

(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  
2012年8月30日(30.08.2012)



(10) 国際公開番号  
WO 2012/114443 A1

- (51) 国際特許分類:  
G06F 11/20 (2006.01)
- (21) 国際出願番号: PCT/JP2011/053732
- (22) 国際出願日: 2011年2月21日(21.02.2011)
- (25) 国際出願の言語: 日本語
- (26) 国際公開の言語: 日本語
- (71) 出願人(米国を除く全ての指定国について): 富士通株式会社(FUJITSU LIMITED) [JP/JP]; 〒2118588 神奈川県川崎市中原区上小田中4丁目1番1号 Kanagawa (JP).
- (72) 発明者; および
- (75) 発明者/出願人(米国についてのみ): 山下 浩一郎(YAMASHITA, Koichiro) [JP/JP]; 〒2118588 神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内 Kanagawa (JP). 山内 宏真(YAMAUCHI, Hiromasa) [JP/JP]; 〒2118588 神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内 Kanagawa (JP). 鈴木 貴久(SUZUKI, Takahisa) [JP/JP]; 〒2118588 神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内 Kanagawa (JP). 栗原 康志(KURIHARA, Koji) [JP/JP]; 〒2118588 神奈川県川崎市中原区上小田

中4丁目1番1号 富士通株式会社内 Kanagawa (JP).

(74) 代理人: 酒井 昭徳(SAKAI, Akinori); 〒1006020 東京都千代田区霞が関3丁目2番5号 霞が関ビルディング20階 酒井総合特許事務所 Tokyo (JP).

(81) 指定国(表示のない限り、全ての種類の国内保護が可能): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PE, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) 指定国(表示のない限り、全ての種類の広域保護が可能): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), ユーラシア (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), ヨーロッパ (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,

[続葉有]

(54) Title: PROCESSOR MANAGEMENT METHOD

(54) 発明の名称: プロセッサ管理方法

[図1]

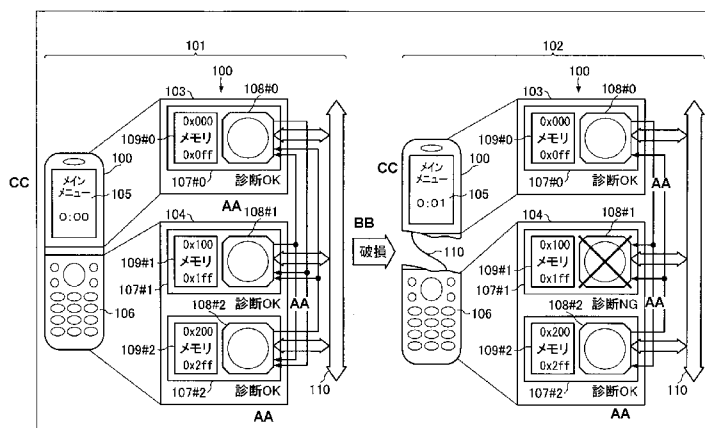


FIG. 1:  
109#0, 109#1, 109#2 Memory  
AA Diagnosis  
BB Breakage  
CC Main menu

(57) Abstract: Provided to users is a low cost, high performance system. The system (100) is constituted of a set of domains (107) each including a memory (109) and an SoC (108) having a processor. The system (100) sets up in the processor in a domain (107#0) a master mechanism for managing domains (107#0-107#2). Similarly, the system (100) sets up in the processor of a domain (107#1) a local master mechanism for managing itself and a virtual master mechanism for mirroring a master context to the memory (109) in the own domain (107#1). The processor in the domain (107#0) notifies to the domains (107#0-107#2) offset values such that the logical addresses of the memories (109#0-109#2) are continuous.

(57) 要約: 安価であり、かつ高性能なシステムをユーザに提供する。システム(100)は、プロセッサを有するSoC(108)とメモリ(109)を含んだドメイン(107)の集合で構築されている。このとき、システム(100)は、ドメイン(107#0)内のプロセッサに、ドメイン(107#0)~ドメ

イン(107#2)を管理するマスタ機構を設定する。同様に、システム(100)は、ドメイン(107#1)のプロセッサに、自身を管理するローカルマスタ機構と、マスタコンテキストを自ドメイン内のメモリ(109)にミラーリングする仮想マスタ機構と、を設定する。ドメイン(107#0)内のプロセッサは、メモリ(109#0)~メモリ(109#2)の論理アドレスが連続するようなオフセット値を、ドメイン(107#0)~ドメイン(107#2)に通知する。

WO 2012/114443 A1

**WO 2012/114443 A1**



---

NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI 添付公開書類:

(BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, — 国際調査報告 (条約第 21 条(3))  
NE, SN, TD, TG).

## 明 細 書

**発明の名称：** プロセッサ管理方法

**技術分野**

[0001] 本発明は、複数のプロセッサを管理するプロセッサ管理方法に関する。

**背景技術**

[0002] 近年、携帯電話等の携帯端末は、用途が広範囲となり、機能も豊富となっている。このように携帯端末に求められる性能要求が高くなってきている。たとえば、高機能な携帯端末を提供できる技術として、携帯端末を画像入出力用ユニットと音声受信ユニットというように、ユニットごとに分離可能な技術が開示されている（たとえば、下記特許文献 1、2 を参照。）。これにより、ユーザは、通話を行いつつ、画像閲覧を行うといった高機能なサービスを楽しむことができる。

[0003] また、各ユニットをドッキング可能な携帯端末について、各ユニットを、バスを介してデバイスと接続可能にする技術が開示されている（たとえば、下記特許文献 3 を参照。）。また、携帯端末の各ユニット内で、異常、故障を診断する自己診断回路を有する技術が開示されている（たとえば、下記特許文献 4 を参照。）。

**先行技術文献**

**特許文献**

[0004] 特許文献1：特開 2 0 0 5 － 1 7 6 3 7 1 号公報

特許文献2：特開平 1 0 － 2 2 8 4 5 8 号公報

特許文献3：特公平 6 － 6 6 0 5 7 号公報

特許文献4：特開平 3 － 2 1 7 1 3 7 号公報

**発明の概要**

**発明が解決しようとする課題**

[0005] しかしながら、上述した従来技術において、特許文献 1 ～ 4 にかかる技術では、高性能が求められる分、各ユニットに搭載されるプロセッサが高性能

のプロセッサとなるため、コストの増加を招いてしまうという問題があった。具体的には、複数のユニットに分割可能な携帯端末は、複数のユニットに分割されたことによるバックアップ処理や、機能のミラーリングを行う場合に、高性能プロセッサを複数配置しておくことが要求されるため、コストが増加してしまっていた。

[0006] 本発明は、上述した従来技術による問題点を解消するため、安価であり、かつ高性能なシステムをユーザに提供できるプロセッサ管理方法を提供することを目的とする。

### 課題を解決するための手段

[0007] 上述した課題を解決し、目的を達成するため、本発明の一側面によれば、複数のプロセッサのうちの一のプロセッサに複数のプロセッサを管理するマスタ機構を設定し、複数のプロセッサのうちの一のプロセッサ以外のプロセッサのそれぞれに当該プロセッサを管理するローカルマスタ機構と仮想マスタ機構とを設定し、マスタ機構によって管理される共有メモリが複数のプロセッサから連続したメモリとしてアクセスされるために、マスタ機構が複数のプロセッサにアドレスのオフセット値を通知するプロセッサ管理方法が提案される。

### 発明の効果

[0008] 本発明の一側面によれば、安価であり、かつ高性能なシステムをユーザに提供できるという効果を奏する。

### 図面の簡単な説明

[0009] [図1] 図1は、本実施の形態にかかるシステム100の破損前と破損後の動作を示す説明図である。

[図2] 図2は、システム100のハードウェアを示すブロック図である。

[図3] 図3は、システム100の機能を示すブロック図である。

[図4] 図4は、システム100のソフトウェアを示すブロック図である。

[図5A] 図5Aは、スレーブドメイン破損時による障害ドメイン検出動作を示す説明図（その1）である。

[図5B]図5Bは、スレーブドメイン破損時による障害ドメイン検出動作を示す説明図（その2）である。

[図6A]図6Aは、スレーブドメイン破損後の復元動作を示す説明図（その1）である。

[図6B]図6Bは、スレーブドメイン破損後の復元動作を示す説明図（その2）である。

[図7A]図7Aは、マスタドメイン破損時の動作を示す説明図（その1）である。

[図7B]図7Bは、マスタドメイン破損時の動作を示す説明図（その2）である。

[図8A]図8Aは、分離状態で破損した場合の動作を示す説明図（その1）である。

[図8B]図8Bは、分離状態で破損した場合の動作を示す説明図（その2）である。

[図9]図9は、ステータステーブル303の記憶内容の一例を示す説明図である。

[図10]図10は、起動処理を示すフローチャートである。

[図11]図11は、マスタドメイン、スレーブドメインの運用時処理を示すフローチャート（その1）である。

[図12]図12は、マスタドメイン、スレーブドメインの運用時処理を示すフローチャート（その2）である。

[図13]図13は、マスタコンテキスト転送処理1を示すフローチャートである。

[図14]図14は、マスタコンテキスト転送処理2を示すフローチャートである。

[図15]図15は、ハードチェックスレッド（マスタ）の処理を示すフローチャートである。

[図16]図16は、ハードチェックスレッド（スレーブ）処理を示すフローチャートである。

ヤートである。

[図17]図17は、障害ドメイン検出処理を示すフローチャートである。

### 発明を実施するための形態

[0010] 以下に添付図面を参照して、開示のプロセッサ管理方法の実施の形態を詳細に説明する。なお、下記説明では、プロセッサ管理方法を適用したシステムについて説明を行う。

[0011] 図1は、本実施の形態にかかるシステム100の破損前と破損後の動作を示す説明図である。符号101で示す説明図では、複数のプロセッサを有するシステム100の破損前の状態を示しており、符号102で示す説明図では、システム100の破損後の状態を示している。

[0012] 始めに、符号101で示す図の説明を行う。システム100は、ディスプレイユニット103と、キーボードユニット104を含む。ディスプレイユニット103は、ディスプレイ105を含むユニットである。キーボードユニット104は、キーボード106を含むユニットである。

[0013] また、システム100は、独立して1つのコンピュータシステムとして動作可能であるドメイン単位で分割されている。具体的には、ディスプレイユニット103にはドメイン107#0が含まれ、キーボードユニット104にはドメイン107#1、ドメイン107#2が含まれる。以下、接尾記号となる“#0”～“#2”が付与されたハードウェアは、対応するドメイン107内に含まれるハードウェアであることを示す。同様に、“#0”～“#2”が付与されたソフトウェアは、対応するドメイン107内のCPU (Central Processing Unit) によって実行されるソフトウェアであることを示す。

[0014] ドメイン107には、SoC (System-on-a-Chip) 108、メモリ109が含まれる。具体的には、ドメイン107#0にはSoC 108#0とメモリ109#0が含まれ、ドメイン107#1にはSoC 108#1とメモリ109#1が含まれ、ドメイン107#2にはSoC 108#2とメモリ109#2が含まれる。なお、ドメイン107、SoC 10

8、メモリ109の詳細については、図2にて後述する。なお、SOC108は、単なる携帯電話機能を実現する簡易的なSOCを想定している。簡易的なSOCであるため、SOC108のコストは低く抑えることができる。

[0015] またドメイン107#0～ドメイン107#2は、バス110で接続されている。バス110は、たとえば、フレキシブルケーブル等を採用することができる。または、ドメイン107#0～ドメイン107#2の接続方法は、有線でなく無線であってもよい。たとえば、ドメイン107#0～ドメイン107#2は、IEEE802.15.1規格に代表される近接無線通信部を含み、近接無線通信部によってドメイン間通信を行ってもよい。

[0016] また、ドメイン107#0～ドメイン107#2は、仮想的に1つのコンピュータシステムをユーザに提供するOSを実行する。これにより、システム100は、3ドメインを含んだシステムとなる。たとえば、SOC108#0～SOC108#2内にCPUが1つずつ存在する場合、システム100は3つのCPUを有するシステム100となる。

[0017] なお、図1では、ドメイン107#0内のCPUに、システム100を管理するマスタ機構が設定され、ドメイン107#1、ドメイン107#2のCPUに、自身を管理するローカルマスタ機構が設定された場合を想定している。

[0018] また、メモリ109#0～メモリ109#2は、連続した共有メモリ空間となるように物理アドレスからのオフセット値が設定されている。具体的なオフセット値については、図4にて後述する。オフセット値によって、メモリ109#0の論理アドレスは0x000～0xfffとなり、メモリ109#1の論理アドレスは0x100～0x1ffとなり、メモリ109#2の論理アドレスは0x200～0x2ffとなる。最終的に、メモリ109#0～メモリ109#2は、論理アドレス0x000～0x2ffを各ドメインに提供することができる。

[0019] このように、システム100は、各ドメインのメモリサイズが小さくとも、複数のドメインを接続することで、各ドメインに、より広いアドレスを提

供することができる。広いアドレスを利用できるようになったシステム１００は、同時に多くのアプリケーションソフトウェア（以下、「アプリ」と称する）を起動することができ、高性能なサービスをユーザに提供できる。

[0020] また、ドメイン１０７＃０～ドメイン１０７＃２は、相互に他のドメインを診断している。具体的には、ドメイン１０７＃０は、ドメイン１０７＃１とドメイン１０７＃２を診断し、ドメイン１０７＃１は、ドメイン１０７＃０とドメイン１０７＃２を診断し、ドメイン１０７＃２は、ドメイン１０７＃０とドメイン１０７＃１を診断している。これにより、ドメイン１０７＃０～ドメイン１０７＃２のうちいずれかのドメインに障害が発生した場合、他のドメインは障害の発生を検出できる。なお、具体的な診断方法としては、図９にて後述する。なお、符号１０１に示す図では、ドメイン１０７＃０～ドメイン１０７＃２は正常動作しているため、診断ＯＫとなる。

[0021] 続けて、符号１０２で示す図の説明を行う。符号１０２で示すシステム１００は、システム１００に加えられた衝撃等により、ディスプレイユニット１０３とキーボードユニット１０４との結合部分が破損した状態である。さらに、システム１００は、衝撃によってＳｏＣ１０８＃１が破損した状態である。ＳｏＣ１０８＃１が破損した結果、ドメイン１０７＃１は、コンピュータシステムとして動作不可能な状態である。なお、バス１１０は、断線が発生しておらず、ドメイン１０７＃０とドメイン１０７＃２間の通信を行える状態である。

[0022] ドメイン１０７＃１の診断結果がＮＧとなり、障害が発生したという情報は、ドメイン１０７＃０による診断、またドメイン１０７＃２による診断によって検出される。検出を受けたシステム１００は、３ドメインから２ドメインの動作に切り替える。これにより、システム１００は、障害発生時に、機能や性能の制限を最小限にしつつ、継続的に利用することができる。

[0023] （システム１００のハードウェア）

図２は、システム１００のハードウェアを示すブロック図である。図２において、システム１００は、ディスプレイ１０５と、キーボード１０６と、

ドメイン107#0～ドメイン107#2と、フラッシュROM (Read-Only Memory) 201と、フラッシュROMコントローラ202と、フラッシュROM203と、を含む。また、ドメイン107#0には、CPU204#0と、ペリフェラル205#0と、電力制御回路206#0と、DBB (Digital Base Band) 207#0と、自己診断回路208#0、メモリ109#0を含む。なお、図2で示すシステム100は、3つのドメインを含むが、2以上のドメインを含んでいてもよい。

[0024] また、メモリ109#0、CPU204#0～自己診断回路208#0は、SOC内部バス209#0で接続されている。同様に、ドメイン107#1には、メモリ109#1、CPU204#1～自己診断回路208#1を含み、SOC内部バス209#1で接続されている。また、ドメイン107#2には、メモリ109#2、CPU204#2～自己診断回路208#2を含み、SOC内部バス209#2で接続されている。

[0025] また、ドメイン107#0について、SOC108#0は、CPU204#0～DBB207#0を含み、自己診断回路208#0とメモリ109#0は、SOC108#0の外部に存在する。ドメイン107#1、ドメイン107#2についても同様である。以下、ドメイン107#0に含まれるハードウェアについて説明を行うが、ドメイン107#1、ドメイン107#2に含まれるハードウェアも同様の機能を有する。

[0026] CPU204#0は、システム100の全体の制御を司る。ペリフェラル205#0は、CPU204#0以外の周辺機器を示している。たとえば、ペリフェラル205#0は、GPU (Graphics Processing Unit) であったり、DMAC (Direct Memory Access Controller) であってもよい。または、ペリフェラル205#0は、ディスプレイ105等を制御するコントローラであったりしてもよい。

[0027] 電力制御回路206#0は、CPU204#0、ペリフェラル205#0

、DBB207#0の電力を制御する回路である。さらに、電力制御回路206#0は、SOC108#0の外部にある自己診断回路208#0と、メモリ109#0の電力を制御してもよい。たとえば、電力制御回路206#0は、ペリフェラル205#0が未使用である場合に、ペリフェラル205#0に提供する電力を停止してもよい。

[0028] DBB207#0は、通信、通話機能を実行する通話装置である。具体的に、DBB207#0は、通信相手を特定する機能や、データに誤り制御符号を付与するといった機能を有する。また、DBB207#0は、通信回線を通じてLAN (Local Area Network)、WAN (Wide Area Network)、インターネットなどのネットワークに接続され、ネットワークを介して他の装置に接続される。なお、DBB207#0は、DBB207#1、DBB207#2とも通信可能である。

[0029] 自己診断回路208#0は、ドメイン107#0内の各装置が正常に動作しているかを診断する機能を有する回路である。たとえば、自己診断回路208#0は、SOC108#0内の電流、電圧、温度を測定し、異常値になった場合に診断NGとして出力する。

[0030] なお、電流、電圧が異常値となる場合としては、たとえば、ドメイン107#0への衝撃、または、装置内の劣化に伴って電気回路が短絡した場合、ドメイン107#0に設計値を超える電流が流れる可能性がある。または、温度が異常値となる場合としては、ドメイン107#0で実行中のソフトウェアによって、CPU204#0が熱暴走した場合である。自己診断回路208#0は、このような状態を検出し、診断NGとして出力する。

[0031] メモリ109#0は、データを記憶できる記憶装置である。たとえば、メモリ109#0は、ROM、RAM (Random Access Memory) 等である。たとえば、ROMは、ブートプログラムなどのプログラムを記憶している。また、RAMは、CPU204#0のワークエリアとして使用される。なお、SOC内部バス209#0は、CPU204#0とワークエリアとなるメモリ109#0を接続しており、CPU204#0の能

力を発揮させるため、高速転送が可能であることが好ましい。

[0032] ディスプレイ 105 は、カーソル、アイコンあるいはツールボックスを始め、文書、画像、機能情報などのデータを表示する。ディスプレイ 105 は、たとえば、TFT (Thin Film Transistor) 液晶ディスプレイなどを採用することができる。キーボード 106 は、文字、数字、各種指示などの入力のためのキーを有し、データの入力を行う。また、キーボード 106 は、タッチパネル式の入力パッドやテンキーなどであってもよい。

[0033] フラッシュROM 201 は、読出し速度が高速なフラッシュROMであり、たとえば、NOR型フラッシュメモリである。フラッシュROM 201 は、OS (Operating System) などのシステムソフトウェアやアプリなどを記憶している。たとえば、OSを更新する場合、システム 100 は、DBB 207 を経由して新しいOSを受信し、フラッシュROM 201 に格納されている古いOSを、受信した新しいOSに更新する。

[0034] フラッシュROMコントローラ 202 は、CPU 204 #0 ~ CPU 204 #2 の制御にしたがってフラッシュROM 203 に対するデータのリード／ライトを制御する。フラッシュROM 203 は、データの保存、運搬を主な目的としたフラッシュROMであり、たとえば、NAND型フラッシュメモリである。また、フラッシュROM 203 は、フラッシュROMコントローラ 202 の制御で書き込まれたデータを記憶する。データの具体例としては、システム 100 を使用するユーザがDBB 207 を通して取得した画像データ、映像データや、本実施の形態であるプロセッサ管理方法を実行するプログラムが格納されていてもよい。フラッシュROM 203 は、たとえば、メモリカード、SDカードなどを採用することができる。

[0035] (システム 100 の機能)

次に、システム 100 の機能について説明する。図 3 は、システム 100 の機能を示すブロック図である。システム 100 は、検出部 311 と、転送部 312 と、判断部 313 と、設定部 314 と、通知部 315 と、割当部 3

１６と、格納部３１７と、設定部３１８と、終了部３１９と、を含む。この制御部となる機能（検出部３１１～終了部３１９）は、記憶装置に記憶されたプログラムをＣＰＵ２０４＃０～ＣＰＵ２０４＃２が実行することにより、その機能を実現する。記憶装置とは、具体的には、たとえば、図１、図２に示したメモリ１０９、フラッシュＲＯＭ２０１、フラッシュＲＯＭ２０３などである。

[0036] なお、図３では、ＣＰＵ２０４＃０がシステム１００全体を制御することを想定している。それに関連して、図３では、各機能部のうち、検出部３１１、割当部３１６は、ＣＰＵ２０４＃０～ＣＰＵ２０４＃２が実行することを想定している。また、転送部３１２、設定部３１４、通知部３１５は、システム１００全体を制御するＣＰＵ２０４＃０が実行することを想定している。また、判断部３１３、格納部３１７、設定部３１８、終了部３１９は、ＣＰＵ２０４＃１、ＣＰＵ２０４＃２が実行することを想定している。したがって、たとえば、ＣＰＵ２０４＃１がシステム１００全体を制御する場合、ＣＰＵ２０４＃１が、転送部３１２、設定部３１４、通知部３１５を有していてもよい。

[0037] また、システム１００は、メモリ１０９内に格納されているマスタコンテキスト３０１、ミラーリングマスタコンテキスト３０２にアクセス可能である。マスタコンテキスト３０１は、システム１００の運用時に要求される情報群が格納されている。たとえば、マスタコンテキスト３０１は、システムの起動情報、時刻情報、ペリフェラル２０５の設定情報、メモリ１０９に割り当てたオフセット値、重要なプロセスのコンテキスト情報等を格納している。ミラーリングマスタコンテキスト３０２は、マスタコンテキスト３０１と同内容が記憶されている。なお、ミラーリングマスタコンテキスト３０２は、転送部３１２、格納部３１７の機能により、マスタコンテキスト３０１と同内容を記憶するように維持される。

[0038] また、マスタコンテキスト３０１、ミラーリングマスタコンテキスト３０２内には、ステータステーブル３０３が格納されている。ステータステーブ

ル 3 0 3 は、自ドメイン内の診断情報として、診断時刻と診断結果が格納されている。ステータステーブル 3 0 3 の詳細については、図 9 にて後述する。

- [0039] 検出部 3 1 1 は、他ドメイン 1 0 7 の障害を検出する機能を有する。たとえば、検出部 3 1 1 # 0 ~ 検出部 3 1 1 # 2 は、他ドメイン 1 0 7 内の自己診断回路 2 0 8 に診断実行を要求して、診断結果を取得して他ドメイン 1 0 7 の障害を検出してもよい。また、検出部 3 1 1 は、マスタ機構のコンテキストデータにタイムスタンプまたは付随情報を所定周期毎に書き込み、タイムスタンプまたは付随情報に基づいて障害を検出してもよい。
- [0040] たとえば、検出部 3 1 1 # 0 ~ 検出部 3 1 1 # 2 は、ステータステーブル 3 0 3 # 0 に自身のドメインの診断結果を行った診断時刻と診断結果を 1 分毎に書き込む。書き込み後、検出部 3 1 1 # 0 ~ 検出部 3 1 1 # 2 は、自ドメイン内のステータステーブル 3 0 3 の診断時刻と診断結果に基づいて障害を検出する。なお、ステータステーブル 3 0 3 # 0 に書き込まれた内容は、転送部 3 1 2、格納部 3 1 7 の機能により、自ドメイン内のステータステーブル 3 0 3 に格納される。
- [0041] 具体的には、検出部 3 1 1 # 0 ~ 検出部 3 1 1 # 2 は、診断時刻が所定周期以上経過したドメインが存在する場合、他ドメインの障害を検出する。または、検出部 3 1 1 # 0 ~ 検出部 3 1 1 # 2 は、診断結果が NG となったドメインが存在する場合、他ドメインの障害を検出する。なお、検出結果は、検出部 3 1 1 の機能を有する CPU 2 0 4 のレジスタ、キャッシュメモリ、メモリ 1 0 9 などの記憶領域に格納されてもよい。
- [0042] 転送部 3 1 2 は、複数のプロセッサのうちの一のプロセッサ以外のプロセッサに更新されたコンテキストデータを転送する機能を有する。たとえば、転送部 3 1 2 は、マスタコンテキスト 3 0 1 のうち更新されたデータと、更新されたアドレスを CPU 2 0 4 # 1、CPU 2 0 4 # 2 に通知する。なお、通知された結果は、CPU 2 0 4 # 0 のレジスタ、キャッシュメモリ、メモリ 1 0 9 # 0 などの記憶領域に格納されてもよい。

- [0043] 判断部 3 1 3 は、検出部 3 1 1 # 1、検出部 3 1 1 # 2 によって障害が検出されるときに、複数のプロセッサを管理する状態にあると判断する機能を有する。たとえば、判断部 3 1 3 # 2 は、障害が検出された場合に、マスタ機構からの応答がない場合、ローカルマスタ機構を実行する CPU のうちいずれか一つの CPU を、複数のプロセッサを管理する状態にあると判断する。また、判断部 3 1 3 は、ローカルマスタ機構を実行する CPU を順位づけしておき、順位が最も高い CPU を複数のプロセッサを管理する状態にあると判断してもよい。
- [0044] また、判断部 3 1 3 は、さらに、マスタ機構に通知して障害がマスタ機構の障害であるか否かを判断してもよい。たとえば、判断部 3 1 3 # 2 は、バス 1 1 0 を経由してマスタ機構に設定された CPU 2 0 4 # 0 からの応答の有無によって、マスタ機構の障害の可能性のあることを判断する。
- [0045] さらに、判断部 3 1 3 は、マスタ機構への通知に対して応答がない場合に、マスタ機構が制御可能な通話装置へ発信してマスタ機構と切断された状態か否かを判断してもよい。具体的には、判断部 3 1 3 は、バス 1 1 0 を経由して CPU 2 0 4 # 0 からの応答がない場合、DBB 2 0 7 # 0 に発信してマスタ機構と切断された状態か否かを判断する。なお、判断結果は、判断部 3 1 3 の機能を有する CPU 2 0 4 のレジスタ、キャッシュメモリ、メモリ 1 0 9 などの記憶領域に格納されてもよい。
- [0046] 設定部 3 1 4 は、複数のプロセッサのうちの一のプロセッサに複数のプロセッサを管理するマスタ機構を設定する機能を有する。たとえば、設定部 3 1 4 は、CPU 2 0 4 # 0 ~ CPU 2 0 4 # 2 のうちドメイン 1 0 7 # 0 の CPU 2 0 4 # 0 にマスタ機構を設定する。なお、マスタ機構が設定された CPU は、マスタ機構スレッドを実行する。
- [0047] また、設定部 3 1 4 は、複数のプロセッサのうちの一のプロセッサ以外のプロセッサのそれぞれに、自身を管理するローカルマスタ機構と仮想マスタ機構とを設定する。たとえば、設定部 3 1 4 は、CPU 2 0 4 # 0 ~ CPU 2 0 4 # 2 のうち CPU 2 0 4 # 1 と CPU 2 0 4 # 2 に、ローカルマスタ

機構と仮想マスタ機構を設定する。なお、ローカルマスタ機構が設定されたCPUは、ローカルマスタ機構を実行する。同様に、仮想マスタ機構が設定されたCPUは、仮想マスタ機構スレッドを実行する。

[0048] また、設定部314は、検出部311#0によって障害が検出されるときに、少なくとも一部の処理を停止させる設定を行う。具体的には、設定部314は、障害が検出されたドメインに対して、該当ドメインのメモリ109に割り当てたメモリ空間解放を設定する。さらに、設定部314は、該当ドメインのメモリ109への書き込み禁止を設定し、さらに該当ドメインへのスレッド生成禁止を設定する。なお、設定された情報は、メモリ109、フラッシュROM201等の記憶領域に記憶される。

[0049] 通知部315は、マスタ機構によって管理される共有メモリが複数のプロセッサから連続したメモリとしてアクセスされるために、マスタ機構が複数のプロセッサにアドレスのオフセット値を通知する機能を有する。たとえば、メモリ109#0～メモリ109#2の物理アドレスが0x00～0xfffであることを想定する。このとき、通知部315は、CPU204#0にオフセット値0を通知し、CPU204#1にオフセット値0x100を通知し、CPU204#2にオフセット値0x200を通知する。なお、通知されるオフセット値は、通知先CPUが管理するメモリ109に格納される。また、通知元となる、CPU204#0のレジスタ、キャッシュメモリ、メモリ109#0などの記憶領域にオフセット値が格納されてもよい。

[0050] 割当部316は、通知部315によってオフセット値が通知された後に、オフセット値に基づいて共有メモリをローカルマスタ機構が管理するメモリ領域へ割り当てる機能を有する。

[0051] たとえば、割当部316#0は、メモリ109#0の物理アドレスに通知されたオフセット値0を加算して、メモリ109#0の論理アドレスを0x000～0x0ffで割り当てる。同様に、割当部316#1は、メモリ109#1の物理アドレスに通知されたオフセット値0x100を加算して、メモリ109#1の論理アドレスを0x100～0x1ffで割り当てる。

また、割当部 3 1 6 # 2 は、メモリ 1 0 9 # 2 の物理アドレスに通知されたオフセット値  $0 \times 2 0 0$  を加算して、メモリ 1 0 9 # 2 の論理アドレスを  $0 \times 2 0 0 \sim 0 \times 2 f f$  で割り当てる。なお、割り当てを行ったという情報は、割当部 3 1 6 の機能を有する CPU 2 0 4 のレジスタ、キャッシュメモリ、メモリ 1 0 9 などの記憶領域に格納されてもよい。

[0052] 格納部 3 1 7 は、転送部 3 1 2 から転送されるコンテキストデータをローカルマスタ機構が管理するメモリ領域に格納する機能を有する。たとえば、格納部 3 1 7 # 1 は、更新されたコンテキストデータをミラーリングマスタコンテキスト 3 0 2 # 1 に格納する。

[0053] また、格納部 3 1 7 は、マスタ機構に設定されたブレークポイントを検出したときに、マスタ機構のコンテキストデータをローカルマスタ機構が管理するメモリ領域に格納してもよい。たとえば、格納部 3 1 7 # 1 は、マスタコンテキスト 3 0 1 に設定されたブレークポイントに対するライトアクセスを検出したときに、更新されたコンテキストデータをミラーリングマスタコンテキスト 3 0 2 # 1 に格納する。

[0054] また、格納部 3 1 7 は、コンテキストデータをメモリ領域に格納する第 1 スレッドが実行される前に、前述のコンテキストデータがさらに更新されたコンテキストデータを格納する第 2 スレッドが実行されることが検出されるとき、第 1 スレッドを中止してもよい。

[0055] たとえば、第 1 スレッドが、アドレス  $0 \times 0 4 0$  に対して更新されたコンテキストデータを、ミラーリングマスタコンテキスト 3 0 2 # 1 に格納するスレッドであることを想定する。このとき、格納部 3 1 7 # 1 は、アドレス  $0 \times 0 4 0$  に対してさらに更新されたコンテキストデータをミラーリングマスタコンテキスト 3 0 2 # 1 に格納する第 2 スレッドが実行されることが検出されると、第 1 スレッドの処理を中止する。なお、格納を行ったという結果は、格納部 3 1 7 の機能を有する CPU 2 0 4 のレジスタ、キャッシュメモリ、メモリ 1 0 9 などの記憶領域に格納されてもよい。

[0056] 設定部 3 1 8 は、判断部 3 1 3 によって複数のプロセッサを管理する状態

にあると判断されるとき、ローカルマスタ機構を新たなマスタ機構に設定する機能を有する。たとえば、設定部 318#1は、CPU204#1がCPU204#0~CPU204#2を管理する状態にあると判断されるときに、CPU204#1で実行するローカルマスタ機構を新たなマスタ機構に設定する。なお、設定結果は、設定部 318の機能を有するCPU204のレジスタ、キャッシュメモリ、メモリ109などの記憶領域に格納されてもよい。

[0057] 終了部 319は、判断部 313によってマスタ機構と切断された状態であると判断された場合に、ローカルマスタ機構に設定されたプロセッサの動作を終了する機能を有する。たとえば、終了部 319#2は、バス110が切断され、CPU204#0と切断された状態であると判断された場合に、CPU204#2をシャットダウンし、動作を終了する。また、シャットダウンとなる対象は、ローカルマスタ機構に設定されたプロセッサが含まれるドメイン内の全ての装置に対してであってもよい。動作を終了するという履歴は、フラッシュROM201等の記憶領域に格納されてもよい。

[0058] 以上、システム100は、プロセッサと記憶領域とを含むドメインの集合のうち、システム100を制御する特定のドメインの障害を、ドメインの集合のうち特定のドメイン以外の他のドメインによって検出する検出部 311を含む。さらに、システム100は、特定のドメイン内の記憶領域に記憶された自システムのコンテキストを、制御結果が更新されるたびに、他のドメインの記憶領域へ転送する転送部 312を含む。検出部 311によって特定のドメインの障害が検出された場合、システム100は、転送部 312によって他のドメインの記憶領域に転送されたコンテキストを用いて、他のドメインを、自システムを制御する新たな特定のドメインに設定する設定部 318を含む。

[0059] 図4は、システム100のソフトウェアを示すブロック図である。システム100は、図2で示したハードウェアを用いて、図4で示すソフトウェアを実行する。CPU204#0は、OSが提供するソフトウェアとして、カ

ーネル４０１＃０と、マスタ機構スレッド４０２＃０と、ＯＳ管理部４０３と、通信連携部４０４と、スケジューラ４０５とを実行する。また、ＣＰＵ２０４＃１は、カーネル４０１＃１と、仮想マスタ機構スレッド４０６＃１と、ローカルマスタ機構スレッド４０７＃１と、ＯＳ管理部４０３とを実行する。同様に、ＣＰＵ２０４＃２は、カーネル４０１＃２と、仮想マスタ機構スレッド４０６＃２と、ローカルマスタ機構スレッド４０７＃２と、ＯＳ管理部４０３を実行する。また、ＯＳ管理部４０３は、プロセス管理部４０８、メモリ管理部４０９を含む。

[0060] また、システム１００は、ＯＳの機能を利用して、ユーザにサービスを提供するアプリとして、プロセス４１０～プロセス４１３を実行している。なお、ＣＰＵ２０４＃０がプロセス４１０を実行し、ＣＰＵ２０４＃１がプロセス４１１、プロセス４１２を実行し、ＣＰＵ２０４＃２がプロセス４１３を実行している。

[0061] カーネル４０１は、ＣＰＵ２０４＃０～ＣＰＵ２０４＃２を制御するプログラムである。カーネル４０１は、ＯＳの中核の機能であり、たとえば、システム１００のリソースを管理し、スレッドなどのソフトウェアがハードウェアにアクセスできるようにする。

[0062] また、カーネル４０１は、ＣＰＵ２０４＃０～ＣＰＵ２０４＃２のうち、いずれか１つのＣＰＵをマスタＣＰＵに設定し、マスタ機構スレッド４０２を起動させ、その他のＣＰＵに仮想マスタ機構スレッド４０６、ローカルマスタ機構スレッド４０７を起動させる。以降の説明では、マスタ機構スレッド４０２を実行しているＣＰＵが含まれるドメイン１０７をマスタドメインと定義し、ローカルマスタ機構スレッド４０７を実行しているＣＰＵが含まれるドメイン１０７をスレーブドメインと定義する。

[0063] また、カーネル４０１は、スレーブドメイン群に対して順位づけを行う。たとえば、カーネル４０１は、ドメイン１０７＃１をスレーブドメイン順位１と設定し、ドメイン１０７＃２をスレーブドメイン順位２と設定する。なお、ドメイン１０７内でのマスタドメイン設定、およびスレーブドメイン設

定は、ドメイン１０７＃０～ドメイン１０７＃２のうち、いずれのドメイン１０７がマスタドメインとなってもよい。たとえば、マスタドメインは、設計時点にて設計者によって設定されていてもよい。また、システム１００は、マスタドメインを、ディスプレイ１０５にメインメニューを表示するドメイン１０７に設定してもよい。

[0064] なお、ＳｏＣ内部バス２０９はバス１１０より高速である。このように、カーネル４０１は、システム１００を、仮想的なマルチコアプロセッサシステムとして、ユーザに提供する。また、カーネル４０１によって１つの仮想的なマルチコアプロセッサシステムとなったシステム１００は、ＣＰＵに専用のメモリを有するという、分散メモリ型のマルチコアプロセッサシステムという形態になる。

[0065] マスタ機構スレッド４０２は、システム１００全体の動作を制御するプログラムである。具体的に、マスタ機構スレッド４０２は、ペリフェラル２０５＃０～ペリフェラル２０５＃２、ディスプレイ１０５、キーボード１０６、フラッシュＲＯＭコントローラ２０２等の初期化処理を行ったり、時刻の設定処理を行ったりする。なお、マスタ機構スレッド４０２は、システム１００の起動情報、時刻情報、ペリフェラル２０５＃０～ペリフェラル２０５＃２等の情報、また、ＯＳ管理部４０３による実行結果を、マスタコンテキスト３０１に記録する。

[0066] ＯＳ管理部４０３は、システム１００を１コンピュータシステムとして提供するために、ドメイン１０７内ローカルの機能を有する。ＯＳ管理部４０３は前述の機能を満たすために、プロセス管理部４０８、メモリ管理部４０９を含む。具体的には、ＣＰＵ２０４＃０は、プロセス管理部４０８＃０とメモリ管理部４０９＃０を実行する。同様に、ＣＰＵ２０４＃１は、プロセス管理部４０８＃１とメモリ管理部４０９＃１を実行し、ＣＰＵ２０４＃２は、プロセス管理部４０８＃２とメモリ管理部４０９＃２を実行する。

[0067] 通信連携部４０４は、マスタドメインが実行する機能であり、自ドメインのＤＢＢ２０７と連携して通信機能を行う機能を有する。図４の例では、通

信連携部４０４は、ＤＢＢ２０７＃０と連携して通信機能を実行する。なお、他ドメインのＤＢＢ２０７は、使用しないため電源の供給を受けていない状態である。

[0068] スケジューラ４０５は、システム１００内のドメイン１０７＃０～ドメイン１０７＃２に割り当てるプロセスを決定する機能を有する。プロセスとは、ＣＰＵの処理するプログラムの実行単位である。具体的に、スケジューラ４０５は、プロセス４１０をドメイン１０７＃０に割り当て、プロセス４１１とプロセス４１２をドメイン１０７＃１に割り当て、プロセス４１３をドメイン１０７＃２に割り当てる。

[0069] 仮想マスタ機構スレッド４０６は、マスタ機構スレッド４０２が更新するマスタコンテキストを自ドメイン内のメモリ１０９にミラーリングする機能を有する。ミラーリングの形式としては、下記２つの方法が適用可能である。１つ目の処理方法は、ドメイン１０７間で同期を取り、マスタドメインが更新部分をスレーブドメインに通知する処理である。２つ目の処理方法は、ドメイン１０７間で同期を取らず、マスタドメインが更新部分をスレーブドメインに通知後他の処理に移り、スレーブドメインは、通知を受けて更新部分を反映する処理である。マスタコンテキスト転送処理１、マスタコンテキスト転送処理２の詳細については、図１３、図１４にて後述する。

[0070] ローカルマスタ機構スレッド４０７は、スレーブドメイン内の管理を行うスレッドである。たとえば、ローカルマスタ機構スレッド４０７＃１、ローカルマスタ機構スレッド４０７＃２は、マスタ機構スレッド４０２＃０から通知された処理内容を実行し、ログとして記録する。

[0071] また、マスタ機構スレッド４０２とローカルマスタ機構スレッド４０７は、自己診断回路２０８から診断ＯＫの通知を受け取り、自ドメインが動作中であることを他ドメインのマスタ機構スレッド４０２、またはローカルマスタ機構スレッド４０７に通知する。さらに、マスタ機構スレッド４０２とローカルマスタ機構スレッド４０７は、他ドメインが診断ＯＫか否かを判断する。

- [0072] なお、具体的には、マスタ機構スレッド４０２とローカルマスタ機構スレッド４０７は、ハードチェックスレッドを起動し、ハードチェックスレッドが定期的に自己診断回路２０８に診断を実行させる。また、ハードチェックスレッドは、自己診断回路２０８によって診断ＯＫを受けたドメインを生存ドメインに設定する。また、ハードチェックスレッドは、自己診断回路２０８によって診断ＮＧを受けたドメイン、または所定周期を経過しても自己診断回路２０８によって診断が実行されていないドメインを障害ドメインに設定する。また、具体的な他ドメインの診断判断方法については、図９にて後述する。
- [0073] プロセス管理部４０８は、ドメイン１０７に割り当てられたプロセスを管理する機能を有する。プロセス管理部４０８の動作モードは、マスタモードとスレーブモードが存在する。スレーブモードで動作するプロセス管理部４０８は、自ドメインに割り当てられたプロセスの生成、実行、終了を管理する。また、マスタモードで動作するプロセス管理部４０８＃０は、スレーブモードで動作する機能に加えて、システム１００のハードウェア等のリソースを管理する。
- [0074] たとえば、スレーブモードで動作するプロセス管理部４０８＃１は、ドメイン１０７＃１に割り当てられたプロセス４１１とプロセス４１２を時分割にしたがってＣＰＵ２０４＃１に割り当てる。また、マスタモードで動作するプロセス管理部４０８＃０は、たとえば、フラッシュＲＯＭコントローラ２０２へのアクセス権を、プロセス４１３に付与する。
- [0075] メモリ管理部４０９は、メモリ１０９＃０～メモリ１０９＃２を仮想共有メモリ空間４１４としてプロセス４１０～プロセス４１３に提供する機能を有する。たとえば、メモリ１０９＃０、メモリ１０９＃１、メモリ１０９＃２の物理アドレスが０×００～０×ｆｆであると想定する。
- [0076] このとき、メモリ管理部４０９＃０は、メモリ１０９＃０の論理アドレスを、メモリ１０９＃１の物理アドレスにオフセット値±０を加算した０×００～０×０ｆｆとしてプロセス４１０に提供する。さらに、メモリ管理部

409#0は、メモリ109#1の論理アドレスを、メモリ109#1の物理アドレスにオフセット値+0x100を加算した0x100~0xfffとしてプロセス410に提供する。

[0077] さらに、メモリ管理部409#0は、メモリ109#2の論理アドレスを、メモリ109#1の物理アドレスにオフセット値+0x200を加算した0x200~0x2ffとしてプロセス410に提供する。これにより、メモリ管理部409#0は、プロセス410に論理アドレスを0x000~0x2ffとなる仮想共有メモリ空間414を提供することができる。

[0078] 同様に、メモリ管理部409#1、メモリ管理部409#2も、メモリ109#0の論理アドレスを、物理アドレスにオフセット値±0を加算した値でプロセス410~プロセス413に提供する。続けて、メモリ管理部409#1、メモリ管理部409#2は、メモリ109#1の論理アドレスを、物理アドレスにオフセット値+0x100を加算した値でプロセス410~プロセス413に提供する。最後に、メモリ管理部409#1、メモリ管理部409#2は、メモリ109#2の論理アドレスを、物理アドレスにオフセット値+0x200を加算した値でプロセス410~プロセス413に提供する。

[0079] これにより、メモリ管理部409#1、メモリ管理部409#2も、プロセス410~プロセス413に論理アドレスを0x00~0x2ffとなる仮想共有メモリ空間414を提供することができる。

[0080] なお、オフセット値の設定方法としては、たとえば、メモリ管理部409は、マスタドメインのメモリ109の論理アドレスを0x00から開始するアドレスに設定する。もし、マスタドメインのメモリ109の物理アドレスが0x00より大きいアドレスで開始する場合、メモリ管理部409は、論理アドレス0x00となるオフセット値を設定する。続けて、メモリ管理部409は、スレーブドメイン順位1のメモリ109の論理アドレスのオフセット値を、マスタドメインのメモリ109の論理アドレスの終了アドレスに設定する。以下、メモリ管理部409は、スレーブドメイン順位nのメモリ

109の論理アドレスのオフセット値を、スレーブドメイン順位 $n-1$ のメモリ109の論理アドレスの終了アドレスに設定する。なお、 $n$ は1以上の整数である。

[0081] 以下、図4に示したソフトウェアの機能を利用して、図5A～図8Bにて、ドメイン107のいずれかが破損した場合の検出動作および復元動作を説明する。

[0082] 図5A、図5Bは、スレーブドメイン破損時による障害ドメイン検出動作を示す説明図である。図5Aでは、システム100が破損された状態を示している。図5Aに示すシステム100は、プロセス411によって生成された画面をディスプレイ105に表示中に、スレーブドメインとなるドメイン107#1が破損した状態を示している。

[0083] このとき、図5Bで示すマスタ機構スレッド402#0とローカルマスタ機構スレッド407#2は、ドメイン107#1が障害ドメインとなったことを検出する。検出後、スレーブドメインに属するローカルマスタ機構スレッド407#2は、バス110を経由して、マスタドメインが生存ドメインであるかを確認する。図5Aに示すシステム100は、マスタドメインとなるドメイン107#0が生存ドメインであるため、ローカルマスタ機構スレッド407#2は、障害ドメイン検出時の処理を終了し、通常処理を続行する。また、マスタドメインに属するマスタ機構スレッド402#0は、破損したドメイン107#1を障害ドメインとして検出すると、復元動作に移行する。後述する図6A、図6Bにて、スレーブドメイン破損後の復元動作について説明する。

[0084] 図6A、図6Bは、スレーブドメイン破損後の復元動作を示す説明図である。図6Aに示すシステム100は、スレーブドメインとなるドメイン107#1が破損し障害ドメインとなったことを、マスタドメインに属するマスタ機構スレッド402#0が検出した状態である。検出後、マスタ機構スレッド402#0は、ドメイン107#1に属するメモリ109#1の論理アドレス $0 \times 100 \sim 0 \times 1ff$ を書き込み禁止に設定する。

- [0085] 書き込み禁止に設定することで、たとえばドメイン１０７＃１がソフトウェア暴走によって障害ドメインとなった場合に、システム１００は、暴走中のソフトウェアによるメモリ破壊、自動リセットによるメモリクリアを防止できる。なお、メモリ１０９＃１への読み込みについては許可される。これにより、たとえばドメイン１０７＃１で暴走していないソフトウェアがメモリ１０９＃１に出力したデータを、生存ドメインで実行中のソフトウェアが読み込むことが可能になるため、生存ドメインで実行中のソフトウェアの処理を継続することができる。
- [0086] 次に、マスタ機構スレッド４０２＃０は、障害ドメインへのスレッド生成禁止処理を実行する。なお、障害ドメインへのスレッド生成禁止は、障害ドメインが修復されるまで継続される。これにより、システム１００は、２ドメインで動作を継続できる。たとえば、ユーザは、システム１００のうち破損したドメインを含むユニットを修理にだしつつ、残ったユニットでシステム１００を稼働し続ける、といった運用が可能になる。図６Ａの例では、システム１００は、図５Ａにて示したプロセス４１１を終了し、メインメニューをディスプレイ１０５に表示している。
- [0087] 図７Ａ、図７Ｂは、マスタドメイン破損時の動作を示す説明図である。図７Ａに示すシステム１００は、ディスプレイユニット１０３に衝撃等が入り、マスタドメインとなるドメイン１０７＃０が破損した状態を示している。
- [0088] このとき、図７Ｂで示すローカルマスタ機構スレッド４０７＃１とローカルマスタ機構スレッド４０７＃２は、ドメイン１０７＃０が破損し、ドメイン１０７＃０が障害ドメインとなったことを検出する。検出後、ローカルマスタ機構スレッド４０７＃１、ローカルマスタ機構スレッド４０７＃２は、それぞれ、バス１１０を経由して、マスタドメインの生存確認を行う。図７Ｂに示すシステム１００ではマスタドメインとなるドメイン１０７＃０が破損しているため、バス１１０を経由した生存確認は失敗する。
- [0089] 続けて、ローカルマスタ機構スレッド４０７＃１、ローカルマスタ機構スレッド４０７＃２は、ＤＢＢ２０７を利用して、マスタドメインの生存確認

を行う。これにより、システム１００は、バス１１０が断線したためマスタドメインと通信が行えなくなっている状態か否かを判断できる。

[0090] 具体的なＤＢＢ２０７を利用した生存確認とは、ローカルマスタ機構スレッド４０７＃１が、ＤＢＢ２０７＃１から、ＤＢＢ２０７＃０に対して通信発信する。ローカルマスタ機構スレッド４０７＃１は、発信に対して応答があればマスタドメインが生存していると判断し、発信に対して応答がなければマスタドメインが破損していると判断する。同様に、ローカルマスタ機構スレッド４０７＃２は、ＤＢＢ２０７＃２から、ＤＢＢ２０７＃０に対して通信発信することで、マスタドメインの生存確認を行う。

[0091] 図７Ｂに示すシステム１００ではマスタドメインとなるドメイン１０７＃０が破損しているため、ＤＢＢ２０７＃０は、ＤＢＢ２０７＃１とＤＢＢ２０７＃２に対する通信発信に応答できない。したがって、ローカルマスタ機構スレッド４０７＃１とローカルマスタ機構スレッド４０７＃２は、マスタドメインが破損していると判断する。判断後、スレーブドメインの順位が最上位となるドメイン１０７＃１が、新たなマスタドメインとして運用を続行することになる。

[0092] 具体的に、ローカルマスタ機構スレッド４０７＃１は、仮想マスタ機構スレッド４０６＃１を終了し、ローカルマスタ機構スレッド４０７＃１の出力先として、ミラーリングマスタコンテキスト３０２＃１を新たなマスタコンテキストに設定する。続けて、ローカルマスタ機構スレッド４０７＃１は、ローカルマスタ機構スレッド４０７＃１をマスタ機構スレッド４０２＃１として起動処理を実行する。これにより、システム１００は、スレーブドメインであったドメイン１０７＃１をマスタドメインとして設定し、運用を続行することができる。

[0093] 図８Ａ、図８Ｂは、分離状態で破損した場合の動作を示す説明図である。図８Ａに示すシステム１００は、スレーブドメインとなるドメイン１０７＃１とバス１１０が破損した状態を示している。なお、図８Ａでは、バス１１０が有線であり、バス１１０が切断された状態を示している。もし、バス１

10として無線が採用されていた場合、図8Aに示すシステム100は、近接無線通信部が破損し、ドメイン間通信が行えない状態として想定する。

[0094] このとき、図8Bに示すマスタ機構スレッド402#0とローカルマスタ機構スレッド407#2は、ドメイン107#1の破損を検出する。検出後、スレーブドメインに属するローカルマスタ機構スレッド407#2は、バス110を経由して、マスタドメインが生存しているかを確認する。図8Bに示すシステム100には、マスタドメインとなるドメイン107#0が生存しているが、バス110が切断されているため、ローカルマスタ機構スレッド407#2は、ドメイン107#0の生存を確認できず、生存していないと判断する。続けて、ローカルマスタ機構スレッド407#2は、DBB207#2を利用して、マスタドメインの生存確認を行う。

[0095] 図8Bに示すシステム100ではマスタドメインとなるドメイン107#0が生存しているため、DBB207#2に対する通信発信に応答できる。したがって、ローカルマスタ機構スレッド407#2は、マスタドメインが生存していると判断する。これにより、ローカルマスタ機構スレッド407#2はバス110が断線したためマスタドメインと通信が行えなくなっている状態であると判断する。この状態では、ドメイン107#2は、マスタドメインとなるドメイン107#0の指示を受けることができなくなるため、シャットダウンし、動作を終了する。

[0096] なお、マスタドメインとバス110に障害が発生し、分離状態となる場合もありうる。たとえば、図8Bの状態、ドメイン107#1がマスタドメインと設定されており、ドメイン107#0がスレーブドメイン順位1と設定されており、ドメイン107#2がスレーブドメイン順位2と設定された状態を想定する。このとき、スレーブドメイン順位1であるドメイン107#0が新たなマスタドメインとなり、起動処理を再実行する。起動時に、ドメイン107#0は、スレーブドメイン順位2であったドメイン107#2と通信できないため、起動処理時にシステム100から切り離されることになる。

- [0097] 図9は、ステータステーブル303の記憶内容の一例を示す説明図である。なお、図9で示すステータステーブル303は、現在時刻が12:34:00以降の状態を示している。ステータステーブル303は、ドメイン種別、診断時刻、診断結果という3つのフィールドを含む。ドメイン種別フィールドには、ドメイン107の種別として、マスタドメインか、スレーブドメインか、またスレーブドメインであればスレーブドメイン順位が格納される。なお、ドメイン種別フィールドは、ドメイン107が一意に特定できればどのような値であってもよいので、たとえば、ドメイン107のID (Identification) であってもよい。
- [0098] 診断時刻フィールドには、自己診断回路208が診断を行った時刻が格納される。診断結果フィールドには、自己診断回路208による診断結果が格納される。たとえば、システム100は、所定周期となる1分毎に自己診断回路208にて診断を行うことを想定している。この状態で、図9に示すステータステーブル303について、マスタドメインの自己診断回路208が、12:34:00にマスタドメインを診断し、診断結果がOKであることを示している。同様に、スレーブドメイン順位1の自己診断回路208が、12:34:00にスレーブドメイン順位1を診断し、診断結果がOKであることを示している。また、スレーブドメイン順位2の自己診断回路208が、12:33:00にスレーブドメイン順位2を診断し、診断結果がOKであることを示している。
- [0099] また、スレーブドメイン順位2について12:34:00に診断が行われていないため、マスタドメインとスレーブドメイン順位1は、スレーブドメイン順位2が障害ドメインとなったことを検出する。また、図9には図示していないが、自己診断回路208の診断の結果、診断結果がNGとなり、診断結果フィールドにNGが記載される場合も存在する。このような場合も、他ドメインは診断結果フィールドにNGと記載されているドメインを障害ドメインとして検出する。
- [0100] 図10は、起動処理を示すフローチャートである。システム100は、電

源投入される（ステップS 1 0 0 1）と、マスタドメイン、スレーブドメイン順位 1 ～スレーブドメイン順位 n をブート開始する（ステップS 1 0 0 2 ～ステップS 1 0 0 4）。以降、マスタドメインがステップS 1 0 0 5 ～ステップS 1 0 0 9 を実行し、スレーブドメイン順位 1 がステップS 1 0 1 0 ～ステップS 1 0 1 4 を実行し、スレーブドメイン順位 n がステップS 1 0 1 5 ～ステップS 1 0 1 9 を実行する。なお、スレーブドメイン順位 n によるステップS 1 0 1 5 ～ステップS 1 0 1 9 の処理は、ステップS 1 0 1 0 ～ステップS 1 0 1 4 と等しいため、説明を省略する。

[0101] マスタドメインは、仮想共有メモリ空間 4 1 4 を初期化する（ステップS 1 0 0 5）。初期化後、マスタドメインは、アドレスのオフセット値を通知する（ステップS 1 0 0 6）。通知後、マスタドメインは、メモリ 1 0 9 の物理アドレスからのオフセット値で設定される論理アドレスにメモリ 1 0 9 を割り当てる（ステップS 1 0 0 7）。割り当て後、マスタドメインは、プロセス管理部 4 0 8 をマスタモードで起動し（ステップS 1 0 0 8）、さらに、マスタ機構スレッド 4 0 2 を起動し（ステップS 1 0 0 9）、起動処理を終了する。マスタドメインは、起動処理終了後、運用時処理に遷移する。

[0102] また、スレーブドメイン順位 1 は、オフセット値の通知を受け付けたか否かを判断する（ステップS 1 0 1 0）。オフセット値の通知を受け付けていない場合（ステップS 1 0 1 0 : N o）、スレーブドメイン順位 1 は、再度ステップS 1 0 1 0 の処理を実行する。オフセット値の通知を受け付けた場合（ステップS 1 0 1 0 : Y e s）、スレーブドメイン順位 1 は、メモリ 1 0 9 の物理アドレスからのオフセット値で設定される論理アドレスにメモリ 1 0 9 を割り当てる（ステップS 1 0 1 1）。割り当て後、スレーブドメイン順位 1 は、プロセス管理部 4 0 8 をスレーブモードで起動する（ステップS 1 0 1 2）。

[0103] 続けて、スレーブドメイン順位 1 は、仮想マスタ機構スレッド 4 0 6 を起動し（ステップS 1 0 1 3）、さらに、ローカルマスタ機構スレッド 4 0 7 を起動し（ステップS 1 0 1 4）、起動処理を終了する。スレーブドメイン

順位 1 は、起動処理終了後、運用時処理に遷移する。

[0104] 起動処理が終了した、マスタドメイン、スレーブドメインは、運用時処理に遷移する。図 1 1、図 1 2 では、マスタドメイン、スレーブドメインの運用時処理を説明する。マスタドメインは、運用時処理をマスタ機構スレッド 4 0 2 による機能によって実行する。同様に、スレーブドメインは、運用時処理をローカルマスタ機構スレッド 4 0 7、仮想マスタ機構スレッド 4 0 6 による機能によって実行する。

[0105] また、ステップ S 1 1 0 3、ステップ S 1 1 0 4、ステップ S 1 1 0 9、ステップ S 1 1 1 0 の処理は、スレーブドメイン順位 1 ~ スレーブドメイン順位 n のうち、いずれかのスレーブドメインが実行する。図 1 1 では、説明の簡略化のため、スレーブドメイン順位 1 が実行すると仮定して説明を行う。なお、図 1 2 で示すステップ S 1 2 0 1、ステップ S 1 2 0 4、ステップ S 1 2 0 5 の処理は、スレーブドメイン順位 1 ~ スレーブドメイン順位 n 全てのスレーブドメインが実行する。

[0106] 図 1 1 は、マスタドメイン、スレーブドメインの運用時処理を示すフローチャート（その 1）である。マスタ機構スレッド 4 0 2 は、他ドメインへのメモリアクセスが発生したか否かを判断する（ステップ S 1 1 0 1）。他ドメインへのメモリアクセスが発生した場合（ステップ S 1 1 0 1 : Y e s）、マスタ機構スレッド 4 0 2 は、メモリ管理部 4 0 9 により、他ドメインのメモリ 1 0 9 へのアクセス要求を通知する（ステップ S 1 1 0 2）。

[0107] ステップ S 1 1 0 3、ステップ S 1 1 0 4 では、他ドメインがスレーブドメイン順位 1 である場合を想定する。ローカルマスタ機構スレッド 4 0 7 # 1 は、自ドメインへのメモリアクセス要求を受け付けたか否かを判断する（ステップ S 1 1 0 3）。メモリアクセス要求を受け付けた場合（ステップ S 1 1 0 3 : Y e s）、ローカルマスタ機構スレッド 4 0 7 # 1 は、メモリアクセス応答を行い（ステップ S 1 1 0 4）、要求元に応答の結果をマスタ機構スレッド 4 0 2 に送信する。応答の結果を送信後、またはメモリアクセス要求を受け付けていない場合（ステップ S 1 1 0 3 : N o）、ローカルマ

タ機構スレッド407#1は、ステップS1109の処理に移行する。

[0108] アクセスに対する応答受信後、または他ドメインへのメモリアクセスが発生していない場合（ステップS1101：No）、マスタ機構スレッド402は、ユーザの操作を受け付けたかを判断する（ステップS1105）。操作を受け付けた場合（ステップS1105：Yes）、マスタ機構スレッド402は、ユーザの操作受付処理を実行する（ステップS1106）。なお、具体的なユーザの操作受付処理とは、たとえば、キーボード106の操作によってGUI上で選択されたアプリの起動処理、またはアプリの終了処理等を実行する。操作受付処理を実行後、マスタ機構スレッド402は、ユーザの操作によってスレッド生成要求が発生したか否かを判断する（ステップS1107）。

[0109] スレッド生成要求が発生した場合（ステップS1107：Yes）、マスタ機構スレッド402は、スケジューラ405により、いずれかのスレーブドメインへスレッド生成要求を通知する（ステップS1108）。なお、スケジューラ405によるドメインの選択方法は、たとえば、スレーブドメイン順位1～スレーブドメイン順位n内のCPU204のうち、最も負荷の低いCPUを含むドメインであってもよい。図11では、スレーブドメイン順位1に通知した場合を想定する。

[0110] ローカルマスタ機構スレッド407#1は、自ドメインへのスレッド生成要求を受け付けたか否かを判断する（ステップS1109）。スレッド生成要求を受け付けた場合（ステップS1109：Yes）、ローカルマスタ機構スレッド407#1は、スレッド生成処理を行う（ステップS1110）。生成終了後、またはスレッド生成要求を受け付けていない場合（ステップS1109：No）、ローカルマスタ機構スレッド407#1は、図12で示すステップS1204に移行する。

[0111] スレッド生成要求通知後か、スレッド生成要求が発生していない場合（ステップS1107：No）か、または操作を受け付けていない場合（ステップS1105：No）、マスタ機構スレッド402は、ステップS1201

の処理に移行する。

- [0112] 図12は、マスタドメイン、スレーブドメインの運用時処理を示すフローチャート（その2）である。マスタ機構スレッド402、仮想マスタ機構スレッド406#1～仮想マスタ機構スレッド406#nは、マスタコンテキスト転送処理を実行する（ステップS1201）。マスタコンテキスト転送処理の詳細は、図13、図14にて後述する。処理終了後、マスタ機構スレッド402は、所定周期が経過したかを判断する（ステップS1202）。所定周期が経過した場合（ステップS1202：Yes）、マスタ機構スレッド402は、ハードチェックスレッド処理（マスタ）を実行する（ステップS1203）。ハードチェックスレッド処理（マスタ）の詳細は、図15にて後述する。
- [0113] 処理終了後、または所定周期が経過していない場合（ステップS1202：No）、マスタ機構スレッド402は、ステップS1101の処理に移行する。仮想マスタ機構スレッド406#1～仮想マスタ機構スレッド406#nは、マスタコンテキスト転送処理の終了後、再びステップS1201の処理を実行する。
- [0114] また、ローカルマスタ機構スレッド407#1～ローカルマスタ機構スレッド407#nは、所定周期が経過したか否かを判断する（ステップS1204）。所定周期が経過した場合（ステップS1204：Yes）、ローカルマスタ機構スレッド407#1～ローカルマスタ機構スレッド407#nは、ハードチェックスレッド処理（スレーブ）を実行する（ステップS1205）。処理終了後、または所定周期が経過していない場合（ステップS1204：No）、ローカルマスタ機構スレッド407#1～ローカルマスタ機構スレッド407#nは、ステップS1103の処理に移行する。
- [0115] 図13、図14では、マスタコンテキスト転送処理の説明を行う。システム100が実行するマスタコンテキスト転送処理は、図13に示すマスタコンテキスト転送処理1か、または図14に示すマスタコンテキスト転送処理2のいずれか一方の処理が適用される。

- [0116] 図13は、マスタコンテキスト転送処理1を示すフローチャートである。マスタ機構スレッド402は、マスタコンテキスト301に対する書込イベントが発生したかを判断する（ステップS1301）。書込イベントが発生した場合（ステップS1301：Yes）、マスタ機構スレッド402は、メモリ更新要求を仮想マスタ機構スレッド406#1～仮想マスタ機構スレッド406#nに通知する（ステップS1302）。通知後、マスタ機構スレッド402は、バリア同期を発行し、メモリロックする（ステップS1303）。
- [0117] なお、バリア同期を行ったマスタ機構スレッド402は、同一のバリア同期グループとなる仮想マスタ機構スレッド406#1～仮想マスタ機構スレッド406#nがバリア同期を実行するまで待機する。具体的に、マスタ機構スレッド402は、仮想マスタ機構スレッド406#1がステップS1305の処理を実行し、かつ、仮想マスタ機構スレッド406#nがステップS1307を実行するまで待機する。
- [0118] 続けて、仮想マスタ機構スレッド406#1は、メモリ更新要求を受け付けたか否かを判断する（ステップS1304）。メモリ更新要求を受け付けていない場合（ステップS1304：No）、仮想マスタ機構スレッド406#1は、再びステップS1304の処理を実行する。メモリ更新要求を受け付けている場合（ステップS1304：Yes）、仮想マスタ機構スレッド406#1は、バリア同期を発行し、メモリロックする（ステップS1305）。同様に、仮想マスタ機構スレッド406#nも、ステップS1306、ステップS1307を実行する。なお、ステップS1306、ステップS1307の処理は、ステップS1304、ステップS1305の処理と等しいため、説明を省略する。
- [0119] マスタ機構スレッド402、仮想マスタ機構スレッド406#1～仮想マスタ機構スレッド406#n全てがバリア同期を発行した場合、マスタ機構スレッド402は、更新されたマスタコンテキストデータを転送する（ステップS1308）。転送先は、仮想マスタ機構スレッド406#1～仮想マ

スタ機構スレッド406#nとなる。転送後、マスタ機構スレッド402は、メモリアンロックし、バリア同期を発行する（ステップS1309）。

[0120] 続けて、仮想マスタ機構スレッド406#1は、受信したマスタコンテキストデータをミラーリングマスタコンテキスト302#1に格納し（ステップS1310）、格納後、メモリアンロックし、バリア同期を発行する（ステップS1311）。同様に、仮想マスタ機構スレッド406#nは、受信したマスタコンテキストデータをミラーリングマスタコンテキスト302#nに格納し（ステップS1312）、格納後、メモリアンロックし、バリア同期を発行する（ステップS1313）。

[0121] マスタ機構スレッド402、仮想マスタ機構スレッド406#1～#n全てがバリア同期を発行した場合、マスタ機構スレッド402、仮想マスタ機構スレッド406#1～仮想マスタ機構スレッド406#nは、マスタコンテキスト転送処理1を終了する。また、書込イベントが発生していない場合（ステップS1301：No）、マスタ機構スレッド402は、マスタコンテキスト転送処理1を終了する。

[0122] このように、マスタコンテキスト転送処理1は、マスタコンテキスト301内の同期部分の転送領域を更新箇所に絞ることで、データ転送量を抑えることを実現している。なお、マスタコンテキスト転送処理1は、メモリロック期間が発生するため、バス110の転送速度が高速であることが好ましい。

[0123] 図14は、マスタコンテキスト転送処理2を示すフローチャートである。なお、図14の説明において、仮想マスタ機構スレッド406#1～仮想マスタ機構スレッド406#nによって実行されるステップS1403～ステップS1405の処理は同一である。したがって、図14の説明では、仮想マスタ機構スレッド406#1の実行例を想定して説明する。

[0124] マスタ機構スレッド402は、マスタコンテキスト301に対する書込イベントが発生したか否かを判断する（ステップS1401）。書込イベントが発生した場合（ステップS1401：Yes）、マスタ機構スレッド40

2は、マスタコンテキスト301を更新する（ステップS1402）。更新後、または書込イベントが発生していない場合（ステップS1401：No）、マスタ機構スレッド402は、マスタコンテキスト転送処理2を終了する。

[0125] また、仮想マスタ機構スレッド406#1は、マスタコンテキスト301の記憶領域に、ブレークポイントを設定する（ステップS1403）。なお、ステップS1403で設定するブレークポイントは、ライトアクセスに反応するブレークポイントであることが好ましい。また、ブレークポイントの設定後、仮想マスタ機構スレッド406#1は、ブレークポイントに対してライトアクセスが発生したか否か判断する（ステップS1404）。

[0126] ライトアクセスが発生していない場合（ステップS1404：No）、仮想マスタ機構スレッド406#1は、再びステップS1404の処理に移行する。ライトアクセスが発生した場合（ステップS1404：Yes）、仮想マスタ機構スレッド406#1は、ライトアクセスが発生した転送領域アドレスと更新されたマスタコンテキストデータを引数として、コンテキストデータ転送スレッドを生成し起動する（ステップS1405）。起動後、仮想マスタ機構スレッド406#1は、マスタコンテキスト転送処理2を終了する。

[0127] なお、コンテキストデータ転送スレッドは、ステップS1405が実行されるたびに新たに生成される。また、コンテキストデータ転送スレッドは各ドメインで個別に生成される。以下の説明では、仮想マスタ機構スレッド406#1が属するスレーブドメインで起動されたコンテキストデータ転送スレッドであることを想定して説明を行う。

[0128] 新たに起動されたコンテキストデータ転送スレッドは、自身のスレッドIDと、転送領域アドレスを、転送領域管理テーブルに格納する（ステップS1406）。なお、転送領域管理テーブルは、スレッドIDフィールドと、転送領域アドレスフィールドを含むテーブルである。格納後、コンテキストデータ転送スレッドは、転送領域アドレスが一致する既存のコンテキストデ

ータ転送スレッドが存在するか否かを判断する（ステップS 1 4 0 7）。

[0129] 既存のコンテキストデータ転送スレッドが存在する場合（ステップS 1 4 0 7 : Y e s）、コンテキストデータ転送スレッドは、既存のコンテキストデータ転送スレッドの終了通知を発行する（ステップS 1 4 0 8）。具体的な発行方法としては、コンテキストデータ転送スレッドは、OS等が提供しているAPI群のうちスレッドを終了するAPIを、転送領域管理テーブルのスレッドIDを引数として実行する。

[0130] 既存のコンテキストデータ転送スレッドが存在しない場合（ステップS 1 4 0 7 : N o）、コンテキストデータ転送スレッドは、受信したマスタコンテキストデータをミラーリングマスタコンテキスト3 0 2 # 1に格納する（ステップS 1 4 0 9）。格納後、コンテキストデータ転送スレッドは、転送領域管理テーブルから自身のスレッドIDが設定されたレコードを削除し（ステップS 1 4 1 0）、スレッド実行を終了する。

[0131] このように、マスタコンテキスト転送処理2は、バス1 1 0の転送速度が遅く、頻繁にステータステーブル3 0 3が更新される場合でも対応可能である。具体的に、マスタコンテキスト転送処理2を適用したシステム1 0 0は、受信したマスタコンテキストデータをミラーリングマスタコンテキスト3 0 2 # 1に格納する前に、マスタコンテキストが書き換わってもよい。

[0132] 図1 5は、ハードチェックスレッド（マスタ）の処理を示すフローチャートである。図1 5では、マスタドメインで実行されるハードチェックスレッドの処理を示す。ハードチェックスレッドは、障害ドメイン検出処理を実行する（ステップS 1 5 0 1）。障害ドメイン検出処理の詳細は、図1 7にて後述する。処理後、ハードチェックスレッドは、障害ドメインが検出されたか否かを判断する（ステップS 1 5 0 2）。障害ドメインが検出された場合（ステップS 1 5 0 2 : Y e s）、ハードチェックスレッドは、障害ドメインのメモリ1 0 9に割り当てた仮想メモリ空間解放を設定する（ステップS 1 5 0 3）。

[0133] 続けて、ハードチェックスレッドは、障害ドメインのメモリ1 0 9への書

き込み禁止を設定し（ステップS 1 5 0 4）、さらに障害ドメインへのスレッド生成禁止を設定し（ステップS 1 5 0 5）、スレッド実行を終了する。なお、障害ドメインが複数検出される場合も存在する。この場合、ハードチェックスレッドは、検出された全ての障害ドメインに対して、ステップS 1 5 0 3～ステップS 1 5 0 5の処理を実行する。また、障害ドメインが検出されなかった場合（ステップS 1 5 0 2 : N o）、ハードチェックスレッドは、スレッド実行を終了する。

[0134] 図 1 6 は、ハードチェックスレッド（スレーブ）処理を示すフローチャートである。図 1 6 では、スレーブドメイン順位 1～スレーブドメイン順位 n で実行されるハードチェックスレッドの処理を示す。ハードチェックスレッドは、障害ドメイン検出処理を実行する（ステップS 1 6 0 1）。障害ドメイン検出処理の詳細は、図 1 7 にて後述する。処理後、ハードチェックスレッドは、障害ドメインが検出されたか否かを判断する（ステップS 1 6 0 2）。

[0135] 障害ドメインが検出された場合（ステップS 1 6 0 2 : Y e s）、ハードチェックスレッドは、マスタドメインがバス 1 1 0 を経由して通信可能か否かを判断する（ステップS 1 6 0 3）。なお、通信可能か否かの判断方法として、ハードチェックスレッドは、1 0 0 プロセッサ間通信である I P C（Inter Processor Communication）等の A P I を用いて判断することができる。

[0136] マスタドメインが通信可能でない場合（ステップS 1 6 0 3 : N o）、ハードチェックスレッドは、自ドメインの D B B 2 0 7 をアクティベートし（ステップS 1 6 0 4）、マスタドメインの D B B 2 0 7 に通信発信する（ステップS 1 6 0 5）。なお、各ドメインの D B B 2 0 7 は、それぞれ固有の I D を有しており、固有の I D を用いて発信することで、システム 1 0 0 内の他の D B B 2 0 7 に発信することができる。

[0137] 発信後、ハードチェックスレッドは、マスタドメインから応答があったか否かを判断する（ステップS 1 6 0 6）。マスタドメインから応答があった

場合（ステップS 1 6 0 6 : Y e s）、ハードチェックスレッドは、自ドメインをシャットダウンする（ステップS 1 6 0 7）。ステップS 1 6 0 7の処理後、自ドメインをシャットダウンするため、ハードチェックスレッドは、スレッド実行を終了する。なお、ステップS 1 6 0 6 : Y e sとなった場合とは、マスタドメインと物理的に切断されてしまったケースである。

[0138] マスタドメインから応答がない場合（ステップS 1 6 0 6 : N o）、ハードチェックスレッドは、自ドメインが最高位の生存ドメインか否かを判断する（ステップS 1 6 0 8）。自ドメインが最高位の生存ドメインである場合（ステップS 1 6 0 8 : Y e s）、ハードチェックスレッドは、仮想マスタ機構スレッド4 0 6を終了させる（ステップS 1 6 0 9）。なお、ステップS 1 6 0 8 : Y e sとなった場合とは、マスタドメインが障害ドメインとなり、自ドメインがマスタドメインの代替となるケースである。

[0139] 続けて、ハードチェックスレッドは、ローカルマスタ機構スレッド4 0 7にミラーリングマスタコンテキスト3 0 2を括り付ける（ステップS 1 6 1 0）。具体的に、ハードチェックスレッドは、ローカルマスタ機構スレッド4 0 7がアクセスするデータを、ローカルマスタ機構スレッド4 0 7が出力していたログからミラーリングマスタコンテキスト3 0 2へと切り替える。

[0140] 括り付け後、ハードチェックスレッドは、ローカルマスタ機構スレッド4 0 7を新たなマスタ機構スレッド4 0 2として起動する（ステップS 1 6 1 1）。なお、新たなマスタ機構スレッド4 0 2は、起動処理として、ステップS 1 0 0 5の処理から実行を開始する。起動後、ハードチェックスレッドは、スレッド実行を終了する。

[0141] 自ドメインが最高位の生存ドメインでない場合（ステップS 1 6 0 8 : N o）、ハードチェックスレッドは、スレッド実行を終了する。なお、ステップS 1 6 0 8 : N oとなった場合とは、マスタドメインが障害ドメインとなり、他のスレーブドメインがマスタドメインとなり、自ドメインは引き続きスレーブドメインとして動作するケースである。

[0142] マスタドメインが通信可能である場合（ステップS 1 6 0 3 : Y e s）、

ハードチェックスレッドは、スレッド実行を終了する。ステップS 1 6 0 3 : Y e s となった場合とは、スレーブドメインのいずれかが障害ドメインとなったケースである。障害ドメインが検出されていない場合（ステップS 1 6 0 2 : N o ）、ハードチェックスレッドは、スレッド実行を終了する。

[0143] 図 1 7 は、障害ドメイン検出処理を示すフローチャートである。障害ドメイン検出処理は、マスタドメイン、スレーブドメイン順位 1 ~スレーブドメイン順位 n が実行するハードチェックスレッドにて実行される。

[0144] ハードチェックスレッドは、自ドメイン内の自己診断回路 2 0 8 を起動し、診断実行させる（ステップS 1 7 0 1 ）。続けて、ハードチェックスレッドは、ステータステーブル 3 0 3 # 0 のうち、ドメイン種別フィールドが自ドメインであるレコードを選択する（ステップS 1 7 0 2 ）。選択後、ハードチェックスレッドは、自己診断回路 2 0 8 の結果を、選択したレコードの診断結果フィールドに書き込む（ステップS 1 7 0 3 ）。書き込み後、ハードチェックスレッドは、自ドメイン内のステータステーブル 3 0 3 のレコードのうち、診断結果が N G となるレコードが存在するかを判断する（ステップS 1 7 0 4 ）。N G となるレコードが存在する場合（ステップS 1 7 0 4 : Y e s ）、ハードチェックスレッドは、存在したレコードに対するドメインを障害ドメインとして検出する（ステップS 1 7 0 5 ）。

[0145] 障害ドメインを検出後、または N G となるレコードが存在しない場合（ステップS 1 7 0 4 : N o ）、ハードチェックスレッドは、診断時刻フィールドが所定周期以上更新されていないレコードが存在するか否かを判断する（ステップS 1 7 0 6 ）。所定周期以上更新されていないレコードが存在しない場合（ステップS 1 7 0 6 : N o ）、ハードチェックスレッドは、障害ドメイン検出処理を終了する。

[0146] または所定周期以上更新されていないレコードが存在する場合（ステップS 1 7 0 6 : Y e s ）、ハードチェックスレッドは、存在したレコードに対するドメインを障害ドメインとして検出する（ステップS 1 7 0 7 ）。出力後、ハードチェックスレッドは、障害ドメイン検出処理を終了する。

- [0147] 以上説明したように、プロセッサ管理方法によれば、安価なS o Cとメモリを含むドメイン群でシステムを構築し、各ドメインのメモリにオフセット値を設定してメモリ空間を連結する。これにより、プロセッサ管理方法を適用したシステムは、各ドメインは広いメモリ空間を利用でき、安価で高性能なシステムをユーザに提供できる。メモリ空間が広くなることで、システムは複数のアプリを同時に実行することが容易になる。
- [0148] また、プロセッサ管理方法によれば、更新されたコンテキストデータをマスタドメインからスレーブドメインに転送し、スレーブドメイン内のメモリに格納してもよい。これにより、マスタドメインに障害が発生した場合、スレーブドメインは、自身のメモリ内にコンテキストデータを用いて、システムを続行することができる。さらに、システムは、更新箇所に絞れるため、転送量を抑えることができる。
- [0149] また、プロセッサ管理方法によれば、マスタドメイン内に設定したブレークポイントに対するライトアクセスを検出したときに、ブレークポイントに対する更新されたコンテキストデータをスレーブドメイン内のメモリに格納してもよい。これにより、マスタドメインに障害が発生した場合、スレーブドメインは、自身のメモリ内にコンテキストデータを用いて、システムを続行することができる。さらに、システムの形態が、ドメイン間のバスが遅く、頻繁にマスタコンテキストが変更になる場合でも対応することができる。
- [0150] また、プロセッサ管理方法によれば、ブレークポイントに対するデータをスレーブドメイン内のメモリに格納する第1スレッドが実行完了する前に、同一のアドレスに対してメモリに格納する第2スレッドを検出した場合、第1スレッドの処理を中止してもよい。これにより、ブレークポイントを利用した更新方法の場合について、余計な処理を行わずに済ませることができる。
- [0151] また、プロセッサ管理方法によれば、他ドメインの障害を検出してもよい。これにより、マスタドメインの障害、または、ドメイン間のバスに障害が発生したことを検出することができる。

- [0152] また、プロセッサ管理方法によれば、自ドメインの診断結果と診断時刻を、マスタコンテキストに書き込み、転送された自ドメイン内のミラーリングされたマスタコンテキスト内の診断結果と診断時刻に基づいて、他ドメインの障害を検出してもよい。これにより、ドメインの診断回数を最小限に抑え、他ドメインの診断結果を参照することができる。
- [0153] また、プロセッサ管理方法によれば、障害が検出されるとき、マスタドメインは障害となったドメインに対して、一部の処理を停止してもよい。具体的には、マスタドメインは、該当ドメインのメモリに割り当てたメモリ空間解放を設定する。これにより、システムは、障害ドメインのメモリが新たに使用されることを防止することができる。また、マスタドメインは、該当ドメインのメモリへの書き込み禁止を設定する。これにより、システムは、該当ドメイン内で暴走中のアプリによるメモリ破壊、または、自動リセットによるメモリクリアを回避できる。また、マスタドメインは、該当ドメインへのスレッド生成禁止を設定する。これにより、システムは、アプリを正常に実行できないドメインを避けてアプリを割り当てることができる。
- [0154] また、プロセッサ管理方法によれば、障害が検出されるときに、ローカルマスタ機構がマスタ機構に通知して障害がマスタ機構の障害であるか否かを判断する。さらに、プロセッサ管理方法によれば、マスタ機構の障害であるときに、ローカルマスタ機構が複数のプロセッサを管理する状態にあるか否かを判断してもよい。これにより、プロセッサ管理方法を適用したシステムは、マスタドメインの障害時に、スレーブドメインの中から、次にドメインとなるドメインを判断することができる。
- [0155] また、プロセッサ管理方法によれば、マスタドメインへの通知に対して応答がない場合、マスタドメインのDBBへ発信してマスタ機構と切断された状態か否かを判断し、切断された状態であれば、自身のドメインの動作を終了してもよい。これにより、マスタドメインが動作しているにも関わらず、障害により分離状態となってしまった場合、分離されたドメインが動作し続けることがなくなり、無駄な消費電力を削減することができる。

[0156] また、プロセッサ管理方法によれば、複数のプロセッサを管理する状態にあると判断されるとき、スレーブドメインを新たなマスタドメインに設定してもよい。これにより、プロセッサ管理方法を適用したシステムは、障害発生時に対して、新たなマスタドメインの実行により機能や性能の制限を最小限にしつつ、継続的に利用することができる。また、これらの動作により、プロセッサ管理方法を適用したシステムを使用するユーザは、障害発生したドメインを含むユニットを修理サービスなどに預けつつ、一方で、残余のドメインにより継続的にシステムを利用することができる。

[0157] なお、本実施の形態で説明したプロセッサ管理方法は、予め用意されたプログラムをパーソナル・コンピュータやワークステーション、携帯電話等のコンピュータで実行することにより実現することができる。本プロセッサ管理方法を実行するプログラムは、ハードディスク、フレキシブルディスク、CD-ROM、MO、DVD、SDカード等のコンピュータで読み取り可能な記録媒体に記録され、コンピュータによって記録媒体から読み出されることによって実行される。また本プロセッサ管理方法を実行するプログラムは、インターネット等のネットワークを介して配布してもよい。

### 符号の説明

- [0158]
- 100 システム
  - 109 メモリ
  - 110 バス
  - 204 CPU
  - 301 マスタコンテキスト
  - 302 ミラーリングマスタコンテキスト
  - 303 ステータステーブル
  - 311 検出部
  - 312 転送部
  - 313 判断部
  - 314 設定部

3 1 5 通知部

3 1 6 割当部

3 1 7 格納部

3 1 8 設定部

3 1 9 終了部

## 請求の範囲

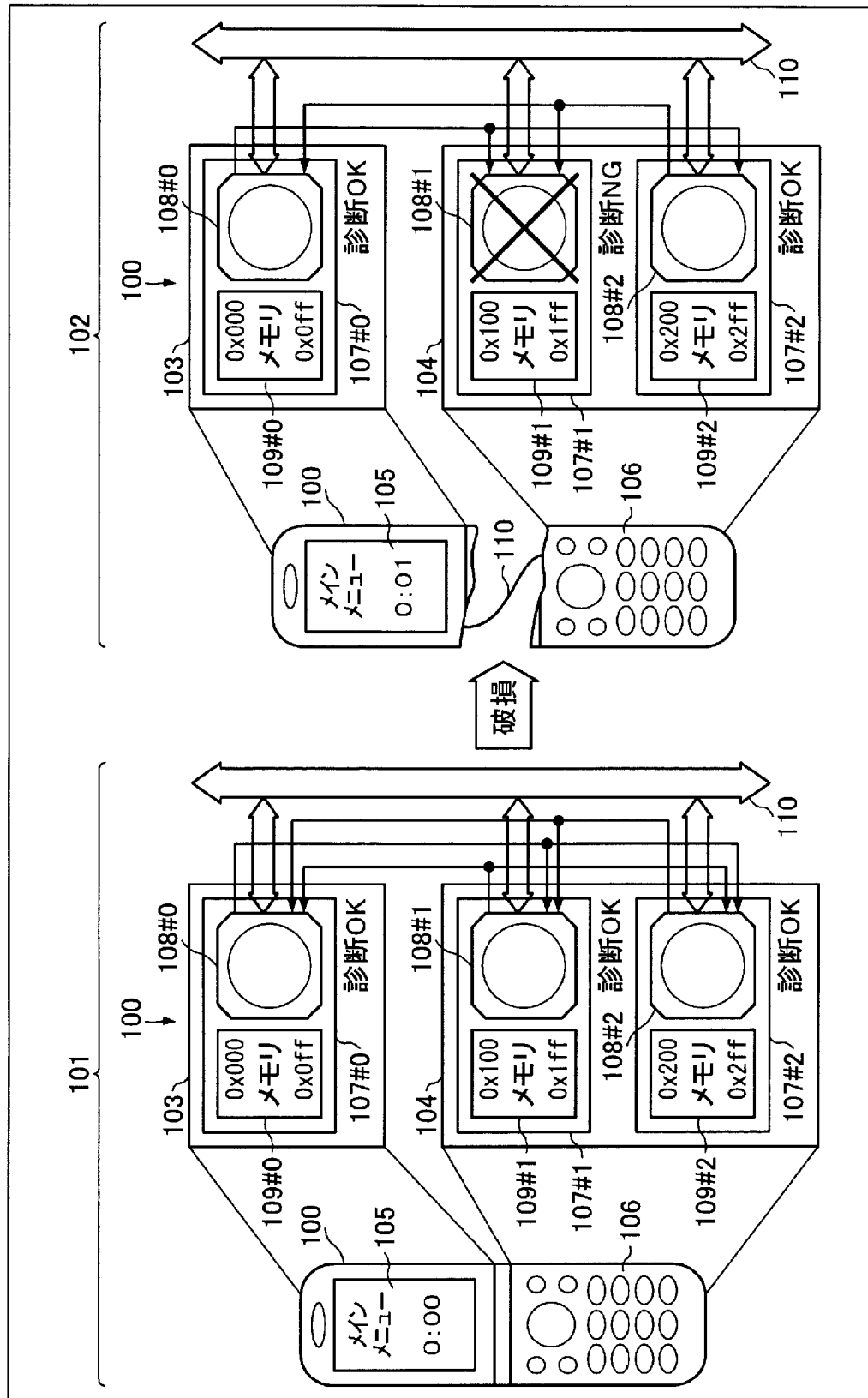
- [請求項1]       複数のプロセッサのうちの一のプロセッサに前記複数のプロセッサを管理するマスタ機構を設定し、
- 前記複数のプロセッサのうちの前記一のプロセッサ以外のプロセッサのそれぞれに当該プロセッサを管理するローカルマスタ機構と仮想マスタ機構とを設定し、
- 前記マスタ機構によって管理される共有メモリが前記複数のプロセッサから連続したメモリとしてアクセスされるために、前記マスタ機構が前記複数のプロセッサにアドレスのオフセット値を通知することを特徴とするプロセッサ管理方法。
- [請求項2]       前記オフセット値が通知された後に、前記ローカルマスタ機構は、前記オフセット値に基づいて前記共有メモリを前記ローカルマスタ機構が管理するメモリ領域へ割り当てること
- を特徴とする請求項1に記載のプロセッサ管理方法。
- [請求項3]       前記仮想マスタ機構は、前記マスタ機構から転送されるコンテキストデータを前記ローカルマスタ機構が管理するメモリ領域に格納すること
- を特徴とする請求項1または請求項2に記載のプロセッサ管理方法。
- [請求項4]       前記仮想マスタ機構は、前記マスタ機構に設定されたブレークポイントを検出したときに、前記マスタ機構のコンテキストデータを前記ローカルマスタ機構が管理するメモリ領域に格納すること
- を特徴とする請求項1または請求項2に記載のプロセッサ管理方法。
- [請求項5]       第1スレッドにおいて前記コンテキストデータを前記メモリ領域に格納するときに、更新された前記コンテキストデータが第2スレッドにおいて検出されるとき、前記第1スレッドの処理を中止すること
- を特徴とする請求項4に記載のプロセッサ管理方法。

- [請求項6] 前記マスタ機構または前記ローカルマスタ機構は、障害を検出するための処理を行うこと
- を特徴とする請求項 1 乃至請求項 5 の何れかーに記載のプロセッサ管理方法。
- [請求項7] 前記マスタ機構または前記ローカルマスタ機構は、前記マスタ機構のコンテキストデータにタイムスタンプまたは付随情報を所定周期毎に書き込み、
- 前記タイムスタンプまたは前記付随情報に基づいて前記障害を検出すること
- を特徴とする請求項 6 に記載のプロセッサ管理方法。
- [請求項8] 前記障害が検出されるとき、前記マスタ機構は少なくとも一部の処理を停止すること
- を特徴とする請求項 6 または請求項 7 に記載のプロセッサ管理方法。
- [請求項9] 前記障害が検出され、前記ローカルマスタ機構は前記マスタ機構に通知して前記障害が前記マスタ機構の障害であるか否かを判断し、
- 前記マスタ機構の障害であると判断されたとき、前記ローカルマスタ機構が前記複数のプロセッサを管理する状態にあるか否かを判断すること
- を特徴とする請求項 6 または請求項 7 に記載のプロセッサ管理方法。
- [請求項10] 前記障害が検出され、前記マスタ機構への通知に対して応答がない場合に、前記ローカルマスタ機構は前記マスタ機構が制御可能な通話装置へ発信して前記マスタ機構と切断された状態か否かを判断し、
- 前記マスタ機構と切断された状態であると判断された場合、前記ローカルマスタ機構に設定されたプロセッサの動作を終了すること
- を特徴とする請求項 6 に記載のプロセッサ管理方法。
- [請求項11] 前記ローカルマスタ機構が前記複数のプロセッサを管理する状態に

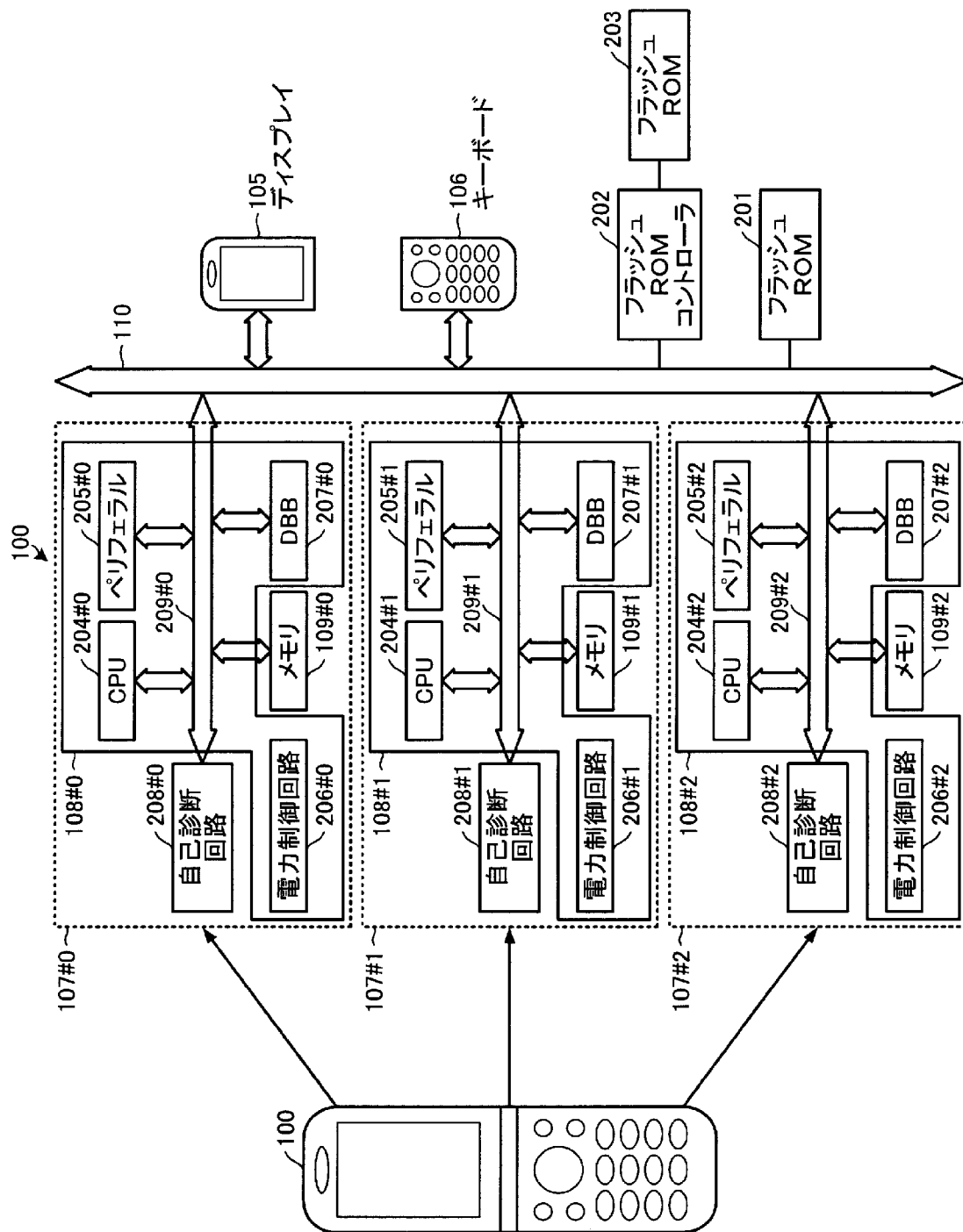
あると判断されるとき、前記ローカルマスタ機構は新たなマスタ機構として設定すること

を特徴とする請求項 6 乃至請求項 9 の何れかーに記載のプロセッサ管理方法。

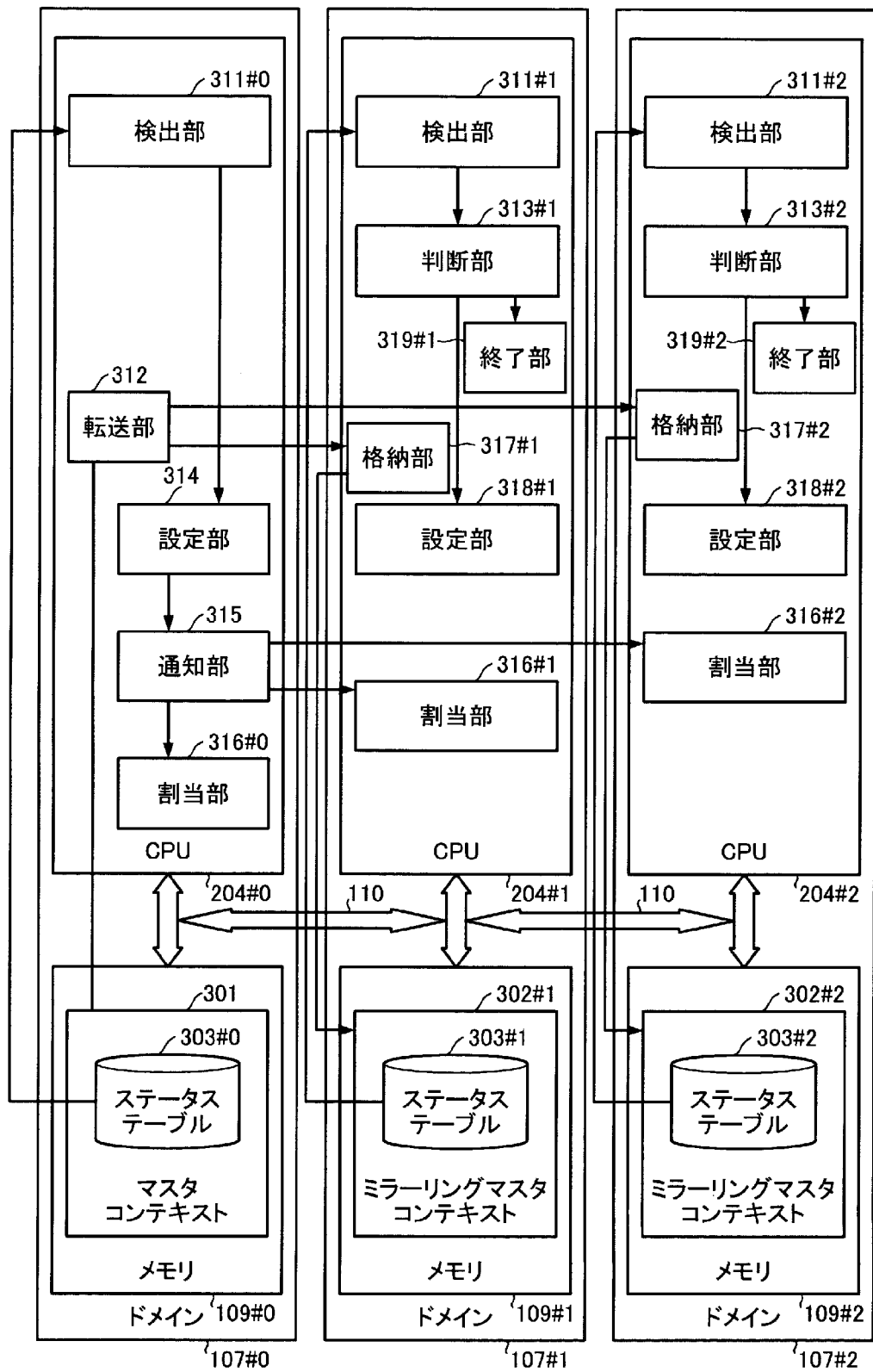
[図1]



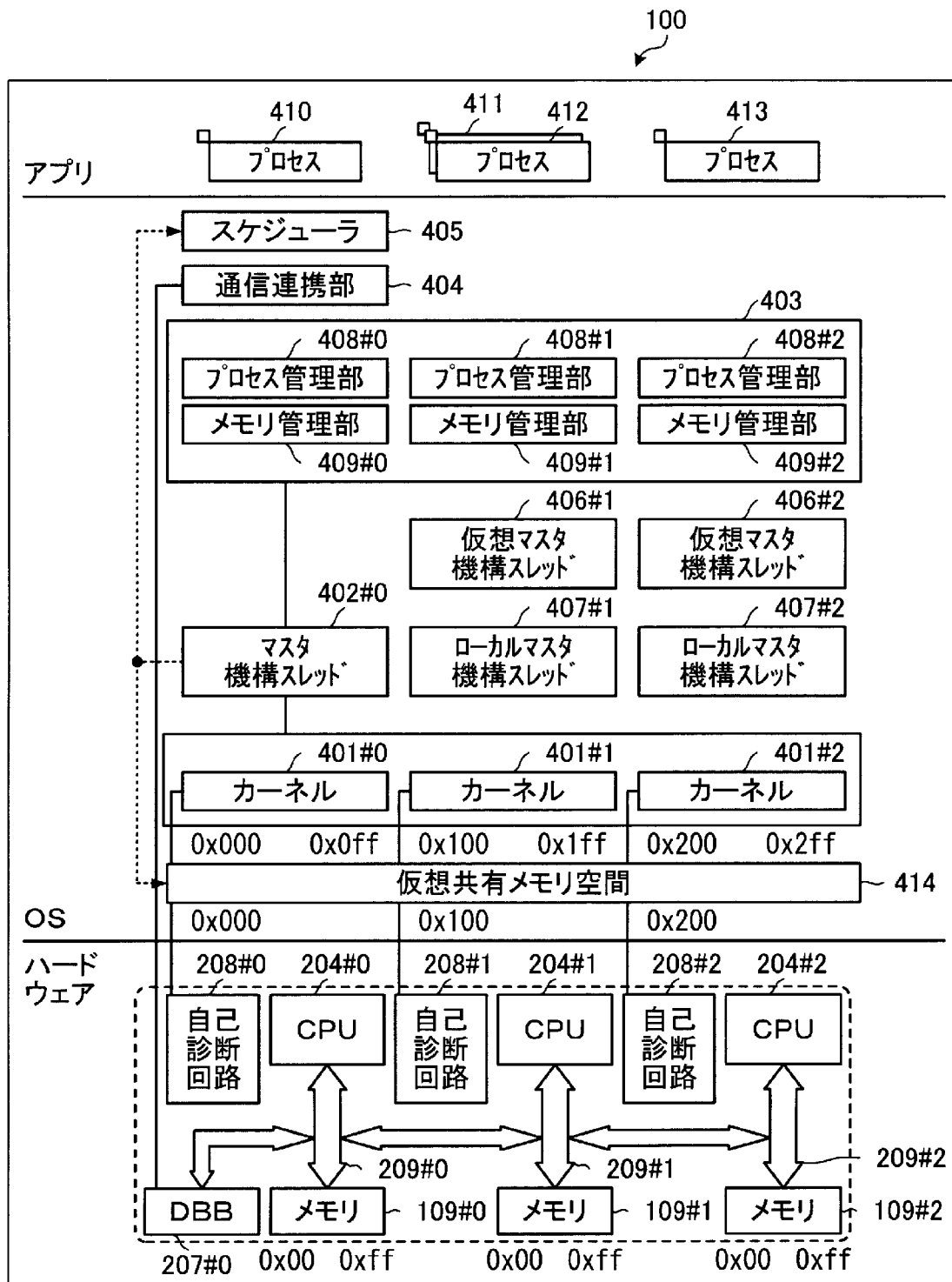
[図2]



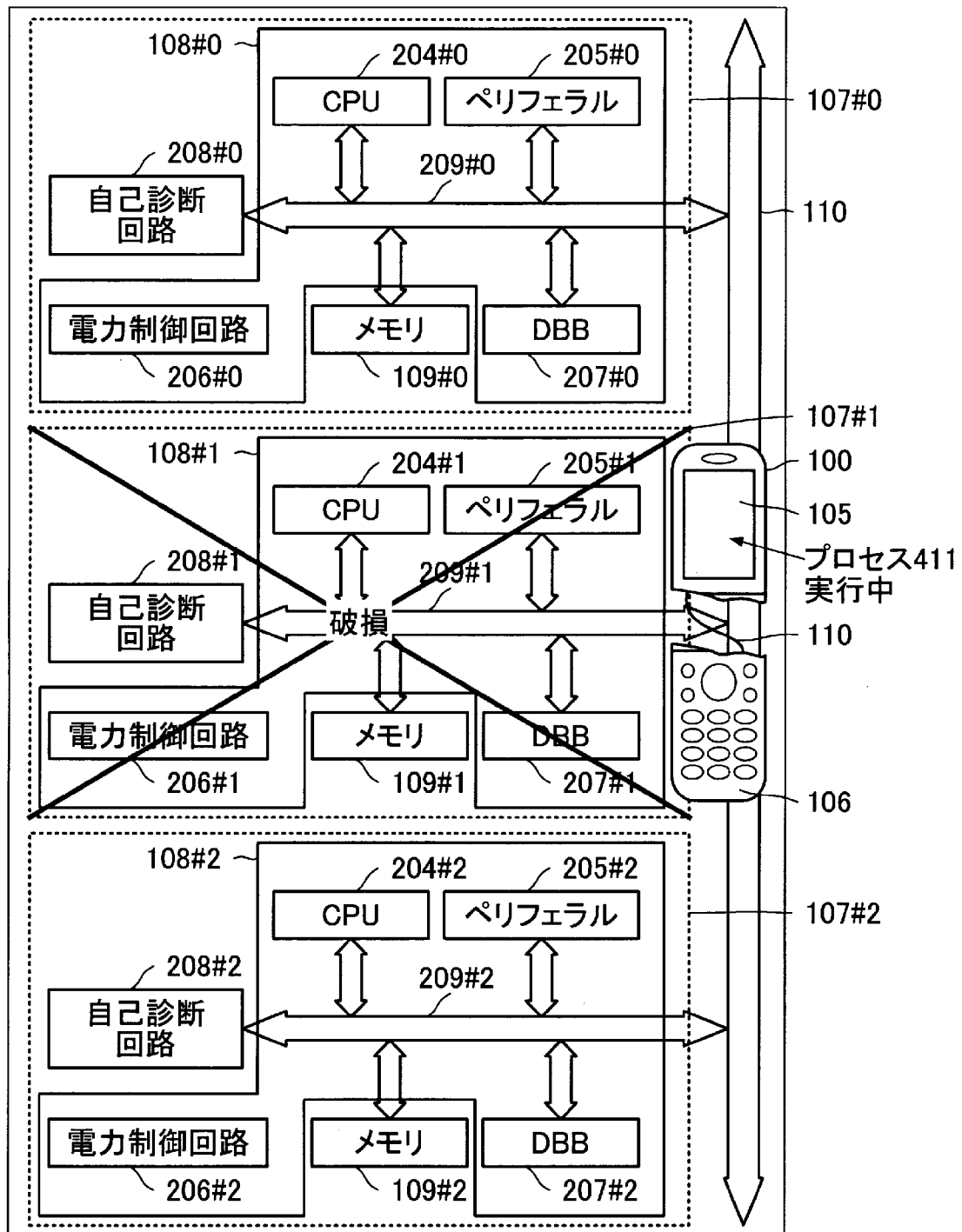
[図3]



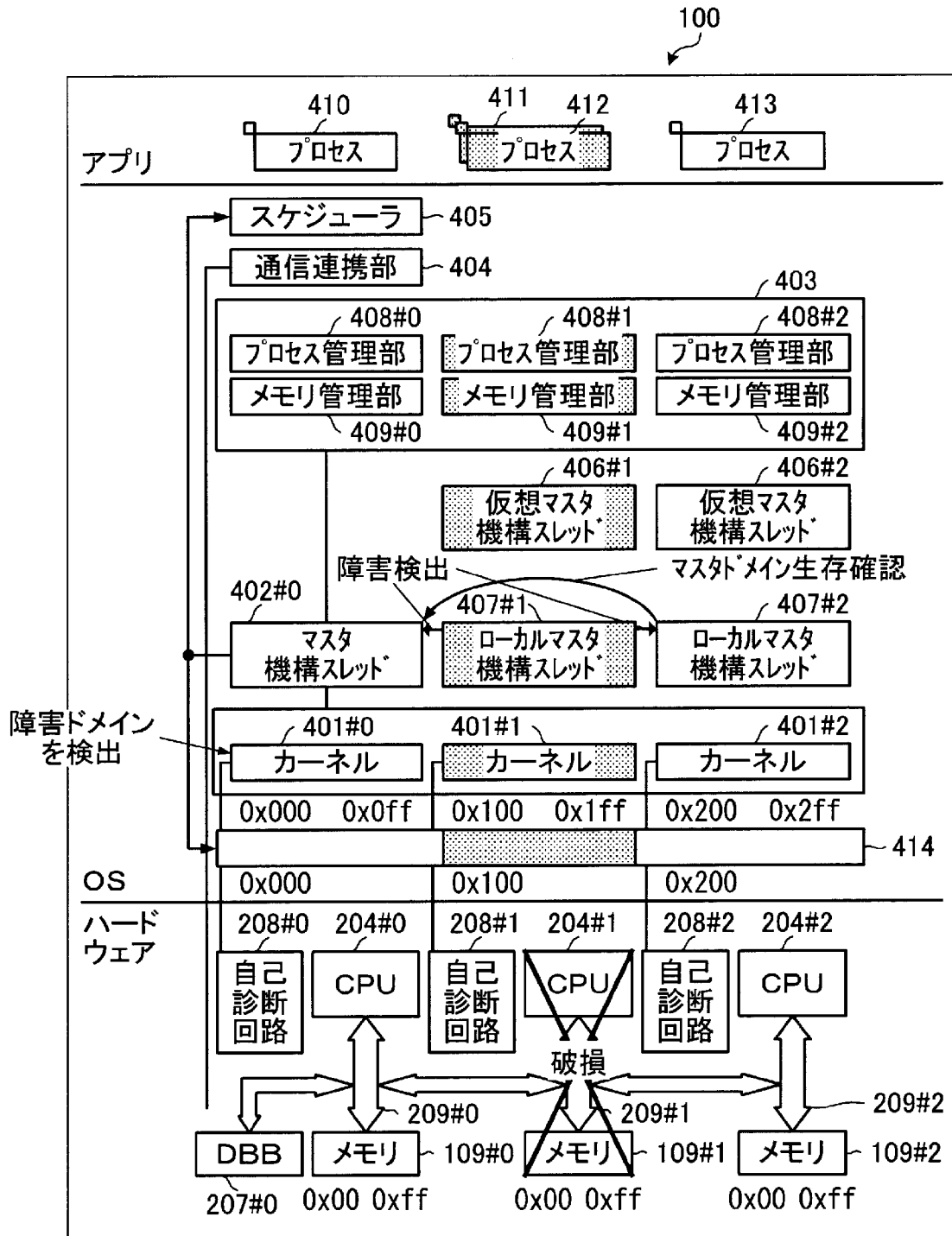
[図4]



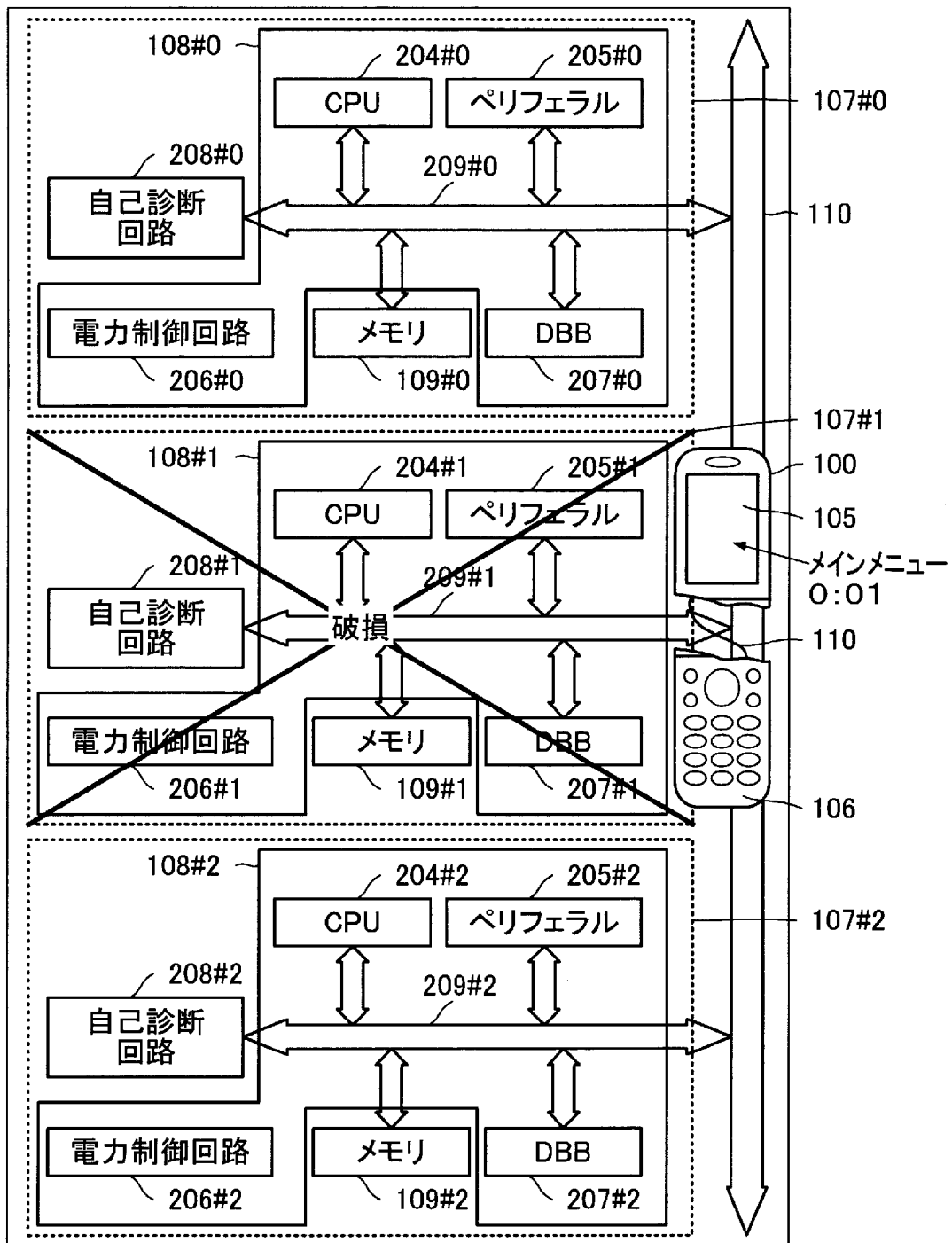
[図5A]



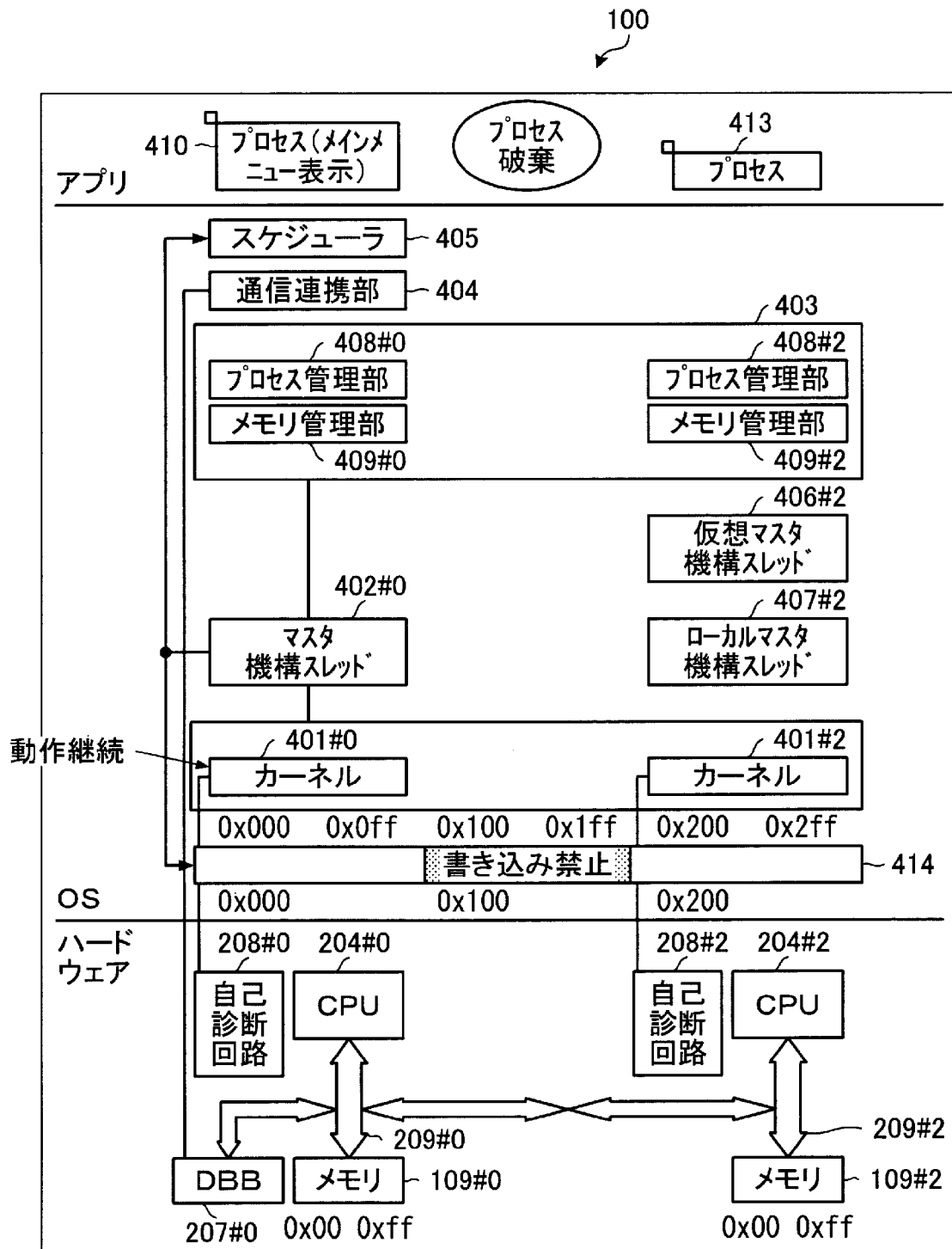
[図5B]



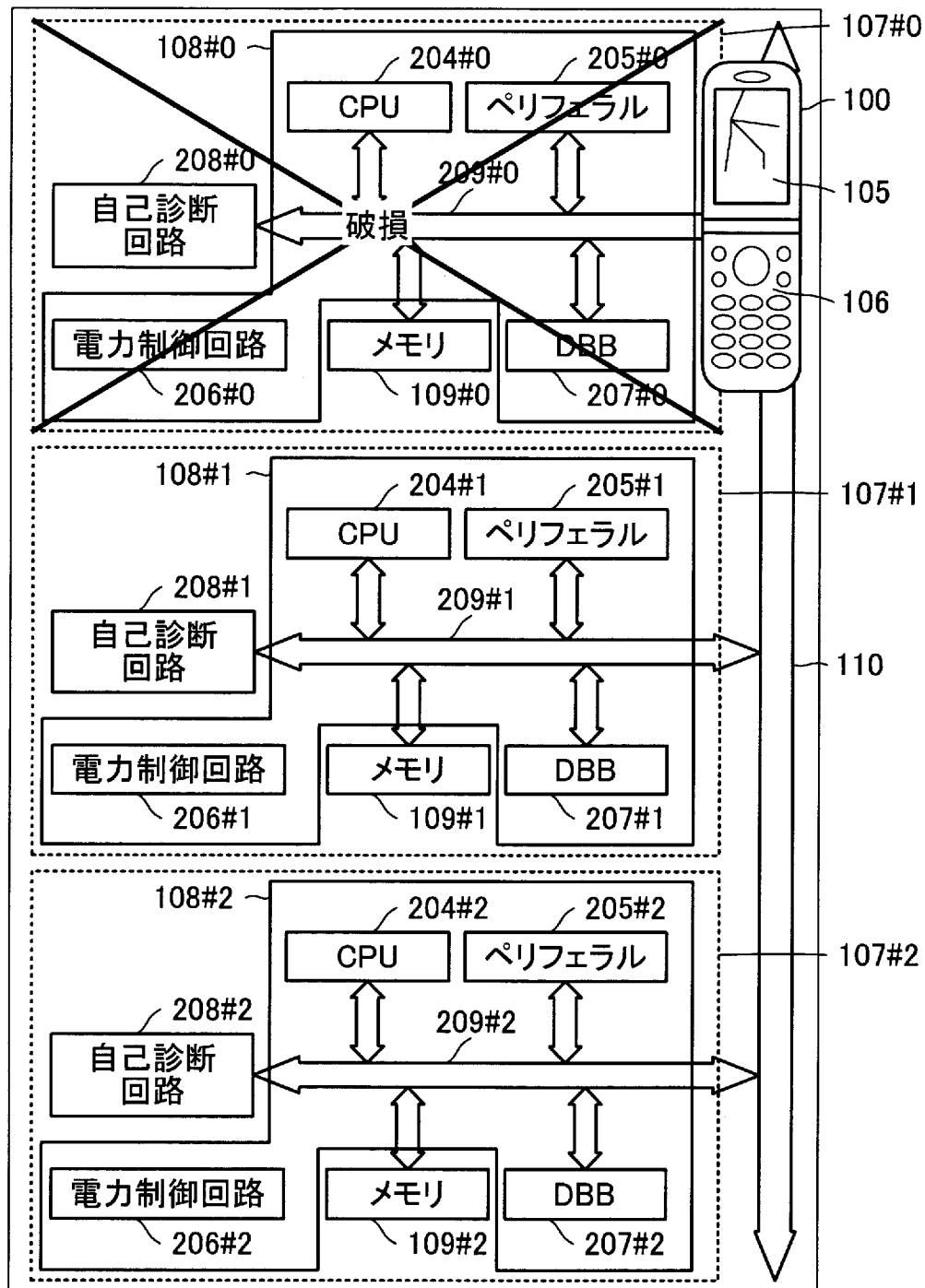
[図6A]



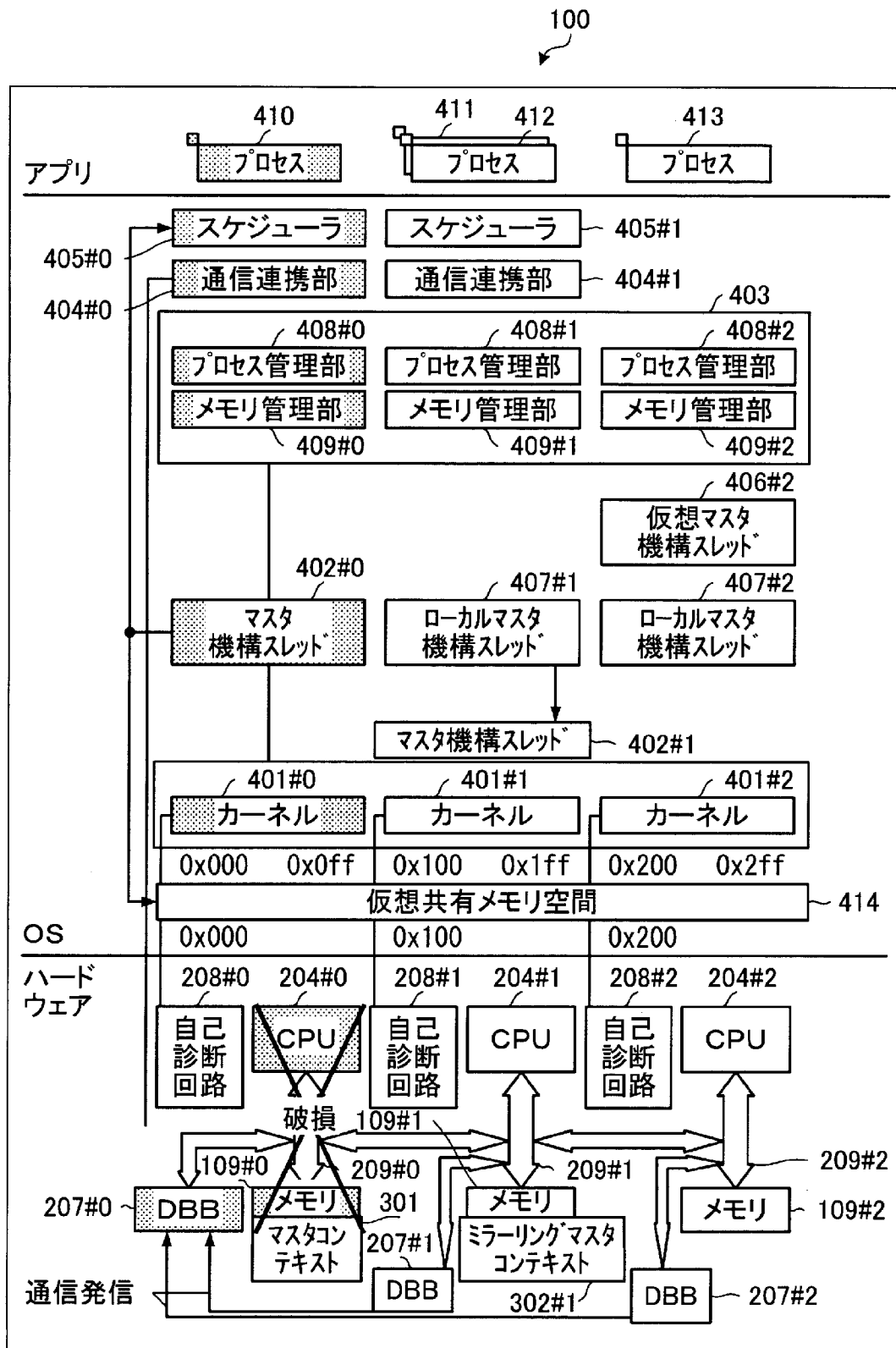
[図6B]



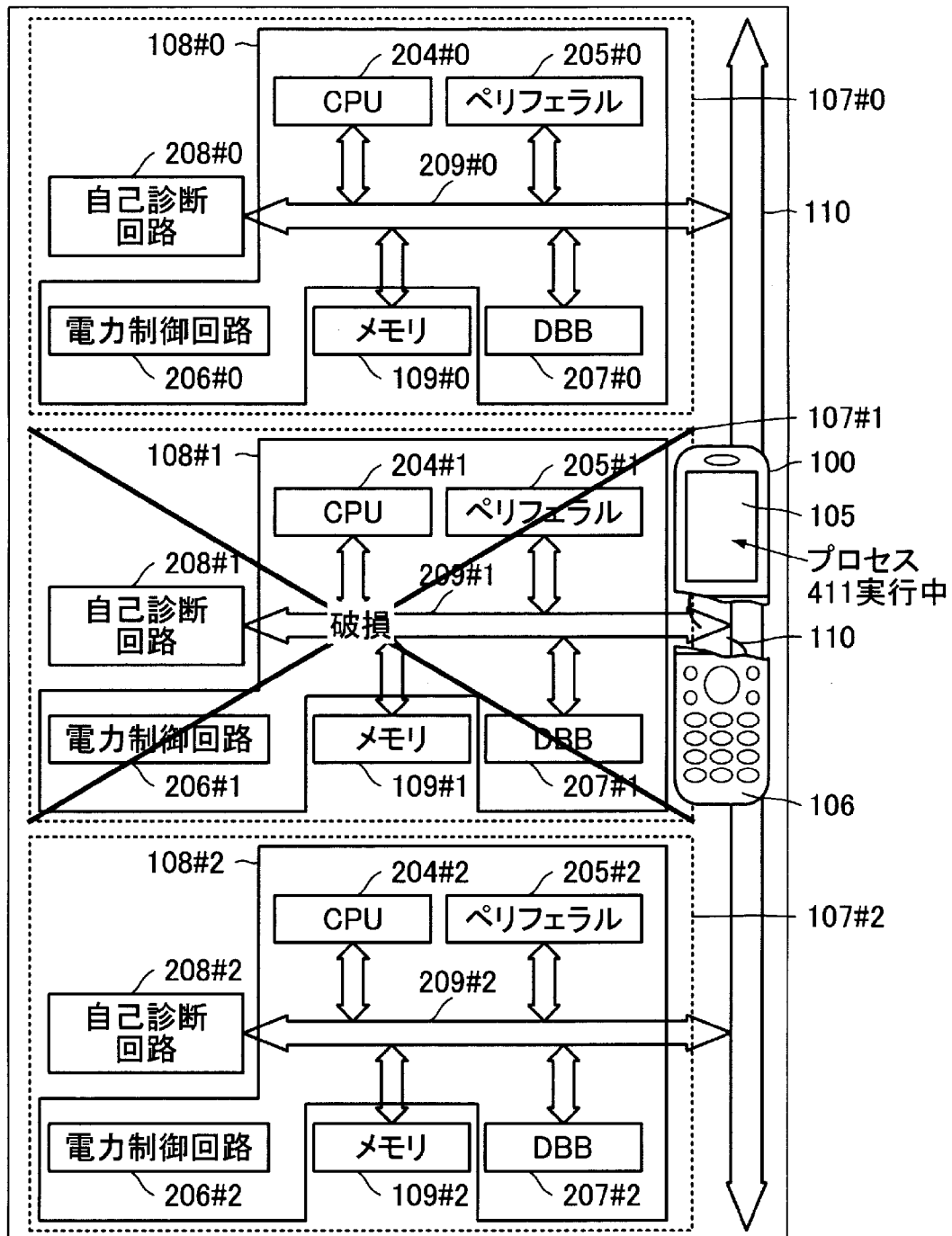
[図7A]



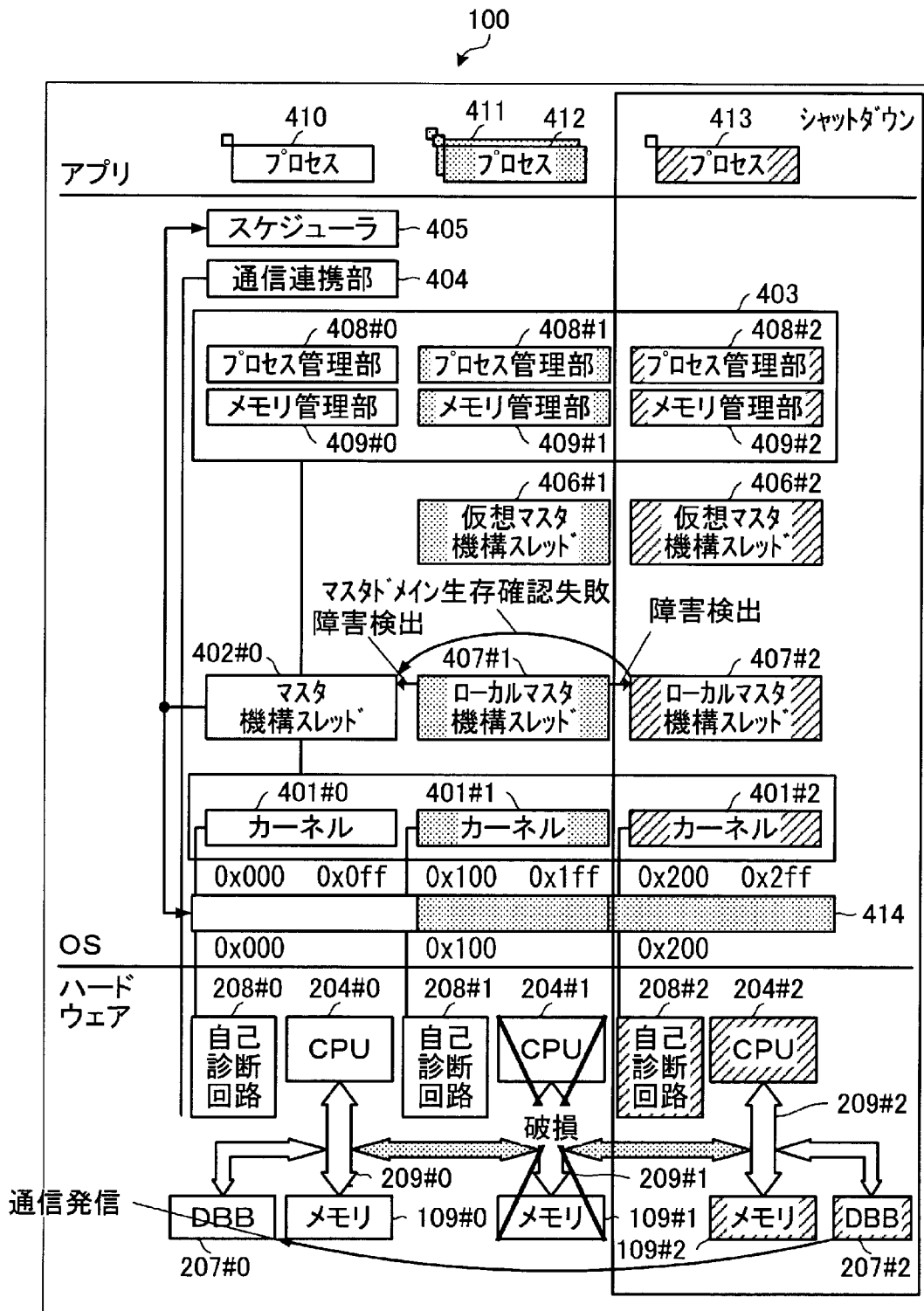
[図7B]



[図8A]



[図8B]



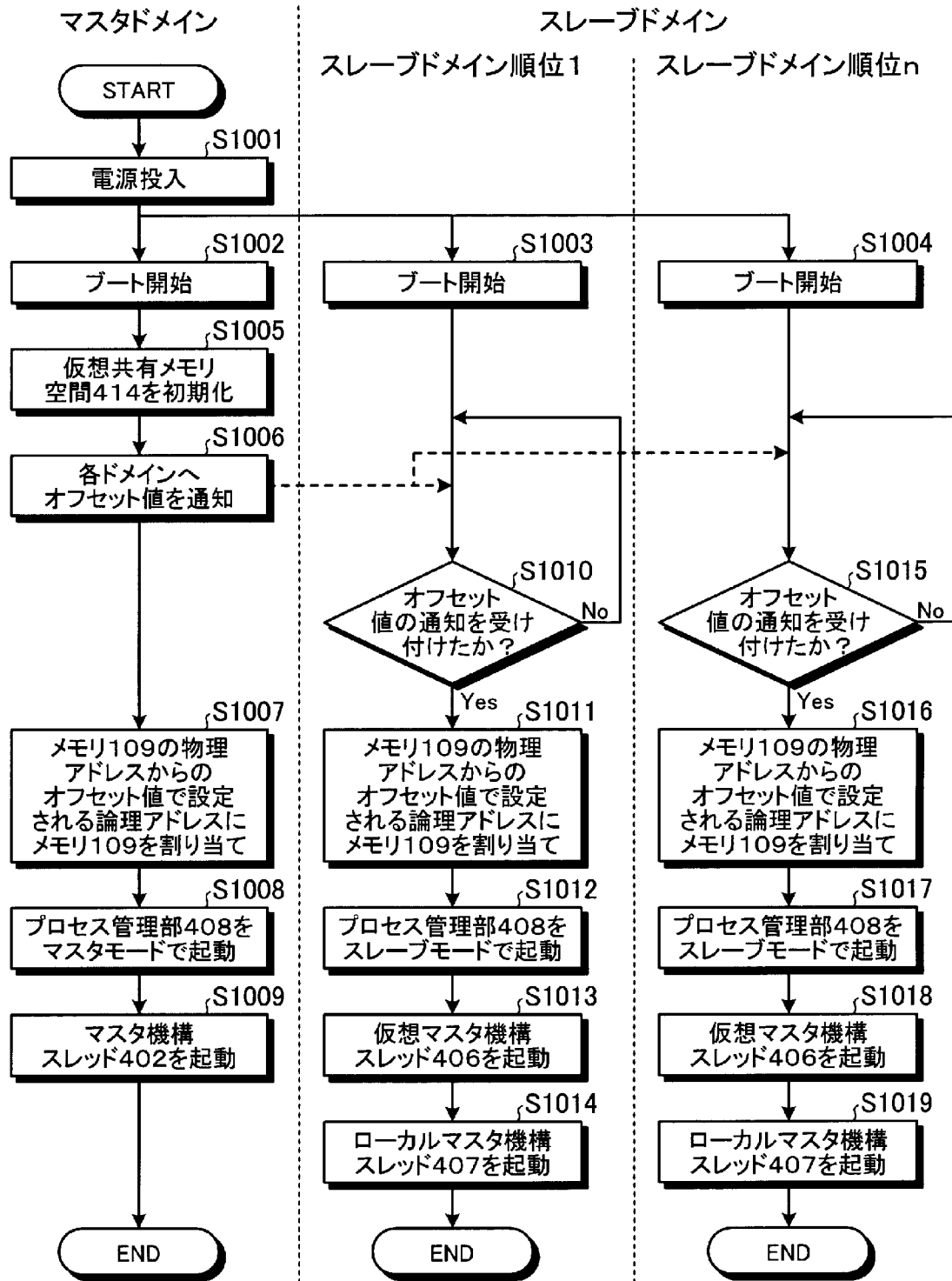
[図9]

|             |          |      |
|-------------|----------|------|
| ドメイン種別      | 診断時刻     | 診断結果 |
| マスタドメイン     | 12:34:00 | OK   |
| スレーブドメイン順位1 | 12:34:00 | OK   |
| スレーブドメイン順位2 | 12:33:00 | OK   |

303

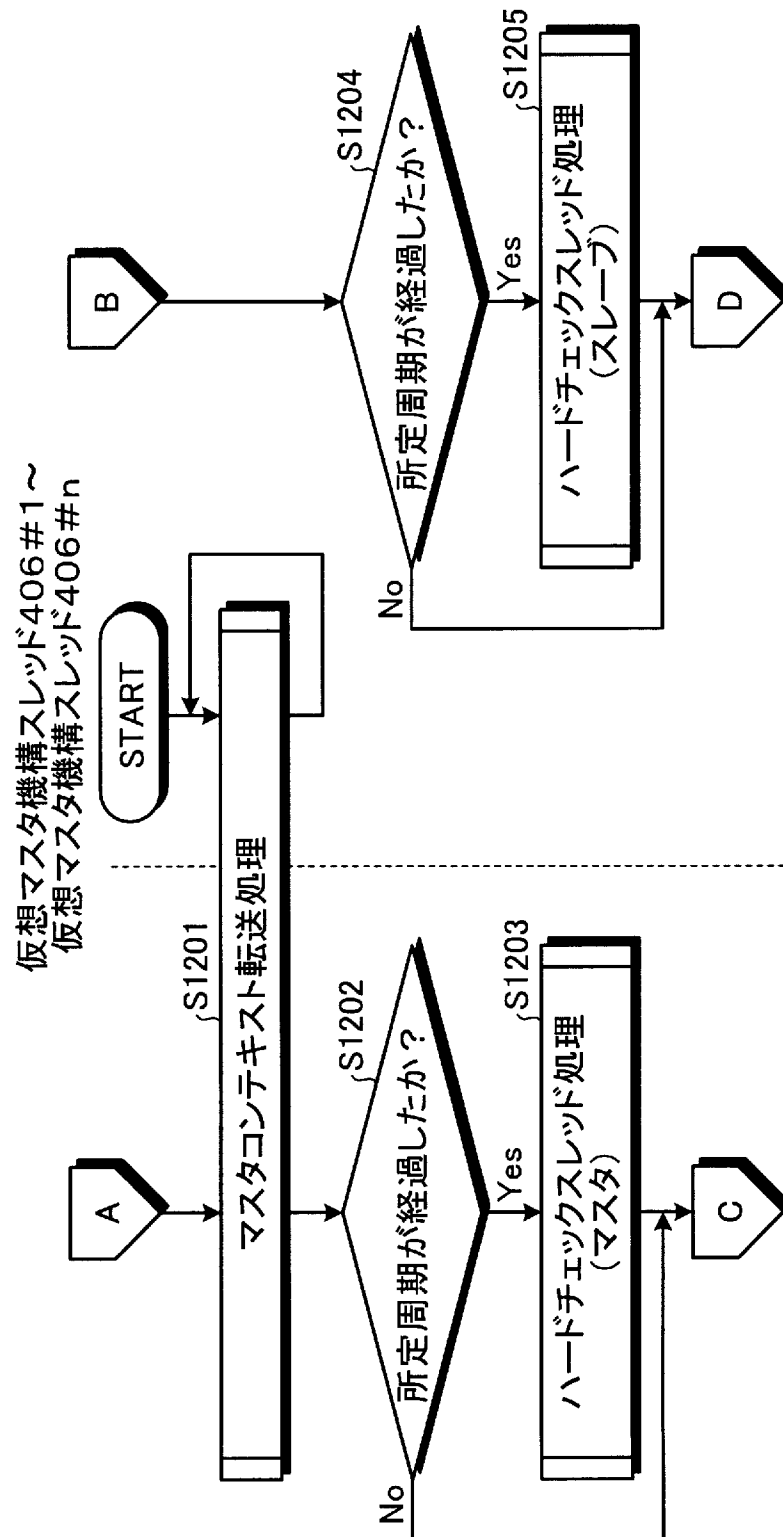
障害検出

[図10]

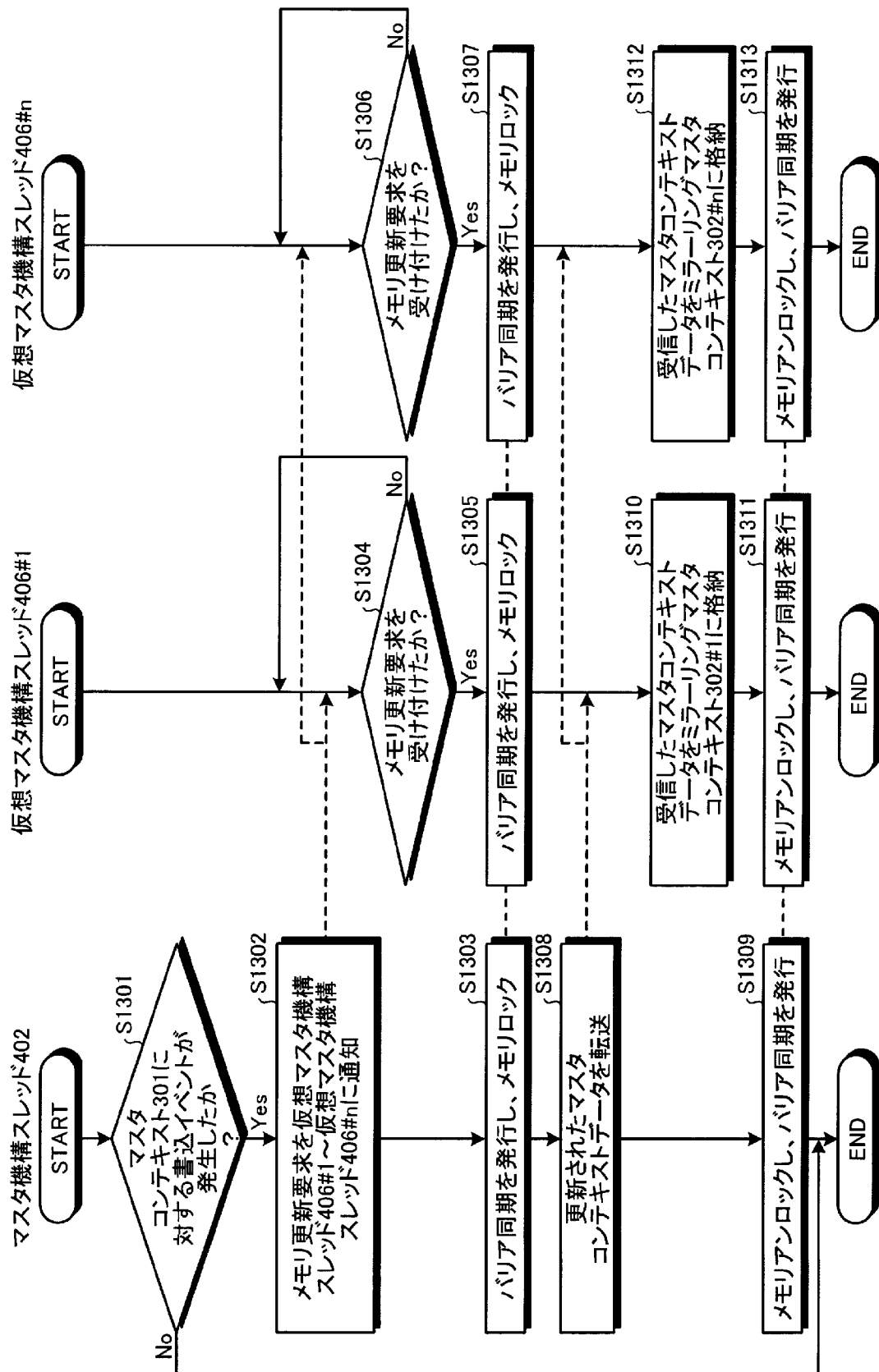




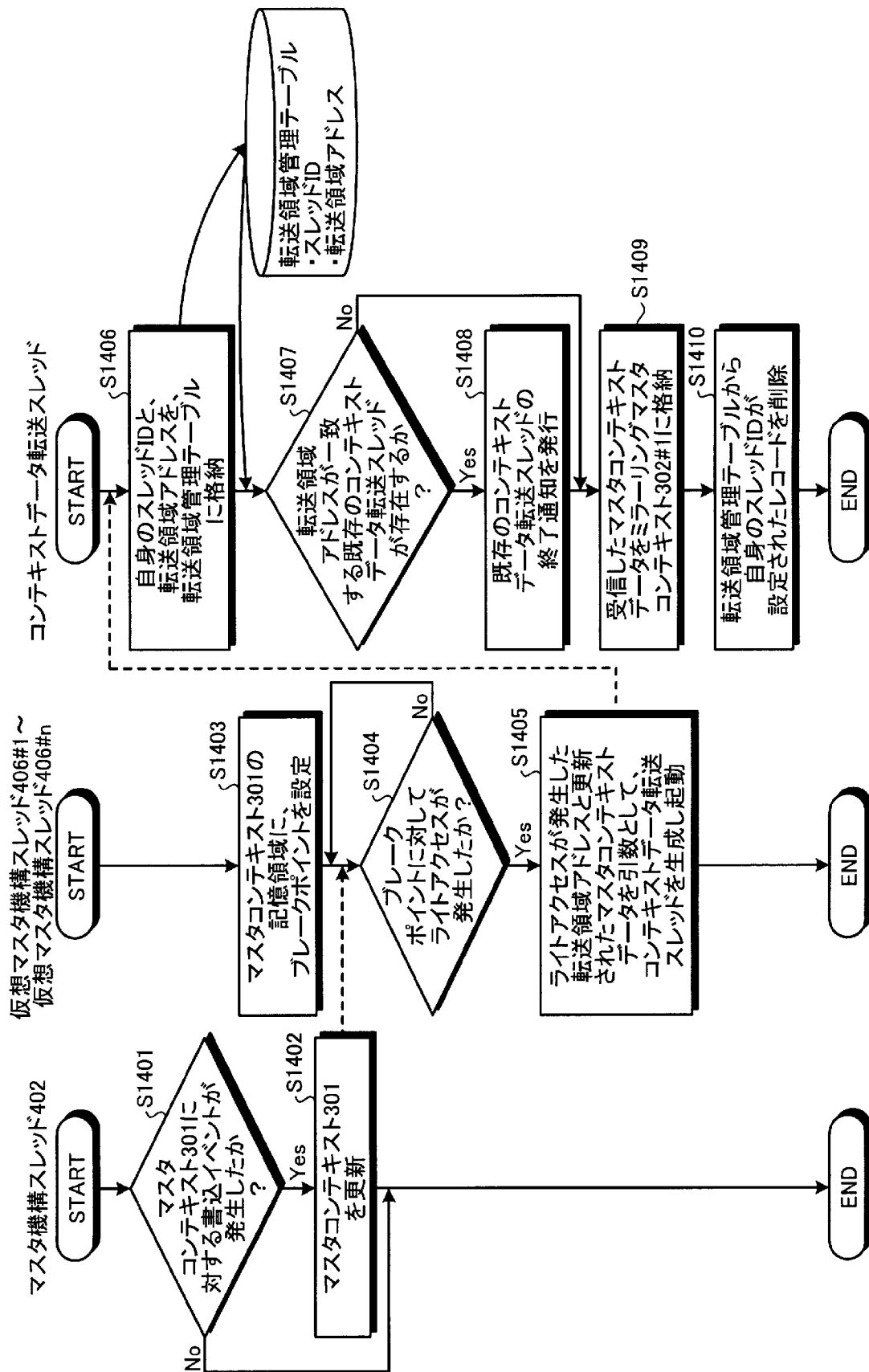
[図12]



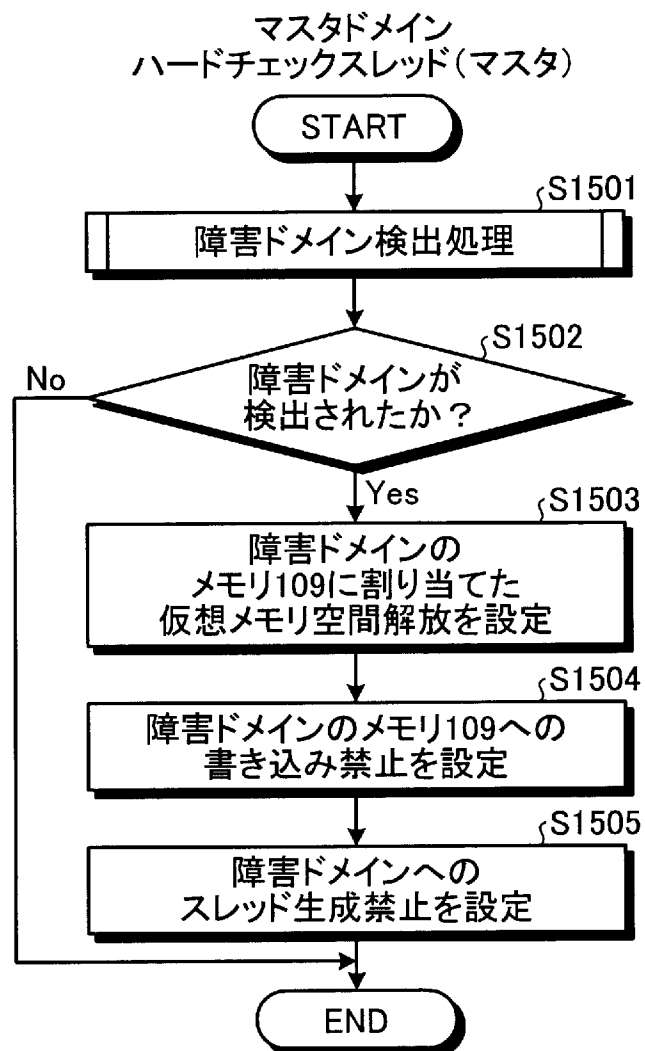
[図13]



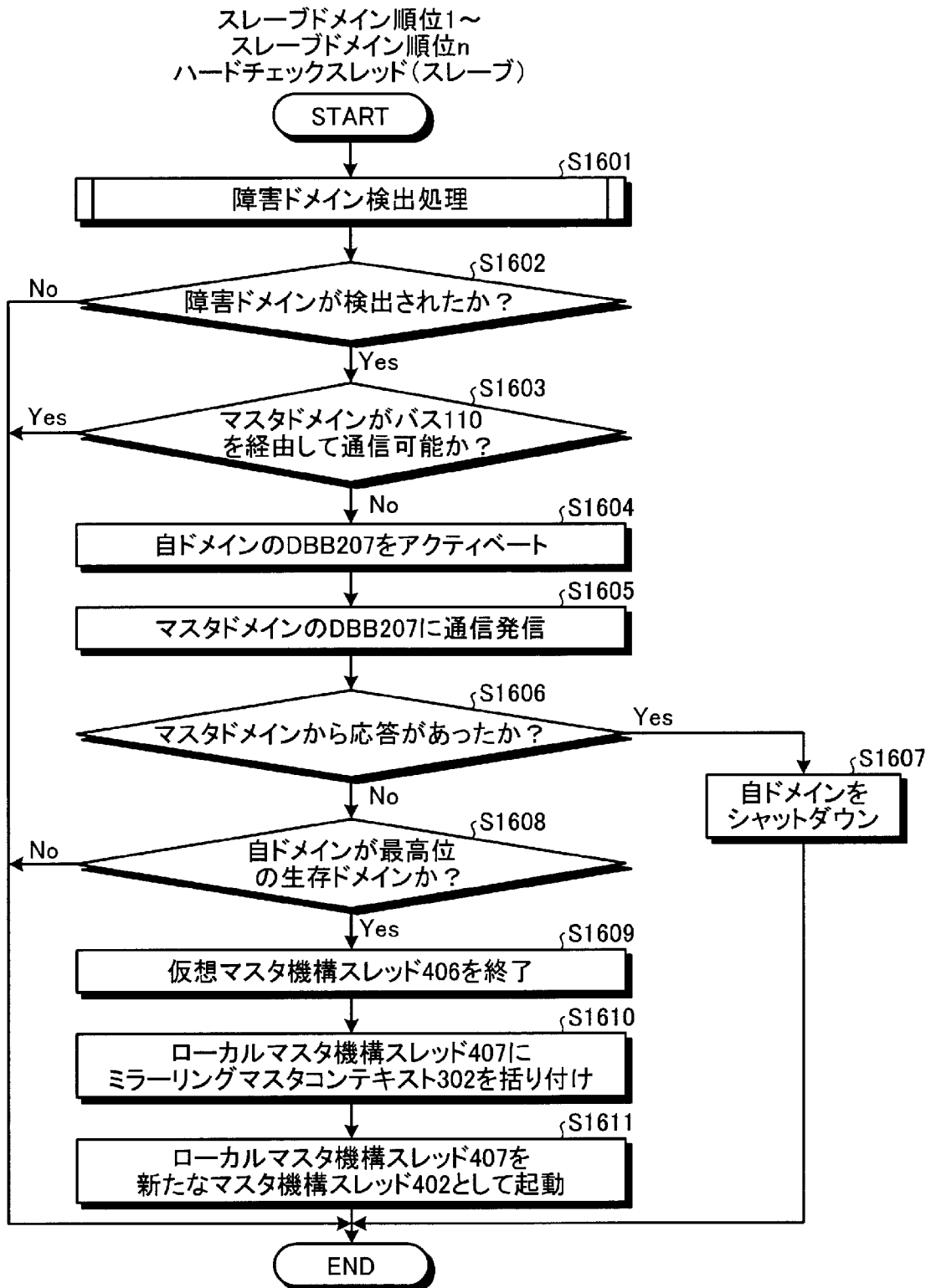
[図14]



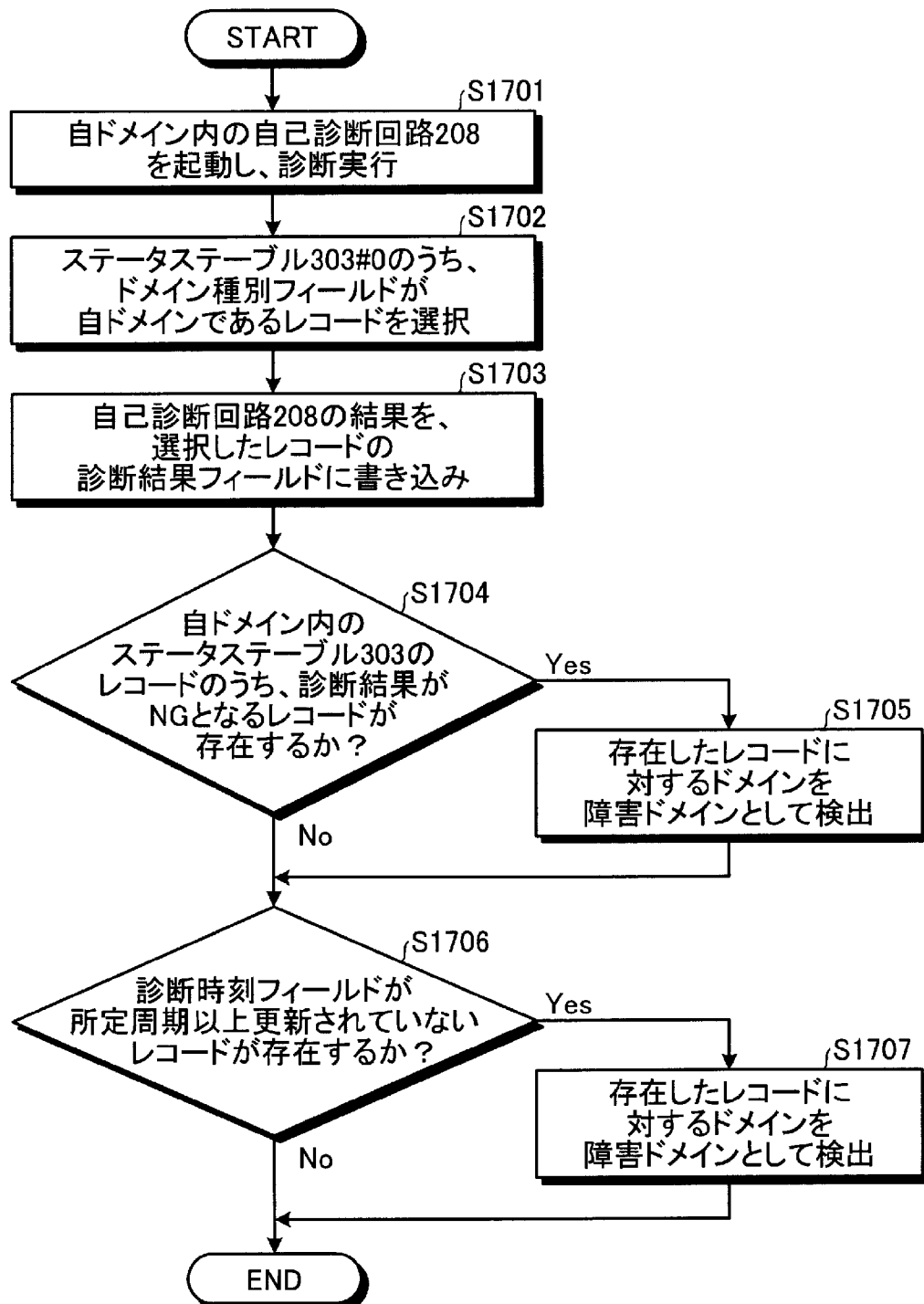
[図15]



[図16]



[図17]



## INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP2011/053732

## A. CLASSIFICATION OF SUBJECT MATTER

G06F11/20 (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)

G06F11/16-11/20, G06F9/46-9/54

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 1996-2011

Kokai Jitsuyo Shinan Koho 1971-2011 Toroku Jitsuyo Shinan Koho 1994-2011

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

## C. DOCUMENTS CONSIDERED TO BE RELEVANT

| Category* | Citation of document, with indication, where appropriate, of the relevant passages                                                                                                      | Relevant to claim No. |
|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------|
| Y<br>A    | JP 2009-211279 A (Hitachi High-Tech Control Systems Corp.),<br>17 September 2009 (17.09.2009),<br>paragraphs [0002], [0015], [0025] to [0041],<br>[0055]; fig. 1 to 6<br>(Family: none) | 1-9, 11<br>10         |
| Y<br>A    | JP 7-84921 A (Ricoh Co., Ltd.),<br>31 March 1995 (31.03.1995),<br>paragraphs [0083] to [0091]; fig. 1, 2, 19 to<br>21<br>(Family: none)                                                 | 1-9, 11<br>10         |



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

"I" 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

"&amp;" document member of the same patent family

Date of the actual completion of the international search

11 May, 2011 (11.05.11)

Date of mailing of the international search report

24 May, 2011 (24.05.11)

Name and mailing address of the ISA/  
Japanese Patent Office

Authorized officer

Facsimile No.

Telephone No.

# INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP2011/053732

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

| Category* | Citation of document, with indication, where appropriate, of the relevant passages                                                                                                                           | Relevant to claim No. |
|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------|
| Y<br>A    | JP 2001-188690 A (Toshiba Corp.),<br>10 July 2001 (10.07.2001),<br>paragraphs [0023], [0048], [0058] to [0065],<br>[0083]; fig. 1, 4 to 6, 11 to 14<br>(Family: none)                                        | 4-5<br>1-3, 6-11      |
| A         | JP 4-156655 A (Fujitsu Ltd.),<br>29 May 1992 (29.05.1992),<br>page 3, upper left column, lines 3 to 13; page<br>5, upper left column, line 7 to upper right<br>column, line 11; fig. 6, 14<br>(Family: none) | 10                    |
| A         | JP 2007-304687 A (Hitachi, Ltd.),<br>22 November 2007 (22.11.2007),<br>paragraphs [0012] to [0014]<br>& US 2007/0288585 A1 & CN 101072125 A                                                                  | 10                    |

## A. 発明の属する分野の分類 (国際特許分類 (I P C))

Int.Cl. G06F11/20(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. G06F11/16-11/20, G06F9/46-9/54

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 1 年 |
| 日本国実用新案登録公報 | 1 9 9 6 - 2 0 1 1 年 |
| 日本国登録実用新案公報 | 1 9 9 4 - 2 0 1 1 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<br>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<br>請求項の番号   |
|-----------------|------------------------------------------------------------------------------------------------------------------|------------------|
| Y<br>A          | JP 2009-211279 A (株式会社日立ハイテクコントロールシステムズ)<br>2009.09.17, 段落【0002】、【0015】、【0025】-【0041】、【0055】、<br>第1-6図 (ファミリーなし) | 1-9, 11<br>10    |
| Y<br>A          | JP 7-84921 A (株式会社リコー) 1995.03.31, 段落【0083】-【0091】、<br>第1, 2, 19-21図 (ファミリーなし)                                   | 1-9, 11<br>10    |
| Y<br>A          | JP 2001-188690 A (株式会社東芝) 2001.07.10, 段落【0023】、【0048】、<br>【0058】-【0065】、【0083】、第1, 4-6, 11-14図 (ファミリーなし)         | 4-5<br>1-3, 6-11 |

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

## \* 引用文献のカテゴリー

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

の日の後に公表された文献

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

国際調査を完了した日

1 1 . 0 5 . 2 0 1 1

国際調査報告の発送日

2 4 . 0 5 . 2 0 1 1

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)

北元 健太

5 B

4 6 8 0

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

| C (続き) . 関連すると認められる文献 |                                                                                                              |                |
|-----------------------|--------------------------------------------------------------------------------------------------------------|----------------|
| 引用文献の<br>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<br>請求項の番号 |
| A                     | JP 4-156655 A (富士通株式会社) 1992. 05. 29,<br>第 3 頁左上欄第 3 行-13 行, 第 5 頁左上欄第 7 行-右上欄第 11 行,<br>第 6, 14 図 (ファミリーなし) | 10             |
| A                     | JP 2007-304687 A (株式会社日立製作所) 2007. 11. 22,<br>段落【0012】 - 【0014】 & US 2007/0288585 A1 & CN 101072125 A        | 10             |