバイスの「COMPLETE FLAG」を値 1 に設定し（S 3 1 4）、「UPDATING LOCK」を解除して排他処理を終了し（S 3 1 5）、S 3 0 5 の処理に戻る。

[0107] S 3 0 6 において、ルートブリッジ装置の配下のデバイスで、ローカル・エラー・レジスタからの情報の読み出しが終わっていないデバイスがなくなると（S 3 0 6 で NO）、S 3 0 1 に戻り、以下、同様に処理を行っていく。

[0108] 以上説明した通り、実施例 1 に係る情報処理装置及びエラー処理方法では、PCI デバイスの位置情報を含むデバイス・リスト 2 3 0 を作成する。そして、作成したデバイス・リスト 2 3 0 の情報に基づいて、PCI デバイス・ツリーに含まれる末端装置（End Point）から優先的にエラー発生の有無を特定する。デバイス・リスト 2 3 0 を参照することで、エラーの特定処理を行う際にデバイスの探索を行う必要がなく、末端装置に短時間でアクセスすることができる。

[0109] また、PCIデバイス・ツリーに含まれる末端装置から優先的にエラー発生の有無を特定することで、より効率的にエラーが発生したPCIデバイスを特定でき、コンピュータに含まれる周辺装置の故障を短時間で特定することができる。

実施例 2

[0110] 実施例2では、複数のCPUコア111～113を用いて、SMMのモードにおいてPCIデバイスのエラーの特定を行う例を説明する。また、PCIデバイスの追加または削除を行った際の処理についても併せて説明する。

[0111] 図11は、実施例2に係るコンピュータ30のシステム構成例を示す図である。図11のシステム構成例において、メモリ200には、ハイパーバイザ250の処理を行うためのプログラムが格納される。また、メモリ200に含まれるSMM領域210には、図3で示される構成要素に加えて、デバイス・リスト更新部242、デバイス変更情報I/F部243を有する。

[0112] ハイパーバイザ250は、CPUコア111～113上で稼働する仮想マシンを管理する。また、ハイパーバイザ250は、ゲストOS511、512のPCIホット・プラグを検出する。ハイパーバイザ250は、デバイス変更通知部251を有し、PCIホット・プラグによるPCIデバイスの追加や削除を検出すると、デバイス変更情報I/F部243を介して、デバイス・リスト更新部242にデバイスの変更があったことを通知する。

[0113] デバイス・リスト更新部242は、デバイス変更通知部251から通知された情報に基づいてデバイス・リスト230の内容を更新する。デバイス・リスト更新部242の処理は、例えば、PCIホット・プラグ等によってデバイスの追加や削除が行われた際に実行される処理である。

[0114] PCIデバイスの追加や削除があると、デバイス変更通知部251は、デバイス変更情報I/F部243に含まれる記憶領域に、変更のあったデバイスの情報を設定した後、各CPUコア111～113に対してSMIを上げる。各CPUコア111～113は、SMIの割り込みを受信するとSMMのモードに移行し、デバイス・リスト更新部242による処理を呼び出す。

デバイス・リスト更新部242は、デバイス変更情報I/F部243の記憶領域に設定された情報を参照して、PCIデバイスの追加や削除があるか否かを判断し、デバイス・リスト230の内容を更新する。

[0115] 図11のシステム構成例では、図2、図3に示される構成要素に加えてROM (Read Only Memory) 280が追加されている。ROMには、デバイス・リスト作成部241の処理を行うプログラムや、IOエラーハンドラ220の各処理を行うプログラム等が格納される。ROMに格納されたプログラムは、コンピュータ30のシステム起動時にメモリ200上の領域に展開され、各CPUコア111~113によって実行される。

[0116] デバイス・リスト作成部241は、メモリ200上に展開されると、コンピュータ30のシステムに含まれるPCIデバイスを探索する。PCIデバイスの探索の結果、PCIデバイスを見つけるたびにデバイス・リスト230に見つけたデバイスに対応するデバイス・リスト・エントリを作成する。デバイス・リスト作成部241によるPCIデバイスの探索とデバイス・リスト・エントリの作成が完了すると、図5に示されるデバイス・リスト230が作成される。

[0117] コンピュータ30のシステム運用中に、例えば図2に示すように末端装置433でエラーが発生したとするとコンピュータ30内の各CPUコア111~113にSMIの割り込みが上がり、各CPUコア111~113はSMMのモードに移行する。各CPUコア111~113がSMMのモードに移行した後にエラーが発生したPCIデバイスの探索が行われるが、エラーを早期に発見するには、1つのCPUコアで探すよりも、複数のCPUコアで分担した方がより早く発見され得る。

[0118] 前述のように、デバイス・リスト230には、各PCIデバイスからのエラー・レジスタの情報収集状況を判断するための「COMPLETE FLAG」と、各CPUコア111~113が排他的に処理を行うための「UPDATING LOCK」のエントリを含む。これらの「COMPLETE FLAG」と「UPDATING LOCK」を用いることによって、複数のCPUコア111~113を用いて並行してエラー・レ

ジスタの情報を収集させることができる。これによって、多くのＰＣＩデバイスからエラー・レジスタを収集する時間を短縮することができる。

[0119] 図１２は、複数のＣＰＵコアを用いてエラー・レジスタ値収集部２２２による処理の全体フローの例を示す図である。図１２の左側のＳ４０１～Ｓ４０３は代表ＣＰＵコアによる処理で、Ｓ４０４～Ｓ４０６はその他のＣＰＵコアによる処理を示す。ここで、代表コアは、例えば、複数のＣＰＵコア１１１～１１３のうちの１つのＣＰＵコアであり、他のＣＰＵコアによるエラー・レジスタ収集処理を制御する。

[0120] まず、モード（Mode）の引数を指定して図１２の処理が呼び出されると、代表ＣＰＵコアは、他のＣＰＵコアに対してエラー・レジスタ値を収集する処理を依頼する要求を送信する（Ｓ４０１）。ここで、代表ＣＰＵコアは、複数の他のＣＰＵコアに対してエラー・レジスタ値を収集する処理を依頼する要求を送信してもよい。そして、代表ＣＰＵコア自身もエラー・レジスタ値を収集する処理を行う（Ｓ４０２）。Ｓ４０２のエラー・レジスタ値を収集する処理では、モード（Mode）の引数を指定して図１０の処理を呼び出す。

[0121] 一方、他のＣＰＵコアは、代表ＣＰＵコアからのエラー・レジスタ値を収集する処理を依頼する要求を待ち（Ｓ４０４）、要求を受信すると、エラー・レジスタ値を収集する処理を行う（Ｓ４０５）。Ｓ４０５のエラー・レジスタ値を収集する処理でも、要求を受信した他のＣＰＵコアが、モード（Mode）の引数を指定して図１０の処理を呼び出す。他のＣＰＵコアは、Ｓ４０５のエラー・レジスタ値の読み出し処理が完了すると、代表ＣＰＵコアからの次の要求が送信されてくるのを待つ（Ｓ４０６）。

[0122] 代表コアは、自身が行うエラー・レジスタ値の収集処理が終わると、他の複数のＣＰＵコアによるエラー・レジスタ値収集処理の終了を待つ（Ｓ４０３）。具体的には、代表ＣＰＵコアは、モード（Mode）を指定して図１３の処理を呼び出す。

[0123] 図１３は、図１２における代表ＣＰＵコアが他の複数のＣＰＵコアによる

エラー・レジスタ値収集処理の終了を待つ処理フローの例を示す図である。
図 1 3 の処理は、図 1 2 の S 4 0 3 から呼ばれる処理を図示したものである。
。

代表 C P U コアは、まず、ルートブリッジ装置のデバイス・リスト・エントリをデバイス・リスト 2 3 0 から取り出す (S 5 0 1)。S 5 0 1 でのルートブリッジ装置のデバイス・リスト・エントリの取り出しは、最初は先頭のルートブリッジ装置のエントリを取り出し、2 回目以降は「NEXT ROOT LINK」をたどることによって取り出すことができる。

[0124] S 5 0 2 において、エラー・レジスタ値の収集状況を確認できていないルートブリッジ装置のデバイス・リスト・エントリがない場合や、ルートブリッジ装置のデバイス・リスト・エントリが取り出せない場合には (S 5 0 2 で N O)、処理を終了する。エラー・レジスタ値の収集結果を確認できていないルートブリッジ装置のデバイス・リスト・エントリがある場合 (S 5 0 2 で Y E S)、次にルートブリッジ装置の「DISABLE FLAG」の値が 0 であるかどうかを確認する (S 5 0 3)。

[0125] 「DISABLE FLAG」の値が 0 でない場合 (S 5 0 3 で N O)、処理は S 5 0 1 に戻り、次のルートブリッジ装置のデバイス・リスト・エントリの取り出し処理を行う。「DISABLE FLAG」の値が 0 の場合 (S 5 0 3 で Y E S)、S 5 0 4 の処理に進む。

[0126] S 5 0 4 の処理は、ルートブリッジ装置のデバイス・リスト・エントリで「GLOBAL ERROR FLAG」の値が 1 であるかどうかを確認することにより、ルートブリッジ装置の配下のデバイスにエラーがあるかどうかを判定する。「GLOBAL ERROR FLAG」の値が 1 でない場合には (S 5 0 4 で N O)、ルートブリッジ装置の配下のデバイスにエラーが見つからなかったことになるので、処理は S 5 0 1 に戻り、次のルートブリッジ装置のデバイス・リスト・エントリの取り出し処理に進む。

[0127] 「GLOBAL ERROR FLAG」の値が 1 の場合 (S 5 0 4 で Y E S)、S 5 0 1 でエントリを取り出したルートブリッジ装置を始点とする P C I バス・ツリー

上のいずれかのデバイスにエラーがあることになる。そのため、ルートブリッジ装置の配下のデバイスからのエラー・レジスタ値の収集の状況を確認するために、ルートブリッジ装置の配下のデバイスのデバイス・リスト・エントリを取り出す（S 5 0 5）。ルートブリッジ装置の配下のデバイスのデバイス・リスト・エントリがない場合（S 5 0 6でNO）、処理はS 5 0 1に戻り、次のルートブリッジ装置のデバイス・リスト・エントリの取り出し処理に進む。

[0128] ルートブリッジ装置の配下のデバイスのデバイス・リスト・エントリがある場合（S 5 0 6でYES）、対象デバイスの「DISABLE FLAG」の設定の有無を確認する（S 5 0 7）。「DISABLE FLAG」の値が0でなければ（S 5 0 7でNO）、エラー・レジスタの収集の対象外とし、S 5 0 5の処理に戻り、ルートブリッジ装置の配下の次のデバイスのデバイス・リスト・エントリを取り出す。

[0129] 対象デバイスの「DISABLE FLAG」の値が0の場合（S 5 0 7でYES）、モードの設定が1（末端装置のみが対象）でかつ対象デバイスの「END DEVICE FLAG」が1であるか、あるいは、モードの設定が2（全てのデバイスが対象）であるかを確認する（S 5 0 8）。S 5 0 8のいずれの場合にも該当しない場合（S 5 0 8でNO）、エラー・レジスタの収集の対象外となるので、S 5 0 5の処理に戻る。

[0130] S 5 0 8のいずれかの場合に該当する場合（S 5 0 8でYES）、対象デバイスの「COMPLETE FLAG」の値が1であるかどうか、すなわち、対象デバイスからのエラー・レジスタ値の収集が完了しているかどうかを確認する（S 5 0 9）。「COMPLETE FLAG」の値が1でなければ（S 5 0 9でNO）、まだ他のCPUコアが対象デバイスからエラー・レジスタ値を収集する処理を完了していないので、「COMPLETE FLAG」の値が1になるまでループして待つ。

[0131] S 5 0 9の処理において、「COMPLETE FLAG」の値が1であることを確認できた場合（S 5 0 9でYES）、S 5 1 0の処理に進み、「READ-FAIL FLAG」の値が1であるかどうかを判定する。「READ-FAIL FLAG」の値が1である

場合（S 5 1 0でYES）、ブリッジ装置のエラー・レジスタの値も参照して総合的に解析する必要があるため、S 5 1 1の処理に進む。S 5 1 1では、対象デバイスが含まれるPCIバス・ツリーの始点にあたるルートブリッジ装置のデバイス・リスト・エントリの「CHECK ALL FLAG」を値1に設定する。そして、S 5 0 5の処理に戻る。

[0132] S 5 0 5では、ルートブリッジ装置の配下の次のデバイスのデバイス・リスト・エントリを取り出す。そして、S 5 0 6において、ルートブリッジ装置の配下のデバイスで、エラー・レジスタ値の収集処理状況の確認が終わっていないデバイスがなくなると（S 5 0 6でNO）、S 5 0 1に戻り、以下、どのように他のデバイスについても処理状況を確認していく。

[0133] 図12、図13を用いて説明したように、代表CPUコアを含む複数のCPUコアを用いてエラー・レジスタ値の収集処理を行うことによって、多くのPCIデバイスからエラー・レジスタ値を収集する時間を短縮することができる。

[0134] なお、図12では、複数のCPUコアで実施する場合を説明したが、1つのCPUコアでエラー・レジスタ値の収集処理を行う場合には、S 4 0 2とS 4 0 3の処理を行うようにすればよい。すなわち、CPUコアが1つの場合でもS 4 0 3の処理において「CHECK ALL FLAG」の更新を行うように実装することができる。

[0135] 図14は、デバイス・リストの初期値の例を示す図である。デバイス・リスト230は、例えば、図14に示されるように各列に各デバイスのデバイス・リスト・エントリを複数の行を用いて記憶するテーブル形式で保持するようにしてもよい。

[0136] デバイス・リスト230の実際のデータはメモリ200上に記憶されるので、デバイス・リスト230中の各デバイスに対してアドレス番地を割り当て、「NEXT DEVICE LINK」や「NEXT ROOT LINK」等のポインタもアドレス番地を格納することによって、ポインタの指し示す先を設定することができる。図14に示される各デバイス・リスト・エントリの初期値は、図6の説明

において説明した初期値の値が格納される。

[0137] 図15は、デバイス・リストの遷移例1を示す図である。また、図16は、デバイス・リストの遷移例2を示す図である。ここで、例えば、図11に示すコンピュータ30のシステム構成において、図2に示すように末端装置433にエラーが発生したとする。また、SMMのモードにおけるエラーの探索処理は、3つのCPUコア111～113を用いて並行して行うものとする。

[0138] コンピュータ30のシステム内の末端装置433にエラーが発生すると、エラー情報がルートブリッジ装置412に通知され、ルートブリッジ装置412からSMIの割り込みが各CPUコア111～113に上げられる。SMIの割り込みを受けた全てのCPUコア111～113は、SMMのモードに移行する。代表CPUコアをCPUコア111とし、代表CPUコア111が、SMMのモードにおいてエラー特定処理を代表して管理するものとする。

[0139] CPUコア111は、SMMのモードにおいて、IOエラーハンドラ220の処理を、図8のフローに従って実行する。CPUコア111は、図8の探索範囲の選択処理(S101)を実行する。ここで、末端装置433(DEVICE21)にエラーが発生したと仮定した場合、エラーの情報は、末端装置433が属するPCIバス・ツリーのルートブリッジ装置412(ROOT BRIDGE02)に通知される。そして、ルートブリッジ装置412のグローバル・エラー・レジスタにエラーがあったことを示す情報(例えば、値1)が設定されることになる。

[0140] 図8のS101の処理、具体的には図9に示す探索範囲選択部221によるルートブリッジ装置のエラー情報収集処理を実行した後のデバイス・リスト230の一例を図15に示す。図15の例では、ルートブリッジ装置411(ROOT BRIDGE01)とルートブリッジ装置412(ROOT BRIDGE02)の「COMPLETE FLAG」が1に設定されている。また、ルートブリッジ装置412(ROOT BRIDGE02)の「GLOBAL ERROR FLAG」が1に設定されている。

- [0141] ルートブリッジ装置412 (ROOT BRIDGE02) の「GLOBAL ERROR FLAG」が1に設定されているため、CPUコア111は、図8のS102においてエラー箇所を特定する処理を継続すると判定する (S102でYES)。そして、CPUコア111は、図8のS103の末端装置のエラー・レジスタ値の収集処理を行う。このとき、CPU111は、モードの引数を1にして図9の処理を呼び出す。
- [0142] また、代表CPUコアであるCPU111は、図12の処理フローに従い、待機している他のCPUコア112、113に対して、エラー・レジスタ値の収集を行う要求を送信する (S401)。そして、代表CPUコア111によるS402のエラー・レジスタ値収集処理と他のCPUコア112、113によるS405のエラー・レジスタ値収集処理とを並行して実行する。
- [0143] 代表CPUコア111が図12のS403までの処理を実行した後のデバイス・リスト230の例を図16に示す。図15において「GLOBAL ERROR FLAG」が1となっているルートブリッジ装置412 (ROOT BRIDGE02) の配下にある末端装置433、434のエラーの有無の特定が行われる。その結果として、末端装置433、434の「COMPLETE FLAG」が値1に設定される。
- [0144] また、末端装置433 (DEVICE21) のエラーが検出され、DEVICE21の「ERROR FLAG」が値1に設定され、「ERROR REGISTER DUMP」にエラー情報が格納される。そして、ルートブリッジ装置412 (ROOT BRIDGE02) の「ERROR DETECTED COUNT」が1となる。
- [0145] その後、代表CPUコア111は、図8のエラー箇所の特定1の処理 (S104) を実行し、収集したエラー・レジスタ値の情報に基づいて、総合的にエラー箇所を特定する。本設例においては末端装置433 (DEVICE21) においてエラーが特定される。
- [0146] 図8のS104においてエラーが特定されたので (S105でYES)、代表CPUコア111は、S108のエラーの対処を行う。エラーの対処では、例えば、代表CPUコア111は、エラーが発生した末端装置433 (D

EVIC21) を使用しないようにハイパーバイザ 250 に指示を送る。また、代表 CPU コア 111 は、不図示の管理端末に末端装置 433 のエラー情報を通知して、末端装置 433 の交換等の保守を依頼する。

[0147] 次に、図 17、図 18 を用いてデバイス変更情報 I/F 部 243 を説明する。図 17 は、デバイス変更情報インタフェース (I/F) 部 243 の例を示す図である。デバイス変更情報 I/F 部 243 に含まれる記憶領域には、「ACTION」と「LOCATION」のフィールドが含まれる。

[0148] 図 17 の「LOCATION」のフィールドには、追加または削除の対象となる PCI デバイスの位置情報が記憶される。位置情報としては、例えば、セグメント番号 (Seg)、バス番号 (Bus)、デバイス番号 (Dev)、ファンクション番号 (Func) などが含まれる。「ACTION」フィールドは、PCI デバイスの変更の種類を示す。

[0149] 図 18 は、デバイス変更情報インタフェース 243 の「ACTION」フィールドの値の例を示す図である。図 18 において、「NONE」は何もしないことを意味し、「ADD」は PCI デバイスの追加を意味し、「REMOVE」は PCI デバイスの削除を意味する。

[0150] まず、「ACTION」が「ADD」である場合の動作を説明する。「ACTION」が「ADD」である場合、デバイス・リスト更新部 242 は、デバイス・リスト 230 に登録されたルートブリッジ装置の配下の「NEXT DEVICE LINK」を辿り、新しいデバイスを追加する PCI バス・ツリーのルートブリッジ装置を特定する。

[0151] デバイス・リスト更新部 242 は、追加されるデバイスのセグメント番号、すなわち、デバイス変更情報 I/F 243 の「LOCATION」フィールドに記憶されたセグメント番号を特定する。そして、デバイス・リスト更新部 242 は、特定したセグメント番号と同じセグメント番号を有するルートブリッジ装置について、そのルートブリッジ装置が扱うバス範囲の情報が格納されたレジスタを読み出す。追加するデバイスのバス番号、すなわち、デバイス変更情報 I/F 243 の「LOCATION」フィールドに記憶されたバス番号が、

ルートブリッジ装置が扱うバス範囲に含まれるかどうかをレジスタから読み出した情報に基づいて判定する。

[0152] 追加するデバイスのバス番号が、特定したルートブリッジ装置が扱うバス範囲に含まれる場合、そのルートブリッジ装置の「NEXT DEVICE LINK」を辿って得られる一連のリストから、「LOCATION」が一致するエントリが既に存在するかどうかを判定する。「LOCATION」が一致するエントリが存在すれば何もせず、存在しなければリストの末尾に、追加するデバイスに対応する新しいエントリを追加する。

[0153] 次に「ACTION」が「REMOVE」である場合の動作を説明する。「ACTION」が「REMOVE」である場合、デバイス・リスト更新部 242 は、デバイス・リスト 230 に登録されたルートブリッジ装置の配下の「NEXT DEVICE LINK」を辿り、削除する対象となるデバイスが属する P C I バス・ツリーのルートブリッジ装置を特定する。

[0154] デバイス・リスト更新部 242 は、削除するデバイスのセグメント番号、すなわち、デバイス変更情報 1 / F 243 の「LOCATION」フィールドに記憶されたセグメント番号を特定する。そして、デバイス・リスト更新部 242 は、特定したセグメント番号と同じセグメント番号を持つルートブリッジ装置について、そのルートブリッジ装置が扱うバス範囲の情報が格納されたレジスタを読み出す。削除するデバイスのバス番号、すなわち、デバイス変更情報 1 / F 243 の「LOCATION」フィールドに記憶されたバス番号が、ルートブリッジ装置が扱うバス範囲に含まれるかどうかをレジスタから読み出した情報に基づいて判定する。

[0155] 削除するデバイスのバス番号が、特定したルートブリッジ装置が扱うバス範囲に含まれる場合、そのルートブリッジ装置の「NEXT DEVICE LINK」を辿って得られる一連のリストから、「LOCATION」が一致するエントリを探し、デバイス・リスト 230 から削除する。

[0156] 尚、デバイス・リスト更新部 242 は、「ADD」や「REMOVE」の処理が終了すると、「ACTION」のフィールドを、例えば値 0 でクリアする。

- [0157] 図19は、デバイス削除後のデバイス・リストの例を示す図である。例えば、エラーが検出された末端装置433 (DEVICE21) がPCIホット・プラグ機能によってコンピュータ30のシステム構成から外された場合を想定する。この場合、末端装置433が外されたことをハイパーバイザ250が検出すると、デバイス変更通知部251は、デバイス変更情報1/F部243の「LOCATION」に末端装置433の位置情報を設定し、「ACTION」に「2」 (REMOVEを示す値) を設定する。そして、デバイス変更通知部251は、各CPUコア111~113に対してSMIの割り込みを上げ、各CPUコア111~113をSMMのモードに遷移させる。
- [0158] 各CPUコア111~113がSMMのモードに遷移した後、デバイス・リスト更新部242が、デバイス変更情報1/F部243の「ACTION」フィールドの情報からデバイスの削除要求であると判断する。そして、デバイス・リスト更新部242は、デバイス変更情報1/F部243の「LOCATION」に設定された位置情報から、削除するデバイスが属するPCIバス・ツリーのルートブリッジ装置がルートブリッジ装置412であることを特定する。
- [0159] デバイス・リスト更新部242は、デバイス・リスト230のエントリのうち、ルートブリッジ装置412 (ROOT BRIDGE02) から辿って得られるリストから、末端装置433 (DEVICE21) に一致するエントリを削除する。このようにして末端装置433が削除されたデバイス・リスト230の例が図19に示した例となる。
- [0160] 図20は、デバイス追加後のデバイス・リストの例を示す図である。例えば、PCIホット・プラグ機能によって新しい末端装置 (DEVICE21b) が追加された場合を想定する。この場合、新しい末端装置 (DEVICE21b) が追加されたことをハイパーバイザ250が検出すると、デバイス変更通知部251は、デバイス変更情報1/F部243の「LOCATION」に新しい末端装置 (DEVICE21b) の位置情報を設定する。また、デバイス変更通知部251は、デバイス変更情報1/F部243の「ACTION」に「1」 (ADDを示す値) を設定する。そして、デバイス変更通知部251は、各CPUコア111~113に対

してSMMの割り込みを上げ、各CPUコア111～113をSMMのモードに遷移させる。

[0161] 各CPUコア111～113がSMMのモードに遷移した後、デバイス・リスト更新部242が、デバイス変更情報1/F部243の「ACTION」フィールドの情報からデバイスの追加要求であると判断する。そして、デバイス・リスト更新部242は、デバイス変更情報1/F部243の「LOCATION」に設定された位置情報から、追加するデバイスが属するPCIバス・ツリーのルートブリッジ装置がルートブリッジ装置412であることを特定する。

[0162] デバイス・リスト更新部242は、デバイス・リスト230のエントリのうち、ルートブリッジ装置412（ROOT BRIDGE02）から辿って得られるリストから、新しい末端装置（DEVICE21b）に一致するエントリを探索する。この新しい末端装置（DEVICE21b）のエントリは、デバイス・リスト230中には存在しないので、ルートブリッジ装置412（ROOT BRIDGE02）から辿って得られるリストの最後に新しい末端装置（DEVICE21b）のエントリを追加する。このようにして新しい末端装置（DEVICE21b）を追加したデバイス・リスト230の例が図20に示した例となる。

[0163] 以上に説明したように、実施例2においては、デバイス・リスト230を用いてデバイスのエラー特定処理の状態管理を行うことで、複数のCPUコアで並列してエラー・レジスタの情報を収集させることができる。これによって、多くのPCIデバイスからエラー・レジスタを収集する時間を短縮することができる。

実施例 3

[0164] 図21は、実施例3に係るサーバシステム1000のシステム構成図の例を示す図である。サーバシステム1000は、複数のシステムボード（SB）30Aと、各システムボード（SB）を管理するマネジメントボード（MMB）を有する。また、サーバシステム1000は、ネットワーク50を介して、管理端末60に接続される。

[0165] 実施例3では、実施例2におけるコンピュータ30のシステム構成が、各

システムボード（SB）30Aに搭載された例となっている。各システムボード（SB）30Aは、バスを介してマネジメントボード（MMB）40と接続され、マネジメントボード40によって管理される。

[0166] いずれかのシステムボード（SB）30Aに含まれるPCIデバイスにエラーが発生し、実施例1や実施例2で説明したエラーの特定処理（例えば、S104やS107の処理）が行われ、エラーを発生させたデバイスが特定された場合を想定する。この場合、エラー対処部224は、特定されたデバイスに対するエラーの対処（S108）を行うとともに、管理端末60に対して、エラーを発生させたデバイスの情報を送信する。

[0167] 管理端末60は、サーバシステム1000からデバイスのエラーに関する情報を受信すると、サーバシステム1000のMMB40に対して何らかのリカバリ処理を指示したり、サーバシステム1000の保守管理者に故障したデバイスの交換要求を通知する。実施例1や実施例2で説明した方法によってPCIデバイスのエラーを早期に特定できた場合、各システムボード（SB）30A上で稼働するゲストOS511、512を早期に再稼働させて、システムの継続運用を行うことができる。

その他

[0168] 以上、本発明の好ましい実施例について説明したが、本発明は特定の実施例に限定されるものではなく、種々の変形や変更が可能である。例えば、実施例では、1つのCPU内に複数のCPUコアを有する構成例として説明したが、複数のCPUがバス等で接続されたシステム構成でも本発明を実施できる。複数のCPU構成においても、先の実施例と同様にSMIに応じてSMMのモードに遷移する場合には本発明のエラー特定方法によって早期にデバイスのエラーを特定することが可能となる。

[0169] また、実施例2において、複数のCPUコア111～113を用いて並行してデバイスのエラー・レジスタ値の収集を行う例を説明した。その際、全てのCPUコア111～113がデバイス・リスト230に含まれる全てのエントリを探索する例で説明したが、これは処理の簡略性を優先した例を説

明したものである。そのため、例えば、CPUコアごとにデバイス・リストを分割して持たせ、デバイスのエラーの探索を独立して平行に行わせることもできる。この場合にも実施例2とほぼ同様の効果が得られる。

符号の説明

- [0170] 10、20、30 コンピュータ
- 30A システムボード
- 40 MMB
- 50 通信回線
- 60 管理端末
- 100 CPU
- 111、112、113 CPUコア
- 121、122、123 SMM
- 200 メモリ
- 210 SMM領域
- 220 IOエラーハンドラ
- 221 探索範囲選択部
- 222 エラー・レジスタ値収集部
- 223 エラー箇所特定部
- 224 エラー対処部
- 230 デバイス・リスト
- 241 デバイス・リスト作成部
- 242 デバイス・リスト更新部
- 243 デバイス変更情報I/F部
- 250 ハイパーバイザ
- 251 デバイス変更通知部
- 300 バス
- 400 PCI周辺ブロック
- 411～413 ルートブリッジ装置

4 2 1 ～ 4 2 4 ブリッジ装置
4 3 1 ～ 4 3 7 末端装置
5 0 1、5 0 2 パーティション
5 1 1、5 1 2 ゲストOS
1 0 0 0 サーバシステム

請求の範囲

[請求項1]

処理装置と、

バスを介して前記処理装置からアクセスされる第1周辺装置を含み、
該第1周辺装置を起点にツリー状に接続されるとともに該第1周辺装置を介して前記処理装置からアクセスされる複数の周辺装置と、

前記バスを介して前記処理装置からアクセスされ、前記複数の周辺装置の接続構成を示す構成情報を記憶する記憶装置と、を備え、

前記処理装置は、前記複数の周辺装置のいずれかで発生したエラーに起因する割り込みに応じて各周辺装置のエラー情報を各周辺装置から読み出してエラーを特定するにあたり、前記記憶装置に記憶された構成情報を参照して前記ツリー状に接続された複数の周辺装置のうち前記第1周辺装置とは逆側の末端に位置する複数の周辺装置である複数の末端装置を特定し、特定した前記複数の末端装置から優先的に、前記エラー情報を読み出してエラーの有無を判定することで、前記エラーを発生させた周辺装置を特定する

ことを特徴とする情報処理装置。

[請求項2]

前記処理装置は、

特定した前記複数の末端装置についてエラーがなければ、各末端装置よりも前記第1周辺装置側に接続された周辺装置のそれぞれから前記エラー情報を読み出してエラーの有無を判定する

ことを特徴とする請求項1に記載の情報処理装置。

[請求項3]

前記構成情報は、前記情報処理装置の起動処理を行う際に、前記処理装置によって前記複数の周辺装置の接続構成を特定して作成され、

前記複数の周辺装置に含まれる周辺装置の削除または追加を行う度に、前記処理装置によって前記構成情報の更新が行われる

ことを特徴とする請求項1又は2に記載の情報処理装置。

[請求項4]

前記複数の周辺装置は、前記バスに接続された前記第1周辺装置を起点としてツリー状に接続された第1周辺装置群と、前記バスに接続

された第2周辺装置を起点としてツリー状に接続された第2周辺装置群とを含み、

前記第1周辺装置は、前記第1周辺装置群に含まれる周辺装置のいずれかでエラーが発生したことを示す第1代表エラー情報を保持し、

前記第2周辺装置は、前記第2周辺装置群に含まれる周辺装置のいずれかでエラーが発生したことを示す第2代表エラー情報を保持し、

前記処理装置は、

前記割り込みを受信すると、前記記憶装置に記憶した構成情報を参照して前記第1周辺装置及び前記第2周辺装置を特定し、

特定した前記第1周辺装置及び前記第2周辺装置から、前記第1代表エラー及び前記第2代表エラーの情報を読み出し、

読み出した前記第1及び第2代表エラーの情報から、前記第1周辺装置群と前記第2周辺装置群のいずれにおいてエラーが発生したかを特定し、

特定した前記第1又は第2周辺装置群に含まれる周辺装置の中から前記複数の末端装置を特定する

ことを特徴とする請求項1～3のいずれか1項に記載の情報処理装置。

[請求項5] 前記処理装置は、各々独立して演算処理を行う複数の処理部を備え、

前記複数の周辺装置の各々のエラー情報の読み出し処理を前記複数の処理部を用いて並行して行う

ことを特徴とする請求項1～4のいずれか1項に記載の情報処理装置。

[請求項6] 前記処理装置が備える複数の処理部のうち、一の処理部が他の処理部に対して、前記複数の周辺装置のエラー情報の読み出しを指示し、

前記一の処理部が他の処理部による前記複数の周辺装置のエラー情報の読み出し処理の完了を待って、前記複数の周辺装置のなかから前

記エラーを発生させた周辺装置を特定する

ことを特徴とする請求項5に記載の情報処理装置。

[請求項7] 前記構成情報は、前記複数の周辺装置の各々について、前記複数の処理部のいずれかによるエラー情報の読み出し処理の経過を示す読み出し経過情報を保持し、

前記複数の処理部の各々は、前記複数の周辺装置に含まれる周辺装置からエラー情報の収集を行うたびに、前記エラー情報を収集した周辺装置に対応する読み出し経過情報を読み出しが完了したことを示す情報に更新し、

前記一の処理部は、前記構成情報に含まれる読み出し経過情報を参照することによって、前記他の処理部による読み出し処理の完了を判定する

ことを特徴とする請求項6に記載の情報処理装置。

[請求項8] 処理装置と、バスを介して前記処理装置からアクセスされる第1周辺装置を含み、該第1周辺装置を起点にツリー状に接続されるとともに該第1周辺装置を介して前記処理装置からアクセスされる複数の周辺装置とを備える情報処理装置におけるエラー処理方法において、

前記複数の周辺装置のいずれかで発生したエラーに起因する割り込みに応じて各周辺装置のエラー情報を各周辺装置から読み出してエラーを特定するにあたり、前記複数の周辺装置の接続構成を示す構成情報を参照して前記ツリー状に接続された複数の周辺装置のうち前記第1周辺装置とは逆側の末端に位置する複数の周辺装置である複数の末端装置を特定し、

特定した前記複数の末端装置から優先的に、前記エラー情報を読み出してエラーの有無を判定することで、前記エラーを発生させた周辺装置を特定する

処理を前記処理装置に実行させるエラー処理方法。

[請求項9] 処理装置と、バスを介して前記処理装置からアクセスされる第1周

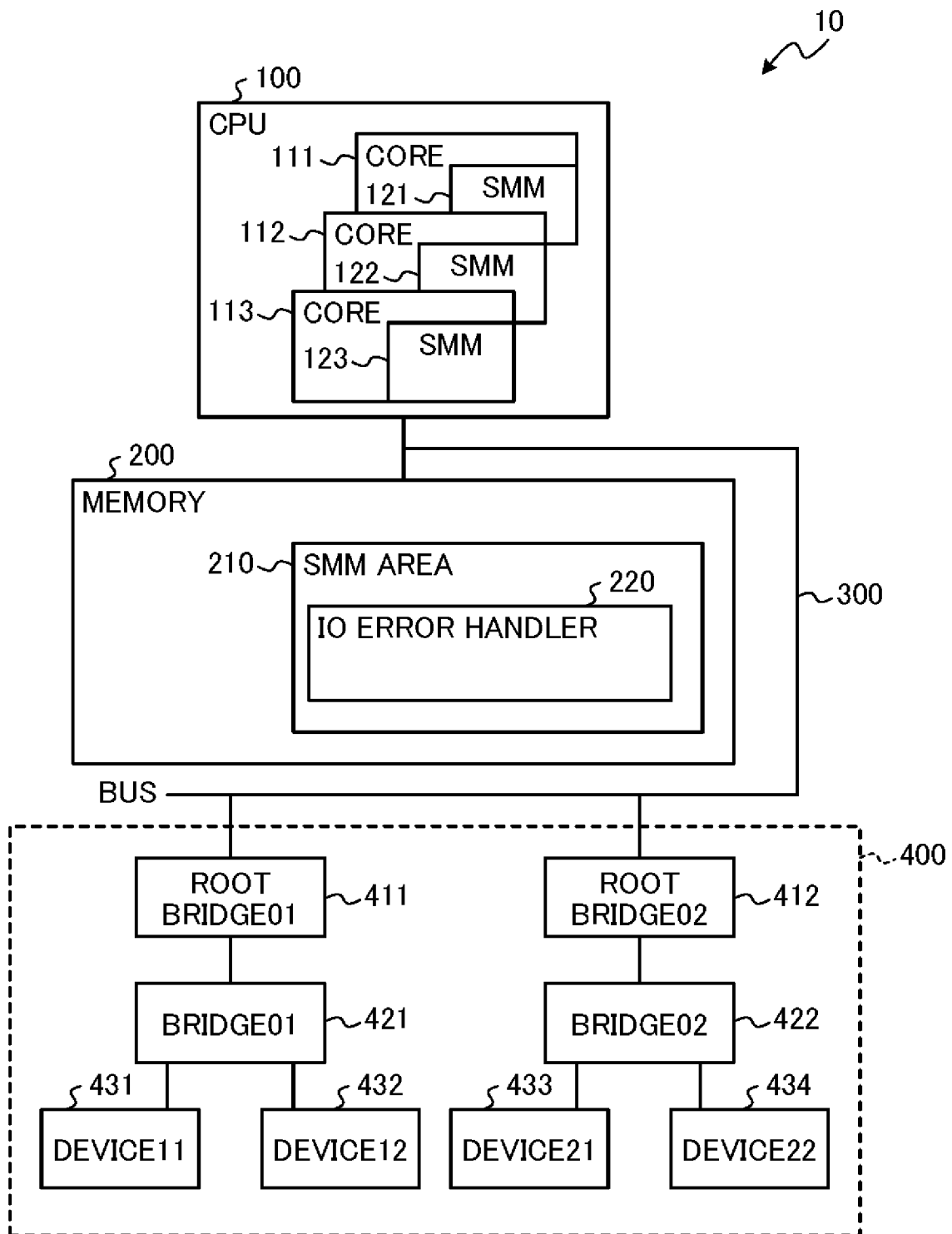
辺装置を含み、該第1周辺装置を起点にツリー状に接続されるとともに該第1周辺装置を介して前記処理装置からアクセスされる複数の周辺装置とを備える情報処理装置におけるエラー処理を行うプログラムであって、

前記複数の周辺装置のいずれかで発生したエラーに起因する割り込みに応じて各周辺装置のエラー情報を各周辺装置から読み出してエラーを特定するにあたり、前記複数の周辺装置の接続構成を示す構成情報を参照して前記ツリー状に接続された複数の周辺装置のうち前記第1周辺装置とは逆側の末端に位置する複数の周辺装置である複数の末端装置を特定し、

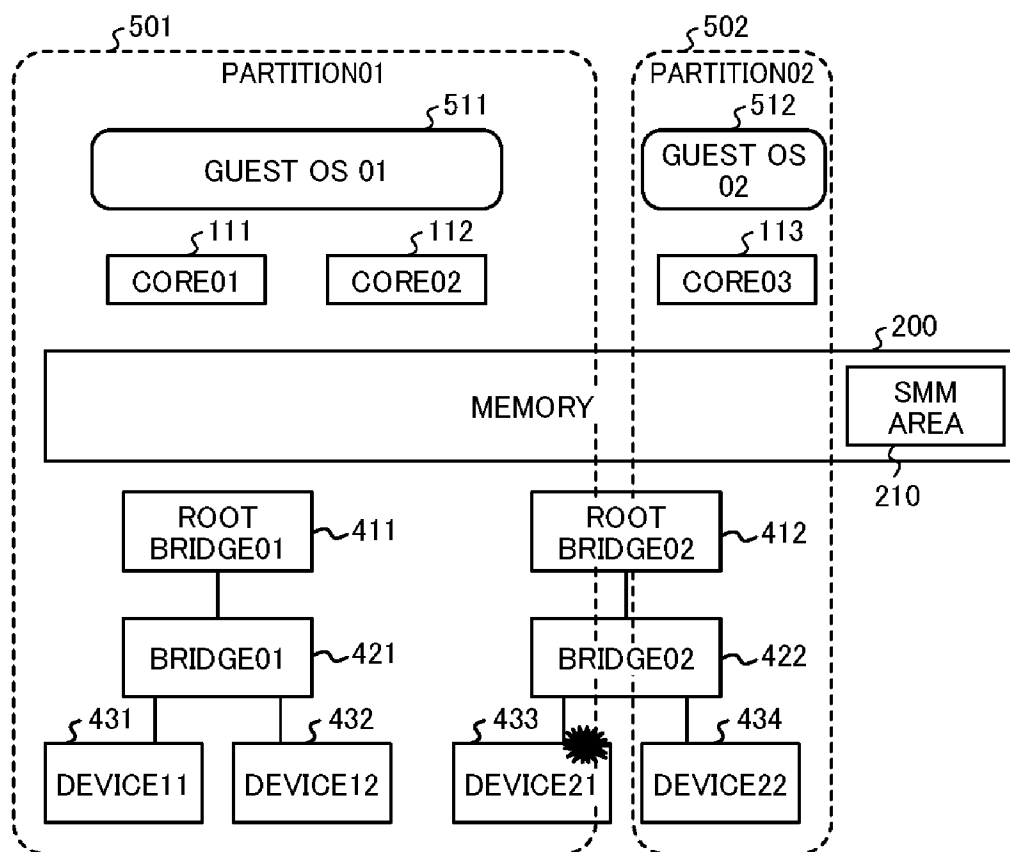
特定した前記複数の末端装置から優先的に、前記エラー情報を読み出してエラーの有無を判定することで、前記エラーを発生させた周辺装置を特定する

処理を前記処理装置に実行させるエラー処理プログラム。

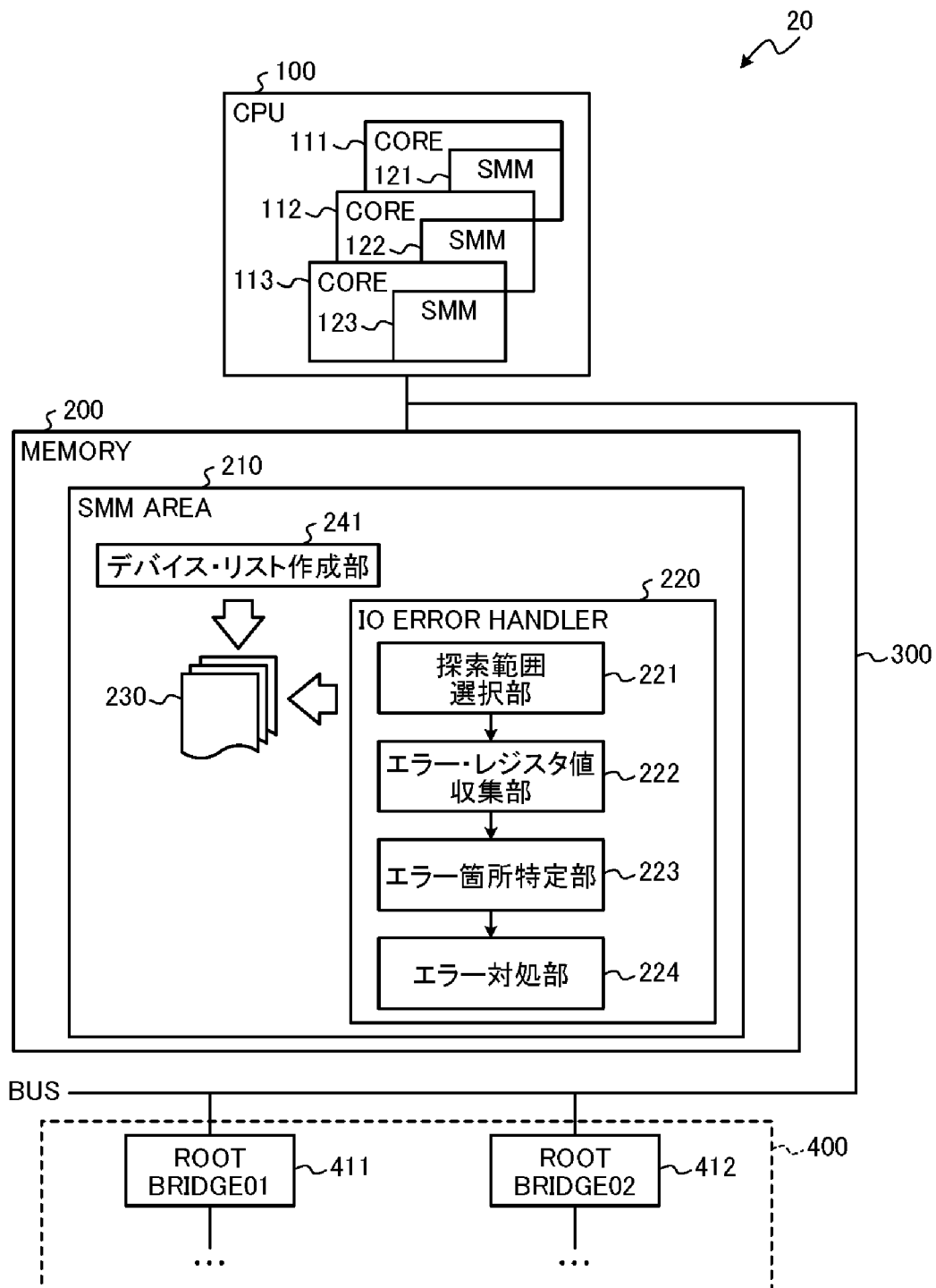
[図1]



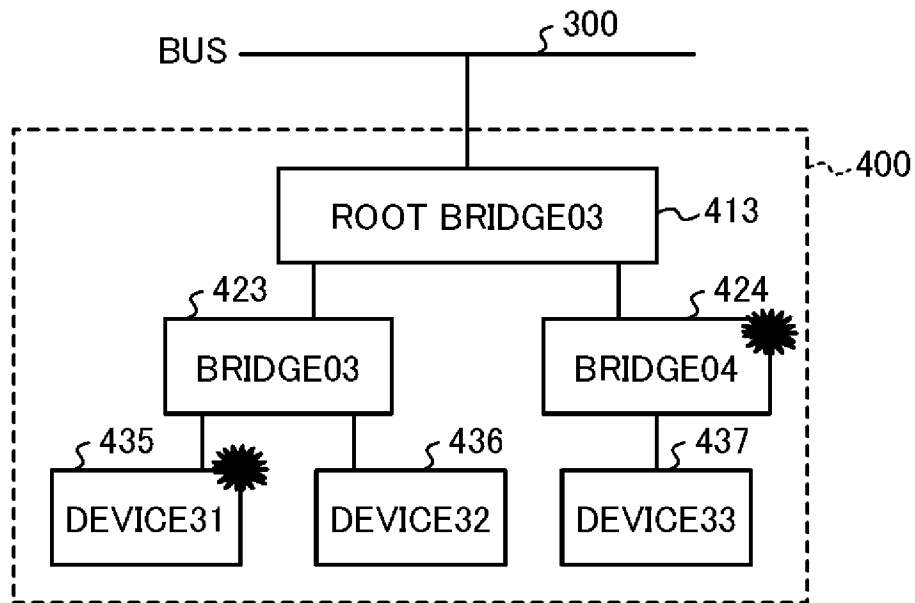
[図2]



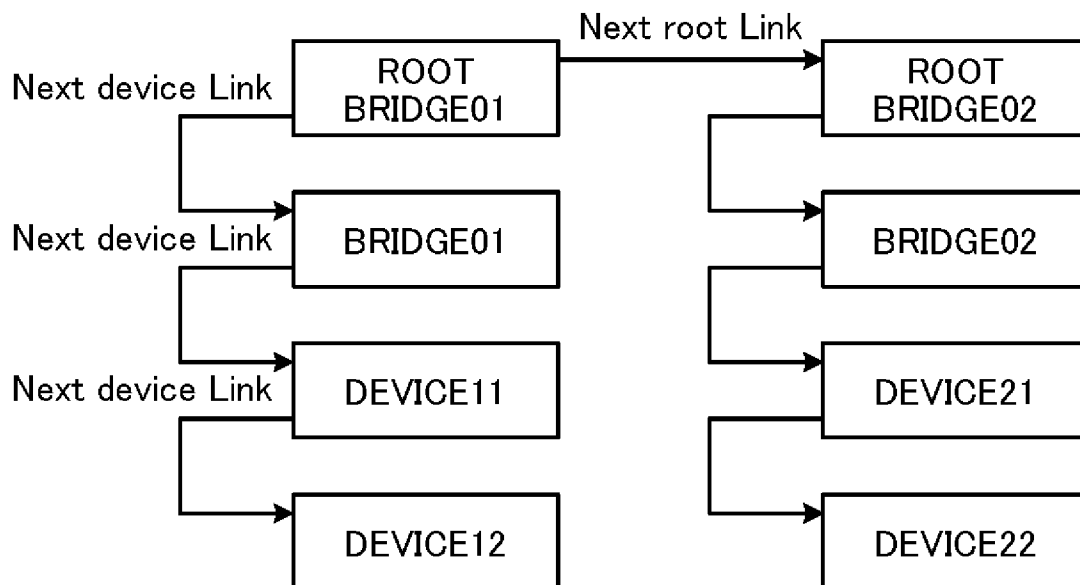
[図3]



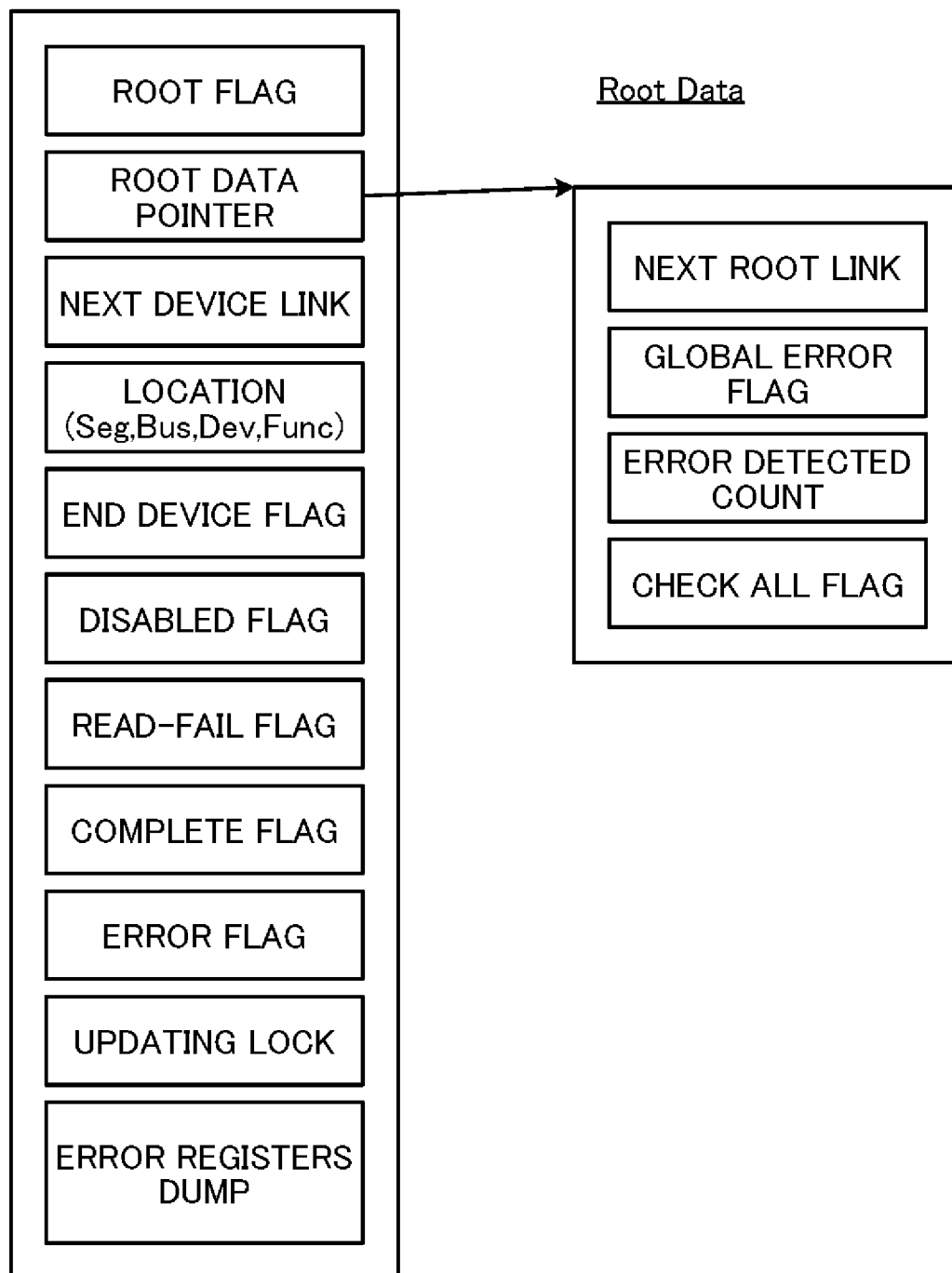
[図4]



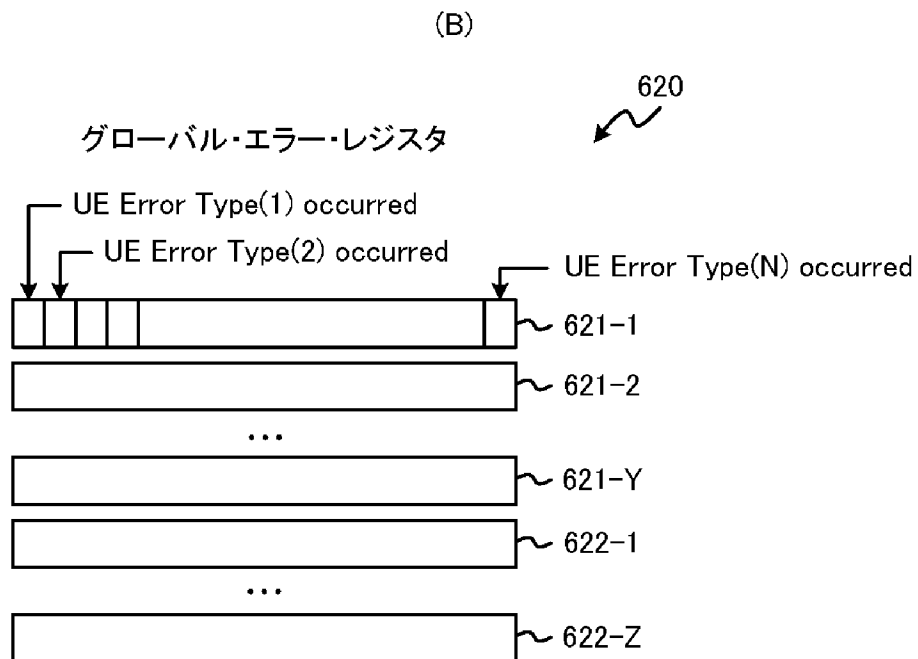
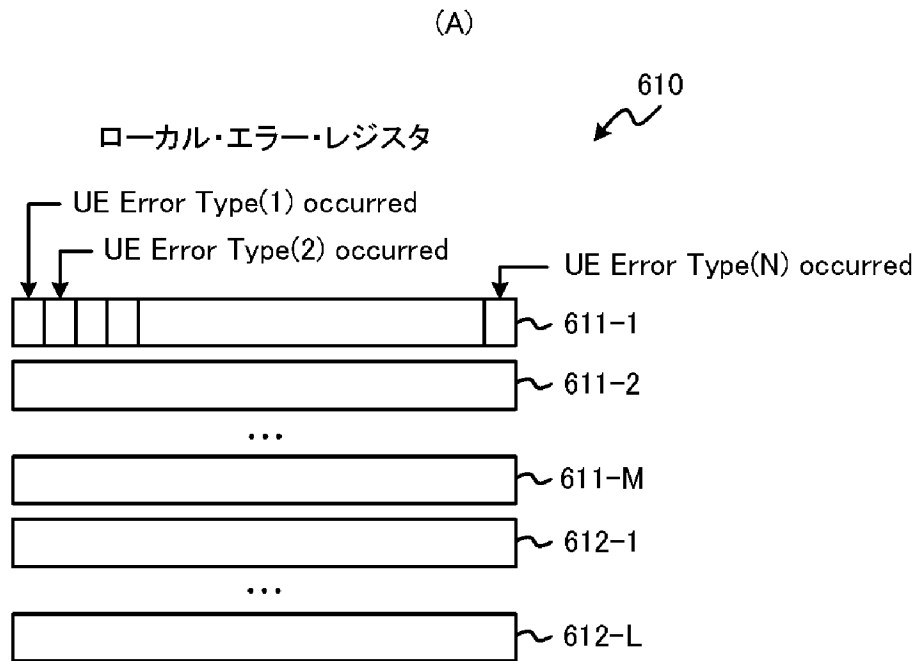
[図5]



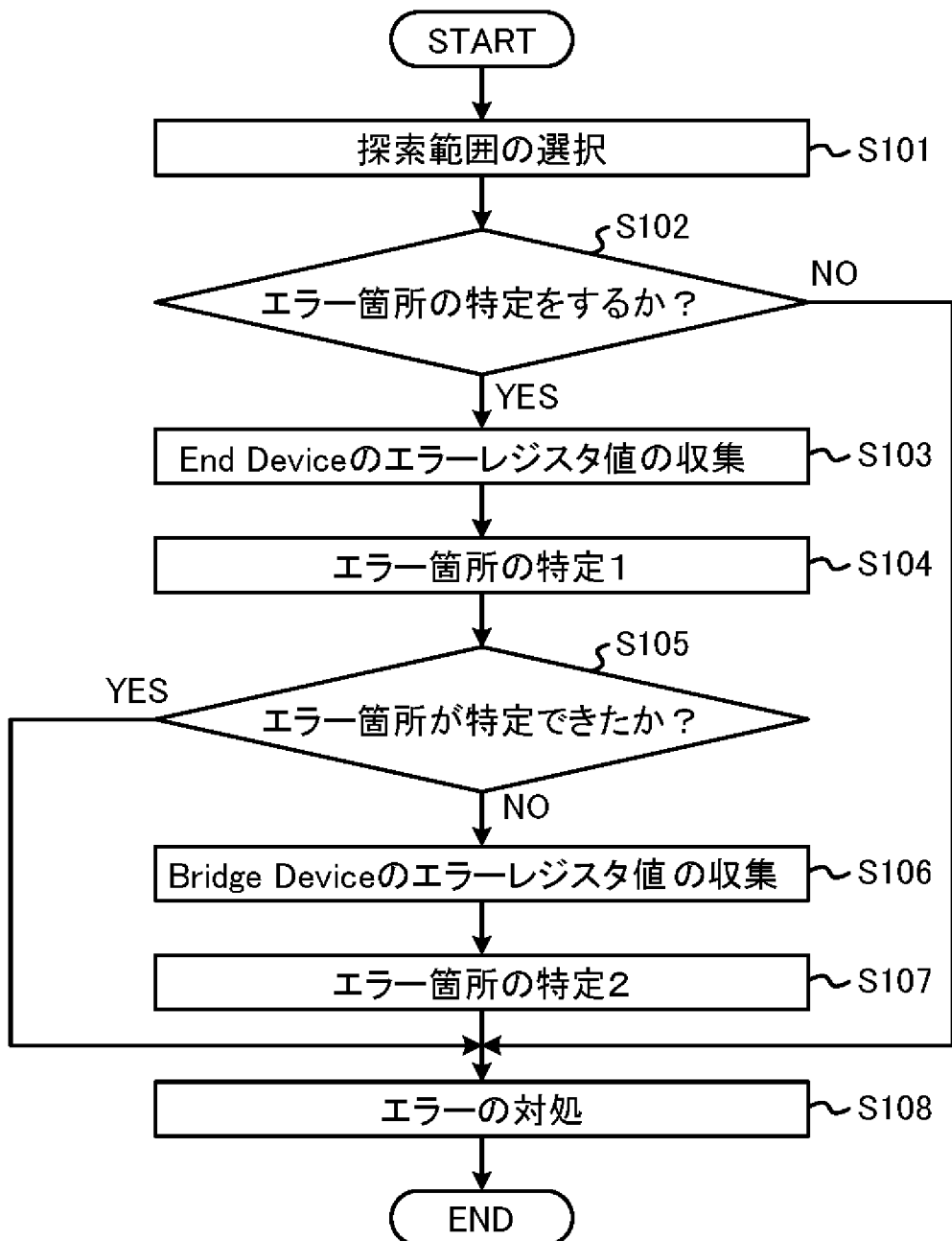
[図6]



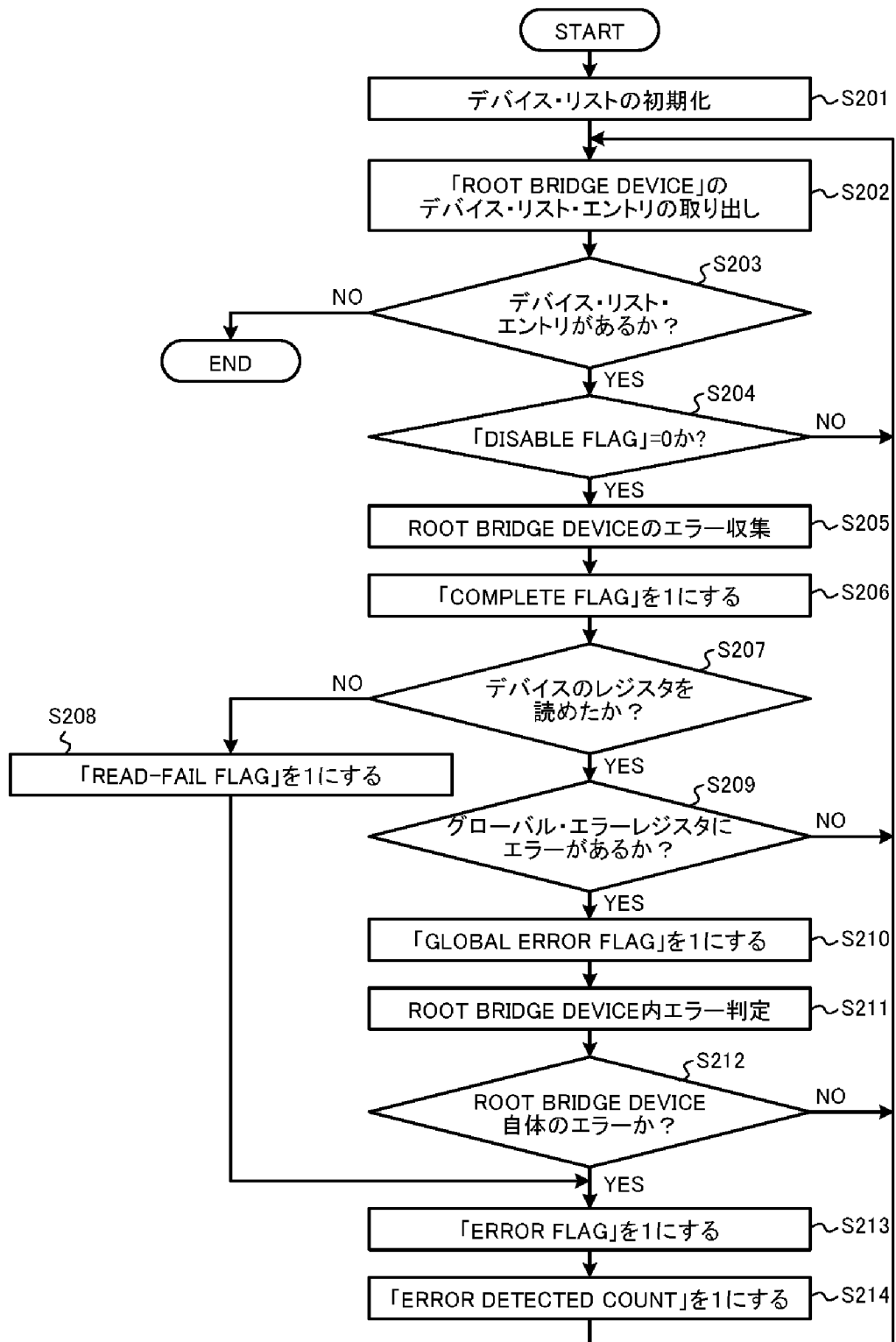
[図7]



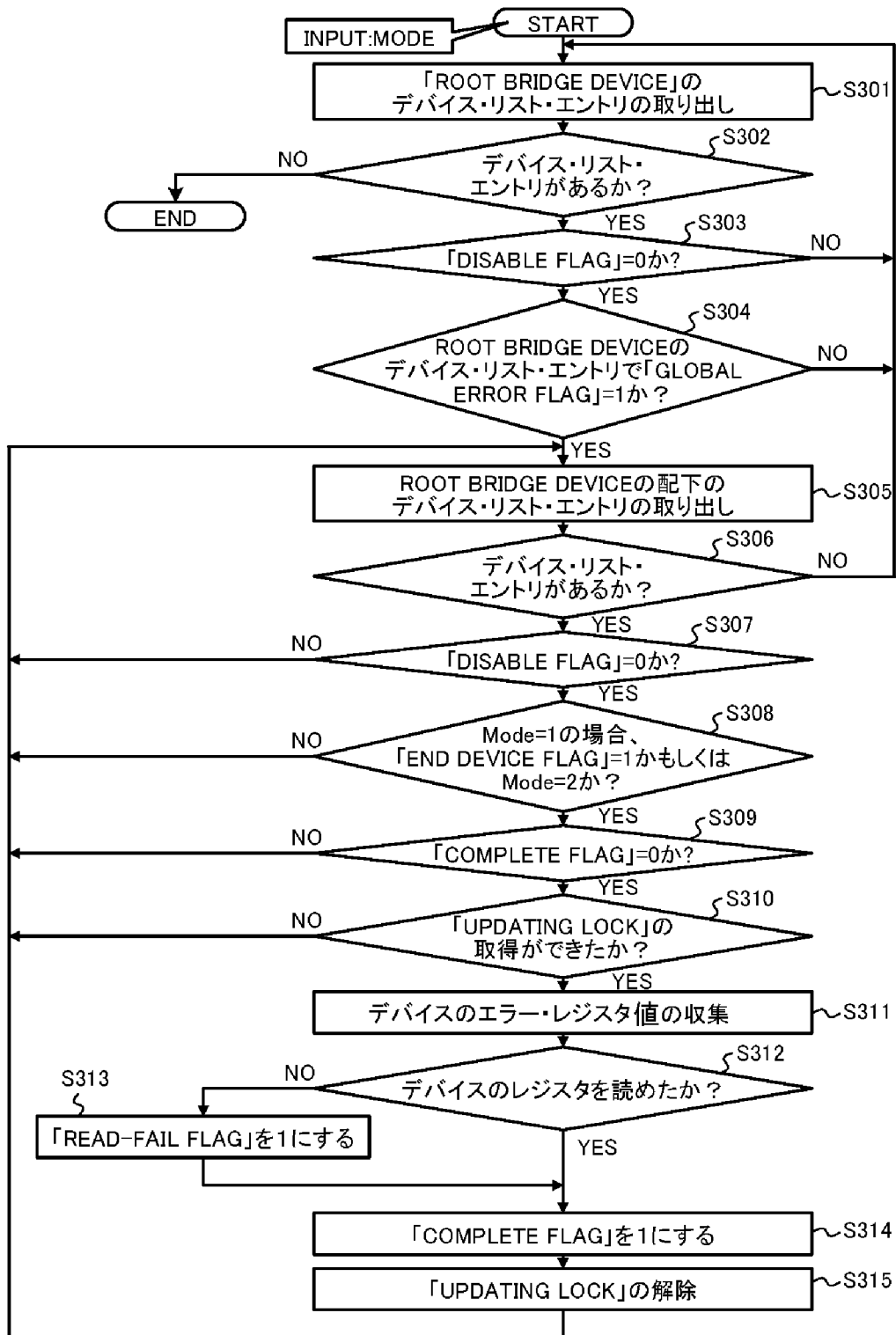
[図8]



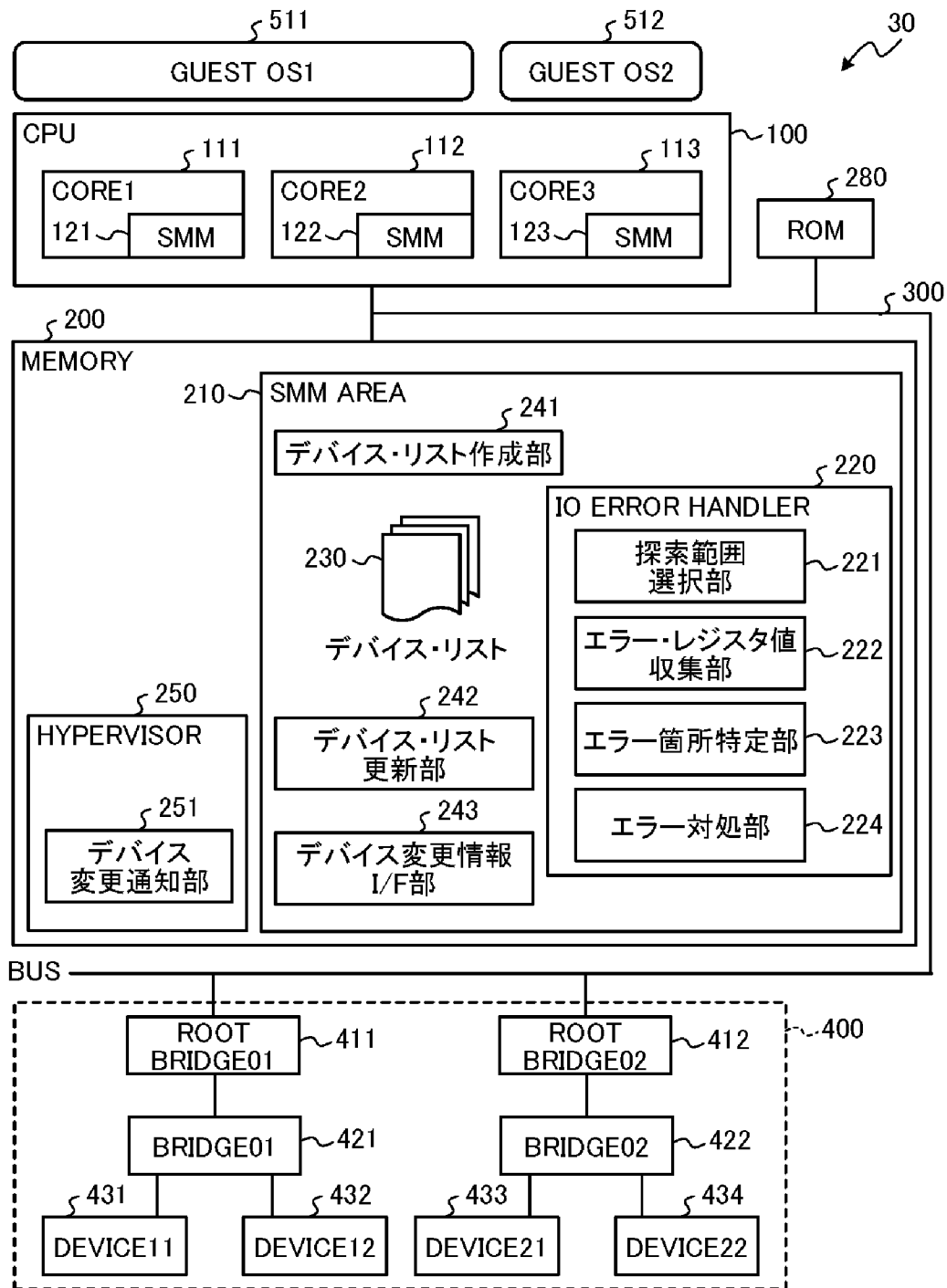
[図9]



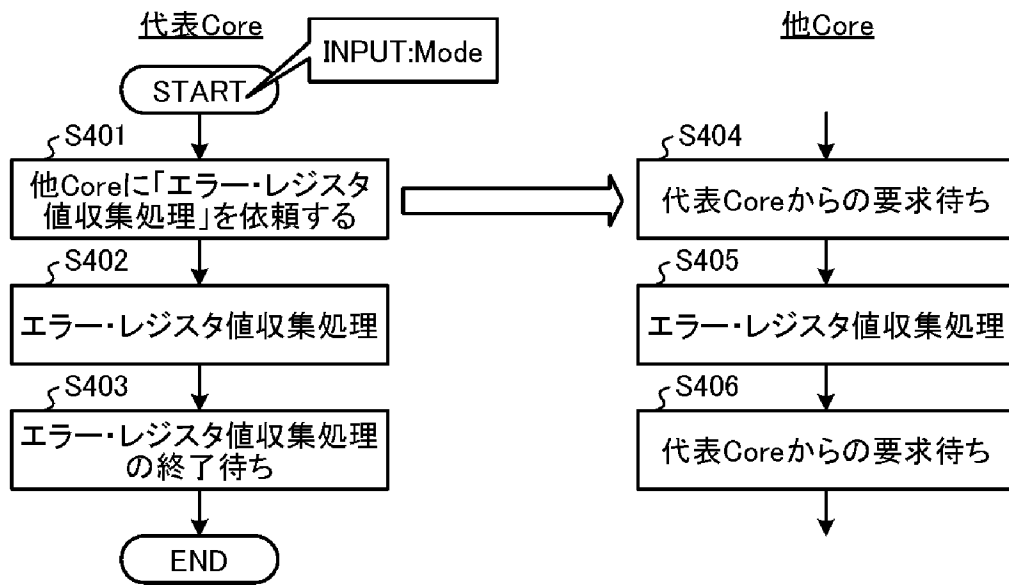
[図10]



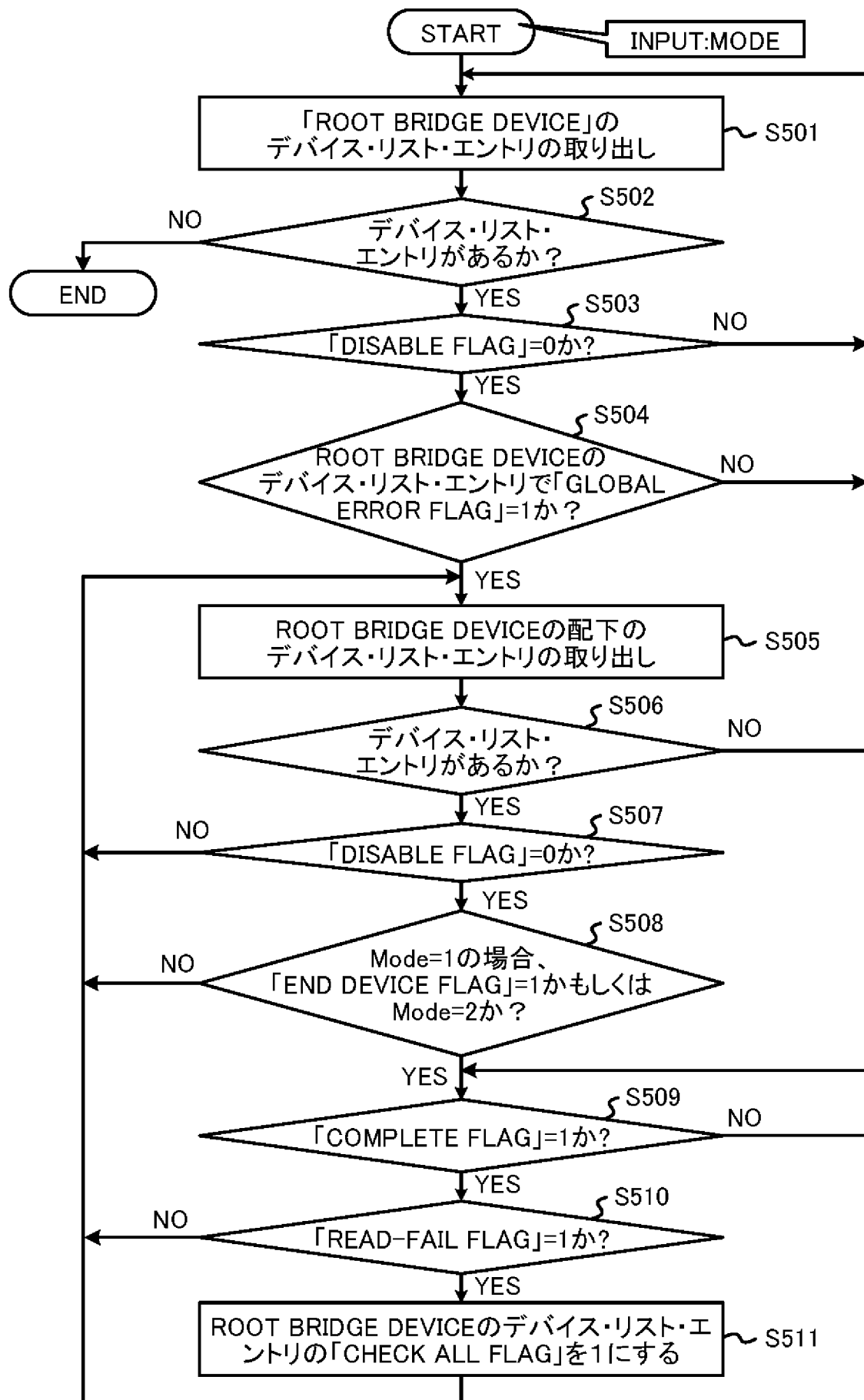
[図11]



[図12]



[図13]



[図14]

ADDRESS	0x6000 ROOT BRIDGE01	0x6100 BRIDGE01	0x6200 DEVICE11	0x6300 DEVICE12	0x6400 ROOT BRIDGE02	0x6500 BRIDGE02	0x6600 DEVICE21	0x6700 DEVICE22
ROOT FLAG	1	0	0	0	1	0	0	0
ROOT DATA POINTER	0x7000	NULL	NULL	NULL	0x7100	NULL	NULL	NULL
NEXT DEVICE LINK	0x6100	0x6200	0x6300	NULL	0x6500	0x6600	0x6700	NULL
LOCATION	(0,0,10,0)	(0,10,5,0)	(0,20,4,0)	(0,20,5,0)	(0,64,10,0)	(0,74,5,0)	(0,84,4,0)	(0,84,5,0)
END DEVICE FLAG	0	0	1	1	0	0	1	1
DISABLED FLAG	0	0	0	0	0	0	0	0
READ-FAIL FLAG	0	0	0	0	0	0	0	0
COMPLETE FLAG	0	0	0	0	0	0	0	0
ERROR FLAG	0	0	0	0	0	0	0	0
UPDATING FLAG	0	0	0	0	0	0	0	0
ERROR REGISTER DUMP	-	-	-	-	-	-	-	-

ADDRESS	0x7000 ROOT BRIDGE01	0x7100 ROOT BRIDGE02
NEXT ROOT LINK	0x7100	NULL
GLOBAL ERROR FLAG	0	0
ERROR DETECTED COUNT	0	0
CHECK ALL FLAG	0	0

[図15]

ADDRESS	0x6000 ROOT BRIDGE01	0x6100 BRIDGE01	0x6200 DEVICE11	0x6300 DEVICE12	0x6400 ROOT BRIDGE02	0x6500 BRIDGE02	0x6600 DEVICE21	0x6700 DEVICE22
ROOT FLAG	1	0	0	0	1	0	0	0
ROOT DATA POINTER	0x7000	NULL	NULL	NULL	0x7100	NULL	NULL	NULL
NEXT DEVICE LINK	0x6100	0x6200	0x6300	NULL	0x6500	0x6600	0x6700	NULL
LOCATION	(0,0,10,0)	(0,10,5,0)	(0,20,4,0)	(0,20,5,0)	(0,64,10,0)	(0,74,5,0)	(0,84,4,0)	(0,84,5,0)
END DEVICE FLAG	0	0	1	1	0	0	1	1
DISABLED FLAG	0	0	0	0	0	0	0	0
READ-FAIL FLAG	0	0	0	0	0	0	0	0
COMPLETE FLAG	1	0	0	0	1	0	0	0
ERROR FLAG	0	0	0	0	0	0	0	0
UPDATING FLAG	0	0	0	0	0	0	0	0
ERROR REGISTER DUMP	-	-	-	-	-	-	-	-

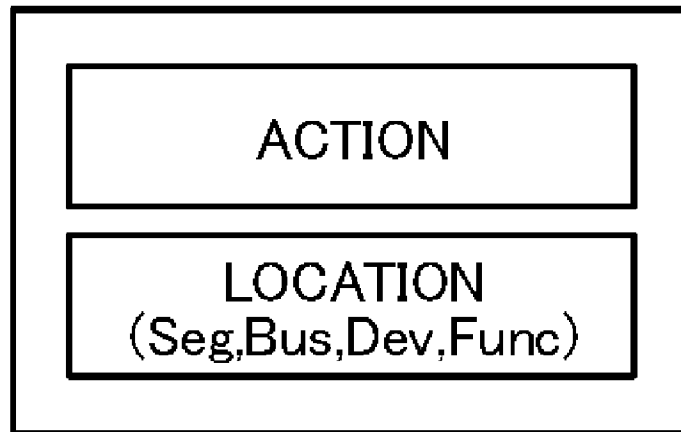
ADDRESS	0x7000 ROOT BRIDGE01	0x7100 ROOT BRIDGE02
NEXT ROOT LINK	0x7100	NULL
GLOBAL ERROR FLAG	0	1
ERROR DETECTED COUNT	0	0
CHECK ALL FLAG	0	0

[図16]

ADDRESS	0x6000 ROOT BRIDGE01	0x6100 BRIDGE01	0x6200 DEVICE11	0x6300 DEVICE12	0x6400 ROOT BRIDGE02	0x6500 BRIDGE02	0x6600 DEVICE21	0x6700 DEVICE22
ROOT FLAG	1	0	0	0	1	0	0	0
ROOT DATA POINTER	0x7000	NULL	NULL	NULL	0x7100	NULL	NULL	NULL
NEXT DEVICE LINK	0x6100	0x6200	0x6300	NULL	0x6500	0x6600	0x6700	NULL
LOCATION	(0,0,10,0)	(0,10,5,0)	(0,20,4,0)	(0,20,5,0)	(0,64,10,0)	(0,74,5,0)	(0,84,4,0)	(0,84,5,0)
END DEVICE FLAG	0	0	1	1	0	0	1	1
DISABLED FLAG	0	0	0	0	0	0	0	0
READ-FAIL FLAG	0	0	0	0	0	0	0	0
COMPLETE FLAG	0	0	0	0	0	0	1	1
ERROR FLAG	0	0	0	0	0	0	1	0
UPDATING FLAG	0	0	0	0	0	0	0	0
ERROR REGISTER DUMP	-	-	-	-	-	-	**	-

ADDRESS	0x7000 ROOT BRIDGE01	0x7100 ROOT BRIDGE02
NEXT ROOT LINK	0x7100	NULL
GLOBAL ERROR FLAG	0	0
ERROR DETECTED COUNT	0	1
CHECK ALL FLAG	0	0

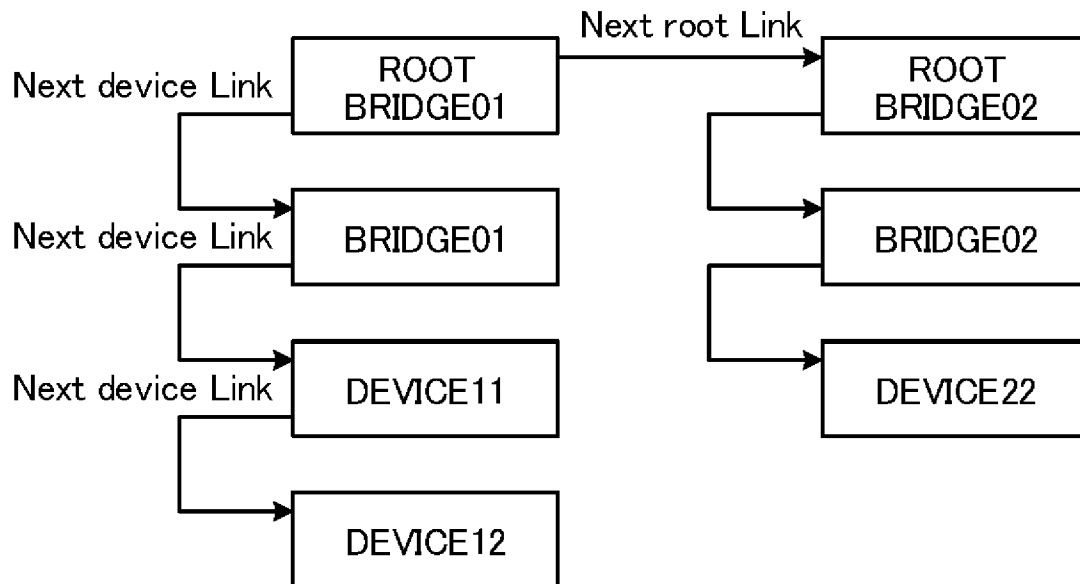
[図17]



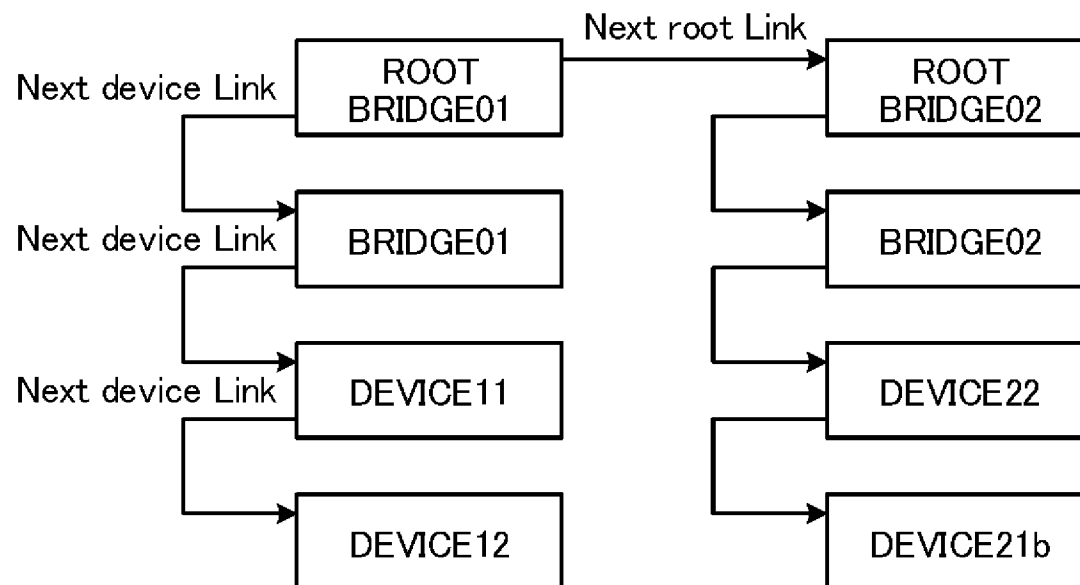
[図18]

ACTION	MEANING
0	NONE
1	ADD
2	REMOVE

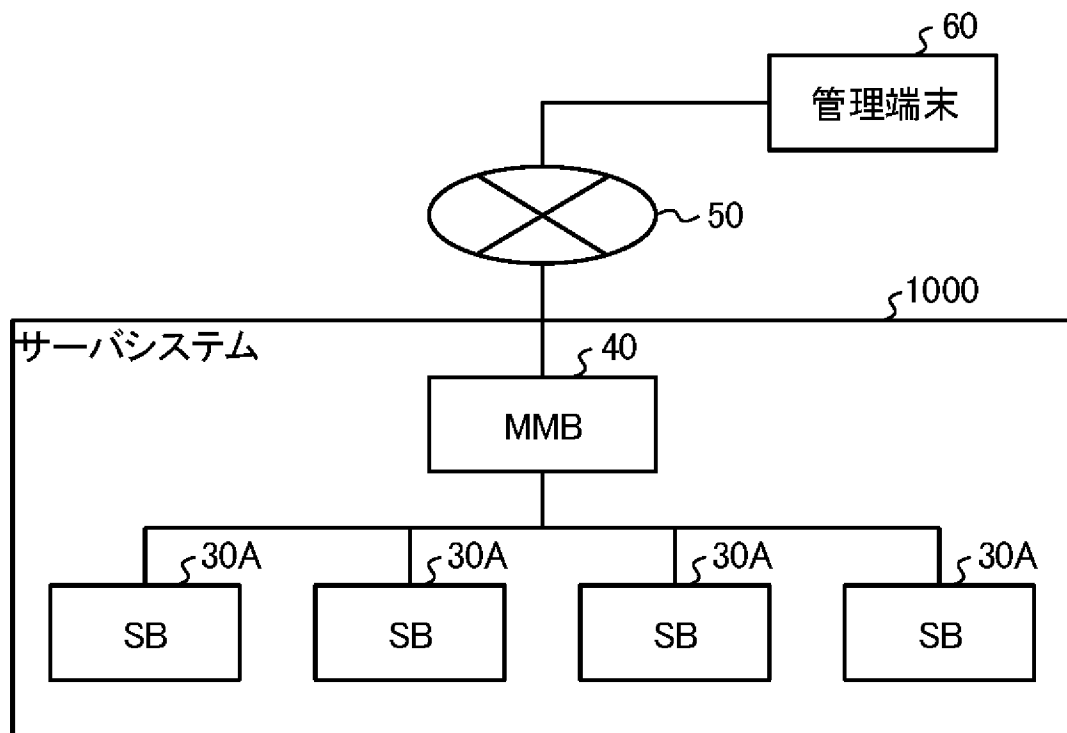
[図19]



[図20]



[図21]



INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP2015/003734

A. CLASSIFICATION OF SUBJECT MATTER

G06F11/30(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)

G06F11/30, G06F11/34, G06F9/455, G06F9/46, G06F13/00, G06F13/14

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	1996-2015
Kokai Jitsuyo Shinan Koho	1971-2015	Toroku Jitsuyo Shinan Koho	1994-2015

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.
A	WO 2012/063358 A1 (Fujitsu Ltd.), 18 May 2012 (18.05.2012), paragraphs [0018] to [0037] & US 2013/0246855 A1 paragraphs [0034] to [0066]	1-9
A	JP 2012-160095 A (Hitachi, Ltd.), 23 August 2012 (23.08.2012), paragraphs [0039] to [0057] & US 2012/0198446 A1 paragraphs [0058] to [0078]	1-9
A	JP 2006-285519 A (Hitachi Global Storage Technologies Netherlands B.V.), 19 October 2006 (19.10.2006), paragraphs [0015] to [0021] (Family: none)	1-9

☐ Further documents are listed in the continuation of Box C.

☐ See patent family annex.

* Special categories of cited documents:

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

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

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

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

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

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

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

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

"&" document member of the same patent family

Date of the actual completion of the international search
08 October 2015 (08.10.15)

Date of mailing of the international search report
20 October 2015 (20.10.15)

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（I P C））

Int.Cl. G06F11/30(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. G06F11/30, G06F11/34, G06F9/455, G06F9/46, G06F13/00, G06F13/14

最小限資料以外の資料で調査を行った分野に含まれるもの

日本国実用新案公報	1 9 2 2 - 1 9 9 6 年
日本国公開実用新案公報	1 9 7 1 - 2 0 1 5 年
日本国実用新案登録公報	1 9 9 6 - 2 0 1 5 年
日本国登録実用新案公報	1 9 9 4 - 2 0 1 5 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
A	WO 2012/063358 A1（富士通株式会社）2012.05.18, 段落 [0018]-[0037] & US 2013/0246855 A1, 段落[0034]-[0066]	1-9
A	JP 2012-160095 A（株式会社日立製作所）2012.08.23, 段落 [0039]-[0057] & US 2012/0198446 A1, 段落[0058]-[0078]	1-9
A	JP 2006-285519 A（ヒタチグローバルストレージテクノロジーズネ ザーランドビービー）2006.10.19, 段落[0015]-[0021]（ファミリー なし）	1-9

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

* 引用文献のカテゴリー

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

の日の後に公表された文献

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

国際調査を完了した日

0 8 . 1 0 . 2 0 1 5

国際調査報告の発送日

2 0 . 1 0 . 2 0 1 5

国際調査機関の名称及びあて先

日本国特許庁（I S A / J P）
郵便番号 1 0 0 - 8 9 1 5
東京都千代田区霞が関三丁目4番3号

特許庁審査官（権限のある職員）

坂庭 剛史

5 B

9 2 8 8

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