(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号

特許第5448496号
(P5448496)

(45)発行日　平成26年3月19日(2014.3.19)　(24)登録日　平成26年1月10日(2014.1.10)

(51)Int.Cl.		F I		
G06F 12/10	(2006.01)	G06F	12/10	557
H04N 1/00	(2006.01)	H04N	1/00	C
G06F 3/12	(2006.01)	G06F	3/12	B
		G06F	12/10	559
		G06F	12/10	509Z

請求項の数 4　(全 17 頁)　最終頁に続く

(21)出願番号　特願2009-37056(P2009-37056)
(22)出願日　平成21年2月19日(2009.2.19)
(65)公開番号　特開2010-191818(P2010-191818A)
(43)公開日　平成22年9月2日(2010.9.2)
審査請求日　平成24年2月20日(2012.2.20)

(73)特許権者　000001007
キヤノン株式会社
東京都大田区下丸子３丁目３０番２号
(74)代理人　100076428
弁理士　大塚　康徳
(74)代理人　100112508
弁理士　高柳　司郎
(74)代理人　100115071
弁理士　大塚　康弘
(74)代理人　100116894
弁理士　木村　秀二
(74)代理人　100130409
弁理士　下山　治
(74)代理人　100134175
弁理士　永川　行光

最終頁に続く

(54)【発明の名称】情報処理装置及びその制御方法

(57)【特許請求の範囲】
【請求項１】
第１ボードと、第２ボードとを有する情報処理装置であって、
前記第１ボードは、
第１制御手段と、
前記第１制御手段のワークメモリとして使用される第１記憶手段とを有し、
前記第２ボードは、
第２制御手段と、
不揮発性の第２記憶手段とを有し、
前記第１制御手段は、オペレーティングシステムが管理する仮想メモリ空間を介して前記第２記憶手段に記憶されたデータにアクセスし、
前記第１制御手段は、前記仮想メモリ空間のアドレスと、前記第１記憶手段の物理アドレスと、前記第２記憶手段の物理アドレスとの対応関係を示す情報を作成して当該情報を前記第２制御手段へ転送し、
前記第２制御手段は、前記情報に基づいて、前記第２記憶手段に記憶されたデータを前記第１記憶手段へ転送することを特徴とする情報処理装置。
【請求項２】
前記第１制御手段は、前記オペレーティングシステムの起動が完了し、前記第２制御手段による前記第１記憶手段へのデータの転送が完了した場合に、アプリケーションの実行を開始することを特徴とする請求項１に記載の情報処理装置。

【請求項３】
前記第２制御手段は、前記第１記憶手段に記憶されるキャッシュデータと前記第２記憶手段に記憶されるデータを同期させる同期イベントの発行に伴って前記第１記憶手段へのデータの転送を実行することを特徴とする請求項１に記載の情報処理装置。
【請求項４】
第１制御手段と、前記第１制御手段のワークメモリとして使用される第１記憶手段とを有する第１ボードと、第２制御手段と、不揮発性の第２記憶手段とを有する第２ボードとを有する情報処理装置を制御する制御方法であって、
前記第１制御手段が、オペレーティングシステムが管理する仮想メモリ空間を介して前記第２記憶手段に記憶されたデータにアクセスするアクセス工程と、
前記第１制御手段が、前記仮想メモリ空間のアドレスと、前記第１記憶手段の物理アドレスと、前記第２記憶手段の物理アドレスとの対応関係を示す情報を作成して当該情報を前記第２制御手段へ転送する第１転送工程と、
前記第２制御手段が、前記情報に基づいて、前記第２記憶手段に記憶されたデータを前記第１記憶手段へ転送する第２転送工程とを有することを特徴とする情報処理装置の制御方法。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、複数のＣＰＵを搭載した情報処理装置及びその制御方法に関するものである。
【背景技術】
【０００２】
情報処理装置に用いられる記憶装置としてフラッシュメモリ等に代表される不揮発メモリがある。不揮発メモリは、電源供給が遮断されても記憶しているデータを保持できるため、組み込み機器等において不意な電源断等に対処するように使用される。
【０００３】
特許文献１には、主メモリ上へキャッシュを応用した発明が記載されている。また特許文献２には、アクセス速度が高速なバッファと低速な不揮発メモリと制御コントローラとを有するユニットで、アクセス速度が低速な不揮発メモリに記憶されたデータをアクセス速度が高速なバッファにキャッシュすることで、高速に不揮発メモリに記憶されたデータを読み出す技術が開示されている。
【先行技術文献】
【特許文献】
【０００４】
【特許文献１】特開２００１－１４７８５５号公報
【特許文献２】特開平７－２４４６１４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、第１制御手段と第２制御手段とを有し、第１制御手段が仮想メモリ領域を管理するオペレーティングシステムで制御されている場合、第２制御手段は仮想メモリ領域に直接アクセスすることができない。このため、第１制御手段により実行されるオペレーティングシステムが、仮想メモリ領域を使用する場合には、第２制御手段が第１制御手段が定義したキャッシュ領域へデータを転送することができない事態が発生する。
【０００６】
本発明は、上述した従来技術の問題点を解決するためになされたものである。
【０００７】
本発明の目的は、仮想メモリ領域を管理するオペレーティングシステムを実行する第１制御手段と第１制御手段と通信可能な第２制御手段を有する情報処理装置において、オペ

レーティングシステムにより管理される仮想メモリ領域の管理外の記憶手段に記憶されたデータ対して、第１制御手段が高速にアクセスできる技術を提供することにある。

【課題を解決するための手段】

【０００８】

上記目的を達成するために本発明の一態様に係る情報処理装置は以下のような構成を備える。即ち、

第１ボードと、第２ボードとを有する情報処理装置であって、

前記第１ボードは、

第１制御手段と、

前記第１制御手段のワークメモリとして使用される第１記憶手段とを有し、

前記第２ボードは、

第２制御手段と、

不揮発性の第２記憶手段とを有し、

前記第１制御手段は、オペレーティングシステムが管理する仮想メモリ空間を介して前記第２記憶手段に記憶されたデータにアクセスし、

前記第１制御手段は、前記仮想メモリ空間のアドレスと、前記第１記憶手段の物理アドレスと、前記第２記憶手段の物理アドレスとの対応関係を示す情報を作成して当該情報を前記第２制御手段へ転送し、

前記第２制御手段は、前記情報に基づいて、前記第２記憶手段に記憶されたデータを前記第１記憶手段へ転送することを特徴とする。

【発明の効果】

【０００９】

本発明によれば、第１制御手段と、前記第１制御手段のワークメモリとして使用される第１記憶手段とを有する第１ボードと、第２制御手段と、不揮発性の第２記憶手段とを有する第２ボードとを有する情報処理装置において、オペレーティングシステムにより管理される仮想メモリ領域の管理外の第２記憶手段に記憶されたデータ対して、第１制御手段が高速にアクセスできる技術を提供することができる。

【図面の簡単な説明】

【００１０】

【図１】本発明の実施形態に係るコントローラが搭載された画像入出力装置（情報処理装置）の機能構成を示すブロック図である。

【図２】本実施形態に係る画像入出力装置のコントローラ３のハードウェア構成を説明するブロック図である。

【図３】本実施形態に係る画像入出力装置の２つのＣＰＵのブートシーケンスを説明するフローチャートである。

【図４】本実施形態に係る画像入出力装置の記憶部に格納されているデータのイメージを示す図である。

【図５】本実施形態に係る画像入出力装置のメインＣＰＵのメモリアクセス速度を説明するための図である。

【図６】本実施形態に係るメインＣＰＵが不揮発メモリをハンドリングする方法を説明する図である。

【図７】本実施形態に係るバスコントローラが管理するメモリ空間を説明する図である。

【図８】本実施形態に係る物理メモリのブロック仕様の一例を示す図である。

【図９】本実施形態に係る画像入出力装置の２つのＣＰＵによるブート処理を説明するフローチャートである。

【図１０】本実施形態に係る画像入出力装置の２つのＣＰＵによる不揮発メモリの書き換え処理を説明するフローチャートである。

【図１１】本実施形態に係る２つのＣＰＵの動作を説明するタイミング図である。

【発明を実施するための形態】

【００１１】

以下、添付図面を参照して本発明の実施形態を詳しく説明する。尚、以下の実施形態は特許請求の範囲に係る本発明を限定するものでなく、また本実施形態で説明されている特徴の組み合わせの全てが本発明の解決手段に必須のものとは限らない。

【００１２】

図１は、本発明の実施形態に係るコントローラが搭載された画像入出力装置（情報処理装置）１００の機能構成を示すブロック図である。尚、この実施例では、この画像入出力装置は、コピー機能、ファクシミリ機能、プリンタ機能等を有する複合機の場合で説明するが、本発明はこれに限定されるものではない。

【００１３】

この画像入出力装置１００は、イーサネット（登録商標）等のＬＡＮ８を介してホストコンピュータ９に接続されている。この画像入出力装置１００は、画像データの読取処理を行うリーダ装置（リーダ部）２と、画像データの印刷処理を行うプリンタ装置（プリンタ部）４、操作部５、画像記憶部（ハードディスク）６及びこれらを制御するコントローラ（制御装置）３とを有している。操作部５は、画像データの入出力操作を行うキーボード及び画像データや各種機能の表示／設定などを行う液晶パネルとを備える。記憶部（ＨＤ）６は、リーダ部２を制御して読み込んだ画像データや、ＬＡＮ８を介してホストコンピュータ９より受信したコードデータから生成される画像データを格納する。コントローラ３は、これら各構成要素に接続され、これら構成要素を制御する。

【００１４】

リーダ部２は、原稿を搬送する原稿給紙ユニット２１と、その原稿を光学的に読み取って電気信号としての画像データに変換するスキャナユニット２２とを有している。ＦＡＸユニット７は、電話回線１０を介してファクシミリデータを送受信する。またプリンタ部４は、シートを収容する複数段の給紙カセットを備えた給紙ユニット４２と、画像データをシートに転写して定着するマーキングユニット４１を有している。更に、印刷されたシートにソート処理やステイプル処理を施して外部に排出する排紙ユニット４３を有している。

【００１５】

コントローラ３は、リーダ部２を制御して原稿の画像データを読込み、プリンタ部４を制御して、その画像データをシートに印刷するコピー機能を提供する。また、リーダ部２で読取った画像データをコードデータに変換し、ネットワーク８を介してホストコンピュータ９へ送信するスキャナ機能を有している。またホストコンピュータ９からネットワーク８を介して受信したコードデータを画像データに変換し、プリンタ部４に出力して印刷するプリンタ機能やその他の機能ブロックを有している。

【００１６】

以上の構成を有する画像入出力装置１００は、大きく分けて複写機能、画像送信機能、画像保存機能及び画像印刷機能を有している。複写機能は、リーダ部２から入力した画像データを記憶部６に記憶するとともに、プリンタ部４により印刷するものである。画像送信機能は、リーダ部２から入力した画像データをＬＡＮ８を介してコンピュータ９に送信する機能である。画像保存機能は、リーダ部２から入力した画像データを記憶部６に記憶し、必要に応じて画像データの送信や印刷を行なう機能である。また画像印刷機能は、コンピュータ９から送信された例えばページ記述言語を解析し、プリンタ部４で印刷する機能である。

【００１７】

図２は、本実施形態に係る画像入出力装置１００のコントローラ３のハードウェア構成を説明するブロック図である。尚、図１と共通する部分は同じ記号で示している。

【００１８】

このコントローラ３は、メインボード２００と、サブボード２２０とを具備している。メインボード２００は所謂汎用のＣＰＵシステムで、ボード２００全体を制御するメインＣＰＵ２０１（第１制御手段）、ブートプログラムが含まれるブートロム２０２、ＣＰＵ２０１がワークメモリとして使用するメモリ２０３（第１記憶手段）を有している。また

、外部バスとのブリッジ機能を持つバスコントローラ２０４、不揮発メモリ２０５（バッテリ等によりバックアップされたメモリであっても良い）を含む。更に、記憶部６を制御するディスクコントローラ２０６と、半導体デバイスで構成された比較的小容量なストレージ装置であるフラッシュディスク（ＳＳＤ等）２０７、ＵＳＢメモリ２０９を制御するＵＳＢコントローラ２０８等を含む。このメインボード２００は、ＵＳＢメモリ２０９、操作部５、記憶部６を接続している。

【００１９】

サブボード２２０は、比較的小さな汎用ＣＰＵシステムと、画像処理用のハードウエアで構成される。サブボード２２０はＣＰＵ２２１（第２制御手段）、ＣＰＵ２２１がワークメモリとして使用するメモリ２２３、外部バスとのブリッジ機能を持つバスコントローラ２２４、不揮発メモリ２２５（第２記憶手段）を有している。更に、リアルタイムでデジタル画像処理を行なう画像処理部２２７とデバイスコントローラ２２６を有する。前述したリーダ部２とプリンタ部４は、デバイスコントローラ２２６を介して画像処理部２２７との間でデジタル画像データの受け渡しを行なう。ＦＡＸユニット７は、ＣＰＵ２２１が直接制御する。

【００２０】

尚、図２は簡略化して示している。例えばＣＰＵ２０１、ＣＰＵ２２１等にはチップセット、バスブリッジ、クロックジェネレータ等のＣＰＵ周辺ハードウェアが多数含まれているが、簡略化のためにこれらを省略して記述している。

【００２１】

次に複写機能を例にして、コントローラ３の動作を説明する。

【００２２】

ユーザが操作部５から複写を指示すると、ＣＰＵ２０１がＣＰＵ２２１を介してリーダ部２に画像読み取り命令を送る。これによりリーダ部２は、原稿を光学的にスキャンしてデジタル画像データに変換してデバイスコントローラ２２６を介して画像処理部２２７に入力する。画像処理部２２７はＣＰＵ２２１を介してメモリ２２３にＤＭＡにより画像データを保存する。

【００２３】

ＣＰＵ２０１はデジタル画像データがメモリ２２３に一定量もしくは全て記憶されたことが確認すると、ＣＰＵ２２１を介してプリンタ部４に印刷指示を出す。ＣＰＵ２２１は画像処理部２２７にメモリ２２３の画像データの位置を教える。そしてプリンタ部４からの同期信号に従ってメモリ２２３上の画像データは、画像処理部２２７とデバイスコントローラ２２６を介してプリンタ部４に送信され、プリンタ部４によりシート上に印刷される。

【００２４】

複数部の印刷を行なう場合、ＣＰＵ２０１がメモリ２２３の画像データを記憶部６に対して保存する。これにより２部目以降は、その記憶部６から画像データを読み出してプリンタ部４に出力して印刷する。

【００２５】

図３は、本実施形態に係る画像入出力装置１００の２つのＣＰＵのブートシーケンスを説明するフローチャートである。本実施形態に係る画像入出力装置１００はマルチコアシステムを採用しているため、ブートシーケンスは各ＣＰＵの制御プログラムで実行される。このフローチャートは、ＣＰＵ２０１が図４のＣＰＵ２０１制御プログラム４０６を実行するまで、またＣＰＵ２２１が図４のＣＰＵ２２１制御プログラム４０７を実行するまでの処理を示している。一般的には、このような複数のボード構成におけるブートシーケンスは、各ＣＰＵがブートのためのＲＯＭを持っている。しかしそのように構成すると、制御プログラムは複数のＣＰＵ間で同一バージョンのものを利用する必要があり、またストレージ装置を２つ持つことによりコストアップの原因となる。そのため本実施形態では、マスタ側（メインボード２００）が最初に起動し、次にメインボード２００がスレーブ側（サブボード２２０）を起動するというシーケンスを採用している。

【００２６】
この処理は、メインボード２００に電源が供給されることにより開始される。電源オンでリセット回路が動作しＣＰＵの周辺ＩＣの初期化を行う。この初期化が完了するとＣＰＵ２０１のリセットが解除されてＣＰＵ２０１が起動する。これによりステップＳ３０１で、ＣＰＵ２０１のブートシーケンスを開始する。ここではＣＰＵ２０１は一般的なＸ８６アーキテクチャによるものと仮定して説明し、ここでのブートデバイスは、ＨＤＤである記憶部６とする。この時の記憶部６データを図４に示す。
【００２７】
図４は、本実施形態に係る画像入出力装置１００の記憶部６に格納されているデータのイメージを示す図である。以下の説明では、この図４のプログラムが参照されている。
【００２８】
次にステップＳ３０２に進み、ＣＰＵ２０１はリセット例外における例外実行が発生しブートロム２０２（例えばＢＩＯＳ）のプログラムコードを実行する。このブートプログラムは、ＣＰＵ２０１のチップセット等の周辺デバイスを逐次初期化し、起動デバイスである記憶部６のＭＢＲ４０１をディスクコントローラ２０６を介して読み込んで実行する。このＭＢＲ(Master Boot record)４０１（図４）に記述可能なプログラムは非常に短いものであり、カーネルローダ１（KernelLoader1）４０２を記憶部６から読んで実行に移すだけである。次にステップＳ３０４に進み、KernelLoader1（４０２）に制御が移る。本実施形態に係るＸ８６はレガシー互換モードを持っており、メモリアクセス制限やＣＰＵの命令実行制限等がある。従って、ここでは最低限のデバイス周りの初期化を行い、レガシー互換で動作しているＣＰＵ２０１のモードを高性能の出る最新のモードへと切り替える。そしてステップＳ３０５で、カーネルローダ２（KernelLoader2）４０３を記憶部６から読み込みジャンプする。次にステップＳ３０６に進み、KernelLoader2（４０３）は、カーネル(kernel)４０４を記憶部６から読み込んで、そのKernelの実行ルーチンにジャンプする。
【００２９】
ここまでの処理は、カーネルが起動する前の状態である。記憶部６等の現在主流とされているストレージ装置は、ＬＢＡ(Logical Block Addressing)と呼ばれる論理アドレスを指定して、セクタ単位でデータの入出力を行なう。しかしプログラマは、膨大な論理アドレスを指定して管理や制御をできないため、カーネル上でファイルという概念を作り出し、そのファイルに対する操作からＬＢＡを求めて所定のデータにアクセスする、という手法を取る。
【００３０】
一般的にストレージ装置に対するアクセスは、全てカーネルのファイルシステムを介して行なわれているといっても良い。また、カーネル上で動作するアプリケーションはファイルシステムで管理されたファイルシステム領域４０５（図４）にしかアクセスできないと考えてよい。
【００３１】
また、ディスクコントローラ２０６では、例えばＡＴＡＰＩやＳＡＴＡ等の規定の標準インターフェースが定義されており、フラッシュディスク２０７や記憶部６はＣＰＵ２０１が意識することなくアクセスできる。また非互換のインターフェースの場合でも、カーネルが起動してしまえば共通のインターフェースとなるようにソフトウエアで対応することが容易に可能となる。こうしてステップＳ３０６でカーネルが起動すると、ＣＰＵ２０１はファイルシステムを使用できるようになる。そしてカーネルの起動時に各デバイスに対するデバイスドライバが組み込まれる。これによりＣＰＵ２０１に接続されているほぼ全てのデバイスはアクセス可能な状態となる。
【００３２】
次にステップＳ３０７に進み、バスコントローラ２０４を介してＣＰＵ２２１内のメモリコントローラを設定する。これによりＣＰＵ２０１は、メモリ２２３に対してアクセス可能な状態を作りだす。次にＣＰＵ２２１の制御プログラム４０７（図４）をファイルシ

ステムを介して記憶部６からメモリ２２３にロードする。ここでＣＰＵ２２１の制御プログラム４０７は、ＣＰＵ２２１のリセット例外からのプログラムを含むようにしておく。そしてＣＰＵ２０１がＣＰＵ２２１のリセット解除を行なうと、ＣＰＵ２２１にリセット例外が発生し、リセットベクタからプログラムを実行する。ＣＰＵ２２１の起動の説明は後述する。

【００３３】

次にステップＳ３０８に進み、ＣＰＵ２０１は、自分の制御プログラムであるＣＰＵ２０１の制御プログラム４０６を記憶部６からメモリ２０３にロードして、そこにジャンプする。次にステップＳ３０９に進み、その制御プログラムに実行が移ってアプリケーションの初期化を行ない、ステップＳ３１０でアイドル状態となる。

【００３４】

次にリセットが解除されてステップＳ３２１で開始するＣＰＵ２２１の処理を説明する。このような２つの独立したＣＰＵを具備する装置では、一般的には、小規模なリアルタイムシステムを想定している。Ｘ８６が過去互換を保つために持つ特別なブートシーケンスと異なり、通常の組み込み用途のＣＰＵは非常にシンプルな起動シーケンスを持つ。

【００３５】

まずステップＳ３２２で、ＣＰＵ２２１はリセットベクタのプログラムを実行する。リセットベクタでは、最低限のハードウェアの初期化を行いカーネルのアドレスへジャンプする。こうしてカーネルの起動が完了すると、ステップＳ３２３に進んで制御プログラム４０７の初期化を行う。そしてステップＳ３２４でアイドル状態となる。

【００３６】

以上説明したように、２つのＣＰＵがそれぞれ異なる制御プログラムを実行することが可能となってコントローラ３が起動される。

【００３７】

図５は、本実施形態に係る画像入出力装置１００のメインＣＰＵ２０１のメモリアクセス速度を説明するための図である。

【００３８】

年々ＣＰＵの性能が向上し、ＣＰＵコア内のクロックは主メモリバスの１０倍以上の周波数であるＣＰＵも存在する。これは主メモリにアクセスせずにＣＰＵ内のキャッシュだけで動作すれば約１０倍の速度で動作できることを示す。そのため各ＣＰＵはキャッシュを多く搭載し、ＣＰＵの処理能力の向上を図っている。

【００３９】

また、プログラムはシーケンシャルに実行する必要があるため、ＣＰＵキャッシュと比べて遅い主メモリのデータアクセスに際して、ＣＰＵ内部の実行パイプラインにより、遅い主メモリへのアクセス中であっても近傍の他命令を並行して実行している。これにより、よりＣＰＵの処理能力を向上させている。このように主メモリの遅さをカバーするように最適化された設計をチップセットとＣＰＵで実現している。

【００４０】

一方で、例えばＰＣＩバスに代表される汎用バスが存在し、バス拡張することで、ＣＰＵが様々なデバイスにアクセスできるようになっている。しかしながら、これらバスは主メモリの速度とは比較とならないほど遅い。このように主メモリに最適化することでＣＰＵの総合的な処理能力を向上させているにも拘らず、拡張されたバスに対する遅いデバイスにＣＰＵがアクセスすることで、ＣＰＵのパイプラインでは吸収できない現象が発生する。この場合ＣＰＵはストールし、何も行なえない状態となる。組み込み等の機器では、拡張バスに対して独自のデバイスを拡張する場合がある。その場合、その拡張バスに対して高頻度でＣＰＵがアクセスしている間、ＣＰＵの能力の１％も使用できないような状況となる場合もある。

【００４１】

次に図２のブロック図の一部と図５とを用いて説明する。

【００４２】

図５において、ＣＰＵ２０１は、内部にＣＰＵコア５００４、一次キャッシュ５００２、二次キャッシュ５００３、バスブリッジ５００１を有している。ＣＰＵ２０１は、システムバス５０１４（第１バス）を介してメモリ２０３と接続されている。ＣＰＵコア５００４がバスアクセスによるデータの読み込みが必要となったとき、おおよそ５０１０～５０１３で示すようなアクセスパターンが考えられる。尚、図５の下部に各々の読み込み時間の概略をグラフで記載する。

【００４３】

５０１０は最も高速なアクセスを示し、一次キャッシュ５００２にデータが存在した場合を示す。５０１１は、その次に高速なアクセスを示し、二次キャッシュ５００３にデータが存在した場合を示す。ここ数年、半導体プロセスの高密度化に伴って二次キャッシュの容量は、４ＫＢ～６ＭＢまでの変化を遂げてきている。５０１２はメインメモリであるメモリ２０３へのバスアクセスを示し、ＣＰＵ内のキャッシュにヒットしなかった場合のアクセスである。この場合は、ＣＰＵコア５００４のクロックでは動作できす、メモリバスのクロックに応じて処理速度が低下することになる。厳密には、メモリ２０３から一次キャッシュ５００２、二次キャッシュ５００３へバーストリードした後にＣＰＵ２０１がそれを参照する。

【００４４】

本実施形態が対象としているのは、５０１３で示す外部バス経由の遅いアクセスである。本実施形態では、ＣＰＵ２０１が、ＰＣＩバスを介してＣＰＵ２２１のバスブリッジ（本実施形態ではＣＰＵ２２１内に実装）経由で、例えば不揮発メモリ２２５等の遅い記録デバイスにアクセスする場合の対処方法を説明する。５０１３で示す遅いアクセスは、バスブリッジ５００１、ＰＣＩバス（２０４/２２４）を介してＣＰＵ２２１が不揮発メモリ２２５からデータを読み込み、同一の経路を通ってＣＰＵ２０１へ届けられるパスを示す。なお、ＣＰＵ２２１は、システムバス５０１５（第２バス）を介して不揮発メモリ２２５と接続されている。

【００４５】

５０１２で示すデータのリードが連続した場合、ＣＰＵ２０１のクロックとパイプラインの構成にもよるが、ＣＰＵコア５００４では、メモリリード待ちの多少のストールが発生する。これに対して、５０１３で示す外部バスアクセスは、５０１０～５０１３で示すようなコア内及びローカルバスのアクセスとは比較とならない程の読み込み時間が必要となる。よって、ＣＰＵ２０１のパイプラインがどんなに大きくなったとしても、同一アドレスに対するメモリリードとメモリライトの命令が連続した場合、データの読み込みが完了しない限り次の書き込み処理が行えない。このためパイプラインはロックし、ほぼ、図５の５０１４で示すようなアクセス時間と等価なＣＰＵ２０１のストールが発生すると考えられる。

【００４６】

従って、高性能なＣＰＵを採用しても、このような遅いバスに接続された、デバイスに対するバスアクセスによるＣＰＵアクセスは、マルチタスク、マルチプロセス環境においてＣＰＵの平均処理能力を大きく低下させる結果となる。

【００４７】

以下にこの問題を解決する手法を説明する。

【００４８】

本実施形態では、複合機などの画像入出力装置を例に挙げて、遅いデバイスとして不揮発メモリを利用した場合で説明する。

【００４９】

図２において、ＣＰＵ２０１が高性能なＣＰＵであり、例えばＬｉｎｕｘ等の仮想メモリをサポートするカーネルを動作させるシステムとし、ＣＰＵ２２１がＲＴＯＳ等のカーネルを動作させリアルタイム性に特化したシステムとする。このときＣＰＵ２０１は高性能であるため、メモリ２０３に対して高速にアクセスできる。しかし、不揮発メモリ２２５に対してはバスコントローラ２０４，２２４と、ＣＰＵ２２１内のバスブリッジを経由

してアクセスする必要があるため、不揮発メモリ２２５は、ＣＰＵ２０１にとって十分低速なデバイスとなる。

【００５０】

図６は、本実施形態に係るメインＣＰＵ２０１が不揮発メモリ２２５をハンドリングする方法を説明する図である。

【００５１】

図において、６００１はＣＰＵ２０１のソフトウエアイメージを示す。メモリ２０３はＣＰＵ２０１のメインメモリである。６００２はＣＰＵ２０１の仮想メモリ空間（領域）を示し、仮想メモリをサポートするＯＳ（オペレーティングシステム）がＣＰＵ２０１のハードウエア上でソフトウエア的に構築している。この仮想メモリ空間は、物理的なメモリ２０３の領域よりもより多くのメモリを仮想的に搭載している如く動作させる技術であり、そのとき本当に必要なメモリだけを物理メモリ上を使用し、必要の無いメモリは記憶部６等に待避する。これによりソフトウエアに対して実際に搭載しているメモリ（物理メモリサイズ）以上のメモリをサービスする仕組みである。ここでは、カーネルは例えば４ＫＢ等のページ単位に物理メモリを分割して管理し、それをＣＰＵ２０１の機能である例えばＭＭＵを利用して、必要な物理メモリをページ単位で仮想メモリ空間上に貼り付けることで再利用している。この結果、仮想メモリ空間６００２のメモリアドレスと、物理メモリ空間上のアドレスが一致しなくなるが、通常ソフトウエアはこれを意識しなくても良い構成になっている。

【００５２】

不揮発メモリ２２５は通常アクセスが遅いため、このような遅いデバイスに対してはＣＰＵ２０１に近い位置にバッファを持つのが通常の使用方法となる。そのためこのバッファを管理するための不揮発ドライバ６００４が必要となる。不揮発ドライバ６００４は、カーネルに対して、例えばmalloc( )等のシステムコールを発行し、メモリ２０３に不揮発メモリバッファを要求する。その結果、カーネルは仮想メモリ空間６００２上に不揮発メモリバッファ６００３を作成し、そのアドレスポインタを返す。不揮発ドライバ６００４は、不揮発メモリ２２５から経路６０１１を介してデータをロードし、経路６０１２で不揮発メモリバッファ６００３にコピーする。これにより不揮発メモリバッファ６００３には不揮発メモリ２２５のデータが記憶されているため、不揮発メモリ２２５のリード要求に対しては不揮発メモリバッファ６００３にキャッシュしてあるデータを読めば良い。また不揮発メモリ２２５へのデータの書き込み要求が発生した場合、不揮発メモリ２２５のデータを更新した後、その更新したデータを不揮発メモリ２２５に転送すれば良い。これらは一般的なキャッシュ技術であり、遅いデバイスに対して、通常、このようなキャッシュシステムを構築してＣＰＵのストールを防いでいる。

【００５３】

本実施形態で着目するのは、図６において、不揮発ドライバ６００４が行なう経路６０１１で示すＣＰＵアクセスであり、不揮発メモリ２２５に対するＣＰＵ２０１のアクセスのストールを改善するのが目的となる。この解決方法として、ＣＰＵ２０１が高速であるため効率が大きく落ちるため、他のデバイスによる転送を行なうことが考えられる。一般的にはＤＭＡＣが考えられる。ＨＤＤ等のストレージデバイスは高速なＤＭＡが搭載されており、ＣＰＵとバス権を取り合いながら低速なデバイスとメモリ２０３との間でデータの転送を行う。この際にＣＰＵ２０１のストールは発生しない（厳密にはメモリバスが混めば若干のメモリ待ちが発生する場合もある）。

【００５４】

図２のような２つのＣＰＵ２０１，２２１を有する構成では、ＣＰＵ２２１による代理転送が考えられる。しかし、ＣＰＵ２０１が仮想メモリ空間をサポートしている場合、６００１で示すＣＰＵ２０１のソフトウェア空間でカーネルが構築する仮想メモリ空間６００２のページ配置はカーネルが管理している。このため、他のシステムからこ仮想のメモリ空間６００２にアクセスする手段が無い。つまりＣＰＵ２２１は、仮想メモリ空間６００２にアクセスする手段を持っておらず、不揮発メモリバッファ６００３に対して代理転

送を行なうことができないことになる。

【００５５】

一方、ＣＰＵ２２１は、不揮発メモリ２２５にアクセスが可能であり、バスコントローラ２０４，２２４を経由してメモリ２０３にアクセスできる。

【００５６】

図７は、本実施形態に係るバスコントローラが管理するメモリ空間を説明する図である。

【００５７】

仮想メモリ空間６００２は、ＣＰＵ２０１により実行されるＯＳのカーネルが管理している仮想メモリ空間である。メモリ２０３のメモリ空間は６００５で示している。ここで、仮想メモリ空間６００２にはメモリ２０３の物理メモリ空間（６００５～６００９）が含まれている。また１２０３は、バスコントローラ２０４，２２４上のメモリ空間である。本実施形態では、汎用バスとして一般的なＰＣＩバスを例に説明する。１２０４は、ＣＰＵ２２１のメモリ空間を示している。ＰＣＩメモリ空間１２０３は、設定を行なうことで用途ごとに複数のウインドウを作成することが可能であり、このウインドウを介して各ＣＰＵは異なるバスシステムに対してアクセスが可能となる。例えば、メモリ空間１２０４にマップされている領域１２０６は、バスコントローラ２０４，２２４のメモリ空間１２０３を介して物理メモリ空間６００５にアクセス可能となる。同様にＣＰＵ２０１は、領域１２０８を介してＣＰＵ２２１のバスシステムに接続されている不揮発メモリ２２５に、ローカルバスに接続されているようにアクセスすることができる。但し、この経路が長くなればバスウエイトが入ることになる。

【００５８】

このように、ＣＰＵ２２１は仮想メモリ空間６００２にアクセスできないが、物理メモリ空間６００５にはアクセスできる。本実施形態では、この特徴を利用している。

【００５９】

仮想メモリ空間上に連続した領域を確保する方法として一般的なものにmalloc( )がある。カーネルの種類によって制御が異なるが、例えば一般的なLinuxの場合、カーネルmalloc( )した時点では仮想メモリ空間上に領域を作成するだけである。そして実際にアクセスが発生した際に、そのとき利用可能な物理空間を動的に割り当て、ゼロ初期化した状態で関連付ける。一度関連付けられた物理メモリは、free( )されない限りその値を保持しなければならないので物理メモリ空間上に存在することになる。しかし使用頻度が低い場合や、他の処理により物理メモリが必要になった場合にスワップアウトされ外部記憶装置上に一時待避される。この待避された状態で再度仮想メモリ空間へのアクセスが発生した場合、ページ例外が発行され、ページ例外処理で空き物理メモリを探し、外部記憶装置から確保した新しい物理メモリへ復元し、これを仮想メモリに接続する。このような仮想メモリの仕組みにより、ユーザプロセスは常に同じ仮想メモリ領域にアクセスすることで、常にそこで自分のデータにアクセスすることが可能となる。

【００６０】

再び、図６を参照して説明する。

【００６１】

仮想メモリ空間６００２の連続したメモリブロックは、カーネルがページングを行って物理メモリ上のバッファ６００５～６００９等の分割されたブロックで確保されている。仮想メモリ空間と物理メモリ空間のメモリブロックの関連は、カーネルのシステムコールを利用することで知ることができる。またシステムコールを発行することで、スワップ非対象とすることも可能である。スワップ非対象とした場合、カーネルが再起動しない限り仮想メモリ空間６００２と物理メモリ空間６００５のバッファ６００６～６００９とは普遍的に関連付けられることになる。

【００６２】

本実施形態では、他のバスに接続された異なるカーネルを持つＣＰＵ２２１が、高速なＣＰＵ２０１に変わってデータを転送することを特徴とする。ＣＰＵ２２１は、メモリ２

０３の物理メモリ空間にはバス的な接続があればアクセスできる。しかし前述したように、仮想メモリシステムを用いたカーネルでは、仮想メモリ空間に対応する物理メモリのデータの配置がシステムの起動毎に異なる。このため、ＣＰＵ２２１が代替して、その仮想メモリ空間に対応する物理メモリ空間に代替送信することができない。

【００６３】

本実施形態では、仮想メモリ空間６００２の不揮発メモリバッファ６００３のＡ，Ｂ，Ｃ，Ｄの領域（第１キャッシュ領域）と、物理メモリ空間６００５のバッファ６００６～６００９（第２キャッシュ領域）を対応付ける情報により物理メモリブロック仕様を作成する。

【００６４】

図８は、本実施形態に係る物理メモリのブロック仕様の一例を示す図である。

【００６５】

図において、８００は、不揮発メモリバッファ６００３の領域Ａ，Ｂ，Ｃ，Ｄの情報である仮想メモリアドレスを示す。８０１は、領域Ａ，Ｂ，Ｃ，Ｄに対応した物理メモリ２０３のバッファ６００６、６００７、６００８、６００９のアドレスを示す。８０２は、バッファＡ，Ｂ，Ｃ，Ｄに対応する不揮発メモリ２２５の各記録領域（ブロック）のアドレスａ，ｂ，ｃ，ｄを示す。８０３は、バッファＡ，Ｂ，Ｃ，Ｄのサイズ（メモリ容量）である。これら情報は、不揮発ドライバ６００４がカーネルの情報を収集して作成可能な仕様情報であり、不揮発メモリバッファ６００３を確保した直後に生成される。

【００６６】

この物理メモリブロック仕様に基づいてＣＰＵ２２１が先に述べたようなバスアクセスにより、メモリ２０３のバッファ６００６～６００９にアクセスできる。これにより、図６に示す経路６０１１による不揮発メモリ２２５へのアクセスをなくすことができ、高性能なＣＰＵ２０１を効率的に使用することが可能となる。

【００６７】

図９は、本実施形態に係る画像入出力装置１００の２つのＣＰＵ２０１，２２１によるブート処理を説明するフローチャートである。尚、前述の図３と共通する部分は同じ記号で示している。

【００６８】

前述ステップＳ３０７で、ＣＰＵ２０１がＣＰＵ２２１を起動した後、ステップＳ９０１に進む。ステップＳ９０１で、ＣＰＵ２０１はオペレーティングシステムのカーネルを起動する。これによりＣＰＵ２２１は、ステップＳ３２２でオペレーティングシステムのカーネルを初期化する。そしてステップＳ９１０に進んで、ＣＰＵ２０１から指示される物理メモリブロックの仕様情報の受信を待つ。

【００６９】

一方、ＣＰＵ２０１は、ステップＳ９０１に続いてステップＳ９０２に進み、仮想メモリ空間６００２に不揮発メモリバッファ６００３を獲得する。このときメモリ２０３の物理メモリ空間６００５で実メモリエリアを確保し、そのエリアをロックすることでロック解除が行なわれるまで、その実メモリエリアが利用されないようにする。次にステップＳ９０３に進み、図８に示すような物理メモリブロックの仕様情報を作成する（仕様作成）。そしてステップＳ９０４に進み、その物理メモリブロックの仕様情報をＣＰＵ２２１へ転送する（仕様データ転送）。そしてステップＳ９０５に進み、デバイスドライバの初期化を行なう。

【００７０】

これによりＣＰＵ２２１は、ステップＳ９１０で、ＣＰＵ２０１から受信した物理メモリブロックの仕様情報を解析する。そして、その仕様情報に従って、不揮発メモリ２２５の各ブロックの内容をメモリ２０３のバッファ（６００６乃至６００９）にコピーする（デバイスデータ転送）。そしてステップＳ９１１で、不揮発メモリ２２５の全てのブロックａ～ｄをメモリ２０３のバッファ６００６～６００９にコピーするとステップＳ９１２に進み、ＣＰＵ２０１に不揮発メモリ２２５のデータのコピーが完了したことを通知する

。そしてステップＳ９１３に進み、アプリケーションの初期化を実行してアイドル状態となる。つまり、アプリケーションの実行は、ＯＳのカーネルの起動が完了し、不揮発メモリ２２５の全てのブロックａ～ｄをメモリ２０３のバッファ６００６～６００９にコピーした場合に開始される。

【００７１】

これによりＣＰＵ２０１は、ステップＳ９０６でＣＰＵ２２１からの処理の完了を待ちを抜けて、ステップＳ９０７に進み、ＣＰＵ２０１はアプリケーションの初期化を行ってアイドル状態となる。これによりＣＰＵ２０１は、仮想メモリ空間６００２で不揮発メモリ２２５にアクセスできるため、不揮発メモリ２２５へのアクセスを高速にできるようになる。

【００７２】

次に本実施形態において、不揮発メモリ２２５の内容を変更する際の処理について説明する。

【００７３】

前述の説明では、遅い不揮発メモリ２２５のためにメモリ２０３のメモリ空間６００２に不揮発メモリバッファ６００３を作成する例を説明した。この場合は、不揮発メモリ２２５からのデータの読み込みに対してはキャッシュ（不揮発メモリバッファ６００３）を参照すれば良い。しかし不揮発メモリ２２５への書き込みイベント（同期イベント）が発生した場合は、不揮発メモリバッファ６００３を書き換えて、不揮発メモリ２２５も書き換える必要がある。この場合は、例えばブロック毎に行なう、或いは一定時間ごとに行なう等、様々な手法が考えられている。

【００７４】

図１０は、本実施形態に係る画像入出力装置１００の２つのＣＰＵによる不揮発メモリ２２５の書き換え処理を説明するフローチャートである。

【００７５】

まずステップＳ１０１１で、不揮発メモリ２２５へのデータ書き込みのイベントの発生を待ち、イベントが発生するとステップＳ１０１２に進む。ステップＳ１０１２では、ＣＰＵ２０１は、そのデータをメモリ２０３の対応するバッファに書き込み、前述の物理メモリブロックの仕様情報をＣＰＵ２２１に通知してＣＰＵ２２１に不揮発メモリ２２５の更新を依頼する。尚、ここでは、仕様情報をＣＰＵ２２１に通知する前に、メモリ２０３のバッファ６００６～６００９をロックしてカーネルが再利用されないようにする必要がある。尚、ここでは、ロックしたまま運用しても良いし、ＣＰＵ２２１に転送を依頼している間だけロックしても良い。これによりＣＰＵ２２１は、ステップＳ１０２１の処理に進み、ＣＰＵ２０１により送られた物理メモリブロックの仕様情報に基づき、メモリ２０３のデータを読み出して不揮発メモリ２２５にデータを書き込む。

【００７６】

図１１は、本実施形態に係る２つのＣＰＵの動作を説明するタイミング図である。尚、図１１では、図９のフローチャートの処理と共通する個所には同じ記号を付している。

【００７７】

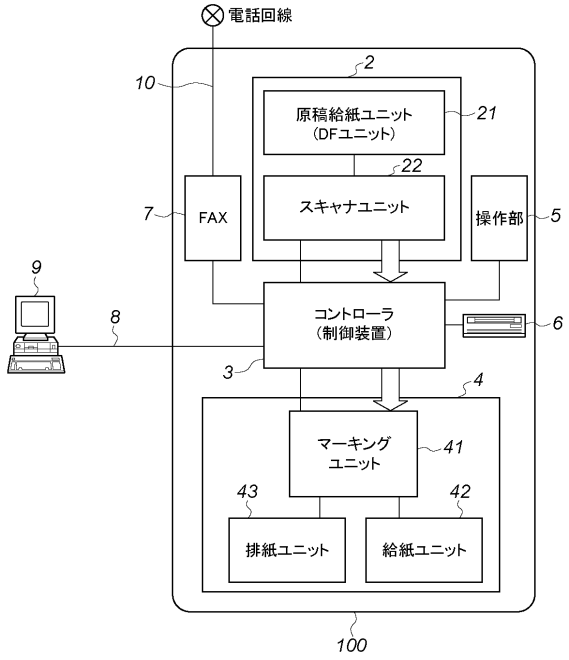
図１１に示すように、ＣＰＵ２２１は、ＣＰＵ２０１から仕様情報を受信すると不揮発メモリ２２５の内容をブロック単位にメモリ２０３に転送して格納する（Ｓ９１０，Ｓ９１１）。これによりＣＰＵ２０１は、図１１の１１００以降の処理で、仮想メモリ空間６００２で不揮発メモリ２２５にアクセスすることができる。１１０１，１１０２は、図１０で説明したように、不揮発メモリ２２５へのデータの書き込み要求が発生した場合を示す。これによりＣＰＵ２２１は、その書き込まれたデータを不揮発メモリ２２５のブロックに格納することができる。

【００７８】

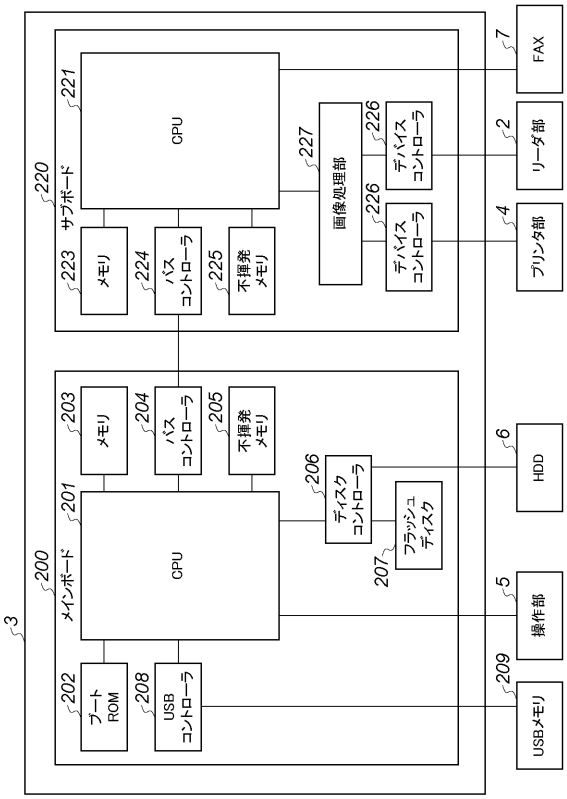
尚、本実施形態では、不揮発メモリ２２５を例に説明したが、図２ではメモリ２２３も外部バス先に接続されており、ＣＰＵ２２１からは高速にアクセスできても、ＣＰＵ２０１からは遅いデバイスとなる。

【００７９】
従って上述した実施形態のように、アクセスの遅いデバイスを不揮発メモリに限定するものではなく、外部バス先に接続されたＣＰＵ２２１がバスアクセス可能なデバイスであれば、前述の実施形態と同様に動作できる。
【００８０】
以上説明したように本実施形態によれば、高性能なＣＰＵが、異なるバスに接続されたアクセスの遅いデバイスに対するアクセス時間を短縮できる。また仮想メモリをサポートしたカーネルにおいて、仮想メモリを有効にしながらも、ＣＰＵの駆動効率を上げることができる。
【００８１】
（他の実施形態）
以上、本発明の実施形態について詳述したが、本発明は、複数の機器から構成されるシステムに適用しても良いし、また一つの機器からなる装置に適用しても良い。
【００８２】
なお、本発明は、前述した実施形態の機能を実現するソフトウェアのプログラムを、システム或いは装置に各種記憶媒体を介して或いは遠隔から供給し、そのシステム或いは装置のコンピュータが該供給されたプログラムを読み出して実行することによっても達成され得る。その場合、プログラムの機能を有していれば、形態は、プログラムである必要はない。

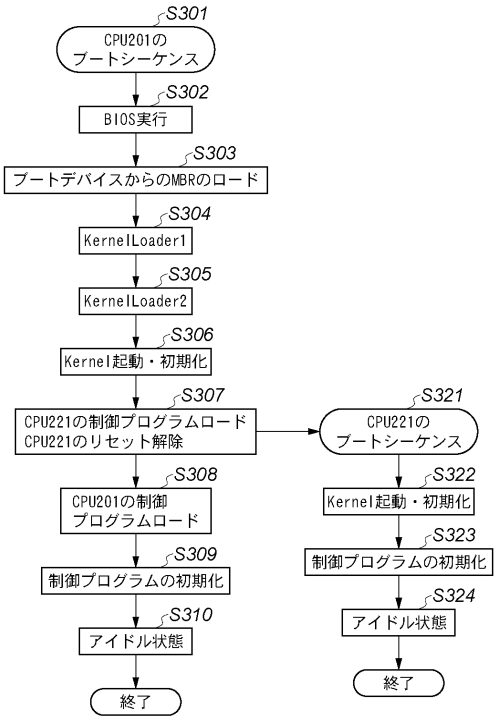
【図１】



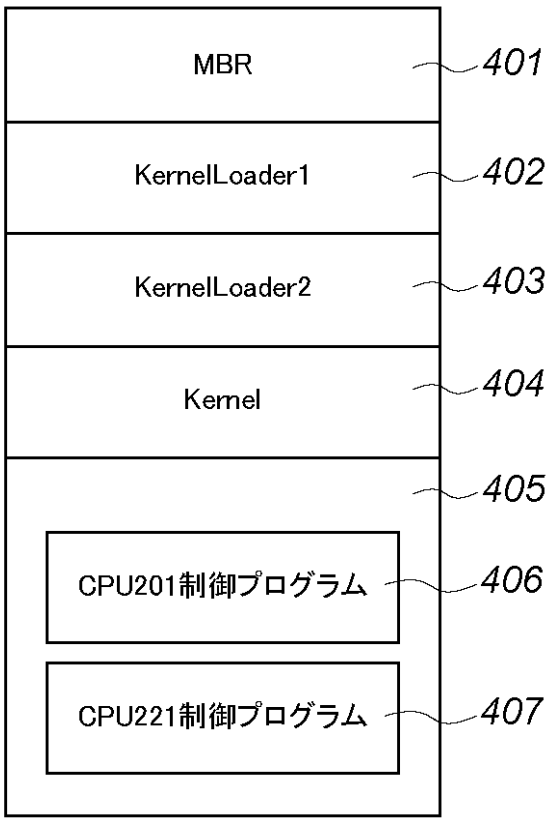
【図２】



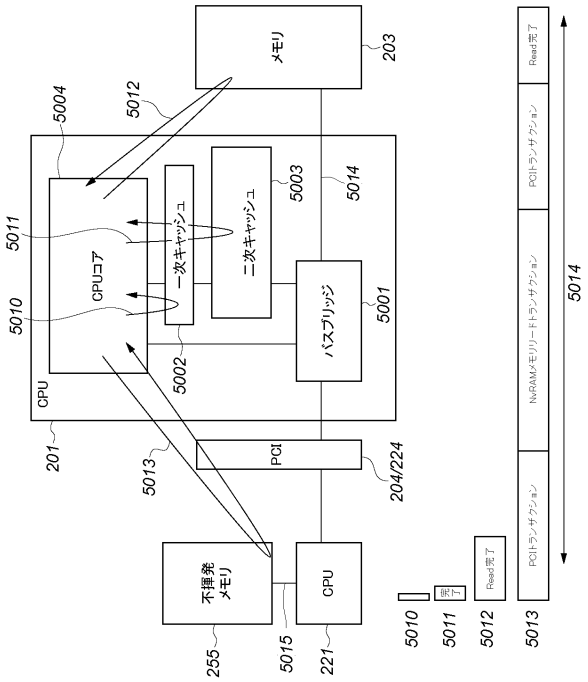
【図３】


S301
CPU201の
ブートシーケンス
S302
BIOS実行
S303
ブートデバイスからのMBRのロード
S304
KernelLoader1
S305
KernelLoader2
S306
Kernel起動・初期化
S307
CPU221の制御プログラムロード
CPU221のリセット解除
S308
CPU201の制御
プログラムロード
S309
制御プログラムの初期化
S310
アイドル状態
終了
S321
CPU221の
ブートシーケンス
S322
Kernel起動・初期化
S323
制御プログラムの初期化
S324
アイドル状態
終了

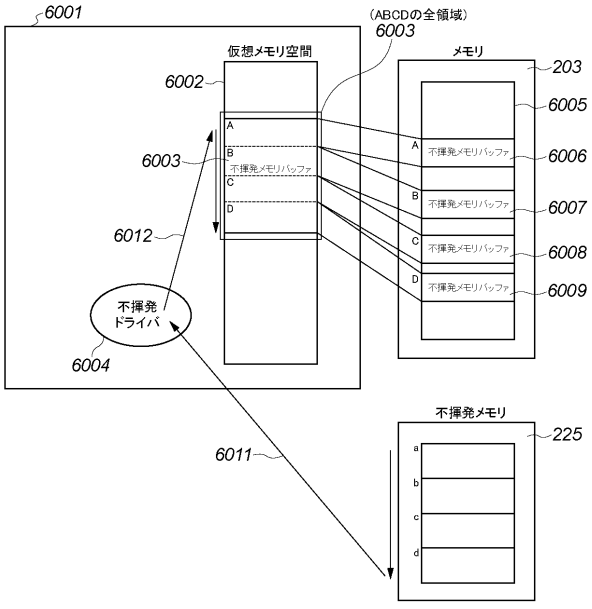
【図４】


MBR
401
KernelLoader1
402
KernelLoader2
403
Kernel
404
405
CPU201制御プログラム
406
CPU221制御プログラム
407

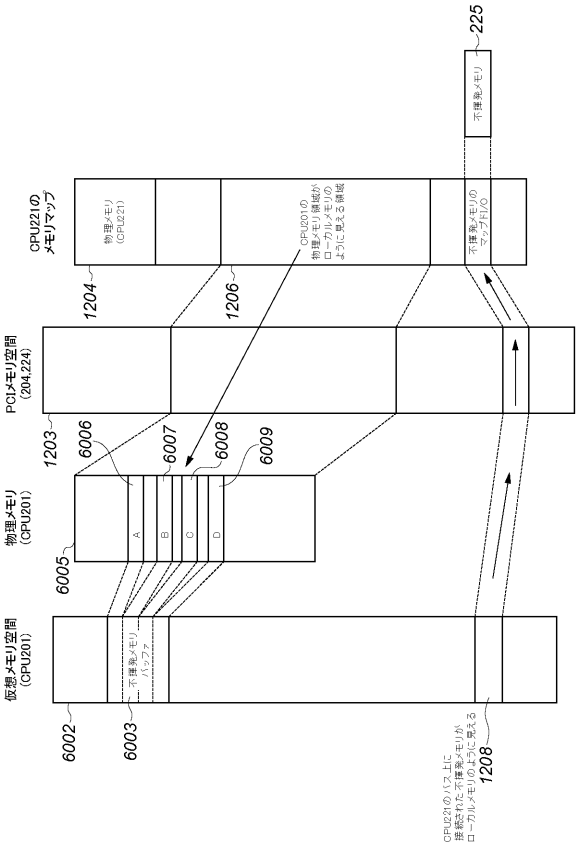
【図５】


メモリ
203
5004
5012
5011
5010
CPUコア
一次キャッシュ
二次キャッシュ
5003
5014
バスブリッジ
5001
5002
CPU
201
5013
PCI
204/224
不揮発
メモリ
255
5015
CPU
221
Read完了
PCIトランザクション
NvRAMメモリリードトランザクション
PCIトランザクション
5014
Read完了
5010
5011
5012
5013

【図６】


6001
(ABCDの全領域)
6003
仮想メモリ空間
メモリ
203
6002
6005
A
B
不揮発メモリバッファ
C
D
6006
6007
6008
6009
6003
6012
不揮発
ドライバ
6004
不揮発メモリ
225
6011
a
b
c
d

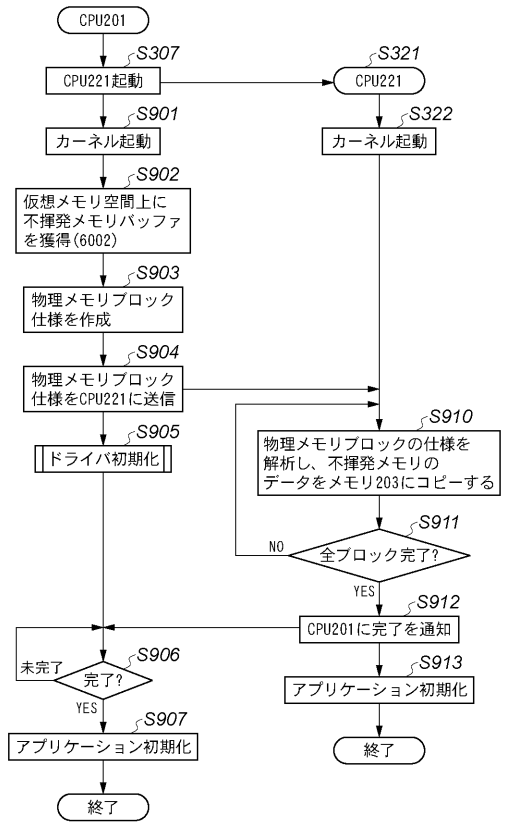
【図７】



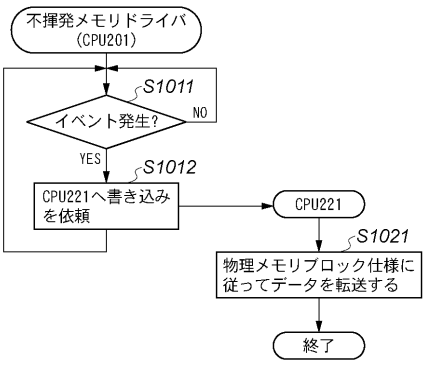
【図８】

800 仮想メモリアドレス	801 物理メモリアドレス	802 デバイスアドレス	803 サイズ
6003のアドレスA	6006のアドレスA	225のアドレスa	←のサイズ
6003のアドレスB	6007のアドレスB	225のアドレスb	←のサイズ
6003のアドレスC	6008のアドレスC	225のアドレスc	←のサイズ
6003のアドレスD	6009のアドレスD	225のアドレスd	←のサイズ

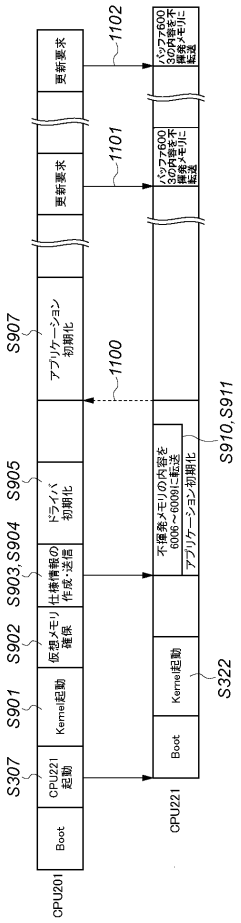
【図９】



【図１０】



【図１１】


CPU201
CPU221
Boot
CPU221 起動
Kernel起動
仮想メモリ確保
仕様情報の作成・送信
ドライバ初期化
アプリケーション初期化
更新要求
更新要求
更新要求
Boot
Kernel起動
不揮発メモリの内容を6006～6009に転送
アプリケーション初期化
バッファ6003の内容を不揮発メモリに転送
バッファ6003の内容を不揮発メモリに転送
S307
S901
S902
S903,S904
S905
S907
S322
S910,S911
1100
1101
1102

フロントページの続き

(51)Int.Cl. FI
G06F 12/10 555
G06F 12/10 503

(72)発明者 原 健二
東京都大田区下丸子3丁目30番2号 キヤノン株式会社内

審査官 桜井 茂行

(56)参考文献 特開昭53－038937（JP, A）
特開2005－157830（JP, A）
特開昭57－162164（JP, A）
特開2008－102850（JP, A）

(58)調査した分野(Int.Cl., DB名)
G06F 12／08－12／10
G06F 3／12
H04N 1／00