

(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
2010年9月2日(02.09.2010)



(10) 国際公開番号
WO 2010/097925 A1

- (51) 国際特許分類:
G06F 9/46 (2006.01)
- (21) 国際出願番号: PCT/JP2009/053546
- (22) 国際出願日: 2009年2月26日(26.02.2009)
- (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): 株式会社日立製作所(HITACHI, LTD.) [JP/JP]; 〒1008280 東京都千代田区丸の内一丁目6番6号 Tokyo (JP).
- (72) 発明者; および
- (75) 発明者/出願人(米国についてのみ): 野尻 徹(NOJIRI, Tohru) [JP/JP]; 〒1858601 東京都国分寺市東恋ヶ窪一丁目280番地 株式会社日立製作所 中央研究所内 Tokyo (JP). 十山 圭介(TOYAMA, Keisuke) [JP/JP]; 〒1858601 東京都国分寺市東恋ヶ窪一丁目280番地 株式会社日立製作所 中央研究所内 Tokyo (JP). 長坂 由子(NAGASAKA, Yoshiko) [JP/JP]; 〒1858601 東京都国分寺市東恋ヶ窪一丁目280番地 株式会社日立製作所 中央研究所内 Tokyo (JP). 佐伯 裕治

(SAEKI, Yuji) [JP/JP]; 〒1858601 東京都国分寺市東恋ヶ窪一丁目280番地 株式会社日立製作所 中央研究所内 Tokyo (JP).

(74) 代理人: 筒井 大和(TSUTSUI, Yamato); 〒1600022 東京都新宿区新宿2丁目3番10号 新宿御苑ビル3階 筒井国際特許事務所 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, 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, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SM, ST, SV, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

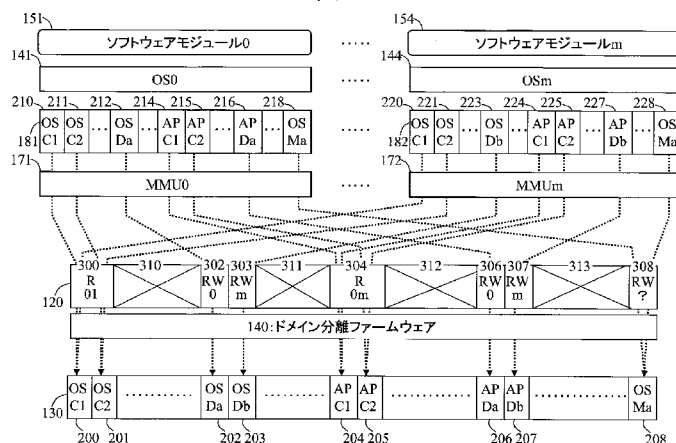
(84) 指定国(表示のない限り、全ての種類の広域保護が可能): ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), ユーラシア (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), ヨーロッパ (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL,

[続葉有]

(54) Title: INFORMATION PROCESSING DEVICE

(54) 発明の名称: 情報処理装置

図 2



151 SOFTWARE MODULE 0
154 SOFTWARE MODULE m
140 DOMAIN SEPARATION FIRMWARE

(57) Abstract: When a plurality of operating systems are mounted, it is preferable to effectively use memory resources without affecting other operating system. Even when the operating systems are different, communication between the operating systems is required since the operating systems are mounted on a single system. In this case, it is necessary to perform data communication without affecting the other operating systems. To cope with this, an information processing device includes firmware and middleware. The firmware allocates as a first domain, a first CPU, a first operating system, and a first region as a part of a memory region while allocating as a second domain, a second CPU, a second operating system, and a second region as a part of the memory region, and performs control so that one of the domains will not access a region allocated to the other domain. When a data communication is required between the first domain and the second domain, the middleware controls the communication.

(57) 要約: 複数OSを搭載した場合、他のOSに影響を与えないように、効率よくメモリ資源を用いることが望まれる。

また、OSが異なるといっても、一つのこの場合、他のOSに影響を与えないようにデータ通信を行うことが必要となる。そこで、情報処理装置を、第1中央処理装置、第1オペレーティングシステム、及び、メモリの領域のうち一部の第1領域を第1ドメインとして割り当て、第2中央処理装置、第2オペレーティングシステム及びメモリの領域のうち一部の第2領域を第2ドメインとして割り当て、一方のドメインが他のドメインに割り当てられた領域にアクセスしないように制御するミドルウェアと、第1ドメイン及び第2ドメイン間でデータの通信が必要な場合に、当該通信を制御するミドルウェアとを有する構成とする。

システム上に搭載されているため、OS間での通信が必要となる。この場合、他のOSに影響を与えないようにデータ通信を行うことが必要となる。そこで、情報処理装置を、第1中央処理装置、第1オペレーティングシステム、及び、メモリの領域のうち一部の第1領域を第1ドメインとして割り当て、第2中央処理装置、第2オペレーティングシステム及びメモリの領域のうち一部の第2領域を第2ドメインとして割り当て、一方のドメインが他のドメインに割り当てられた領域にアクセスしないように制御するミドルウェアと、第1ドメイン及び第2ドメイン間でデータの通信が必要な場合に、当該通信を制御するミドルウェアとを有する構成とする。

WO 2010/097925 A1

WO 2010/097925 A1



NO, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, 添付公開書類:

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

明 細 書

情報処理装置

技術分野

- [0001] 本発明は、複数個の中央演算装置と内蔵メモリを備え、或いは異なるオペレーティングシステムの管理下でプログラムの実行が可能なデータプロセッサに関し、例えば一個の半導体チップに形成されたマルチCPU(中央演算処理装置)を持つ半導体集積回路化されたマイクロプロセッサに適用して有効な技術に関する。

背景技術

- [0002] 近年、マイクロプロセッサにおける集積化が進み、単一の中央演算処理装置を搭載したマイクロプロセッサを複数搭載したシステムが集積化され、複数の中央演算処理装置を搭載したマイクロプロセッサが開発されている。複数の中央演算処理装置を搭載したマイクロプロセッサにおいては、主にLSIの面積に対する端子数の制約によって、各中央演算処理装置が利用するメモリや入出力装置については、マイクロプロセッサ内で共用化されることが一般的であり、主にマイクロプロセッサ内のバスやコントローラ的设计によって実現されている。特に、メモリを共用した場合に例えば、制御ソフトウェアのバグなどにより不正アクセスが発生し、メモリ領域のアクセスの衝突が起こると画像データが正しく処理されないといった問題が発生するので、メモリ領域のアクセスの衝突を回避することが重要になる。回路量のオーバーヘッドが少ない不正アドレスへのアクセスの見地、遮断回路を提供する技術が、特許文献1で示されている。
- [0003] 一方、複数の中央演算処理装置を搭載したマイクロプロセッサ上でプログラムを動作させる場合、基本システムであるオペレーティングシステム(OS)も複数になり、それらのOS間でのデータのやりとりや、別のOS上にある機能を複数のCPUやOSの存在を意識させないで実行する方式が必要になってくる。このような状況において、上記のようなマルチプロセッサシステム上でOSとアプリケーションの実行を制御するシステムとしては、例えば、特許文献2に開示されるように、マルチプロセッサ上でシングルスプロセッサ向けOSと既存のアプリケーションを動作させ並列処理を実現できるようにする従来技術がある。このように物理的なCPUなどシステム構成資源とOSや

アプリケーションとの結びつきを自由にする機構は仮想化と呼ばれ、サーバなどの分野で広く用いられるようになってきている。

特許文献1:特開2004-334410号公報

特許文献2:特開2003-345614号公報

発明の開示

発明が解決しようとする課題

[0004] 単一チップ上に多数のCPUやIPが集積されたマルチコアシステムでは、複雑化した機能が互いに関連し、同時に動作するため、一つの物理的なデバイスを複数のコアおよびそれらコア上で動作するOSで共有する状況が起こる。この共有において、共有の主体となるOSは、時分割で動作していて一時期には一つのOSだけがアクティブ状態であるということではなく、それぞれが常にアクティブで並列に動作してI/Oデバイスを使用するという状況である。そのような場合、実デバイスを制御するための制御レジスタ群は一つであるので、何らかの形でそれらに対するアクセスを調停し、1セットの制御レジスタ群を仮想的に複数に見せて、複数のOSからのアクセスを処理する必要がある。上記のように、マルチコア環境においては、複数のCPUが同時に稼動し、そのそれぞれが別のOS下にあるということが一般的である。一方、I/O装置やアクセラレータIPなどのデバイスは個数が限られており、それら複数のCPUやOSで共有を行いたいという要求がある。そのような共有の状況としては、例えばハードディスクを装備したカーナビゲーションシステムにおいて、一つのハードディスク装置を地図データの格納と音楽データの格納に用い、それらを別々のプロセッサで動作する経路探索と音楽再生で同時にアクセスしたい場合などがある。

[0005] 特に、複数OSを搭載した場合、一つのOSが管理するハードウェア資源は、そのOS以外からアクセスされないことを前提として作られており、そのOSに割り当てられた内蔵メモリの一部領域もしくは外部メモリの一部領域に対する、そのOS以外からアクセス、特にデータが書き換えられる様な想定外の事象は、そのOS並びにその上で実行するソフトウェアモジュールに対して異常動作を引き起こす要因となる。従って、内蔵メモリまたは外部メモリにおいては、複数のOSで共有しないように分離して管理するのが望ましい。しかしながら、内蔵メモリや外部メモリは、コスト等の観点から無限に

搭載することはできない。従って、他のOSに影響を与えないように、効率よくメモリ資源を用いることが望まれる。また、OSが異なるといっても、一つのシステム上に搭載されているため、OS間での通信が必要となる。この場合、他のOSに影響を与えないようにデータ通信を行うことが必要となる。

課題を解決するための手段

- [0006] 上記の課題に対して本発明においては、第1中央処理装置と、第2中央処理装置と、第1中央処理装置にて実行される第1オペレーティングシステムと、第2中央処理装置にて実行される第2オペレーティングシステムと、第1中央処理装置及び第2中央処理装置にアクセスされるメモリとを有する情報処理装置であって、情報処理装置を、第1中央処理装置、第1オペレーティングシステム、及び、メモリの領域のうち一部の第1領域を第1ドメインとして割り当て、第2中央処理装置、第2オペレーティングシステム及びメモリの領域のうち一部の第2領域を第2ドメインとして割り当て、一方のドメインが他のドメインに割当てられた領域にアクセスしないように制御するファームウェアと、第1ドメイン及び第2ドメイン間でデータの通信が必要な場合に、当該通信を制御するミドルウェアとを有する。
- [0007] 更に望ましくは、ファームウェアからの設定に基づいて、一方のドメインが他方のドメインに割当てられた領域にアクセスする際に、そのアクセスを遮断するアクセス制御モジュールを更に具備する。
- [0008] また、第1オペレーティングシステムのコードと第2オペレーティングシステムのコードが共有できる場合において、メモリは、メモリの領域の一部の第3領域に、共有できるコードを保持し、ファームウェアは、第3領域に対し、第1ドメイン及び第2ドメインからリードアクセスを許可し、ライトアクセスを許可しないように制御する。
- [0009] また、メモリは、第1ドメイン及び第2ドメイン間でデータの通信を行う場合に、通信用データを格納する第4領域を含み、ファームウェアは、第1及び第2ドメインから通信の要求がない場合には、第4領域を第1ドメイン及び第2ドメインからアクセスできないように制御し、ミドルウェアは、第1ドメインが第2ドメインにデータの通信を行う旨の要求を受けた場合に、第1ドメインに第4領域へのアクセスを許可し、第2ドメインに第4領域へのアクセスを許可しないようにファームウェアを制御した状態で、第1ドメイン

に第4領域へのアクセスを許可する。

- [0010] また、メモリは、第1ドメインが第2ドメインにデータの通信を行う場合に、通信用のデータを書き込む第5領域と、第2ドメインが第1ドメインにデータの通信を行う場合に、通信用のデータを書き込む第6領域とを有し、ファームウェアは、第5領域に対して第1ドメインがアクセス可能で、第2ドメインはアクセス不可能になるように制御し、第6領域に対して第2ドメインがアクセス可能で、第1ドメインがアクセス不可能となるように制御し、ミドルウェアは、第1ドメインが第2ドメインにデータの通信を行う旨の要求を受けた場合に、第1ドメインが第5領域に通信用のデータを書き込んだ後、第5領域及び第6領域が第1及び第2ドメインからアクセス不可能となるようにファームウェアに指示し、その後、第5領域から第6領域へのデータを転送する。

発明の効果

- [0011] 本願において開示される発明によれば、マルチコア環境において、ソフトウェア開発に要する期間が短縮でき、また入出力処理の効率化が可能になる。

図面の簡単な説明

- [0012] [図1]ソフトウェアシステムの構成を示した図である。
[図2]OS間のドメイン間通信及び各メモリ領域について説明した図である。
[図3]他のOS間のドメイン間通信及び各メモリ領域について説明した図である。
[図4]他のOS間のドメイン間通信及び各メモリ領域について説明した図である。
[図5]他のOS間のドメイン間通信及び各メモリ領域について説明した図である。

符号の説明

- [0013] 100:マルチコアプロセッサ、101~108:CPU、110:プロセッサ内バス、120及び121:アクセス制御ハードウェア、130:外部メモリ、140:ドメイン分離ファームウェア、141~4:OS、150:ドメイン連携ミドルウェア、151~4:ソフトウェアモジュール、160:内蔵メモリ、171及び172:MMU(メモリ管理ユニット)、181及び182:仮想メモリ空間、200~208:外部メモリの領域、210~218及び220~229:各OSの仮想メモリ空間の領域、250:内蔵メモリ管理テーブル、260~269:内蔵メモリの領域、300~309及び310~319:アクセス制御ハードウェアの設定領域。

発明を実施するための最良の形態

[0014] 図1は、本発明における情報処理装置のハードウェアとソフトウェアの構成の一例である。

[0015] 本実施例における情報処理装置のハードウェアの構成要素であるマルチコアプロセッサ(100)は、 n 個の中央演算処理装置(以下、「CPU」と呼ぶ。)(CPU0(101)～CPU n (108))と外部メモリ(130)と内蔵メモリ(160)、並びにアクセス制御ハードウェア(120)とがプロセッサ内バス(110)で接続された構成を備え、各CPUからは、プロセッサ内バス(110)を介して、外部メモリ(130)ならびに内蔵メモリ(160)の全ての領域に対するアクセスを可能とする。また、図示していないが、外部メモリを制御するメモリ制御装置やシリアルIO等の入出力制御装置がプロセッサ内バスに接続される。なお、入出力制御装置は、外部メモリ(130)や内蔵メモリ(160)と同様に一つのメモリ空間の中に含まれ、その制御は同じになるため、外部メモリや内蔵メモリに代表されるものとし、以下、入出力制御装置に関する説明は省略する。

[0016] アクセス制御ハードウェア(120)は、 n 個のCPU(CPU0(101)～CPU n (108))のそれぞれがプロセッサ内バス(110)を介して外部メモリ(130)、内蔵メモリ(160)をアクセスする際に、プロセッサ内バス(110)上に生成するトランザクションを監視し、個々のトランザクションに対して、あらかじめ指定されたアクセスの可否を定めたルールに基づきアクセスの可否を判断する。ここで、前記アクセスの可否を定めたルールにもとづき許可されたアクセスに関するトランザクションに関しては通常通り処理されるが、前記ルールにもとづき許可されていないアクセスに関するトランザクションに関しては、アクセス違反として無効化(遮断)されるとともに、アクセス違反の発生に対する例外処理を起動するために、当該トランザクションを生成したCPUに対して割込み信号を送ることを特徴とする。また、アクセス制御ハードウェア(120)は、個々のCPUから内蔵メモリと外部メモリの領域に対するアクセスの可否をその先頭アドレスと領域のサイズにより指定される。また、アクセスの可否は、各OS毎にリードアクセス/ライトアクセスの夫々について指定される。なお、各領域に対する設定は、リードアクセス/ライトアクセスが許可されるOSを指定するのみとし、設定されていないOSについては、アクセスを許可しないようにする。これにより、アクセス制御ハードウェア(120)

を小さくすることが可能となる。

[0017] 本実施例の情報処理装置におけるソフトウェア構成は、ドメイン分離ファームウェア(140)が、前記マルチコアプロセッサ(100)上で m 個のOS(OS0(141)、OS1(142)、OS2(143)～OS m (144))を実行するために必要となる環境を提供するためにハードウェア資源の管理を実施することを特徴とする。

[0018] 具体的には、本実施例では、前記ドメイン分離ファームウェア(140)及び各OS0～ m (141～144)は、ハードウェア資源の管理のため、以下のことを行う。

(1)ドメイン分離ファームウェア(140)は、OS0(141)に対し、CPU0(101)とCPU1(102)の2つのCPU、並びに内蔵メモリ(160)の一部領域と外部メモリ(130)の一部領域をハードウェア資源として割り当て、さらにOS0(141)は、割り当てられたハードウェア資源を管理し、ソフトウェアモジュール0(151)に対してプログラムを実行するための環境を提供する。

(2)ドメイン分離ファームウェア(140)は、OS1(142)に対し、CPU2(103)とCPU3(104)の2つのCPU、並びに内蔵メモリ(160)の一部領域と外部メモリ(130)の一部領域をハードウェア資源として割り当て、さらにOS1(142)は、割り当てられたハードウェア資源を管理し、ソフトウェアモジュール1(152)に対してプログラムを実行するための環境を提供する。

(3)ドメイン分離ファームウェア(140)は、OS2(143)に対し、CPU4(105)とCPU5(106)の2つのCPU、並びに内蔵メモリ(160)の一部領域と外部メモリ(130)の一部領域をハードウェア資源として割り当て、さらにOS2(143)は、割り当てられたハードウェア資源を管理し、ソフトウェアモジュール2(153)に対してプログラムを実行するための環境を提供する。

(4)以下同様に、ドメイン分離ファームウェア(140)は、OS m (144)に対し、CPU $n-1$ (107)とCPU n (108)の2つのCPU、並びに内蔵メモリ(160)の一部領域と外部メモリ(130)の一部領域をハードウェア資源として割り当て、さらにOS m (144)は、割り当てられたハードウェア資源を管理し、ソフトウェアモジュール m (154)に対してプログラムを実行するための環境を提供する。

[0019] また、前記ドメイン分離ファームウェア(140)は、前記アクセス制御ハードウェア(12

0)に対しアクセスの可否を判断するルールの実施する。

[0020] ここで、通常のOSは、そのOSが管理するハードウェア資源は、そのOS以外からアクセスされないことを前提として作られており、そのOSに割り当てられた内蔵メモリ(160)の一部領域もしくは外部メモリ(130)の一部領域に対する、そのOS以外からアクセス、特にデータが書き換えられる様な想定外の事象は、そのOS並びにその上で実行するソフトウェアモジュールに対して異常動作を引き起こす要因となる。

[0021] この要因を防ぐために本実施例では、ドメイン分離ファームウェア(140)は、アクセス制御ハードウェア(120)に対し、アクセスの可否を判断するルールを以下のように設定する。

(1) OS0(141)に割り当てられた内蔵メモリ(160)の一部領域もしくは外部メモリ(130)の一部領域に対して、OS1(142)並びにソフトウェアモジュール1(152)からアクセス、さらにはデータの書き換えが不可能とするよう、CPU2(103)とCPU3(104)から当該内部メモリ(160)の一部領域並びに当該外部メモリ(130)の一部領域に対してアクセス、特にデータの書き換えを不可能とする。また、OS2(143)並びにソフトウェアモジュール2(153)からアクセス、さらにはデータの書き換えが不可能とするよう、CPU4(105)とCPU5(106)から当該内部メモリ(160)の一部領域並びに当該外部メモリ(130)の一部領域に対してアクセス、特にデータの書き換えを不可能とする。以下同様にして、OS_m(144)並びにソフトウェアモジュール_m(154)からアクセス、さらにはデータの書き換えが不可能とするよう、CPU_{n-1}(107)とCPU_n(108)から当該内部メモリ(160)の一部領域並びに当該外部メモリ(130)の一部領域に対してアクセス、特にデータの書き換えを不可能とする。

(2) OS1(142)に割り当てられた内蔵メモリ(160)の一部領域もしくは外部メモリ(130)の一部領域に対して、OS0(141)並びにソフトウェアモジュール0(151)からアクセス、さらにはデータの書き換えが不可能とするよう、CPU0(101)とCPU1(102)から当該内部メモリ(160)の一部領域並びに当該外部メモリ(130)の一部領域に対してアクセス、特にデータの書き換えを不可能とする。また、OS2(143)並びにソフトウェアモジュール2(153)からアクセス、さらにはデータの書き換えが不可能とするよう、CPU4(105)とCPU5(106)から当該内部メモリ(160)の一部領域並びに当該

外部メモリ(130)の一部領域に対してアクセス、特にデータの書き換えを不可能とする。以下同様にして、OS_m(144)並びにソフトウェアモジュール_m(154)からアクセス、さらにはデータの書き換えが不可能とするよう、CPU_{n-1}(107)とCPU_n(108)から当該内部メモリ(160)の一部領域並びに当該外部メモリ(130)の一部領域に対してアクセス、特にデータの書き換えを不可能とする。

(3) OS₂(143)に割り当てられた内蔵メモリ(160)の一部領域もしくは外部メモリ(130)の一部領域に対して、OS₀(141)並びにソフトウェアモジュール₀(151)からアクセス、さらにはデータの書き換えが不可能とするよう、CPU₀(101)とCPU₁(102)から当該内部メモリ(160)の一部領域並びに当該外部メモリ(130)の一部領域に対してアクセス、特にデータの書き換えを不可能とする。また、OS₁(142)並びにソフトウェアモジュール₁(152)からアクセス、さらにはデータの書き換えが不可能とするよう、CPU₂(103)とCPU₃(104)から当該内部メモリ(160)の一部領域並びに当該外部メモリ(130)の一部領域に対してアクセス、特にデータの書き換えを不可能とする。以下同様にして、OS_m(144)並びにソフトウェアモジュール_m(154)からアクセス、さらにはデータの書き換えが不可能とするよう、CPU_{n-1}(107)とCPU_n(108)から当該内部メモリ(160)の一部領域並びに当該外部メモリ(130)の一部領域に対してアクセス、特にデータの書き換えを不可能とする。

(4) 以下同様にしてOS_m(144)に割り当てられた内蔵メモリ(160)の一部領域もしくは外部メモリ(130)の一部領域に対して、OS₀(141)並びにソフトウェアモジュール₀(151)からアクセス、さらにはデータの書き換えが不可能とするよう、CPU₀(101)とCPU₁(102)から当該内部メモリ(160)の一部領域並びに当該外部メモリ(130)の一部領域に対してアクセス、特にデータの書き換えを不可能とする。また、OS₁(142)並びにソフトウェアモジュール₁(152)からアクセス、さらにはデータの書き換えが不可能とするよう、CPU₂(103)とCPU₃(104)から当該内部メモリ(160)の一部領域並びに当該外部メモリ(130)の一部領域に対してアクセス、特にデータの書き換えを不可能とする。以下同様にして、OS₂(143)並びにソフトウェアモジュール₂(153)からアクセス、さらにはデータの書き換えが不可能とするよう、CPU₄(105)とCPU₅(106)から当該内部メモリ(160)の一部領域並びに当該外部メモリ(130)の一

部領域に対してアクセス、特にデータの書き換えを不可能とする。

- [0022] このように設定することにより、各OSに割当てられた資源は、基本的に他のOSからアクセスされないため、他のOSの不正なアクセスによる異常動作を防止することが可能となる。
- [0023] また、本実施例では、OS間の通信を行うためにドメイン連携ミドルウェア(150)を実装する。ドメイン連携ミドルウェア(150)は、本実施例のアクセス制御ハードウェア(120)とドメイン分離ファームウェア(140)を活用し、OS0(141)とソフトウェアモジュール0(151)とが実行するためのハードウェア資源、OS1(142)とソフトウェアモジュール1(152)とが実行するためのハードウェア資源、OS2(143)とソフトウェアモジュール2(153)とが実行するためのハードウェア資源、並びに以下同様にOS_m(144)とソフトウェアモジュール_m(154)とが実行するためのハードウェア資源が分離されているシステム構成において、ソフトウェアモジュール(ソフトウェアモジュール0(151)、ソフトウェアモジュール1(152)、ソフトウェアモジュール2(153)～ソフトウェアモジュール_m(154))のそれぞれの間で処理の連携を実施するための通信機能を提供する。
- [0024] このように新たにドメイン連携ミドルウェア(150)を設ける理由は、実装を容易化するためである。即ち、ドメイン分離ファームウェアは、基本的に各OS間を確実に分離することにある。一方、OS間での連携は、分離した各OS間をつなぐことになる。このように全く機能が異なるものをドメイン分離ファームウェアに取り込むと、OS間通信が不要な場合であっても分離した各OSの間に穴をあけられることになり、好ましくない。一方、OSにOS間通信を行うための機能を実装しようとする、他のOSの状況等を考慮するように変更しなければならず、その改変に時間がかかる。それに対し、本願のようにドメイン連携ミドルウェアを新たに設け、OSからOS間通信を行いたい旨の要求を出すようにし、全体を把握しているドメイン連携ミドルウェアでその可否を判断し、通信に必要な設定をドメイン分離ファームウェアにする構成とすることにより、OS間通信が必要な場合のみ、当該ドメイン連携ファームウェアを実装すればよいことになる。
- [0025] なお、ドメイン分離ファームウェア(140)及びドメイン連携ミドルウェア(150)は、OSが必要な場合に読み出す構成となっており、その読み出したOSが所属するドメイン

内のCPUにより実行されてもよいし、異なるCPUで実行しても良い。

[0026] <実施例1>

図2は、図1に示した実施例の中で、個々のソフトウェアモジュール(ソフトウェアモジュール0(151)～ソフトウェアモジュールm(154))に対して実行するためのハードウェア資源としてCPU(CPU0(101)～CPU_n(108))と外部メモリ(130)に限定し、ドメイン分離ファームウェア(140)は各OS(OS0(141)～OS_m(144))に対し、CPU(CPU0(101)～CPU_n(108))と外部メモリ(130)を分割したものを割り当てた、一実施例において、OS0(141)とソフトウェアモジュール0(151)、並びにOS_m(144)とソフトウェアモジュールm(154)とに注目し示したものである。

[0027] 仮想メモリ空間(181)は、OS0(141)とソフトウェアモジュール0(151)が実行するためのメモリ領域を構成したものであり、仮想メモリ空間(182)は、OS_m(144)とソフトウェアモジュールm(154)が実行するためのメモリ領域を構成したものであり、それぞれの実態は外部メモリ(130)に存在する。

[0028] OS0(141)とソフトウェアモジュール0(151)が実行するためのメモリ領域として外部メモリ(130)に割り当てられたものの実態は、外部メモリ(130)上のOS code 1(OSC1, 200)、OS code 2(OSC2, 201)、OS個別data a(OSDa, 202)、AP code 1(APC1, 204)、AP code 2(APC2, 205)、AP個別data a(APDa, 206)、OS間通信data a(OSMa, 208)で、これらは、MMU0(171)により仮想メモリ空間(181)上のOS code 1(OSC1, 210)、OS code 2(OSC2, 211)、OS個別data a(OSDa, 212)、AP code 1(APC1, 214)、AP code 2(APC2, 215)、AP個別data a(APDa, 216)、OS間通信data a(OSMa, 218)にアドレス変換されて、OS0(141)とソフトウェアモジュール0(151)からアクセスされる。

[0029] 同様にOS_m(144)とソフトウェアモジュールm(154)が実行するためのメモリ領域として割り当てられたものの実態は、外部メモリ(130)上のOS code 1(OSC1, 200)、OS code 2(OSC2, 201)、OS個別data b(OSDb, 203)、AP code 1(APC1, 204)、AP code 2(APC2, 205)、AP個別data b(APDb, 207)、OS間通信data a(OSMa, 208)で、これらは、MMU1(172)により仮想メモリ空間(182)上のOS code 1(OSC1, 220)、OS code 2(OSC2, 221)、OS個別data

b(OSDb, 223)、AP code 1(APC1, 224)、AP code 2(APC2, 225)、AP個別data b(APDb, 227)、OS間通信data a(OSMa, 228)にアドレス変換されて、OSm(144)とソフトウェアモジュールm(154)からアクセスされる。

[0030] この様なハードウェア資源のソフトウェアに対する割り当てに対して、本実施例におけるアクセス制御ハードウェア(120)は、MMU0(171)およびMMU1(172)により変換されたアドレスとCPUとの組み合わせにより、アクセスの可否を判断する。具体的には、アクセス制御ハードウェア(120)がアクセスの可否を判断するルールは、以下の通りとなり、それぞれドメイン分離ファームウェア(140)が設定する。

(1)OS code 1(OSC1, 200)とOS code 2(OSC2, 201)に対するアクセス可否ルールエントリ(300)には、OS0(141)とソフトウェアモジュール0(151)、並びにOSm(144)とソフトウェアモジュールm(154)、を実行しているCPUからの読みアクセス可。

(2)OS個別data a(OSDa, 202)に対するアクセス可否ルールエントリ(302)には、OS0(141)とソフトウェアモジュール0(151)を実行しているCPUからの読み書きアクセス可。

(3)OS個別data b(OSDb, 203)に対するアクセス可否ルールエントリ(303)には、OSm(144)とソフトウェアモジュールm(154)を実行しているCPUからの読み書きアクセス可。

(4)AP code 1(APC1, 204)とAP code 2(205)に対するアクセス可否ルールエントリ(304)には、OS0(141)とソフトウェアモジュール0(151)、並びにOSm(144)とソフトウェアモジュールm(154)、を実行しているCPUからの読みアクセス可。

(5)AP個別data a(APDa, 206)に対するアクセス可否ルールエントリ(306)には、OS0(141)とソフトウェアモジュール0(151)を実行しているCPUからの読み書きアクセス可。

(6)AP個別data b(APDb, 207)に対するアクセス可否ルールエントリ(307)には、OSm(144)とソフトウェアモジュールm(154)を実行しているCPUからの読み書きアクセス可。

(7)それ以外の外部メモリの領域に対するアクセス可否ルールエントリ(アクセス可否ルールエントリ(310)、アクセス可否ルールエントリ(311)、アクセス可否ルールエントリ(312)、アクセス可否ルールエントリ(313))には、OS0(141)とソフトウェアモジュール0(151)、並びにOSm(144)とソフトウェアモジュールm(154)、を実行しているCPUからの全てのアクセス不可。

[0031] また、OS間通信data a(OSMa, 208)に対するアクセス可否ルールエントリ(308)には、初期状態として、OS0(141)とソフトウェアモジュール0(151)、並びにOSm(144)とソフトウェアモジュールm(154)、を実行しているCPUからの全てのアクセス不可、とドメイン分離ファームウェア(140)が設定する。

[0032] なお、アクセス制御モジュール(120)に記載された記号の意味は、“R”がリードアクセス許可、“W”がライトアクセス許可、“0”がOS0に管理されるCPUからのアクセスが対象、“m”がOSmに管理されているCPUからのアクセスが対象である。従って、例えば、“R0m”となっているものは、OS0及びOSmが管理しているCPUからリードアクセスが可能(ライトアクセスは不可能)という意味となる。

[0033] ここで、特徴的なものは、OS code 1及び2、AP code 1及び2の領域(310, 313)に対し、OS0とOSmの両方が管理しているCPUからリードアクセスできるようにアクセス制御ハードウェア(120)が設定されていることである。上述したとおり、各OSは、夫々が単独で動作し、他のOSからアクセスされないことが前提となっている。従って、夫々が管理する領域が完全に分離していることが望ましい。しかしながら、他のOSからのアクセスが異常動作の原因となるのは、データが書き換わってしまうことに起因する。ここで、CPUの動作を考えると、OS用のコードやアプリケーション用のコードは、各CPUからはリードアクセスのみで充分である。そこで、本発明では、OS用のコードとアプリケーション用のコードを共有できるものは、複数のOS間で共有し、かつ、アクセス制御ハードウェア(120)にて、リードアクセスのみ可と設定することにより、異常動作の要因となるライトアクセスを遮断し、安全に、かつ、効率よくメモリ領域を使用することを可能としている。なお、この点については、他の実施例も同様である。

[0034] 続いて、OS間の通信について詳しく言及する。OS0(141)とソフトウェアモジュール0(151)、もしくはOSm(144)とソフトウェアモジュールm(154)の何れかがOS間

通信のためにOS間通信Data a(OSMa, 208)に対し、データの読み書きを実施する際は、それに先立ちドメイン連携ミドルウェア(150)に対して当該領域に対する読み書きアクセス要求を出す。当該領域に対する読み書き要求を受け取ったドメイン連携ミドルウェア(150)は、現在OS間通信を行ってよいかを判断し、ドメイン分離ファームウェア(140)を介して、OS間通信Data a(OSMa, 208)に対するアクセス可否ルールエントリ(308)に、当該要求を出したOSとソフトウェアモジュールを実行しているCPUからの読み書きアクセス可と設定し、その際、同時にOS0(141)とソフトウェアモジュール0(151)、並びにOSm(144)とソフトウェアモジュールm(154)の実行する双方のCPUからの読み書きアクセス可とすることのないよう制御する。その後、書き込み要求を出したOSとソフトウェアモジュールに対し、読み書きアクセス要求の受理を返す。

[0035] OS間通信Data a(OSMa, 208)に対する読み書きアクセス要求の受理を受け取ったOSとソフトウェアモジュールは、OS間通信Data a(OSMa, 208)に対しデータの読み書きを実施した後、ドメイン連携ミドルウェア(150)に対し、当該領域に対する読み書き完了を出す。

[0036] 当該領域に対する読み書き完了を受け取ったドメイン連携ミドルウェア(150)は、ドメイン分離ファームウェア(140)を介して、OS間通信Data a(OSMa, 208)に対するアクセス可否ルールエントリ(308)に、初期状態と同様に、OS0(141)とソフトウェアモジュール0(151)、並びにOSm(144)とソフトウェアモジュールm(154)、を実行しているCPUからの全てのアクセス不可、を設定する。

[0037] 以上の通り、本実施例では、OS間通信で用いる領域(208)を全てのOS(CPU)に対しデフォルトでリードアクセス/ライトアクセスを不可とし、必要に応じてドメイン連携ミドルウェア及びドメイン分離ファームウェアを用いて、要求を出したOS(CPU)のみにアクセス許可を与える。これにより、通信用に用いるデータを他のOS(CPU)がアクセスし、情報を読み出す、又は、情報を破壊するということから守ることが可能となる。

[0038] <実施例2>

図3は、図2と同様に、図1に示した実施例の中で、個々のソフトウェアモジュール(

ソフトウェアモジュール0(151)～ソフトウェアモジュールm(154))に対して実行するためのハードウェア資源としてCPU(CPU0(101)～CPU_n(108))と外部メモリ(130)に限定し、ドメイン分離ファームウェア(140)は各OS(OS0(141)～OS_m(144))に対し、CPU(CPU0(101)～CPU_n(108))と外部メモリ(130)を分割したものを割り当てた、一実施例において、OS0(141)とソフトウェアモジュール0(151)、並びにOS_m(144)とソフトウェアモジュールm(154)とに注目し示したものである。実施例1では、OS間通信を行うメモリ領域(208)をOS0とOS_mで共有していたが、本実施例では、夫々個別の領域(208、209)を有している。以下、本実施例について説明する。

[0039] 仮想メモリ空間(181)は、OS0(141)とソフトウェアモジュール0(151)が実行するためのメモリ領域を構成したものであり、仮想メモリ空間(182)は、OS_m(144)とソフトウェアモジュールm(154)が実行するためのメモリ領域を構成したものであり、それぞれの実態は外部メモリ(130)に存在する。

[0040] OS0(141)とソフトウェアモジュール0(151)が実行するためのメモリ領域として外部メモリ(130)に割り当てられたものの実態は、外部メモリ(130)上のOS code 1(OSC1, 200)、OS code 2(OSC2, 201)、OS個別data a(OSDa, 202)、AP code 1(APC1, 204)、AP code 2(APC2, 205)、AP個別data a(APDa, 206)、OS間通信data a(OSMa, 208)で、これらは、MMU0(171)により仮想メモリ空間(181)上のOS code 1(OSC1, 210)、OS code 2(OSC2, 211)、OS個別data a(OSDa, 212)、AP code 1(APC1, 214)、AP code 2(APC2, 215)、AP個別data a(APDa, 216)、OS間通信data a(OSMa, 218)にアドレス変換されて、OS0(141)とソフトウェアモジュール0(151)からアクセスされる。

[0041] 同様に、OS_m(144)とソフトウェアモジュールm(154)が実行するためのメモリ領域として割り当てられたものの実態は、外部メモリ(130)上のOS code 1(OSC1, 200)、OS code 2(OSC2, 201)、OS個別data b(OSDb, 203)、AP code 1(APC1, 204)、AP code 2(APC2, 205)、AP個別data b(APDb, 207)、OS間通信data b(OSMb, 209)で、これらは、MMU1(172)により仮想メモリ空間(182)上のOS code 1(OSC1, 220)、OS code 2(OSC2, 221)、OS個別

data b(OSDb, 223)、AP code 1(APC1, 224)、AP code 2(APC2, 225)、AP個別data b(APDb, 227)、OS間通信data b(OSMb, 229)にアドレス変換されて、OSm(144)とソフトウェアモジュールm(154)からアクセスされる。

[0042] この様なハードウェア資源のソフトウェアに対する割り当てに対して、本実施例におけるアクセス制御ハードウェア(120)は、MMU0(171)およびMMU1(172)により変換されたアドレスとCPUとの組み合わせにより、アクセスの可否を判断する。従って、アクセス制御ハードウェア(120)がアクセスの可否を判断するルールは、以下の通りとなり、それぞれドメイン分離ファームウェア(140)が設定する。

(1)OS code 1(OSC1, 200)とOS code 2(OSC2, 201)に対するアクセス可否ルールエントリ(300)には、OS0(141)とソフトウェアモジュール0(151)、並びにOSm(144)とソフトウェアモジュールm(154)、を実行しているCPUからの読みアクセス可。

(2)OS個別data a(OSDa, 202)に対するアクセス可否ルールエントリ(302)には、OS0(141)とソフトウェアモジュール0(151)を実行しているCPUからの読み書きアクセス可。

(3)OS個別data b(OSDb, 203)に対するアクセス可否ルールエントリ(303)には、OSm(144)とソフトウェアモジュールm(154)を実行しているCPUからの読み書きアクセス可。

(4)AP code 1(APC1, 204)とAP code 2(APC2, 205)に対するアクセス可否ルールエントリ(304)には、OS0(141)とソフトウェアモジュール0(151)、並びにOSm(144)とソフトウェアモジュールm(154)、を実行しているCPUからの読みアクセス可。

(5)AP個別data a(APDa, 206)に対するアクセス可否ルールエントリ(306)には、OS0(141)とソフトウェアモジュール0(151)を実行しているCPUからの読み書きアクセス可。

(6)AP個別data b(APDb, 207)に対するアクセス可否ルールエントリ(307)には、OSm(144)とソフトウェアモジュールm(154)を実行しているCPUからの読み書きアクセス可。

(7)それ以外の外部メモリの領域に対するアクセス可否ルールエントリ(アクセス可否ルールエントリ(310)、アクセス可否ルールエントリ(311)、アクセス可否ルールエントリ(312)、アクセス可否ルールエントリ(313))には、OS0(141)とソフトウェアモジュール0(151)、並びにOSm(144)とソフトウェアモジュールm(154)、を実行しているCPUからの全てのアクセス不可。

(8)OS間通信Data a(OSMa, 208)に対するアクセス可否ルールエントリ(308)には、OS0(141)とソフトウェアモジュール0(151)を実行しているCPUからの読み書きアクセス可。

(9)OS間通信data b(OSMb, 209)に対するアクセス可否ルールエントリ(309)には、OSm(144)とソフトウェアモジュールm(154)を実行しているCPUからの読み書きアクセス可。

[0043] 本実施例では、ソフトウェアモジュール0(151)からソフトウェアモジュールm(154)へOS間通信を実施する際は、ソフトウェアモジュール0(151)が外部メモリ(130)上のOS間通信Data a(OSMa, 208)にデータの書き込みが完了した時点で、ドメイン連携ミドルウェア(150)に転送要求を出す。

[0044] 転送要求を受け取ったドメイン連携ミドルウェア(150)は、ドメイン分離ファームウェア(140)を用いてOS間通信Data a(OSMa, 208)に対するアクセス可否ルールエントリ(308)とOS間通信Data b(OSMb, 209)に対するアクセス可否ルールエントリ(309)とに対しOS0(141)とソフトウェアモジュール0(151)、並びにOSm(144)とソフトウェアモジュールm(154)、を実行しているCPUからの全てのアクセス不可、と設定する。その後、ドメイン連携ミドルウェア(150)は、ドメイン分離ファームウェア(140)を用いてOS間通信Data a(OSMa, 208)に対するアクセス可否ルールエントリ(308)とOS間通信Data b(OSMb, 209)に対するアクセス可否ルールエントリ(309)に、ドメイン連携ミドルウェア(150)自身を実行しているCPUからの読み書きアクセス可と設定し、OS間通信Data a(OSMa, 208)に書き込まれたデータをOS間通信Data b(OSMb, 209)の指定された場所へ書き出す。その後、再びドメイン分離ファームウェア(140)を用いてOS間通信Data a(OSMa, 208)に対するアクセス可否ルールエントリ(308)とOS間通信Data b(OSMb, 209)に対する

アクセス可否ルールエントリ(309)に、ドメイン連携ミドルウェア(150)を実行しているCPUからの全てのアクセス不可と設定し、完了した後、ドメイン分離ファームウェア(140)を用いてOS間通信Data a(OSMa, 208)に対するアクセス可否ルールエントリ(308)には、OS0(141)とソフトウェアモジュール0(151)を実行しているCPUからの読み書きアクセス可、OS間通信data b(OSMb, 209)に対するアクセス可否ルールエントリ(309)には、OSm(144)とソフトウェアモジュールm(154)を実行しているCPUからの読み書きアクセス可、と、設定し、ソフトウェアモジュール0(151)に対し、転送完了を出す。

[0045] ソフトウェアモジュールm(154)からソフトウェアモジュール0(151)へOS間通信を実施する際は、ソフトウェアモジュールm(154)が外部メモリ(130)上のOS間通信Data b(OSMb, 209)にデータの書き込みが完了した時点で、ドメイン連携ミドルウェア(150)に転送要求を出す。

[0046] 転送要求を受け取ったドメイン連携ミドルウェア(150)は、ドメイン分離ファームウェア(140)を用いて、OS間通信Data a(OSMa, 208)に対するアクセス可否ルールエントリ(308)とOS間通信Data b(OSMb, 209)に対するアクセス可否ルールエントリ(309)とに対しOS0(141)とソフトウェアモジュール0(151)、並びにOSm(144)とソフトウェアモジュールm(154)、を実行しているCPUからの全てのアクセス不可、と設定する。その後、ドメイン連携ミドルウェア(150)は、ドメイン分離ファームウェア(140)を用いてOS間通信Data a(OSMa, 208)に対するアクセス可否ルールエントリ(308)とOS間通信Data b(OSMb, 209)に対するアクセス可否ルールエントリ(309)に、ドメイン連携ミドルウェア(150)自身を実行しているCPUからの読み書きアクセス可と設定し、OS間通信Data b(OSMb, 209)に書き込まれたデータをOS間通信Data a(OSMa, 208)の指定された場所へ書き出す。その後、再びドメイン分離ファームウェア(140)を用いてOS間通信Data a(OSMa, 208)に対するアクセス可否ルールエントリ(308)とOS間通信Data b(OSMb, 209)に対するアクセス可否ルールエントリ(309)に、ドメイン連携ミドルウェア(150)を実行しているCPUからの全てのアクセス不可と設定し、完了した後、ドメイン分離ファームウェア(140)を用いてOS間通信Data a(OSMa, 208)に対するアクセス可否ルールエントリ

リ(308)には、OS0(141)とソフトウェアモジュール0(151)を実行しているCPUからの読み書きアクセス可、OS間通信data b(OSMb, 209)に対するアクセス可否ルールエントリ(309)には、OSm(144)とソフトウェアモジュールm(154)を実行しているCPUからの読み書きアクセス可、と、設定し、ソフトウェアモジュールm(154)に対し、転送完了を出す。

[0047] 以上の通り、本実施例においても、OS間通信用のデータを転送している間は、すべてのOS(CPU)からのアクセスを不可とするため、通信用データが破壊されない。また、実施例1に対し、各OSは、通信の許可を受ける前に、通信用のデータを自分の領域に書き込むことができるため、次の処理に移ることが可能となり、高速に動作することが可能となる。

[0048] <実施例3>

図4は、図1に示した実施例の中で、個々のソフトウェアモジュール(ソフトウェアモジュール0(151)～ソフトウェアモジュールm(154))に対して実行するためのハードウェア資源としてCPU(CPU0(101)～CPU_n(108))と外部メモリ(130)と内蔵メモリ(160)を持ち、ドメイン分離ファームウェア(140)は各OS(OS0(141)～OSm(144))に対し、CPU(CPU0(101)～CPU_n(108))、外部メモリ(130)を分割したもの、内蔵メモリ(160)を分割したものを割り当てた、一実施例において、OS0(141)とソフトウェアモジュール0(151)、並びにOSm(144)とソフトウェアモジュールm(154)とに注目し示したものである。即ち、本実施例では、実施例1に対し、外部メモリ(130)から内蔵メモリ(160)に必要なデータをコピーし、その内蔵メモリ(160)に対して、CPUがアクセスすることにより処理速度を向上させ、加えて内蔵メモリを用いてOS間通信を行う点が異なる。なお、紙面の都合上、OSとソフトウェアモジュールについては記載を省略しているが、この部分は、図2又は図3と同様である。

[0049] OS0(141)とソフトウェアモジュール0(151)が実行するためのメモリ領域として外部メモリ(130)に割り当てられたものの実態は、外部メモリ(130)上のOS code 1(OSC1, 200)、OS code 2(OSC2, 201)、OS個別data a(OSDa, 202)、AP code 1(APC1, 204)、AP code 2(APC2, 205)、AP個別data a(APDa, 206)、OS間通信data a(OSMa, 208)で、これらは、内蔵メモリ(160)上のOS c

ode 1(OSC1, 260)、OS code 2(OSC2, 261)、OS個別data a(OSDa, 262)、AP code 1(APC1, 264)、AP code 2(APC2, 265)、AP個別data a(APDa, 266)、OS間通信data a(OSMa, 268)に実態のコピーが移され、MMU 0(171)により仮想メモリ空間(181)上のOS code 1(OSC1, 230)、OS code 2(OSC2, 231)、OS個別data a(OSDa, 232)、AP code 1(APC1, 234)、AP code 2(APC2, 235)、AP個別data a(APDa, 236)、OS間通信data a(OSMa, 238)にアドレス変換されて、OS0(141)とソフトウェアモジュール0(151)からアクセスされる。

[0050] 同様に、OSm(144)とソフトウェアモジュールm(154)が実行するためのメモリ領域として割り当てられたものの実態は、外部メモリ(130)上のOS code 1(OSC1, 200)、OS code 2(OSC2, 201)、OS個別data b(OSDb, 203)、AP code 1(APC1, 204)、AP code 2(APC2, 205)、AP個別data b(APDb, 207)、OS間通信data a(OSMa, 208)で、これらは、内蔵メモリ(160)上のOS code 1(OSC1, 260)、OS code 2(OSC2, 261)、OS個別data b(OSDb, 263)、AP code 1(APC1, 264)、AP code 2(APC2, 265)、AP個別data b(APDb, 267)、OS間通信data a(OSMa, 268)に実態のコピーが移され、MMU1(172)により仮想メモリ空間(182)上のOS code 1(OSC1, 240)、OS code 2(OSC2, 241)、OS個別data b(OSDb, 243)、AP code 1(APC1, 244)、AP code 2(APC2, 245)、AP個別data b(APDb, 247)、OS間通信data a(OSMa, 248)にアドレス変換されて、OSm(144)とソフトウェアモジュールm(154)からアクセスされる。

[0051] 内蔵メモリ管理テーブル(250)は、内蔵メモリ(160)上の空き領域、外部メモリ(130)上のメモリ領域の実態(OS code 1(OSC1, 200)、OS code 2(OSC2, 201)、OS個別data a(OSDa, 202)、OS個別data b(OSDb, 203)、AP code 1(APC1, 204)、AP code 2(APC2, 205)、AP個別data a(APDa, 206)、AP個別data b(APDb, 207)、OS間通信data a(OSMa, 208)等)のコピーが、内部メモリ(160)上のどこ(OS code 1(OSC1, 260)、OS code 2(OSC2, 261)、OS個別data a(OSDa, 262)、OS個別data b(OSDb, 263)、AP code

1 (APC1, 264)、AP code 2 (APC2, 265)、AP個別data a (APDa, 266)、AP個別data b (APDb, 267)、OS間通信data a (OSMa, 268)等)に移されたか、を管理する。

[0052] ここで、通常内蔵メモリ(160)の容量は、外部メモリの容量に比べて小さく、外部メモリ(130)上にメモリ領域として割り当てられたものの実態のコピーを、全てを内蔵メモリ(160)に保持することは可能ではない。従って、ソフトウェアモジュールとOSのアクセスに応じて、内蔵メモリ(160)上に領域を確保し、そのアクセスの対象である外部メモリ(130)上にメモリ領域の実態のコピーを、当該領域に割り付ける。ここで、内蔵メモリ管理テーブル(TBL, 250)により内蔵メモリ(160)に空きがないことが明らかな場合は、既に内蔵メモリ(160)に割り付けられているメモリ領域の実態のコピー群の中から一つを選択し、当該メモリ領域の実態のコピーを外部メモリ(130)の該当する領域に退避し、領域を確保する。

[0053] この様なハードウェア資源のソフトウェアに対する割り当てに対して、本実施例におけるアクセス制御ハードウェア(120)は、MMU0(171)およびMMU1(172)により変換されたアドレスとCPUとの組み合わせにより、アクセスの可否を判断する。従って、アクセス制御ハードウェア(120)がアクセスの可否を判断するルールは、以下の通り、それぞれドメイン分離ファームウェア(140)が設定する。

(1) OS code 1 (OSC1, 200)とOS code 2 (OSC2, 201)に対するアクセス可否ルールエントリ(300)には、OS0(141)とソフトウェアモジュール0(151)、並びにOSm(144)とソフトウェアモジュールm(154)、を実行しているCPUからの読みアクセス可。

(2) OS個別data a (OSDa, 202)に対するアクセス可否ルールエントリ(302)には、OS0(141)とソフトウェアモジュール0(151)を実行しているCPUからの読み書きアクセス可。

(3) OS個別data b (OSDb, 203)に対するアクセス可否ルールエントリ(303)には、OSm(144)とソフトウェアモジュールm(154)を実行しているCPUからの読み書きアクセス可。

(4) AP code 1 (APC1, 204)とAP code 2 (APC2, 205)に対するアクセス可

否ルールエントリ(304)には、OS0(141)とソフトウェアモジュール0(151)、並びにOSm(144)とソフトウェアモジュールm(154)、を実行しているCPUからの読みアクセス可。

(5)AP個別data a(APDa, 206)に対するアクセス可否ルールエントリ(306)には、OS0(141)とソフトウェアモジュール0(151)を実行しているCPUからの読み書きアクセス可。

(6)AP個別data b(APDb, 207)に対するアクセス可否ルールエントリ(307)には、OSm(144)とソフトウェアモジュールm(154)を実行しているCPUからの読み書きアクセス可。

(7)それ以外の外部メモリの領域に対するアクセス可否ルールエントリには、OS0(141)とソフトウェアモジュール0(151)、並びにOSm(144)とソフトウェアモジュールm(154)、を実行しているCPUからの全てのアクセス不可。

[0054] 更に、内蔵メモリ(160)を対象とするアクセス制御ハードウェア(121)がアクセスの可否を判断するルールは以下の通り、それぞれドメイン分離ファームウェア(140)が設定する。

(1)OS code 1(OSC1, 260)とOS code 2(OSC2, 261)に対するアクセス可否ルールエントリ(310)には、OS0(141)とソフトウェアモジュール0(151)、並びにOSm(144)とソフトウェアモジュールm(154)、を実行しているCPUからの読みアクセス可。

(2)OS個別data a(OSDa, 262)に対するアクセス可否ルールエントリ(311)には、OS0(141)とソフトウェアモジュール0(151)を実行しているCPUからの読み書きアクセス可。

(3)OS個別data b(OSDb, 263)に対するアクセス可否ルールエントリ(312)には、OSm(144)とソフトウェアモジュールm(154)を実行しているCPUからの読み書きアクセス可。

(4)AP code 1(APC1, 264)とAP code 2(APC2, 265)に対するアクセス可否ルールエントリ(313)には、OS0(141)とソフトウェアモジュール0(151)、並びにOSm(144)とソフトウェアモジュールm(154)、を実行しているCPUからの読みアク

セス可。

(5) AP個別data a (APDa, 266)に対するアクセス可否ルールエントリ(314)には、OS0(141)とソフトウェアモジュール0(151)を実行しているCPUからの読み書きアクセス可。

(6) AP個別data b (APDb, 267)に対するアクセス可否ルールエントリ(315)には、OSm(144)とソフトウェアモジュールm(154)を実行しているCPUからの読み書きアクセス可。

[0055] なお、本実施例では、全ての内蔵メモリの領域を用いているため、図面上、存在していないが、もし、使用しない内蔵メモリの領域があるのであれば、それ以外の内蔵メモリの領域に対するアクセス可否ルールエントリには、OS0(141)とソフトウェアモジュール0(151)、並びにOSm(144)とソフトウェアモジュールm(154)、を実行しているCPUからの全てのアクセス不可、と設定する。

[0056] ここで、OS間通信data a (OSMa, 268)に対するアクセス可否ルールエントリ(316)には、初期状態として、OS0(141)とソフトウェアモジュール0(151)、並びにOSm(144)とソフトウェアモジュールm(154)、を実行しているCPUからの全てのアクセス不可、とドメイン分離ファームウェア(140)が設定する。

[0057] OS0(141)とソフトウェアモジュール0(151)、もしくはOSm(144)とソフトウェアモジュールm(154)の何れかがOS間通信のためにOS間通信Data a (OSMa, 268)に対し、データの読み書きを実施する際は、それに先立ちドメイン連携ミドルウェア(150)に対して当該領域に対する読み書きアクセス要求を出す。

[0058] 当該領域に対する読み書き要求を受け取ったドメイン連携ミドルウェア(150)は、ドメイン分離ファームウェア(140)を用いてOS間通信Data a (OSMa, 268)に対するアクセス可否ルールエントリ(316)に、当該要求を出したOSとソフトウェアモジュールを実行しているCPUからの読み書きアクセス可と設定し、その際、同時にOS0(141)とソフトウェアモジュール0(151)、並びにOSm(144)とソフトウェアモジュールm(154)の実行する双方のCPUからの読み書きアクセス可とすることのないよう制御する。その後、書き込み要求を出したOSとソフトウェアモジュールに対し、読み書きアクセス要求の受理を返す。

- [0059] OS間通信Data a(OSMa, 268)に対する読み書きアクセス要求の受理を受け取ったOSとソフトウェアモジュールは、OS間通信Data a(OSMa, 268)に対しデータの読み書きを実施した後、ドメイン連携ミドルウェア(150)に対し、当該領域に対する読み書き完了を出す。
- [0060] 当該領域に対する読み書き完了を受け取ったドメイン連携ミドルウェア(150)は、ドメイン分離ファームウェア(140)を用いてOS間通信Data a(OSMa, 268)に対するアクセス可否ルールエントリ(316)に、初期状態と同様に、OS0(141)とソフトウェアモジュール0(151)、並びにOSm(144)とソフトウェアモジュールm(154)、を実行しているCPUからの全てのアクセス不可、を設定する。
- [0061] 以上の通り、本実施例では、内蔵メモリ(160)に必要なデータをコピーし、その内蔵メモリ(160)に対して、CPUがアクセスすることにより処理速度を向上させることが可能となる。この場合、アクセス制御ハードウェア及びドメイン分離ファームウェアでの管理を内蔵メモリ及び外部メモリの両方に設けることにより、各CPUは、内蔵メモリのみならず外部メモリへも直接アクセスすることが可能となり、内蔵メモリを介さない処理を行うことが可能となる。また、安全なOS間通信を行う際には、ドメイン分離ファームウェアによりアクセスを許可するOS(CPU)を変更する必要がある。そのため、OS間通信に関する外部メモリの領域は、全てのOS(CPU)からの直接アクセスは遮断し、内蔵メモリの領域のみで行う構成とする。これにより、ドメイン連携ミドルウェアにおけるOS間通信の管理が簡単になる。
- [0062] <実施例4>
- 図5は、図1に示した実施例の中で、個々のソフトウェアモジュール(ソフトウェアモジュール0(151)～ソフトウェアモジュールm(154))に対して実行するためのハードウェア資源としてCPU(CPU0(101)～CPU_n(108))と外部メモリ(130)と内蔵メモリ(160)を持ち、ドメイン分離ファームウェア(140)は各OS(OS0(141)～OSm(144))に対し、CPU(CPU0(101)～CPU_n(108))、外部メモリ(130)を分割したもの、内蔵メモリ(160)を分割したものを割り当てた、一実施例において、OS0(141)とソフトウェアモジュール0(151)、並びにOSm(144)とソフトウェアモジュールm(154)とに注目し示したものである。本実施例では、実施例2に対し、外部メモリ(130)から

内蔵メモリ(160)に必要なデータをコピーし、その内蔵メモリ(160)に対して、CPUがアクセスすることにより処理速度を向上させ、加えて内蔵メモリを用いてOS間通信を行うことが相違する。なお、本図では、紙面の都合上、OSとソフトウェアモジュールの記載は省略しているが、図2及び図3と同様である。

[0063] OS0(141)とソフトウェアモジュール0(151)が実行するためのメモリ領域として割り当てられたものの実態は、外部メモリ(130)上のOS code 1(OSC1, 200)、OS code 2(OSC2, 201)、OS個別data a(OSDa, 202)、AP code 1(APC1, 204)、AP code 2(APC2, 205)、AP個別data a(APDa, 206)、OS間通信data a(OSMa, 208)で、これらは、内蔵メモリ(160)上のOS code 1(OSC1, 260)、OS code 2(OSC2, 261)、OS個別data a(OSDa, 262)、AP code 1(APC1, 264)、AP code 2(APC2, 265)、AP個別data a(APDa, 266)、OS間通信data a(OSMa, 268)に実態のコピーが移され、MMU0(171)により仮想メモリ空間(181)上のOS code 1(OSC1, 230)、OS code 2(OSC2, 231)、OS個別data a(OSDa, 232)、AP code 1(APC1, 234)、AP code 2(APC2, 235)、AP個別data a(APDa, 236)、OS間通信data a(OSMa, 238)にアドレス変換されて、OS0(141)とソフトウェアモジュール0(151)からアクセスされる。

[0064] 同様に、OSm(144)とソフトウェアモジュールm(154)が実行するためのメモリ領域として割り当てられたものの実態は、外部メモリ(130)上のOS code 1(OSC1, 200)、OS code 2(OSC2, 201)、OS個別data b(OSDb, 203)、AP code 1(APC1, 204)、AP code 2(APC2, 205)、AP個別data b(APDb, 207)、OS間通信data b(OSMb, 209)で、これらは、内蔵メモリ(160)上のOS code 1(OSC1, 260)、OS code 2(OSC2, 261)、OS個別data b(OSDb, 263)、AP code 1(APC1, 264)、AP code 2(APC2, 265)、AP個別data b(APDb, 267)、OS間通信data b(OSMb, 269)に実態のコピーが移され、MMU1(172)により仮想メモリ空間(182)上のOS code 1(OSC1, 240)、OS code 2(OSC2, 241)、OS個別data b(OSDb, 243)、AP code 1(APC1, 244)、AP code 2(APC2, 245)、AP個別data b(APDb, 247)、OS間通信data b(OSMb, 249)にアドレス変換されて、OSm(144)とソフトウェアモジュールm(154)からア

セスされる。

[0065] 内蔵メモリ管理テーブル(250)は、内蔵メモリ(160)上の空き領域、外部メモリ(130)上のメモリ領域の実態(OS code 1(OSC1, 200)、OS code 2(OSC2, 201)、OS個別data a(OSDa, 202)、OS個別data b(OSDb, 203)、AP code 1(APC1, 204)、AP code 2(APC2, 205)、AP個別data a(APDa, 206)、AP個別data b(APDb, 207)、OS間通信data a(OSMa, 208)、OS間通信data b(OSMb, 209)等)のコピーが、内部メモリ(160)上のどこ(OS code 1(OSC1, 260)、OS code 2(OSC2, 261)、OS個別data a(OSDa, 262)、OS個別data b(OSDb, 263)、AP code 1(APC1, 264)、AP code 2(APC2, 265)、AP個別data a(APDa, 266)、AP個別data b(APDb, 267)、OS間通信data a(OSMa, 268)、OS間通信data b(OSMb, 269)等)に移されたか、を管理する。

[0066] ここで、通常内蔵メモリ(160)の容量は、外部メモリの容量に比べて小さく、外部メモリ(130)上にメモリ領域として割り当てられたものの実態のコピーを、全てを内蔵メモリ(160)に保持することは可能ではない。従って、ソフトウェアモジュールとOSのアクセスに応じて、内蔵メモリ(160)上に領域を確保し、そのアクセスの対象である外部メモリ(130)上にメモリ領域の実態のコピーを、当該領域に割り付ける。ここで、内蔵メモリ管理テーブル(250)により内蔵メモリ(160)に空きがないことが明らかな場合は、既に内蔵メモリ(160)に割り付けられているメモリ領域の実態のコピー群の中から一つを選択し、当該メモリ領域の実態のコピーを外部メモリ(130)の該当する領域に退避し、領域を確保する。

[0067] この様なハードウェア資源のソフトウェアに対する割り当てに対して、本実施例におけるアクセス制御ハードウェア(120)は、MMU0(171)およびMMU1(172)により変換されたアドレスとCPUとの組み合わせにより、アクセスの可否を判断する。従って、アクセス制御ハードウェア(120)がアクセスの可否を判断するルールは、以下の通り、それぞれドメイン分離ファームウェア(140)が設定する。

(1) OS code 1(OSC1, 200)とOS code 2(OSC2, 201)に対するアクセス可否ルールエントリ(300)には、OS0(141)とソフトウェアモジュール0(151)、並びに

OSm(144)とソフトウェアモジュールm(154)、を実行しているCPUからの読みアクセス可。

(2)OS個別data a(OSDa, 202)に対するアクセス可否ルールエントリ(302)には、OS0(141)とソフトウェアモジュール0(151)を実行しているCPUからの読み書きアクセス可。

(3)OS個別data b(OSDb, 203)に対するアクセス可否ルールエントリ(303)には、OSm(144)とソフトウェアモジュールm(154)を実行しているCPUからの読み書きアクセス可。

(4)AP code 1(APC1, 204)とAP code 2(APC2, 205)に対するアクセス可否ルールエントリ(304)には、OS0(141)とソフトウェアモジュール0(151)、並びにOSm(144)とソフトウェアモジュールm(154)、を実行しているCPUからの読みアクセス可。

(5)AP個別data a(APDa, 206)に対するアクセス可否ルールエントリ(306)には、OS0(141)とソフトウェアモジュール0(151)を実行しているCPUからの読み書きアクセス可。

(6)AP個別data b(APDb, 207)に対するアクセス可否ルールエントリ(307)には、OSm(144)とソフトウェアモジュールm(154)を実行しているCPUからの読み書きアクセス可。

(7)OS間通信Data a(OSMa, 208)に対するアクセス可否ルールエントリ(308)には、OS0(141)とソフトウェアモジュール0(151)を実行しているCPUからの読み書きアクセス可。

(8)OS間通信data b(OSMb, 209)に対するアクセス可否ルールエントリ(309)には、OSm(144)とソフトウェアモジュールm(154)を実行しているCPUからの読み書きアクセス可。

(9)それ以外の外部メモリの領域に対するアクセス可否ルールエントリには、OS0(141)とソフトウェアモジュール0(151)、並びにOSm(144)とソフトウェアモジュールm(154)、を実行しているCPUからの全てのアクセス不可。

[0068] 更に、内蔵メモリ(160)を対象とするアクセス制御ハードウェア(121)がアクセスの

可否を判断するルールは、以下の通り、それぞれドメイン分離ファームウェア(140)が設定する。

(1) OS code 1(OSC1, 260)とOS code 2(OSC2, 261)に対するアクセス可否ルールエントリ(310)には、OS0(141)とソフトウェアモジュール0(151)、並びにOSm(144)とソフトウェアモジュールm(154)、を実行しているCPUからの読みアクセス可。

(2) OS個別data a(OSDa, 262)に対するアクセス可否ルールエントリ(311)には、OS0(141)とソフトウェアモジュール0(151)を実行しているCPUからの読み書きアクセス可。

(3) OS個別data b(OSDb, 263)に対するアクセス可否ルールエントリ(312)には、OSm(144)とソフトウェアモジュールm(154)を実行しているCPUからの読み書きアクセス可。

(4) AP code 1(APC1, 264)とAP code 2(APC2, 265)に対するアクセス可否ルールエントリ(313)には、OS0(141)とソフトウェアモジュール0(151)、並びにOSm(144)とソフトウェアモジュールm(154)、を実行しているCPUからの読みアクセス可。

(5) AP個別data a(APDa, 266)に対するアクセス可否ルールエントリ(314)には、OS0(141)とソフトウェアモジュール0(151)を実行しているCPUからの読み書きアクセス可。

(6) AP個別data b(APDb, 267)に対するアクセス可否ルールエントリ(315)には、OSm(144)とソフトウェアモジュールm(154)を実行しているCPUからの読み書きアクセス可。

(7) OS間通信Data a(OSMa, 268)に対するアクセス可否ルールエントリ(316)には、OS0(141)とソフトウェアモジュール0(151)を実行しているCPUからの読み書きアクセス可。

(8) OS間通信data b(OSMb, 269)に対するアクセス可否ルールエントリ(317)には、OSm(144)とソフトウェアモジュールm(154)を実行しているCPUからの読み書きアクセス可。

- [0069] なお、本実施例では、全ての内蔵メモリの領域を用いているため、図面上、存在していないが、もし、使用しない内蔵メモリの領域があるのであれば、それ以外の内蔵メモリの領域に対するアクセス可否ルールエントリには、OS0(141)とソフトウェアモジュール0(151)、並びにOSm(144)とソフトウェアモジュールm(154)、を実行しているCPUからの全てのアクセス不可と、設定する。
- [0070] ソフトウェアモジュール0(151)からソフトウェアモジュールm(154)へOS間通信を実施する際は、ソフトウェアモジュール0(151)が内蔵メモリ(160)上のOS間通信Data a(OSMa, 268)にデータの書き込みが完了した時点で、ドメイン連携ミドルウェア(150)に転送要求を出す。
- [0071] 転送要求を受け取ったドメイン連携ミドルウェア(150)は、ドメイン分離ファームウェア(140)を用いて、OS間通信Data a(OSMa, 268)に対するアクセス可否ルールエントリ(316)とOS間通信Data b(OSMb, 269)に対するアクセス可否ルールエントリ(317)とに対しOS0(141)とソフトウェアモジュール0(151)、並びにOSm(144)とソフトウェアモジュールm(154)、を実行しているCPUからの全てのアクセス不可、と設定する。その後、ドメイン連携ミドルウェア(150)は、ドメイン分離ファームウェア(140)を用いてOS間通信Data a(OSMa, 268)に対するアクセス可否ルールエントリ(316)とOS間通信Data b(OSMb, 269)に対するアクセス可否ルールエントリ(317)とに対し、ドメイン連携ミドルウェア(150)を実行しているCPUからの読み書きアクセス可と設定し、OS間通信Data a(OSMa, 268)に書き込まれたデータをOS間通信Data b(OSMb, 269)の指定された場所へ書き出す。その後、再びドメイン分離ファームウェア(140)を用いてOS間通信Data a(OSMa, 268)に対するアクセス可否ルールエントリ(316)とOS間通信Data b(OSMb, 269)に対するアクセス可否ルールエントリ(317)とに対し、ドメイン連携ミドルウェア(150)を実行しているCPUからの全てのアクセス不可と設定し、完了した後、ドメイン分離ファームウェア(140)を用いてOS間通信Data a(OSMa, 268)に対するアクセス可否ルールエントリ(316)には、OS0(141)とソフトウェアモジュール0(151)を実行しているCPUからの読み書きアクセス可、OS間通信data b(OSMb, 269)に対するアクセス可否ルールエントリ(317)には、OSm(144)とソフトウェアモジュールm(154)を実行

しているCPUからの読み書きアクセス可、と設定し、ソフトウェアモジュール0(151)に対し転送完了を出す。

[0072] ソフトウェアモジュールm(154)からソフトウェアモジュール0(151)へOS間通信を実施する際は、ソフトウェアモジュールm(154)が外部メモリ(130)上のOS間通信Data b(OSMa, 269)にデータの書き込みが完了した時点で、ドメイン連携ミドルウェア(150)に転送要求を出す。

[0073] 転送要求を受け取ったドメイン連携ミドルウェア(150)は、ドメイン分離ファームウェア(140)を用いてOS間通信Data a(OSMa, 268)に対するアクセス可否ルールエントリ(316)とOS間通信Data b(OSMb, 269)に対するアクセス可否ルールエントリ(317)とに対しOS0(141)とソフトウェアモジュール0(151)、並びにOSm(144)とソフトウェアモジュールm(154)、を実行しているCPUからの全てのアクセス不可、と設定する。その後、ドメイン連携ミドルウェア(150)は、OS間通信Data b(OSMb, 269)に書き込まれたデータをOS間通信Data a(OSMa, 268)の指定された場所へ書き出し、完了した後、ドメイン分離ファームウェア(140)を用いてOS間通信Data a(OSMa, 268)に対するアクセス可否ルールエントリ(316)には、OS0(141)とソフトウェアモジュール0(151)を実行しているCPUからの読み書きアクセス可、OS間通信data b(OSMb, 269)に対するアクセス可否ルールエントリ(317)には、OSm(144)とソフトウェアモジュールm(154)を実行しているCPUからの読み書きアクセス可、と設定し、ソフトウェアモジュールm(154)に対し転送完了を出す。

[0074] 以上の通り、本実施例では、実施例2と比較して内蔵メモリを用いてデータ処理を行うため高速に動作することが可能となる。

産業上の利用可能性

[0075] 本発明は、マルチプロセッサシステムに適用して特に有効である。

請求の範囲

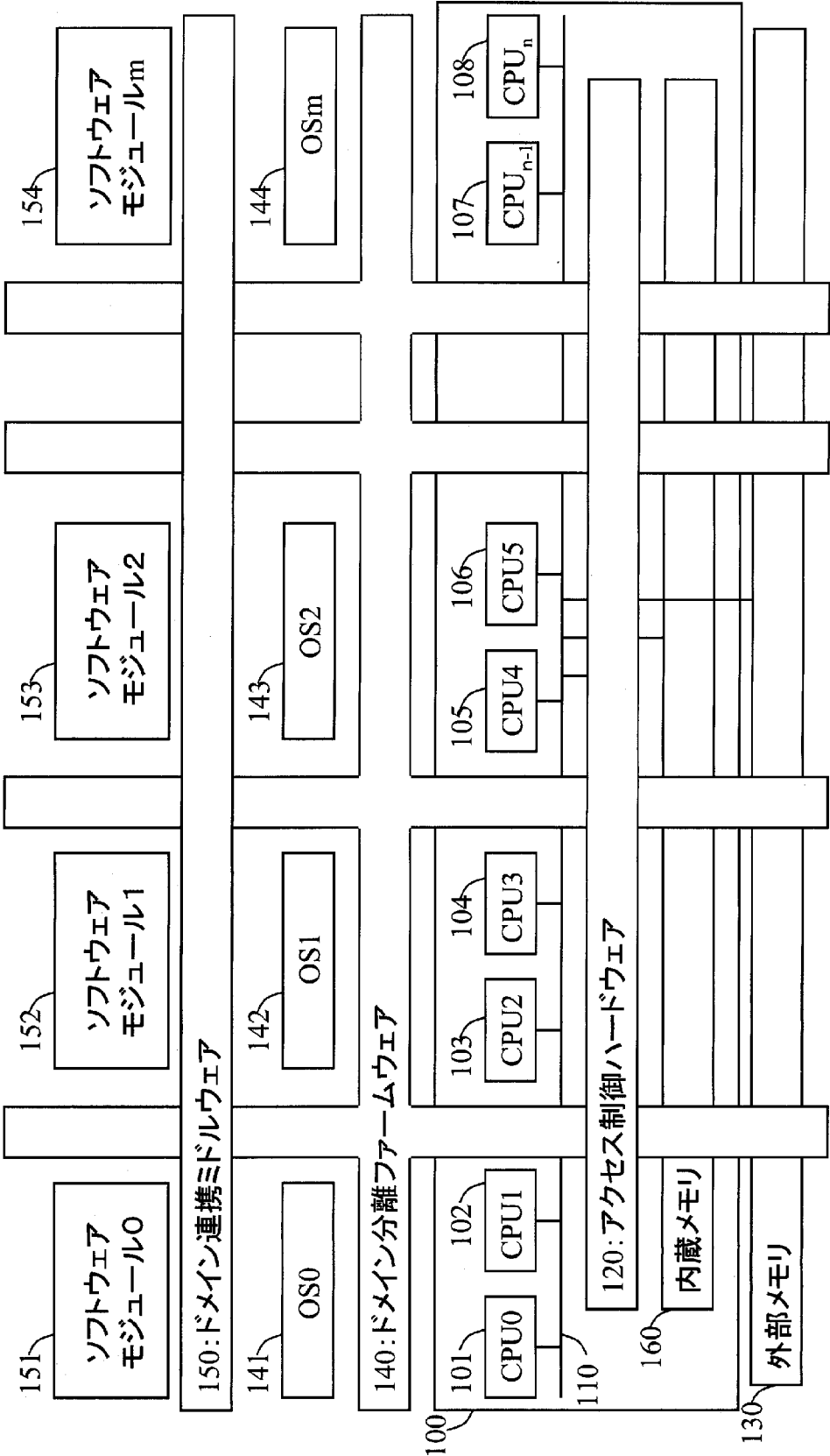
- [1] 第1中央処理装置と、第2中央処理装置と、前記第1中央処理装置にて実行される第1オペレーティングシステムと、前記第2中央処理装置にて実行される第2オペレーティングシステムと、前記第1中央処理装置及び前記第2中央処理装置にアクセスされるメモリとを有する情報処理装置であって、
- 前記情報処理装置を、前記第1中央処理装置、前記第1オペレーティングシステム、及び、前記メモリの領域のうち一部の第1領域を第1ドメインとして割り当て、前記第2中央処理装置、前記第2オペレーティングシステム及び前記メモリの領域のうち一部の第2領域を第2ドメインとして割り当て、一方のドメインが他のドメインに割当てられた領域にアクセスしないように制御するファームウェアと、
- 前記第1ドメイン及び前記第2ドメイン間でデータの通信が必要な場合に、当該通信を制御するミドルウェアとを有することを特徴とする情報処理装置。
- [2] 請求項1において、
- 前記情報処理装置は、前記ファームウェアからの設定に基づいて、一方のドメインが他方のドメインに割当てられた領域にアクセスする際に、そのアクセスを遮断するアクセス制御モジュールを更に具備することを特徴とする情報処理装置。
- [3] 請求項2において、
- 前記ファームウェアは、前記第1領域の先頭アドレス、前記第1領域のサイズ、及び、前記第1領域の割り当てが前記第1ドメインであることを対にして前記アクセス制御モジュールに設定し、前記第2領域の先頭アドレス、前記第2領域のサイズ、及び、前記第2領域の割り当てが前記第2ドメインであることを対にして前記アクセス制御モジュールに設定することを特徴とする情報処理装置。
- [4] 請求項1において、
- 前記第1オペレーティングシステムのコードと前記第2オペレーティングシステムのコードが共有できる場合において、前記メモリは、前記メモリの領域の一部の第3領域に、前記共有できるコードを保持し、前記ファームウェアは、前記第3領域に対し、前記第1ドメイン及び前記第2ドメインからリードアクセスを許可し、ライトアクセスを許可しないように制御することを特徴とする情報処理装置。

- [5] 請求項1において、
前記メモリは、前記第1ドメイン及び前記第2ドメイン間でデータの通信を行う場合に、通信用データを格納する第4領域を含み、
前記ファームウェアは、前記第1及び第2ドメインから通信の要求がない場合には、前記第4領域を前記第1ドメイン及び前記第2ドメインからアクセスできないように制御し、
前記ミドルウェアは、前記第1ドメインが前記第2ドメインにデータの通信を行う旨の要求を受けた場合に、前記第1ドメインに前記第4領域へのアクセスを許可し、前記第2ドメインに前記第4領域へのアクセスを許可しないように前記ファームウェアを制御した状態で、前記第1ドメインに前記第4領域へのアクセスを許可することを特徴とする情報処理装置。
- [6] 請求項1において、
前記メモリは、前記第1ドメインが前記第2ドメインにデータの通信を行う場合に、通信用のデータを書き込む第5領域と、前記第2ドメインが前記第1ドメインにデータの通信を行う場合に、通信用のデータを書き込む第6領域とを有し、
前記ファームウェアは、前記第5領域に対して前記第1ドメインがアクセス可能で、前記第2ドメインはアクセス不可能になるように制御し、前記第6領域に対して前記第2ドメインがアクセス可能で、前記第1ドメインがアクセス不可能となるように制御し、
前記ミドルウェアは、前記第1ドメインが前記第2ドメインにデータの通信を行う旨の要求を受けた場合に、前記第1ドメインが前記第5領域に通信用のデータを書き込んだ後、前記第5領域及び前記第6領域が前記第1及び前記第2ドメインからアクセス不可能となるように前記ファームウェアに指示し、その後、前記第5領域から前記第6領域へのデータを転送することを特徴とする情報処理装置。
- [7] 請求項1において、
前記第1ドメインは、仮想メモリ空間から前記メモリの領域を含む物理メモリ空間へのアドレス変換を行う第1メモリマネージメントユニットを有し、
前記第2ドメインは、仮想メモリ空間から前記メモリの領域を含む物理メモリ空間へのアドレス変換を行う第2メモリマネージメントユニットを有し、

前記ファームウェアは、前記物理メモリ空間に対して、アクセス制御を行うことを特徴とする情報処理装置。

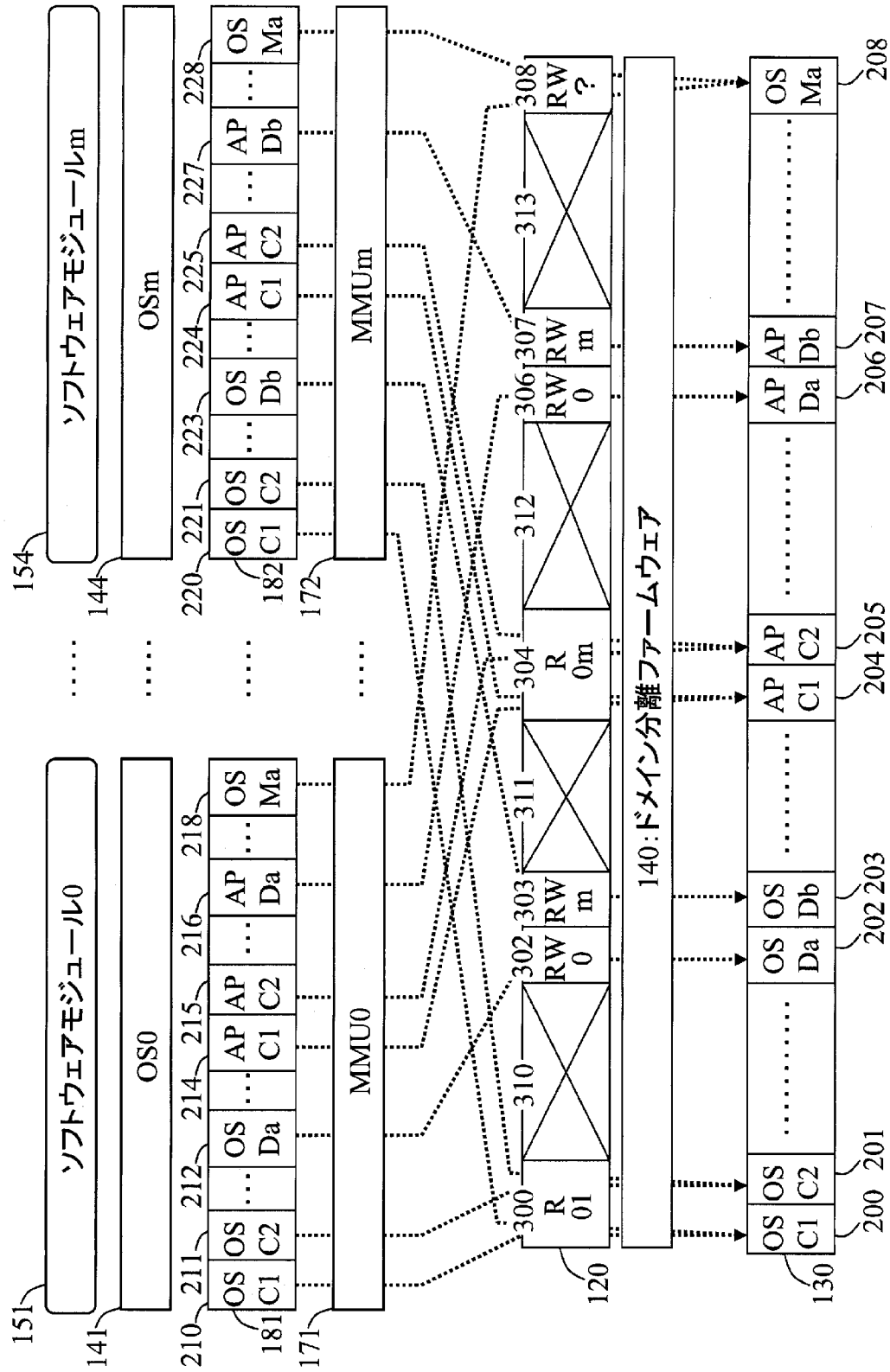
[図1]

図 1



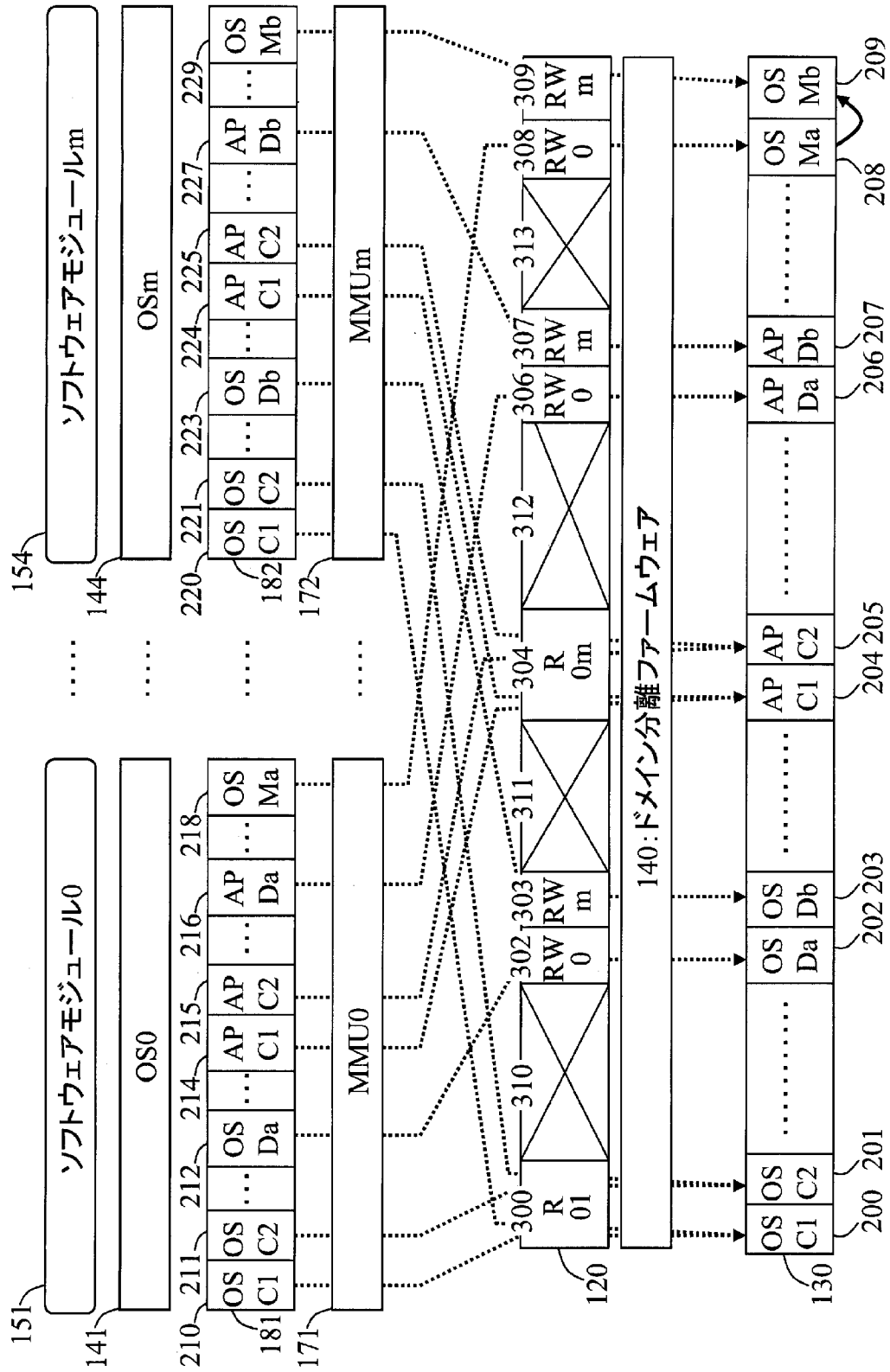
[図2]

図 2



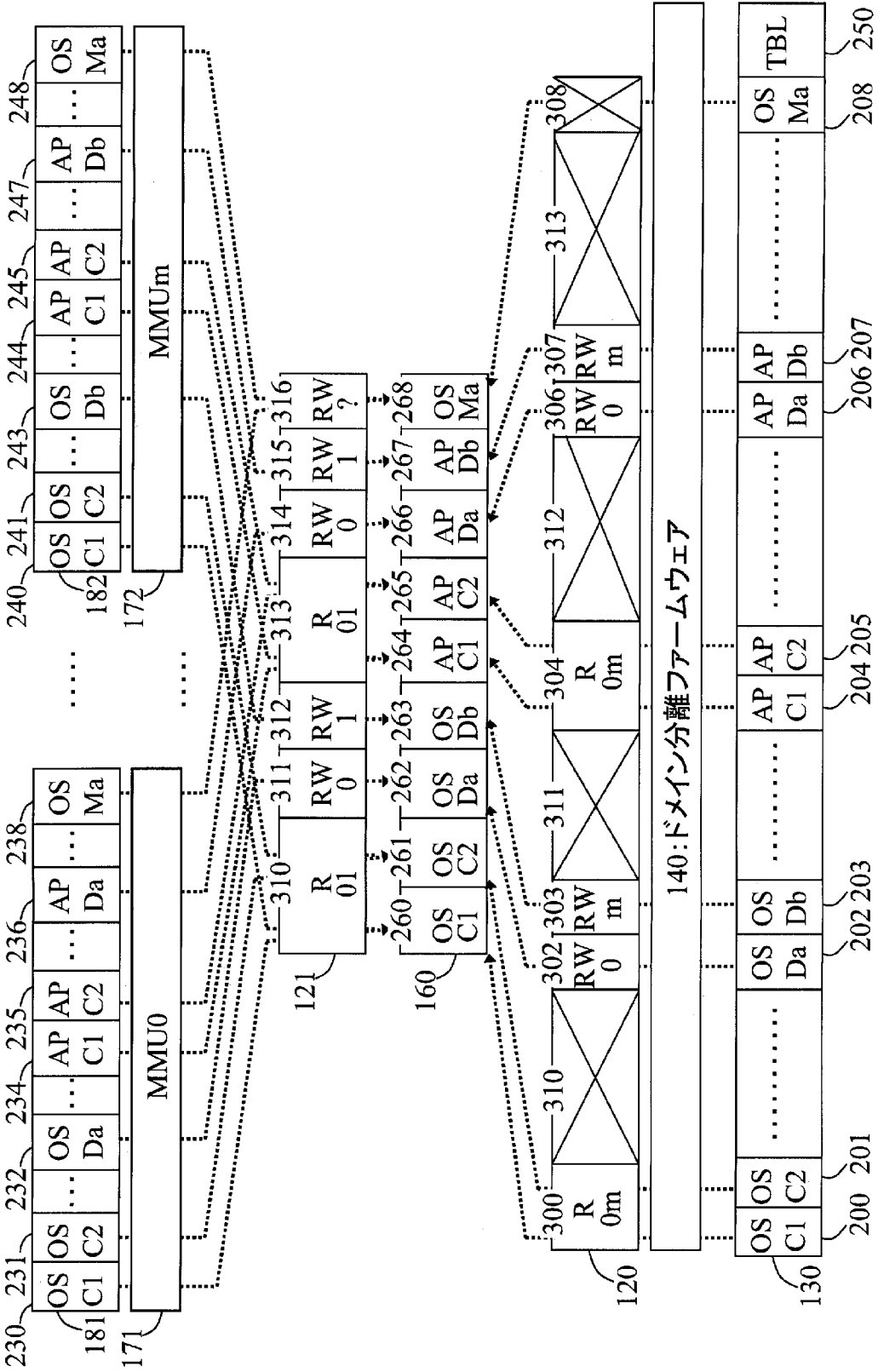
[図3]

図 3



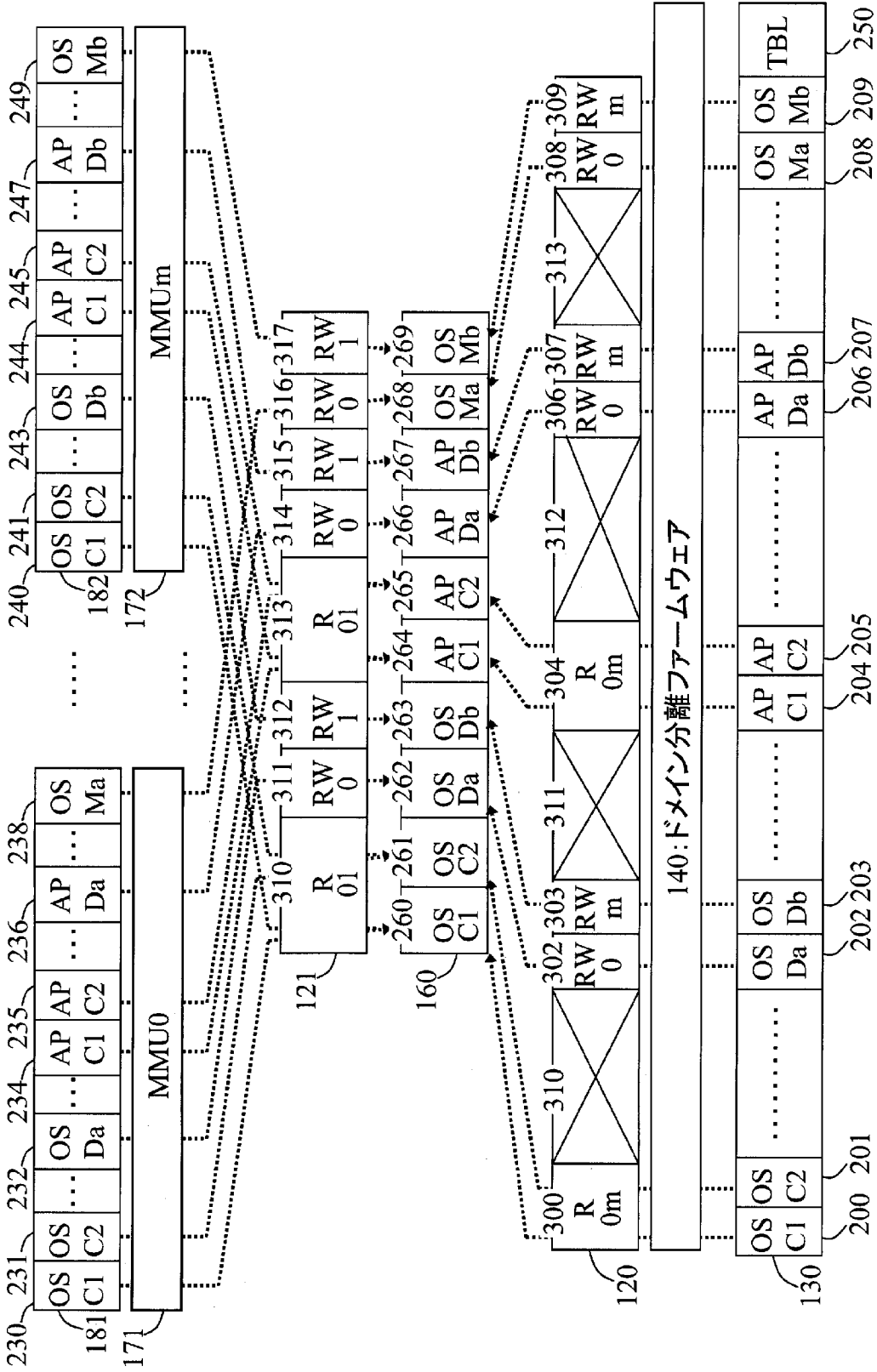
[図4]

図 4



[図5]

図 5



INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP2009/053546

A. CLASSIFICATION OF SUBJECT MATTER

G06F9/46(2006.01) i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06F9/46-9/54

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 1996-2009

Kokai Jitsuyo Shinan Koho 1971-2009 Toroku Jitsuyo Shinan Koho 1994-2009

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

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X Y	JP 2001-331333 A (Hitachi, Ltd.), 30 November, 2001 (30.11.01), Par. Nos. [0012] to [0017], [0068] to [0079]; Figs. 1, 2, 14	1, 2, 5, 6 3, 4, 7
Y	JP 2004-258698 A (Hitachi, Ltd.), 16 September, 2004 (16.09.04), Par. Nos. [0012] to [0019]; Figs. 1 to 3	3, 4, 7
A	JP 2005-513644 A (Nokia Corp.), 12 May, 2005 (12.05.05), Par. Nos. [0017] to [0024]; Figs. 2 to 4	1-7
A	JP 2008-186173 A (Toyota Motor Corp.), 14 August, 2008 (14.08.08), Par. Nos. [0011] to [0020]; Fig. 1	1-7

☒ 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
01 June, 2009 (01.06.09)Date of mailing of the international search report
16 June, 2009 (16.06.09)Name and mailing address of the ISA/
Japanese Patent Office

Authorized officer

Facsimile No.

Telephone No.

INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP2009/053546

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	JP 2002-157133 A (Hitachi, Ltd.), 31 May, 2002 (31.05.02), Full text; all drawings	3, 4, 7

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/JP2009/053546

JP 2001-331333 A	2001.11.30	US 2001/0044817 A1 US 6931640 B
JP 2004-258698 A	2004.09.16	(Family: none)
JP 2005-513644 A	2005.05.12	BR 215054 A CN 1613059 A EP 1456755 A1 FI 20012527 A FI 114663 B1 KR 10-2004-0068600 A US 2003/0120706 A1 WO 03/054698 A1
JP 2008-186173 A	2008.08.14	(Family: none)
JP 2002-157133 A	2002.05.31	JP 3874603 B

A. 発明の属する分野の分類（国際特許分類（I P C））

Int.Cl. G06F9/46(2006.01) i

B. 調査を行った分野

調査を行った最小限資料（国際特許分類（I P C））

Int.Cl. 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 0 9 年
日本国実用新案登録公報	1 9 9 6 - 2 0 0 9 年
日本国登録実用新案公報	1 9 9 4 - 2 0 0 9 年

国際調査で使用した電子データベース（データベースの名称、調査に使用した用語）

C. 関連すると認められる文献

引用文献の カテゴリー*	引用文献名 及び一部の箇所が関連するときは、その関連する箇所の表示	関連する 請求項の番号
X Y	JP 2001-331333 A（株式会社日立製作所）2001.11.30, 段落[0012] - [0017], [0068] - [0079], 第1, 2, 14 図	1, 2, 5, 6 3, 4, 7
Y	JP 2004-258698 A（株式会社日立製作所）2004.09.16, 段落[0012] - [0019], 第1-3 図	3, 4, 7
A	JP 2005-513644 A（ノキア コーポレイション）2005.05.12, 段落[0017] - [0024], 第2-4 図	1-7

☒ 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 1 . 0 6 . 2 0 0 9

国際調査報告の発送日

1 6 . 0 6 . 2 0 0 9

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）

大塚 良平

5 B

3 8 5 6

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 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	JP 2008-186173 A (トヨタ自動車株式会社) 2008.08.14, 段落[0011]－[0020], 第1図	1－7
A	JP 2002-157133 A (株式会社日立製作所) 2002.05.31, 全文, 全図	3, 4, 7

JP 2001-331333 A	2001. 11. 30	US 2001/0044817 A1 US 6931640 B
JP 2004-258698 A	2004. 09. 16	ファミリーなし
JP 2005-513644 A	2005. 05. 12	BR 215054 A CN 1613059 A EP 1456755 A1 FI 20012527 A FI 114663 B1 KR 10-2004-0068600 A US 2003/0120706 A1 WO 03/054698 A1
JP 2008-186173 A	2008. 08. 14	ファミリーなし
JP 2002-157133 A	2002. 05. 31	JP 3874603 B