(19)日本国特許庁(JP)　(12)公開特許公報(A)　(11)特許出願公開番号
特開2004-220581
(P2004-220581A)
(43)公開日　平成16年8月5日(2004.8.5)

(51)Int.Cl.[7]	F I		テーマコード（参考）
G06F 13/14	G06F 13/14	320H	5B005
G06F 12/02	G06F 12/02	570M	5B014
G06F 12/08	G06F 12/08	531Z	5B060
G06F 12/10	G06F 12/08	551C	
	G06F 12/08	553Z	

審査請求 有　請求項の数 23　ＯＬ　（全 29 頁）　最終頁に続く

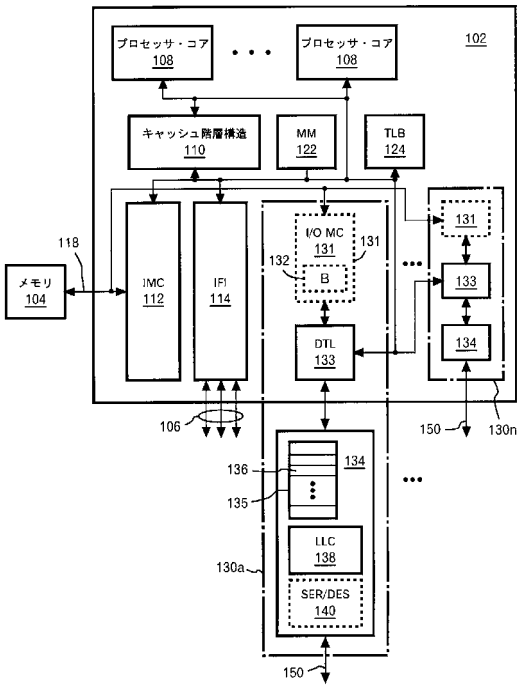
(21)出願番号　特願2003-421698(P2003-421698)
(22)出願日　平成15年12月18日(2003.12.18)
(31)優先権主張番号　10/339724
(32)優先日　平成15年1月9日(2003.1.9)
(33)優先権主張国　米国(US)

(71)出願人　390009531
インターナショナル・ビジネス・マシーンズ・コーポレーション
ＩＮＴＥＲＮＡＴＩＯＮＡＬ　ＢＵＳＩＮＥＳＳ　ＭＡＳＣＨＩＮＥＳ　ＣＯＲＰＯＲＡＴＩＯＮ
アメリカ合衆国１０５０４　ニューヨーク州　アーモンク　ニュー　オーチャード　ロード
(74)代理人　100086243
弁理士　坂口　博
(74)代理人　100091568
弁理士　市位　嘉宏
(74)代理人　100108501
弁理士　上野　剛史

最終頁に続く

(54)【発明の名称】入出力（Ｉ／Ｏ）通信のハードウェア・アクセラレーションを実現するデータ処理システム

(57)【要約】
【課題】改善された処理装置、データ処理システム、およびデータ処理方法を提供すること。
【解決手段】処理装置などの集積回路は、基板、および基板中に形成された集積回路要素を含む。集積回路要素は、命令を実行するプロセッサ・コアと、プロセッサ・コアに結合され、プロセッサ・コアと集積回路の外部のシステム相互接続の間での通信をサポートする相互接続インターフェースと、プロセッサ・コアに結合され、入出力通信リンクを介して入出力通信をサポートする外部通信アダプタの少なくとも一部分を含む。
【選択図】図４



【特許請求の範囲】
【請求項１】
基板と、前記基板中に形成された集積回路要素とを備える集積回路であって、前記集積回路要素が、
命令を実行するプロセッサ・コアと、
前記プロセッサ・コアに結合され、前記プロセッサ・コアと前記集積回路の外部のシステム相互接続との間の通信をサポートする相互接続インターフェースと、
前記プロセッサ・コアに結合され、入出力通信リンクを介して入出力通信をサポートする外部通信アダプタの少なくとも一部分と
を含む集積回路。
【請求項２】
前記集積回路要素がさらに、
前記プロセッサ・コアに結合されたキャッシュ階層構造と、
前記キャッシュ階層構造に結合された集積化メモリ制御装置と
を備える、請求項１に記載の集積回路。
【請求項３】
前記外部通信アダプタを含めて複数の外部通信アダプタの少なくとも一部分を含み、前記複数の外部通信アダプタのうちの少なくとも２つがそれぞれ異なる入出力通信プロトコルを実施する、請求項１に記載の集積回路。
【請求項４】
前記集積回路要素が、通信要求で指定されるアドレスに基づいて、前記集積回路要素内の通信コマンドを前記相互接続インターフェースおよび前記外部通信アダプタに経路指定するためのメモリ・マップを含む、請求項１に記載の集積回路。
【請求項５】
前記外部通信アダプタに結合された変換索引バッファをさらに備え、前記変換索引バッファが、入出力（Ｉ／Ｏ）コマンドで指定される実効アドレスを実アドレスに変換する、請求項１に記載の集積回路。
【請求項６】
前記外部通信アダプタの前記少なくとも一部分が、Ｉ／Ｏデータにアクセスするためにメモリにアクセスする入出力（Ｉ／Ｏ）メモリ制御装置を含む、請求項１に記載の集積回路。
【請求項７】
前記外部通信アダプタの前記少なくとも一部分が、
前記プロセッサ・コアおよびリンク・レイヤ制御装置に結合され、前記プロセッサ・コアによるＩ／Ｏコマンドに応答して、入出力（Ｉ／Ｏ）データ転送を制御するデータ転送ロジックを備える、請求項１に記載の集積回路。
【請求項８】
前記外部通信アダプタの前記少なくとも一部分が、リンク・レイヤ制御装置をさらに備える、請求項１に記載の集積回路。
【請求項９】
前記外部通信アダプタが、Ｉ／Ｏデータ転送の完了を示すソフトウェア・アクセス可能な標識をセットする手段を含む、請求項１に記載の集積回路。
【請求項１０】
前記集積回路内の前記外部通信アダプタの前記部分が、着信Ｉ／Ｏデータと発信Ｉ／Ｏデータの少なくとも一方をバッファするバッファを含む、請求項１に記載の集積回路。
【請求項１１】
請求項１に記載の少なくとも１つの集積回路と、
前記相互接続インターフェースに結合されたシステム相互接続と、
前記少なくとも１つの集積回路に結合されたメモリ・システムと
を備えるデータ処理システム。

【請求項１２】
前記集積回路が、第１の集積回路を備え、
前記外部通信アダプタの前記少なくとも一部分が、第１の部分を備え、
前記データ処理システムがさらに、前記第１の集積回路に接続された第２の集積回路を含み、前記外部通信アダプタが、前記第２の集積回路内で実施される第２の部分を含む、請求項１１に記載のデータ処理システム。
【請求項１３】
前記第２の部分が、リンク・レイヤ制御装置を備える、請求項１２に記載のデータ処理システム。
【請求項１４】
基板と、前記基板中に形成された集積回路要素とを含む第１の集積回路であって、前記集積回路要素が
命令を実行するプロセッサ・コアと、
前記プロセッサ・コアに結合された相互接続インターフェースであって、前記プロセッサ・コアが前記少なくとも１つのプロセッサ・コアと前記第１の集積回路の外部のシステム相互接続との間での通信をサポートする相互接続インターフェースと、
前記プロセッサ・コアに結合され、入出力通信リンクを介して入出力通信をサポートする外部通信アダプタの第１の部分とを含む、第１の集積回路と、
前記第１の集積回路のピンに接続された第２の集積回路とを備え、前記外部通信アダプタが前記第２の集積回路内に実施される第２の部分を含むシステム。
【請求項１５】
前記外部通信アダプタの前記第１の部分が、前記プロセッサ・コアに結合され、プロセッサ・コアによるＩ／Ｏコマンドに応答して入出力（Ｉ／Ｏ）データ転送を制御するデータ転送ロジックを備え、
前記外部通信アダプタの前記第２の部分が、リンク・レイヤ制御装置を備える、請求項１４に記載のシステム。
【請求項１６】
プロセッサ・コアを含む集積回路を動作させる方法であって、
前記集積回路内の相互接続インターフェースを利用して、プロセッサ・コアと前記集積回路の外部のシステム相互接続との間での通信をサポートするステップと、
前記集積回路内の外部通信アダプタを利用して、入出力（Ｉ／Ｏ）通信リンクを介して入出力通信をサポートするステップであって、集積回路内の通信だけを利用して、前記プロセッサ・コアから前記外部通信アダプタにＩ／Ｏ通信コマンドを伝送することを含むステップと
を含む方法。
【請求項１７】
前記プロセッサ・コアがアクセスする可能性が高いデータおよび命令を前記集積回路内のキャッシュ階層構造にキャッシュするステップと、
前記集積回路内の集積化メモリ制御装置を利用して外部メモリにアクセスするステップと
をさらに含む、請求項１６に記載の方法。
【請求項１８】
前記集積回路が、第１および第２の外部通信アダプタの少なくとも一部分を含み、前記Ｉ／Ｏ通信をサポートするステップが、
前記集積回路内の前記第１の外部通信アダプタを利用して、第１のリンク・レイヤ・プロトコルを使用するＩ／Ｏ通信をサポートするステップと、
前記集積回路内の前記第２の外部通信アダプタを利用して、異なる第２のリンク・レイヤ・プロトコルを使用するＩ／Ｏ通信をサポートするステップと
を含む、請求項１６に記載の方法。
【請求項１９】

前記通信要求で指定されるアドレスに基づいて、前記集積回路内の通信コマンドを前記相互接続インターフェースおよび前記外部通信アダプタに経路指定するステップをさらに含む、請求項１６に記載の方法。

【請求項２０】

Ｉ／Ｏ通信をサポートする前記ステップが、前記Ｉ／Ｏコマンド内で指定される実効アドレスを、メモリ位置を識別するために利用される実アドレスに変換するステップをさらに含む、請求項１６に記載の方法。

【請求項２１】

Ｉ／Ｏ通信をサポートする前記ステップが、前記外部通信アダプタ内の入出力（Ｉ／Ｏ）メモリ制御装置を利用して、Ｉ／Ｏデータにアクセスするためにメモリにアクセスするステップを含む、請求項１６に記載の方法。

【請求項２２】

Ｉ／Ｏ通信をサポートする前記ステップが、Ｉ／Ｏデータ転送の完了を示すソフトウェア・アクセス可能な標識をセットするステップを含む、請求項１６に記載の方法。

【請求項２３】

前記Ｉ／Ｏ通信をサポートするステップが、着信Ｉ／Ｏデータと発信Ｉ／Ｏデータの少なくとも一方を前記集積回路内にバッファするステップを含む、請求項１６に記載の方法。

【発明の詳細な説明】

【技術分野】

【０００１】

本発明は、一般にデータ処理に関し、少なくとも一態様では、データ処理システムによる入出力（Ｉ／Ｏ）通信に関する。

【背景技術】

【０００２】

従来のデータ処理システムにおいては、入出力（Ｉ／Ｏ）通信は、一般に、１本または複数の内部バスによってデータ処理システムの１つまたは複数の処理装置に結合されるメモリ・マップＩ／Ｏアダプタによって容易にされている。たとえば、図１は、イーサネット（Ｒ）通信リンク５２を介してリモート・コンピュータ６０とのＩ／Ｏ通信をサポートするＰＣＩ（Peripheral Component Interconnect、周辺装置相互接続）Ｉ／Ｏアダプタ５０を含む従来技術のＳＭＰ（Symmetric Multiprocessor、対称型マルチプロセッサ）データ処理システム８を示している。

【０００３】

図に示すように、従来技術のＳＭＰデータ処理システム８は、ＳＭＰシステム・バス１１により通信できるように結合される複数の処理装置１０を含む。ＳＭＰシステム・バスは、たとえば８バイト幅のアドレス・バスおよび１６バイト幅のデータ・バスを含むことができ、５００ＭＨｚで動作することができる。各処理装置１０は、プロセッサ・コア１４およびキャッシュ階層構造１６を含み、高速（たとえば５３３ＭＨｚの）専用メモリ・バス２０を介して、外部システム・メモリ１２用の関連メモリ制御装置（ＭＣ）１８と通信する。処理装置１０は、一般に先端のカスタム集積回路（ＩＣ）技術を利用して製造され、２ＧＨｚ以上のプロセッサ・クロック周波数で動作することができる。

【０００４】

処理装置１０間の通信は、完全にキャッシュ・コヒーレントである。すなわち、各処理装置１０内のキャッシュ階層構造１６は、従来のＭＥＳＩ（Modified, Exclusive, Shared, Invalid）プロトコル、またはその変形を用いて、その処理装置１０によってアクセスされる現行のキャッシュ・メモリの各グラニュール（granule）が、他の処理装置１０またはシステム・メモリ１２あるいはその両方の中の対応するメモリのグラニュールに対してどういう関係にあるかを追跡する。

【０００５】

ＳＭＰシステム・バス１１には、メザニン（mezzanine）Ｉ／Ｏバス制御装置３０が、

またオプションで、１つまたは複数の追加のメザニン・バス制御装置３２が結合される。メザニンＩ／Ｏバス制御装置３０（および他のメザニン・バス制御装置３２のそれぞれ）は、通信のために、それぞれのメザニン・バス４０をＳＭＰシステム・バス１１にインターフェースする。典型的な実装では、メザニン・バス４０は、ＳＭＰシステム・バス１１よりも幅がずっと狭く、より低い周波数で動作する。たとえば、メザニン・バス４０は、（アドレスとデータを多重化した）８バイト幅とすることができ、２００ＭＨｚで動作することができる。

【０００６】

図に示すように、メザニン・バス４０は、ＭＣＡ（MicrochannelArchitecture、マイクロチャネル・アーキテクチャ）ＩＯＣＣ４２、ＰＣＩエクスプレス（３ＧＩＯ）ＩＯＣＣ４４、およびＰＣＩ　ＩＯＣＣ４６を含めて、いくつかのＩＯＣＣ（I/O channel controllers、Ｉ／Ｏチャネル制御装置）の接続をサポートする。各ＩＯＣＣ４２～４６は、固定した最大数の装置の接続をサポートするスロットを提供するバス４７～４９のそれぞれに結合される。ＰＣＩ　ＩＯＣＣ４６の場合には、接続される装置には、Ｉ／Ｏ通信リンク５２を介してネットワーク５４およびリモート・コンピュータ６０との通信をサポートするＰＣＩ　Ｉ／Ｏアダプタ５０が含まれる。

【０００７】

データ処理システム８内のＩ／Ｏデータと「ローカル」データが、異なるコヒーレンシ・ドメインに属することに留意されたい。すなわち、処理装置１０のキャッシュ階層構造１６は、コヒーレンシを維持するために、従来のＭＥＳＩプロトコルまたはその変形を利用するが、リモート・コンピュータ６０に転送するためにメザニンＩ／Ｏバス制御装置３０にキャッシュされるデータ・グラニュールは、通常、共用状態として、またはデータ・グラニュールが後でデータ処理システム８内で修正される場合には無効状態として記憶される。ほとんどのシステムでは、非排他的状態、修正された状態、または同様の排他的状態が、データ処理システム８内でＩ／Ｏデータに対してサポートされる。さらに、着信Ｉ／Ｏデータ転送はすべて、データを修正するために処理装置１０によって使用されるような、リード・ビフォア・ライト（たとえば、ＲＷＩＴＭ（read-with-intent-to modify、修正のためのリード）およびＤＣＬＡＩＭ）動作ではなくてストア・スルー動作である。

【０００８】

前述の一般的なハードウェア実装では、ＳＭＰデータ処理システム８がＩ／Ｏ通信リンク５２上でデータを伝送するために用いる典型的な方法は、アプリケーション・プロセス、動作環境ソフトウェア（たとえばＯＳおよび関連デバイス・ドライバ）、およびＩ／Ｏアダプタ（および他のハードウェア）のそれぞれが１部分を実施する、３つの部分からなる動作として説明することができる。

【０００９】

所与の任意の時点で、ＳＭＰデータ処理システム８の処理装置１０は、一般に多数のアプリケーション・プロセスを同時に実行する。これらのプロセスの１つがシステム・メモリ１２からリモート・コンピュータ６０へとＩ／Ｏチャネル５２を介してデータを伝送する必要があるという最も簡単な場合には、このプロセスは、まずＩ／Ｏアダプタ５０用のロックを取得するために他のプロセスと競合しなければならない。使用する予定の伝送プロトコルの信頼性およびその他の要因に応じて、データ・グラニュールが伝送に先立って他のプロセスによって修正されないことを保証するために、このプロセスがまた、１つまたは複数の伝送すべきデータ・グラニュールに対して１つまたは複数のロックを取得しなければならないこともある。

【００１０】

このプロセスは、Ｉ／Ｏアダプタ５０用のロック（および伝送すべきデータ・グラニュール用の多分１つまたは複数のロック）を取得した後に、ＯＳソケット・インターフェースを介して、そのオペレーティング・システム（ＯＳ）に１つまたは複数のコールを行う。これらのソケット・インターフェース・コールは、ソケットを初期化し、ポート・アドレスにソケットをバインドし、接続受入れ準備状態を示し、データを送信または受信しあ

るいはその両方を行い、ソケットをクローズすることをオペレーティング・システムに求める要求を含む。これらのソケット・コールでは、呼出し（calling）プロセスは、一般的に利用すべきプロトコル（たとえばＴＣＰ、ＵＤＰなど）、アドレッシングの方法、伝送すべきデータ・グラニュールのベースＥＡ（effective address、実効アドレス）、データ・サイズ、およびリモート・コンピュータ６０内の宛先メモリ位置を示すフォーリン・アドレス（foreign address）を指定する。

【００１１】

次に動作環境ソフトウェアを参照すると、ブートに続いて、ＯＳは、処理装置１０によって内部で使用される仮想（または実効）アドレス空間とは別のＩ／Ｏアドレス空間を割り付けること、およびシステム・メモリ１２中にＴＣＥ（Translation Control Entry、変換制御エントリ）テーブル２４を作成することを含めて、Ｉ／Ｏ通信用のリソースを作成するための様々な動作を実施する。ＴＣＥテーブル２４は、Ｉ／Ｏ装置によって生成されるＩ／Ｏアドレスとシステム・メモリ１２中のＲＡの間で変換を行うＴＣＥを提供することにより、Ｉ／Ｏ通信を実施するのに利用されるＤＭＡ（Direct Memory Access、ダイレクト・メモリ・アクセス）サービスをサポートする。

【００１２】

これらおよびその他のリソースの作成に続いて、ＯＳは、Ｉ／Ｏ通信をサポートするサービスを提供することにより様々なプロセスのソケット・インターフェース・コールに応答する。たとえば、ＯＳは、まずソケット・インターフェース・コールに含まれるＥＡをＲＡ（real address、実アドレス）に変換し、次いで、たとえばＲＡをハッシングすることにより、ＲＡにマッピングすべきＰＣＩ　Ｉ／Ｏアドレス空間のページを決定する。さらに、ＯＳは、システム・メモリ１２中のＴＣＥテーブル２４を、要求されたＩ／Ｏ通信を実施するために利用されるＤＭＡサービスがサポートされるように動的に更新する。もちろん、ＴＣＥテーブル２４内のＴＣＥが現在使用可能でない場合、ＯＳは、ＴＣＥテーブル２４中のあるＴＣＥを犠牲にし、影響を受けるプロセスにそのＤＭＡが終了したことを知らせるか、あるいは必要なＴＣＥを解放するプロセスを要求する必要がある。

【００１３】

次いで、ほとんどのデータ処理システムでは、ＯＳは、Ｉ／Ｏアダプタ５０によるデータ転送のパラメータを指定するＣＣＢ（Command Control Block、コマンド制御ブロック）２２をメモリ１２内に作成する。たとえば、ＣＣＢ２２は、システム・メモリ１２内で位置を指定する１つまたは複数のＰＣＩアドレス空間アドレス、かかる各アドレスに関連するデータ・サイズ、およびリモート・コンピュータ６０内のＣＣＢのフォーリン・アドレスを含むことができる。データ転送のためのＴＣＥおよびＣＣＢ２２を確立した後、ＯＳは、ＣＣＢ２２のベース・アドレスをその呼出しプロセスに返す。使用されるプロトコルによっては、ＯＳはまた、（たとえば、ヘッダでデータをカプセル化することや、フロー制御を提供することなどにより）追加のデータ処理サービスを提供することもできる。

【００１４】

ＣＣＢ２２のベース・アドレスを受け取るとそれに応答して、このプロセスは、ＣＣＢ２２のベース・アドレスをＰＣＩ　Ｉ／Ｏアダプタ５０内のレジスタに書き込むことにより、システム・メモリ１２からリモート・コンピュータ６０へのデータ転送を開始する。この呼出しに応答して、ＰＣＩ　Ｉ／Ｏアダプタ５０は、そのレジスタ中にその呼出しプロセスによって書き込まれるベース・アドレスを利用して、ＣＣＢ２２のＤＭＡ読出しを実施する。（一部の簡単なシステムでは、ＣＣＢ２２が非変換アドレス領域内にあるので、アドレス変換は、ＣＣＢ２２のＤＭＡ読出しにとって必要ではない。ただし、よりハイエンドのサーバー・クラス・システムでは、一般にＣＣＢ２２のＤＭＡ読出しのためにアドレス変換が実施される）。次いで、アダプタ５０はＣＣＢ２２を読み出し、リモート・コンピュータ６０に伝送すべき第１のデータ・グラニュールの（ＣＣＢ２２から読み出された）ベースＰＣＩアドレス空間アドレスを目標とするＤＭＡ読出し動作を発行する。

【００１５】

ＰＣＩアダプタ５０からのＤＭＡ読出し動作の受取りに応答して、ＰＣＩ　ＩＯＣＣ４

６はその内部ＴＣＥキャッシュにアクセスして、指定された目標アドレスに対する変換を見つけ出す。ＴＣＥキャッシュ・ミスに応答して、ＰＣＩ　ＩＯＣＣ４６は、ＴＣＥテーブル２４の読出しを実施して、関連するＴＣＥを得る。ＰＣＩ　ＩＯＣＣ４６が必要なＴＣＥを得た後に、ＰＣＩ　ＩＯＣＣ４６は、ＴＣＥを参照してＤＭＡ読出し動作内で指定されるＰＣＩアドレス空間アドレスをＲＡに変換し、システム・メモリ１２のＤＭＡ読出しを実施し、要求されたＩ／Ｏデータを ＰＣＩ　Ｉ／Ｏアダプタ５０に返す。ＰＣＩ　Ｉ／Ｏアダプタ５０による（たとえばリンク・レイヤ・プロトコルの要件を満たすための）さらなる可能な処理の後、ＰＣＩ　Ｉ／Ｏアダプタ５０は、データ・グラニュールを、リモート・コンピュータ６０のシステム・メモリ内でのデータ・グラニュールの記憶を制御する、リモート・コンピュータ６０内のＣＣＢのフォーリン・アドレスと一緒に、Ｉ／Ｏ通信リンク５２およびネットワーク５４を介してリモート・コンピュータ６０に伝送する。

【００１６】

ＤＭＡ読出し動作およびデータ伝送の前述のプロセスは、ＰＣＩ　Ｉ／Ｏアダプタ５０が、ＣＣＢ２２内で指定されるデータをすべて伝送してしまうまで続行する。その後、ＰＣＩ　Ｉ／Ｏアダプタ５０は割込みをアサートして、データ転送が完了したことを知らせる。当業者には理解されるように、ＰＣＩ　Ｉ／Ｏアダプタ５０による割込みのアサートによって、処理装置１０の１つによるコンテキスト切換えおよびＦＬＩＨ（first-level interrupt handler、第１レベル割込みハンドラ）の実行がトリガされる。次いで、ＦＬＩＨは、（たとえばメザニンＩ／Ｏバス制御装置３０内の）システム割込み制御レジスタを読み取って、その割込みがＰＣＩ　ＩＯＣＣ４６に端を発したと判断し、ＰＣＩ　ＩＯＣＣ４６の割込み制御レジスタを読み取って、その割込みがＰＣＩ　Ｉ／Ｏアダプタ５０によって生成されたと判断し、その後、ＰＣＩ　Ｉ／Ｏアダプタ５０のＳＬＩＨ（second-level interrupt handler、第２レベル割り込みハンドラ）を呼び出して、ＰＣＩ　Ｉ／Ｏアダプタ５０の割込み制御レジスタを読み取って、多分複数のＤＭＡのうちどれが完了したかを判定する。次いで、ＦＬＩＨは、Ｉ／Ｏデータ転送が完了したことを呼出しプロセスに指示するために、ポーリング・フラグをセットする。

【発明の開示】

【発明が解決しようとする課題】

【００１７】

本発明は、以上で概要を説明した従来のＩ／Ｏ通信が非効率的であることを認識したものである。先に指摘したように、ＯＳがメモリ中にＴＣＥテーブルを提供し、それによってＩＯＣＣがシステム・メモリ中でＩ／Ｏドメインからのアドレスを実アドレスに変換することが可能になる。システム・メモリ中でのＴＣＥテーブルの作成と管理に関連するオーバーヘッドは、オペレーティング・システムの性能を低下させ、ＩＯＣＣによるＩ／Ｏアドレスの変換は、各Ｉ／Ｏデータ転送の待ち時間を増加させる。複数のプロセスによるＩ／Ｏアダプタおよびシステム・メモリへのアクセスを同期させるためにロックを使用すること、ならびにそれらへのアクセスについて調停すること、ならびにＩ／Ｏ（たとえば、ＰＣＩ）バス、メザニン・バス、およびＳＭＰシステム・バスによって実施されるプロトコル間で変換を行うことによってさらなる待ち時間が生じる。さらに、ＳＭＰシステム・バス上でのＩ／Ｏデータ転送の伝送は、そうでなければ処理装置間の（たとえば読出し要求および同期動作の）多分性能にとってクリティカルな通信に利用することができたはずの帯域幅を消費する。

【００１８】

従来のデータ処理システムの性能は、Ｉ／Ｏアダプタと呼出しプロセスの間の通信を可能にする割り込みハンドラの使用によってさらに低下する。先に指摘したように、従来の実装においては、データ転送が完了したときＩ／Ｏアダプタが割込みをアサートし、割り込みハンドラは、データ転送が完了したことを呼出しプロセスに知らせるためにシステム・メモリ中でポーリング・フラグをセットする。Ｉ／Ｏアダプタと呼出しプロセスの間の通信を容易にするための割込みの使用は、それによって各データ転送ごとに２つのコンテ

キスト切換えが必要となり、有用な仕事を実施するのではなく割込みハンドラを実行してプロセッサ・サイクルが消費されるので非効率的である。

【００１９】

本発明はさらに、データ処理システム内の他のデータと異なるコヒーレンシ・ドメイン内でＩ／Ｏデータを管理することが多くの場合に望ましくないことを認識したものである。

【００２０】

本発明はまた、データ処理システム性能は、たとえばＩ／Ｏ通信を実施するために用いられる不必要な命令をバイパスすることによりさらに改善できることも認識したものである。たとえば、複数レイヤのプロトコル（たとえば、ＴＣＰ／ＩＰ）を使用するＩ／Ｏ通信では、コンピュータ間でのデータグラムの伝送は、送信側コンピュータでも、受信側コンピュータでも、プロトコル・スタックをトラバースするデータグラムを必要とする。多くのデータ転送では、しばしば結果として得られるアドレス・ポインタ、データ値、またはその他の実行結果が変化することなく、プロトコル・レイヤのうち少なくとも一部の中の命令が繰り返して実行される。したがって、本発明は、Ｉ／Ｏ性能、より一般的にはデータ処理システム性能が、かかる反復的コード・シーケンス内の命令をバイパスすることによって、かなり改善できることを認識したものである。

【課題を解決するための手段】

【００２１】

本発明では、改善された処理装置、データ処理システム、およびデータ処理方法を提供することにより、当技術分野の前記その他の欠点に対処する。本発明の少なくとも一実施形態では、処理装置などの集積回路は、基板および基板中に形成された集積回路要素を含む。この集積回路要素は、命令を実行するプロセッサ・コアと、このプロセッサ・コアに結合され、プロセッサ・コアと集積回路の外部のシステム相互接続との間の通信をサポートする相互接続インターフェースと、プロセッサ・コアに結合され、入出力通信リンクを介して入出力通信をサポートする外部通信アダプタの少なくとも一部分とを含む。

【００２２】

本発明の目的、特徴、および利点は、すべて以下の詳細に記述された説明で明らかとなろう。

【００２３】

本発明の特徴と考えられる新しい特徴は、添付の特許請求の範囲中に記載されている。しかしながら、本発明自体、ならびに好ましい使用形態、そのさらなる目的および利点は、実例としての実施形態についての以下の詳細な説明を添付図面と併せ参照することによって最も良く理解されよう。

【発明を実施するための最良の形態】

【００２４】

再び図面を、具体的には図２を参照すると、本発明を有利に利用することができるネットワーク・システム７０の一例が示されている。図に示すように、ネットワーク・システム７０は、データ通信ができるようにネットワーク７４によって結合された少なくとも２つのコンピュータ・システム（すなわち、ワークステーション・コンピュータ・システム７２およびサーバ・コンピュータ・システム１００）を含む。ネットワーク７４は、任意の数の通信プロトコルを使用する、１つまたは複数の有線、無線、または光によるローカル・エリア・ネットワーク（たとえば、企業のイントラネット）または広域ネットワーク（たとえば、インターネット）を含むことができる。さらに、ネットワーク７４はパケット交換サブネットワークおよび回路交換サブネットワークの一方または両方を含むことができる。以下に詳細に説明するように、本発明によれば、データは、入出力（Ｉ／Ｏ）データ通信のための革新的な方法、システム、および機器を利用して、ワークステーション７２およびサーバ１００により、あるいはワークステーション７２とサーバ１００の間でネットワーク７４を介して転送することができる。

【００２５】

次に、図３を参照すると、本発明による、改善されたＩ／Ｏ通信を含めて改善されたデータ処理をサポートする、マルチプロセッサ（ＭＰ）サーバ・コンピュータ・システム１００の実施形態の一例が示されている。図に示すように、サーバ・コンピュータ・システム１００は、複数の処理装置１０２を含み、これらの複数の処理装置は、それぞれメモリ１０４の１つに結合されている。各処理装置１０２はさらに、諸処理装置１０２間でのデータ、命令、および制御情報の通信をサポートする、統合された分散スイッチング機構１０６に結合される。各処理装置１０２は、集積回路要素がその上に形成された半導体基板を備える単一の集積回路として実施することが好ましい。複数の処理装置１０２、およびスイッチング機構１０６の少なくとも一部分は、有利なことに、共通のバックプレーンまたはチップ・キャリア上に一緒にパッケージ化することができる。

【００２６】

図３にさらに示すように、本発明によれば、１つまたは複数の処理装置１０２は、スイッチング機構１０６とは独立したＩ／Ｏ通信を行うためにＩ／Ｏ通信リンク１５０に結合される。さらに以下で説明するように、通信リンク１５０に処理装置１０２を結合することにより、Ｉ／Ｏ通信をかなり簡単にし、性能を改善することができる。

【００２７】

データ処理システム１００が図示されていない追加の多くの構成要素を含むことができることが当業者には理解されよう。かかる追加の構成要素は本発明を理解するためには必要がないので、これらは図３に示さず、また本明細書でさらに論じてはいない。しかしながら、本発明によって提供されるＩ／Ｏ通信に対する拡張は、任意のシステム・アーキテクチャのデータ処理システムに適用可能であり、一般化したＭＰアーキテクチャまたは図３に示すＳＭＰシステム構造だけに決して限定されないことも理解すべきである。

【００２８】

次に図４を参照すると、サーバ・コンピュータ・システム１００内における処理装置１０２の実施形態の一例のより詳細なブロック図が示されている。図に示すように、処理装置１０２内の集積回路要素は、１つまたは複数の命令スレッドをそれぞれ独立かつ同時に実行できる１つまたは複数のプロセッサ・コア１０８を含む。処理装置１０２はさらに、プロセッサ・コア１０８に結合され、プロセッサ・コア１０８によってアクセスされる可能性の高いデータおよび命令用の待ち時間の小さな記憶域を提供するキャッシュ階層構造１１０をさらに含む。キャッシュ階層構造１１０は、たとえば、別々の分岐した、各プロセッサ・コア１０８ごとのレベル１（Ｌ１）の命令キャッシュおよびデータ・キャッシュ、および複数のプロセッサ・コア１０８によって共用されるレベル２（Ｌ２）の大きなキャッシュを含むことができる。かかる各キャッシュは、従来の（または従来と異なった）キャッシュ・アレイ、キャッシュ・ディレクトリ、およびキャッシュ制御装置を含むことができる。キャッシュ階層構造１１０は、キャッシュされたデータおよび命令のコヒーレンシ状態を追跡するために、そのキャッシュ・ディレクトリ内で、周知のＭＥＳＩ（Modified, Exclusive, Shared, Invalid）キャッシュ・コヒーレンシ・プロトコルまたはそれらの変形プロトコルを実施することが好ましい。

【００２９】

キャッシュ階層構造１１０はさらに、高周波数の広い帯域幅のメモリ・バス１１８によって処理装置１０２に結合されたメモリ１０４へのアクセスを制御するＩＭＣ（integrated memory controller、集積化メモリ制御装置）１１２に結合される。すべての処理装置１０２のメモリ１０４がまとまって、サーバ・コンピュータ・システム１００内の最下レベルの揮発性メモリ（しばしば「システム・メモリ」と呼ばれる）を形成し、このメモリは、一般にすべての処理装置１０２からアクセス可能である。

【００３０】

処理装置１０２はさらに、スイッチング機構１０６用のＩＦＩ（integrated fabric interface、集積化機構インターフェース）１１４を含む。ＩＦＩ１１４は、ＩＭＣ１１２にもキャッシュ階層構造１１０にも結合されており、スイッチング機構１０６に対してプロセッサ・コア１０８が要求する動作を支配するマスタ回路、ならびに（たとえば、コヒ

ーレンシを維持するためにキャッシュ階層構造１１０に対する動作をスヌープすることにより、または関連するメモリ１０４から要求されたデータを検索することにより）スイッチング機構１０６から受け取る動作に応答するスヌープ回路を含む。

【００３１】

処理装置１０２は、プロセッサ・コア１０８およびメモリ・バス１１８に結合された１つまたは複数のＥＣＡ（external communication adapters、外部通信アダプタ）１３０も有する。各ＥＣＡ１３０は、処理装置１０２がその一部を構成するＭＰサブシステムの外部の（または任意選択でサーバ・コンピュータ・システム１００の外部の）装置またはシステムとのＩ／Ｏ通信をサポートする。様々なＩ／Ｏ通信オプションを実現するために、処理装置１０２には、それぞれあるいは全体として（たとえばイーサネット（Ｒ）、ＳＯＮＥＴ、PCIExpress、InfiniBandなどの）様々な通信プロトコルを実施するＥＣＡ１３０を設けることができる。

【００３２】

好ましい実施形態では、ＩＭＣ１１２、ＩＦＩ１１４、およびＥＣＡ１３０のそれぞれは、１つまたは複数のオペレーティング・システムによって割り当てられた実効（または実）アドレスを有するメモリ・マッピングされたリソースである。かかる実施形態では、処理装置１０２は、ＩＭＣ１１２、ＩＦＩ１１４、およびＥＣＡ１３０へのアドレスの割当てを記録するＭＭ（memory map、メモリ・マップ）１２２を備えている。したがって、各処理装置１０２は、コマンドのタイプまたはメモリ・マップ１２２内で提供されるアドレス・マッピングあるいはその両方に基づいて、ＩＭＣ１１２、ＩＦＩ１１４、およびＥＣＡ１３０のどれかにコマンド（たとえば、Ｉ／Ｏ書込みコマンドまたはメモリ読出し要求）を経路指定することが可能である。好ましい実施形態では、ＩＭＣ１１２およびＥＣＡ１３０は、同じダイ中に一体化された個々のプロセッサ・コア１０８とはどのような密接な関係ももたず、任意の処理装置１０２の任意のプロセッサ・コア１０８からアクセス可能になっていることに留意されたい。さらに、ＥＣＡ１３０は、Ｉ／Ｏ読出し動作およびＩ／Ｏ書込み動作を実施するために、サーバ・コンピュータ・システム１００内の任意のメモリ１０４にアクセスすることができる。

【００３３】

より具体的にＥＣＡ１３０について検討すると、各ＥＣＡ１３０は、少なくともＤＴＬ（data-transfer logic、データ転送ロジック）１３３およびプロトコル・ロジック１３４を含み、任意選択のＩ／Ｏ　ＭＣ（I/O memory controller、Ｉ／Ｏメモリ制御装置）１３１をさらに含むことができる。ＤＴＬ１３３は、プロセッサ・コア１０８間で通信リンク１５０へのアクセスの調停を行い、プロセッサ・コア１０８によるＩ／Ｏ読出しコマンドおよびＩ／Ｏ書込みコマンドに応答して、通信リンク１５０とメモリ１０４の間でのデータ転送を制御する制御回路を含む。メモリ１０４にアクセスするために、ＤＴＬ１３３は、任意のＩＭＣ１１２にメモリ読出し要求およびメモリ書込み要求を発行し、あるいは、専用Ｉ／Ｏ　ＭＣ１３１にかかるメモリ・アクセス要求を発行することにより、メモリ１０４にアクセスすることができる。Ｉ／Ｏ　ＭＣ１３１は、複数のメモリ・アクセス要求または着信Ｉ／Ｏデータもしくは発信Ｉ／Ｏデータあるいはその両方をバッファするための任意選択のバッファ記憶装置１３２を含むことができる。

【００３４】

各ＥＣＡ１３０のＤＴＬ１３３はさらに、ＴＬＢ（Translation Lookaside Buffer、変換索引バッファ）１２４に結合され、これらのバッファは、プロセッサ・コア１０８によって使用される実効アドレス（ＥＡ）を実アドレス（ＲＡ）に変換するのに利用されるＰＴＥ（Page Table Entries、ページ・テーブル・エントリ）のサブセットのコピーをバッファする。本明細書中では、実効アドレス（ＥＡ）は、仮想アドレス空間にマッピングされたメモリ記憶位置またはその他のリソースを識別するアドレスとして定義される。他方、実アドレス（ＲＡ）は、本明細書において、実メモリ記憶位置またはその他の実リソースを識別する、実アドレス空間内のアドレスとして定義される。ＴＬＢ１２４は、１つまたは複数のプロセッサ・コア１０８で共用することができ、あるいは１つまたは複数のＤ

ＴＬ１３３によって専用に使用される別個のＴＬＢを備えることができる。

【００３５】

本発明の重要な一態様によれば、図７を参照して以下に詳細に説明するように、ＤＴＬ１３３は、ＴＬＢ１２４にアクセスして、Ｉ／Ｏ動作で転送すべきＩ／Ｏデータのソース・アドレスまたは宛先アドレスとしてプロセッサ・コア１０８によって指定される目標ＥＡを、ＲＡに変換する。したがって、本発明によって、Ｉ／Ｏアドレス変換を実施するために従来技術のようにＴＣＥ２４（図１参照）を使用する必要が全くなくなり、システム・メモリ中でＴＣＥを作成し管理するための付随的なＯＳオーバーヘッドが完全に取り除かれる。

【００３６】

再びＥＣＡ１３０を参照すると、プロトコル・ロジック１３４は、着信Ｉ／Ｏデータおよび発信Ｉ／Ｏデータをバッファするための複数のエントリ１３６を含むデータ・キュー１３５を含む。以下で説明するように、これらのハードウェア・キューは、バッファ１３２またはメモリ１０４あるいはその両方の中の仮想キューで補完することができる。さらに、プロトコル・ロジック１３４は、通信リンク１５０のレイヤ２プロトコルに則って発信Ｉ／Ｏデータを処理し、着信Ｉ／Ｏデータを処理して、たとえばレイヤ２ヘッダを取り除き、その他のデータ・フォーマット化を実施するＬＬＣ（link layer controller、リンク・レイヤ制御装置）１３８を含む。典型的な応用例では、プロトコル・ロジック１３４は、通信リンク１５０上で伝送すべき発信データを直列化し、通信リンク１５０から受信する着信データを非直列化するＳＥＲ／ＤＥＳ（serializer/desirializer、並直列変換器／直並列変換器）１４０をさらに含む。

【００３７】

理解を容易にするために、各ＥＣＡ１３０は完全に別個の回路を有するものとして図４に示してあるが、一部の実施形態では、ダイ面積の効率的な使用を促進するために複数のＥＣＡ１３０が共通の回路を共用できることを理解されたい。たとえば、複数のＥＣＡ１３０が、１つのＩ／Ｏ　ＭＣ１３１を共用することができる。その代わりにまたはそれに加えて、プロトコル・ロジック１３４の複数のインスタンスを、ＤＴＬ１３３の１つのインスタンスによって制御し、１つのインスタンスに接続することもできる。かかる代替実施形態も本発明の範囲に含まれると理解すべきである。

【００３８】

図４にさらに示すように、処理装置１０２内に集積された各ＥＣＡ１３０の部分は実装特有（implementation-specific）であり、本発明の異なる実施形態によって変わるものである。たとえば、実施形態の一例では、ＥＣＡ１３０ａのＩ／Ｏ　ＭＣ１３１およびＤＴＬ１３３は、処理装置１０２内に一体化されるが、ＥＣＡ１３０ａのプロトコル・ロジック１３４は、処理装置１０２のピン数およびダイ・サイズを減少させるためにオフチップのＡＳＩＣ（Application Specific Integrated Circuit、特定用途向けＩＣ）として実施される。それとは対照的に、ＥＣＡ１３０ｎはすべて、処理装置１０２の基板内に一体化される。

【００３９】

各ＥＣＡ１３０が、従来技術のＩ／Ｏアダプタ（たとえば、図１のＰＣＩ　Ｉ／Ｏアダプタ５０）と比較してかなり簡略化されていることに留意されたい。具体的には、従来技術のＩ／Ｏアダプタは、一般にＳＭＰバス・インターフェース・ロジック、ならびに様々なアクティブ・セッションおよび「動作中（in flight）の」バス・トランザクションの状態を維持するための１つまたは複数のハードウェアまたはファームウェアの状態機械を含む。Ｉ／Ｏ通信は、従来のＳＭＰバス上では経路指定されないので、ＥＣＡ１３０は、従来のＳＭＰバス・インターフェース回路を必要としない。さらに、図５および図７に関して以下に詳細に論じるように、かかる状態機械は、メモリ内にセッション状態の情報をＩ／Ｏデータと共に記憶することによって、ＥＣＡ１３０内では規模縮小または除去される。

【００４０】

さらに、処理装置１０２内にＩ／Ｏハードウェアを組み込むことにより、Ｉ／Ｏデータ通信を、スイッチング機構１０６上のデータ通信と同様に完全にキャッシュ・コヒーレントとすることができることに留意されたい。すなわち、各処理装置１０２内のキャッシュ階層構造１１０が、キャッシュ可能なデータを転送するＩ／Ｏ読出しおよび書込み動作の検出に応答して、キャッシュされたデータ・グラニュールのコヒーレンシ状態を適切に更新することが好ましい。たとえば、キャッシュ階層構造１１０は、Ｉ／Ｏ読出し動作中に指定されるアドレスと一致するアドレスを有するキャッシュ済みのデータ・グラニュールを無効にする。同様にキャッシュ階層構造１１０は、キャッシュされたデータ・グラニュールのアドレスと一致するアドレスを指定するＩ／Ｏ書込み動作に応答して、キャッシュ階層構造１１０内にキャッシュされるデータ・グラニュールのコヒーレンシ状態を、排他的なキャッシュ・コヒーレンシ状態（たとえば、ＭＥＳＩの排他的状態または修正状態）から共用状態（たとえばＭＥＳＩの共用状態）に更新する。さらに、Ｉ／Ｏ書込み動作中に伝送されるデータ・グラニュールは、共用状態および無効状態だけに限定されるのではなく、修正状態（たとえばＭＥＳＩの修正状態）または排他的状態（たとえば、ＭＥＳＩの排他的状態または修正状態）として伝送することもできる。かかるデータ転送のスヌープに応答して、キャッシュ階層構造１１０は、対応するキャッシュ・ラインを無効化（あるいは対応するキャッシュ・ラインのコヒーレンシ状態を更新）することになる。

【００４１】

多くの場合、キャッシュ済みのデータのコヒーレンシ状態に影響を及ぼすＩ／Ｏ通信は、スイッチング機構１０６を介してメモリ１０４とＥＣＡ１３０の間でＩ／Ｏデータが通信されるために、複数の処理装置１０２のキャッシュ階層構造１１０によってスヌープされることになる。しかしながら、場合によっては、特定のＩ／Ｏ通信セッションに関与するＥＣＡ１３０およびメモリ１０４が、共に同じ処理装置１０２に関連づけられることもある。したがって、Ｉ／Ｏセッション中のＩ／Ｏ読出し動作およびＩ／Ｏ書込み動作は、処理装置１０２内で内部伝送され、他の処理装置１０２からは見えないものとなる。そうした場合、Ｉ／Ｏデータ転送のマスタ（たとえば、ＥＣＡ１３０）またはスヌープ機構（たとえば、ＩＦＩ１１４またはＩＭＣ１１２）のいずれかが、スイッチング機構１０６上に１つまたは複数のアドレスのみのデータ・キル（data kill）動作またはデータ共用のコヒーレンシ動作を伝送して、他の処理装置１０２中のキャッシュ階層構造１１０に、Ｉ／Ｏデータに関連するディレクトリ・エントリを適切なキャッシュ・コヒーレンシ状態に更新させることが好ましい。

【００４２】

次に図５を参照すると、サーバ・コンピュータ・システム１００内の処理装置１０２に結合されたメモリ１０４の記憶内容のより詳細なブロック図が示されている。メモリ１０４は、たとえば１つまたは複数のＤＲＡＭ（dynamic random access memory、ダイナミック・ランダム・アクセス・メモリ）デバイスを備えることができる。

【００４３】

図に示すように、ハードウェアまたはソフトウェアあるいはその両方により、メモリ１０４中の利用可能な記憶域を、関連する処理装置１０２のプロセッサ・コア１０８に割り付けられた少なくとも１つのプロセッサ領域２４９と、関連する処理装置１０２の１つまたは複数のＥＣＡ１３０に割り付けられた少なくとも１つのＩ／Ｏ領域２５０と、サーバ・コンピュータ・システム１００内の処理装置１０２のすべてに割り付けられ、処理装置１０２のすべてからアクセス可能な共用領域２５２とに分割することが好ましい。プロセッサ領域２４９は、関連する処理装置１０２の各プロセッサ・コア１０８が実行する命令を列挙する任意選択の命令トレース・ログ２６０を記憶する。所望の実装形態によっては、すべてのプロセッサ・コア１０８の命令トレース・ログを、同じプロセッサ領域２４９に記憶することができる。あるいは、各プロセッサ・コア１０８が、それ自体の専用プロセッサ領域２４９中にそれぞれその命令トレース・ログ２６０を記憶することもできる。

【００４４】

Ｉ／Ｏ領域２５０は、それぞれがＩ／Ｏデータ転送用のパラメータを指定する１つまた

は複数のＤＴＣＢ（Data Transfer Control Blocks、データ転送制御ブロック）２５３を記憶することができる。Ｉ／Ｏ領域２５０はさらに、各ＥＣＡ１３０または各Ｉ／Ｏセッションごとに、プロトコル・ロジック１３４内の物理ハードウェア・キュー１３５を補完する仮想キュー２５４と、着信Ｉ／Ｏデータまたは発信Ｉ／Ｏデータの一時記憶を実現するＩ／Ｏデータ・バッファ２５５と、Ｉ／ＯセッションまたはＥＣＡ１３０の制御状態情報をバッファする制御状態バッファ２５６とを含むことが好ましい。たとえば、制御状態バッファ２５６は、かかるコマンドがＤＴＬ１３３によって処理される準備ができるまで、１つまたは複数のＩ／Ｏコマンドをバッファすることができる。さらに、セッション状態の概念を使用するＩ／Ｏ接続では、制御状態バッファ２５６は、セッション状態情報を、多分、ポインタ、またはＩ／Ｏデータ・バッファ２５５中に記憶されるＩ／Ｏデータとの他の構造化関連付けとともに記憶することができる。

【００４５】

図５にさらに示すように、共用領域２５２は、様々な処理装置１０２が実行できるソフトウェア１５８の少なくとも一部分と、処理装置１０２の１つが受信または送信するＩ／Ｏデータ２６２を含むことができる。さらに、共用領域２５２は、以前に論じたように、実効アドレス（ＥＡ）と実アドレス（ＲＡ）の間で変換するために利用されるＰＴＥ（Page Table Entries、ページ・テーブル・エントリ）の少なくとも一部分を含むＯＳ作成ページ・テーブル２６４をさらに含む。

【００４６】

次に図６を参照すると、図２～図３のサーバ・コンピュータ・システム１００のソフトウェア構成１５８の一例のソフトウェア・レイヤ図が示されている。図に示すように、ソフトウェア構成は、その最下位レベルに、システム・スーパバイザ（またはハイパーバイザ）１６０を有し、このスーパバイザ１６０は、データ処理システム８中で同時実行される１つまたは複数のオペレーティング・システム１６２にリソースを割り付ける。オペレーティング・システム１６２の各インスタンスに割り付けられるリソースをパーティションと呼ぶ。したがって、たとえば、ハイパーバイザ１６０は、２つの処理装置１０２をオペレーティング・システム１６２ａのパーティションに、４つの処理装置１０２をオペレーティング・システム１６２ｂのパーティションに、別の処理装置１０２を（タイム・スライシングまたはマルチスレッディングによって）複数のパーティションに、などと割り付けることができ、ある範囲の実アドレス空間および実効アドレス空間を各パーティションにそれぞれ割り付けることができる。

【００４７】

オペレーティング・システム１６２、ミドルウェア１６３、およびアプリケーション・プログラム１６４が、ハイパーバイザ１６０上で実行される。当業者には良く理解されるように、各オペレーティング・システム１６２は、ハイパーバイザ１６０が各オペレーティング・システム１６２に割り付けたリソースのプールからのアドレスおよび他のリソースを様々なハードウェア構成要素およびソフトウェア・プロセスに割り付け、そのパーティションに割り付けられたハードウェアの動作を独立に制御し、ページ・テーブル２６４を作成し管理し、そのアプリケーション・プログラム１６４がオペレーティング・システム・サービスにアクセスするための様々なＡＰＩ（application programming interfaces、アプリケーション・プログラム・インターフェース）を提供する。これらのＯＳ　ＡＰＩには、ソケット・インターフェース、およびＩ／Ｏデータ転送をサポートするその他のＡＰＩが含まれる。

【００４８】

アプリケーション・プログラム１６４は、広範な様々な計算処理、制御、通信、データ管理、およびプレゼンテーションの諸機能のどれかを実施するようにプログラムすることができ、いくつかのユーザ・レベル・プロセス１６６を含む。先に指摘したように、Ｉ／Ｏデータ転送を実施するために、プロセス１６６は、ＯＳ　ＡＰＩを介してその下にあるＯＳ１６２にコールを行って、Ｉ／Ｏデータ転送をサポートする様々なＯＳサービスを要求する。

【００４９】
図７を参照すると、本発明による、Ｉ／Ｏデータ通信方法の一例の高レベルの論理フローチャートが示されている。図７に示すプロセスを、図４に示すハードウェアおよび図５に示すメモリ・ダイアグラムをさらに参照して説明することにする。
【００５０】
図に示すように、図７のプロセスは、ブロック１８０で開始し、次いでブロック１８１へと進む。このブロック１８１は、Ｉ／Ｏ読出し動作およびＩ／Ｏ書込み動作のためのＩ／Ｏ要求を発行する要求プロセス（たとえば、アプリケーション、ミドルウェア、またはＯＳプロセス）を示す。重要なことであるが、要求プロセスが要求されたＩ／Ｏ動作のためにアダプタまたはメモリのロックを取得する必要はない。というのは、ＥＣＡ１３０が処理装置１０２内に一体化されていること、およびそれが提供する通信により、ＥＣＡ１３０は、Ｉ／Ｏコマンドのサービスができるまでプロセッサ・コア１０８によるＩ／Ｏコマンドを「ホールド・オフ」することが可能であり、その代わりにまたはそれに加えて、それに続く処理のために多数のＩ／Ｏコマンドを、バッファ１３２または制御状態バッファ２５６あるいはその両方にバッファすることが可能だからである。以下に論じるように、この「ホールド・オフ」時間がある場合、バッファ１３２または２５５の１つにＩ／Ｏデータをローカルにバッファすることにより、それを最小にすることができる。
【００５１】
所望のプログラミング・モデルに応じて、要求プロセスからのＩ／Ｏ要求をＯＳの関与の下で、またはＯＳの関与なしに処理することができる（このＯＳの関与についてはＩ／Ｏ要求内のフィールドに応じて選択的なものとすることができる）。Ｉ／Ｏ要求をＯＳが処理すべき場合、Ｉ／Ｏ要求は、ＯＳ１６２にＩ／Ｏ通信サービスを要求するＡＰＩコールであることが好ましい。ＡＰＩコールに応答して、ＯＳ１６２は、ブロック１８２に示すように、要求されたＩ／Ｏ転送用のパラメータを指定するＤＴＣＢ（Data Transfer Control Block、データ転送制御ブロック）を構築する。次いで、ＯＳ１６２は、ＤＴＣＢの記憶位置（たとえば、ベースＥＡ）の指示を要求側プロセスに戻すことができる。
【００５２】
あるいは、Ｉ／Ｏ要求をＯＳの関与なしに処理すべき場合には、ブロック１８２に示すように、プロセスがＤＴＣＢを構築することが好ましく、この構築はブロック１８１でＩ／Ｏ要求を発行するより前またはそれと同時に行うことができる。この場合、Ｉ／Ｏ要求は、ＥＣＡ１３０にＤＴＣＢのベースＥＡを供給するために、選択されたＥＣＡ１３０のＤＴＬ１３３にプロセッサ・コア１０８から伝送されるＩ／Ｏコマンドであることが好ましい。
【００５３】
図５に示すように、ＤＴＣＢを、処理装置１０２のローカル・メモリ１０４内の参照番号２５３の位置に構築することができる。あるいは、ＤＴＣＢをプロセッサ・コア１０８内の専用の記憶位置または汎用のレジスタ・セット中に構築することもできる。実施形態の一例では、ＤＴＣＢは、少なくとも、（１）Ｉ／Ｏデータ転送が、着信Ｉ／ＯデータのＩ／Ｏ読出しかそれとも発信Ｉ／ＯデータのＩ／Ｏ書込みかを示すフィールドと、（２）Ｉ／Ｏ動作によりＩ／Ｏデータが転送される転送元または転送先の（たとえば、システム・メモリ１０４中の）１つまたは複数の記憶位置を識別する１つまたは複数の実効アドレス（ＥＡ）を示すフィールドと、（３）Ｉ／Ｏデータを受信または供給するリモート装置、システム、またはメモリ位置を識別するフォーリン・アドレス（たとえば、ＩＰ（Internet Protocol、インターネット・プロトコル）アドレス）の少なくとも一部分を示すフィールドとを含む。
【００５４】
その後、図７に示すプロセスは、ブロック１８３に進み、このブロックでは、選択されたＥＣＡ１３０のＤＴＬ１３３にＤＴＣＢを渡す。理解されるように、プロセッサ・コア１０８がＤＴＣＢをＤＴＬ１３３に「プッシュ」することができ、あるいは、たとえばＩ／Ｏ　ＭＣ１３１またはＩＭＣ１１２に対して１つまたは複数のメモリ読出し動作を発行

することにより、ＤＴＬ１３３がＤＴＣＢを「プル」することもできる。（かかるメモリ読出し動作には、ＴＬＢ１２４を利用したＥＡ－ＲＡ変換が必要となることもある。）ＤＴＣＢの受信に応答して、ＤＴＬ１３３はＤＴＣＢを検査して、要求されたＩ／Ｏ動作がＩ／Ｏ読出しか、それともＩ／Ｏ書込みかを判定する。ＤＴＣＢがＩ／Ｏ読出し動作を指定する場合、図７に示すプロセスは、ブロック１８４から以下に説明するブロック２１０へと進む。しかし、ＤＴＣＢがＩ／Ｏ書込み動作を指定する場合には、図７のプロセスは、ブロック１８４からブロック１９０へと進む。

【００５５】

ブロック１９０では、ＤＴＣＢ内で指定されるＩ／Ｏデータの１つまたは複数のＥＡを、１つまたは複数のメモリ１０４中のＩ／Ｏデータにアクセスするために利用できるＲＡに変換するために、ＤＴＬ１３３のＴＬＢ１２４（図４参照）にアクセスする。実効アドレスから実アドレスへの変換を実施するのに必要なＰＴＥがＴＬＢ１２４中に存在する場合、ブロック１９２でＴＬＢのヒットが起こり、ＴＬＢ１２４はＤＴＬ１３３に対応するＲＡを供給する。次いで、プロセスは、ブロック１９２から以下に説明するブロック２００へと進む。しかし、必要なＰＴＥが、ＴＬＢ１２４に現在バッファされていない場合には、ブロック１９２でＴＬＢミスが起こり、プロセスはブロック１９４に進む。ブロック１９４で、ＯＳは、実効アドレス－実アドレス変換を実施するのに必要なＰＴＥをページ・テーブル２６４からＴＬＢ１２４中にロードするために、従来型のＴＬＢ再ロード動作を実施する。次いで、プロセスは、ブロック２００に進む。

【００５６】

ブロック２００で、ローカル・メモリ１０４からＩ／Ｏデータを得るために、実アドレスを含む読出し要求をＩ／Ｏ　ＭＣ１３１（または、Ｉ／Ｏ　ＭＣが実装されていない場合はＩＭＣ１１２）に発行することにより、また他のメモリ１０４からＩ／Ｏデータを得るために、実アドレスを含む読出し要求をＩＦＩ１１４に発行することにより、ＤＴＬ１３３は、ＤＴＣＢ中で識別される、システム・メモリ１０４からＩ／Ｏデータにアクセスする。Ｉ／Ｏデータが伝送を待つ間、ＤＴＬ１３３は、発信Ｉ／Ｏデータを１つまたは複数のバッファ１３２および２５５に一時的にバッファすることができる。重要なことであるが、このようにデータをバッファすることにより、ＤＴＬ１３３（または要求中のプロセス）がＩ／Ｏデータ用のロックを取得することを必要とせずに、バッファ済みのＩ／Ｏデータが、伝送以前に変更されないように保護され、それによって、システム・メモリ１０４中のデータのコピーに１つまたは複数のプロセスがアクセスしそれを変更することが可能になる。その後、ブロック２０２に示すように、ＤＴＬ１３３は、プロトコル特有のデータグラムおよびメッセージを利用して、発信Ｉ／Ｏデータをキュー１３５およびＬＬＣ１３８（および必要ならＳＥＲ／ＤＥＳ１４０）を介して通信リンク１５０に伝送する。かかる伝送は、ＤＴＣＢが指定するデータがすべて送信されるまで継続する。その後、プロセスは、以下に説明するブロック２４２に進む。

【００５７】

再び図７のブロック１８４を参照すると、ＤＴＬ１３３が、ＤＴＣＢ中で指定されたＩ／Ｏ動作がＩ／Ｏ読出し動作であると判定したのに応答して、プロセスはブロック２１０に進む。このブロック２１０では、Ｉ／Ｏデータの受信の準備が整ったことを示すために、ＤＴＬ１３３が、プロトコル・ロジック１３４および通信リンク１５０を介してネットワーク７４上にＩ／Ｏ読出し要求を送信開始（launch）する。次いで、ブロック２１２では、データグラムがネットワーク７４から受信されるまでプロセスを繰り返す。

【００５８】

プロトコル・ロジック１３４がネットワーク７４からデータグラムを受信するのに応答して、データグラムがＤＴＬ１３３に渡されるが、このＤＴＬ１３３が、バッファ１３２、２５５の１つにデータグラムをバッファすることが好ましい。さらに、ブロック２１４に示すように、ＤＴＬ１３３は、ＴＬＢ１２４にアクセスして、データグラムが指定するＥＡの変換結果を得ようとする。ＥＡを変換するための関連ＰＴＥがＴＬＢ１２４中にバッファされている場合、ＴＬＢヒットがブロック２１６で起こり、ＤＴＬ１３３は目標メ

モリ位置のＲＡを受け取り、プロセスは、以下に説明するブロック２４０に進む。しかし、ブロック２１６でＴＬＢミスであった場合はそれに応答して、プロセスはブロック２２０に進む。ブロック２２０で、指定されたＥＡを変換するために必要なＰＴＥを得るために、ＯＳがシステム・メモリ１０４中のページ・テーブル２６４にアクセスする。ブロック２３０～２３２に示すように、ＴＬＢの再ロード動作の完了を待つ間、Ｉ／Ｏ読出しを停止することができ、あるいは、着信データを１つまたは複数のバッファ１３２および２５５にバッファしながらＩ／Ｏ読出しを続行することもできる。ＴＬＢの再ロード動作が完了し、Ｉ／Ｏ読出し動作のためのＲＡが得られた後、プロセスは、ブロック２４０に進む。このブロック２４０では、ＲＡを指定する１つまたは複数のメモリ書込み動作を発行することにより、ＤＴＬ１３３はＩ／Ｏ読出しデータを（たとえば、１つまたは複数のバッファ１３２、２５５から）メモリ１０４の１つに記憶する。

【００５９】

たとえば、Ｉ／Ｏ読出し動作で大量のデータを読み出す場合、スイッチング機構１０６が多用される場合、あるいはスイッチング機構１０６を介したメモリ記憶動作に関連する待ち時間が望ましくないほど長い場合など、一部のケースでは、スイッチング機構１０６を介して伝送されるＩ／Ｏデータの量を最小限に抑えるのが望ましいこともある。したがって、ブロック２１４～２４０に示すアドレス変換プロセスの改善策として、ＯＳは、Ｉ／ＯデータをＥＣＡ１３０にとってローカルなメモリ１０４に記憶させるように選択的に決定することができる。その場合には、ＯＳは、着信Ｉ／ＯデータグラムにＥＡを変換するためのページ・テーブル２６４をローカル・メモリ１０４中の記憶位置に関連するＲＡで更新する。その結果、ブロック２４０に示す記憶ステップでは、ブロック２１４と２３２の一方で行われるＥＡ－ＲＡ変換に基づいて、着信Ｉ／Ｏデータのすべてをローカル・メモリ１０４の共用メモリ領域２５２内のメモリ位置に記憶する。

【００６０】

プロセスは、ブロック２０２またはブロック２４０のいずれかからブロック２４２に進む。ブロック２４２で、ＥＣＡ１３０がＩ／Ｏデータの完了の指示を要求プロセスに提供する。完了指示は、たとえば、ＤＴＣＢ内の完了フィールド、ＥＣＡ１３０中のメモリ・マッピングされた記憶位置、あるいはプロセッサ・コア１０８内の条件レジスタ・ビットなどその他の完了指示を含むことができる。要求プロセスは、Ｉ／Ｏデータ転送が完了したことを検出するために（たとえば、読出し要求を発行することにより）完了指示をポーリングすることができ、あるいは、完了指示の状態変化によって、ローカルな（すなわち、オンチップの）割込みをトリガすることもできる。重要なことであるが、本発明では、Ｉ／Ｏデータ転送が完了したことを、要求プロセスに信号で知らせるための従来からのＩ／Ｏ割込みは必要でない。その後、図７に示すプロセスは、ブロック２５０で終了する。

【００６１】

次に図８を参照すると、本発明によるプロセッサ・コア１０８の実施形態の一例のより詳細なブロック図が示されている。図に示すように、プロセッサ・コア１０８は、ＩＳＵ（instruction sequence unit、命令シーケンス・ユニット）２７０およびいくつかの実行ユニット２８２～２９０を含む、命令パイプラインを含む。ＩＳＵ２７０は、ＩＭＭＵ（instruction memory management unit、命令メモリ管理ユニット）２７２によって実施されるＥＲＡＴ（effective-to-real address translation、実効－実アドレス変換）によって得られる実アドレスを利用して、処理用の命令をＬ１　Ｉ－キャッシュ２７４からフェッチする。もちろん、要求された命令のキャッシュ・ラインがＬ１　Ｉ－キャッシュ２７４に存在しない場合には、ＩＳＵ２７０は、Ｉ－キャッシュ再ロード・バス２７６を介してキャッシュ階層構造１１０（またはより下のレベルの記憶装置）中のＬ２キャッシュに関連する命令キャッシュ・ラインを要求する。

【００６２】

命令がフェッチされ、前処理が行われる場合はそれが実施された後で、ＩＳＵ２７０は、命令タイプに基づいて、命令バス２８０を介して実行ユニット２８２～２９０に、命令を、おそらく順不同にディスパッチする。すなわち、条件－レジスター修正（condition-

register-modifying）命令および分岐命令は、それぞれＣＲＵ（condition register unit、条件レジスタ・ユニット）２８２およびＢＥＵ（branch execution unit、分岐実行ユニット）２８４にディスパッチされ、固定小数点命令およびロード／ストア命令は、それぞれＦＸＵ（fixed-point unit、固定小数点ユニット）２８６およびＬＳＵ（load-store unitロード・ストア・ユニット）２８８にディスパッチされ、浮動小数点命令は、ＦＰＵ（floating-point unit、浮動小数点ユニット）２９０にディスパッチされる。

【００６３】

好ましい一実施形態では、各ディスパッチされた命令はさらに、関連メモリ１０４（図５参照）中の命令トレース・ログ２６０に記録するために、トレース・バス２８１を介してＩＭＣ１１２に伝送される。代替実施形態では、ＩＳＵ２７０は、プロセッサ・コア１０８の設計（architected）状態にコミットした完了命令だけをトレース・バス２８１を経由して伝送することができ、あるいは、命令トレース・ログ２６０に記録するためにどの命令（たとえば、命令なし、ディスパッチされた命令または完了命令または特定の命令タイプのみあるいはこれらの組合せ）をメモリ１０４に伝送するかの選択を可能にする、関連するソフトウェアまたはハードウェア選択可能なモード・セレクタ２７３を有することができる。さらなる改良では、トレース・バス２８１がディスパッチされた命令のすべてをメモリ１０４に伝えること、およびＩＳＵ２７０がディスパッチされた命令のどれが実際に完了したかを示す完了指示をメモリ１０４に伝送する。これらの実施形態のすべてで、アプリケーションその他のソフトウェア・プログラムの完全な命令トレースを、あまり回路に手を入れずに（non-intrusively）、またプロセッサ・コア１０８の性能を実質的に低下させずに達成することができる。

【００６４】

可能なキューイングおよびバッファリングの後、ＩＳＵ２７０によってディスパッチされた命令が、機会があれば実行ユニット２８０～２９０によって実行される。命令「実行」は、本明細書では、プロセッサの論理回路が、命令オペレーション・コード（opcode）、および関連オペランドがある場合はそれも調査し、それに応答して、データまたは命令をデータ処理システム中（たとえば、システム・メモリ位置間、レジスタもしくはバッファとメモリの間など）で移動し、またはデータに対して論理演算または数値演算を実施するプロセスと定義する。メモリ・アクセス（すなわちロード・タイプまたはストア・タイプの）命令では、実行は、一般的に命令オペランドから目標ＥＡを計算することを含む。

【００６５】

実行ユニット２８２～２９０の１つの中での実行中に、命令は、入力オペランドがある場合はそれを、実行ユニットに結合されたレジスタ・ファイル３００～３０４内の１つまたは複数の設計レジスタまたはリネーム・レジスタあるいはその両方から受け取ることができる。同様に、命令実行のデータ結果（すなわち宛先オペランド）がある場合はそれが、レジスタ・ファイル３００～３０４内の命令で指定される位置に実行ユニット２８２～２９０によって書き込まれる。たとえば、ＦＸＵ２８６は、入力オペランドをＧＰＲＦ（general-purpose register file、汎用レジスタ・ファイル）３０２から受け取り、宛先オペランド（すなわちデータ結果）をＧＰＲＦ３０２に記憶し、ＦＰＵ２９０は入力オペランドをＦＰＲＦ（floating-point register file、浮動小数点レジスタ・ファイル）３０４から受け取り、宛先オペランドをＦＰＲＦ３０４に記憶し、ＬＳＵ２８８は、入力オペランドをＧＰＲＦ３０２から受け取り、データをＬ１　Ｄ－キャッシュ３０８とＧＰＲＦ３０２およびＦＰＲＦ３０４との間で転送させる。同様に、条件－レジスター修正命令または条件－レジスター依存（condition-register-dependent）命令を実行するときは、ＣＲＵ２８２およびＢＥＵ２８４は、ＣＲＦ（controlregister file、制御レジスタ・ファイル）３００にアクセスし、このＣＲＦ３００は、好ましい一実施形態では、それぞれの条件レジスタ、リンク・レジスタ、カウント・レジスタ、およびリネーム・レジスタを含む。ＢＥＵ２８４は、パス・アドレスを得るための条件分岐を解決するために条件レジスタ、リンク・レジスタおよびカウント・レジスタの値にアクセスする。ＢＥＵ２８４は、このパス・アドレスを、それが指示するパスに沿って命令フェッチを開始するために命

令シーケンス・ユニット２７０に供給する。実行ユニットは、命令の実行を終了した後、命令シーケンス・ユニット２７０に知らせ、この命令シーケンス・ユニット２７０では、プログラム順の命令の完了、および、データ結果がある場合はそのデータ結果のプロセッサ・コア１０８の設計状態へのコミットメントをスケジュールする。

【００６６】

図８にさらに示すように、プロセッサ・コア１０８はさらに、取込みロジック３２２と、バイパスＣＡＭ（content addressable memory、連想記憶装置）３２４を備える命令バイパス回路３２０を含む。図１０を参照して以下に説明するように、バイパス回路３２０は、プロセッサ・コア１０８が、Ｉ／Ｏデータ転送を実施するために利用されるものを含めて、反復的コード・シーケンスをバイパスするのを可能にし、それによってシステム性能がかなり改善される。

【００６７】

次に図９を参照すると、命令バイパスＣＡＭ３２４のより詳細なブロック図が示されている。図に示すように、命令バイパスＣＡＭ３２４は、命令ストリーム・バッファ３４０、ユーザ・レベル設計状態ＣＡＭ３４３、およびメモリ・マップ・アクセスＣＡＭ３４６を含む。

【００６８】

命令ストリーム・バッファ３４０は、それぞれがスヌープ・キル・フィールド３４１および命令アドレス・フィールド３４２を含むいくつかのバッファ・エントリを含む。命令アドレス・フィールド３４２は、コード・シーケンス中の命令のアドレス（または少なくとも上位アドレス・ビット）を含み、スヌープ・キル・フィールド３４１は、命令アドレスを対象とするストア・オペレーションまたはその他の無効オペレーションが、Ｉ／Ｏチャネル１５０、ローカル・プロセッサ・コア１０８またはスイッチング機構１０６からスヌープされたかどうかを示す。したがって、命令ストリーム・バッファ３４０の内容は、命令シーケンス中の命令のどれかが、その最後の実行以来変更されたかどうかを示す。

【００６９】

ユーザ・レベル設計状態ＣＡＭ３４３は、それぞれがプロセッサ・コア１０８のユーザ・レベルの設計状態の一部分を形成するそれぞれのレジスタに対応するいくつかのＣＡＭエントリを含む。各ＣＡＭエントリはレジスタ値フィールド３４５を含み、このレジスタ値フィールド３４５は、命令ストリーム・バッファ３４０中に記録されるコード・シーケンスの始めと終りの時点での（たとえば、レジスタ・ファイルＣＲＦ３００、ＧＰＲＦ３０２、およびＦＰＲＦ３０４内の）対応するレジスタの値を記憶する。したがって、ＣＡＭエントリのレジスタ値フィールドは、１つがコード・シーケンスの始めで取り込まれ、第２のものがコード・シーケンスの終りで取り込まれる、プロセッサ・コア１０８のユーザ・レベル設計状態の２つの「スナップ・ショット」を含む。各ＣＡＭエントリには、使用フラグ３４４が関連づけられており、これは、レジスタ値フィールド３４５内の関連レジスタ値が書込み前のコード・シーケンスの途中で読み出されたかどうか（すなわち、初期レジスタ値が、コード・シーケンスの正しい実行にとって重要であるかどうか）を示す。この情報は、後で、ＣＡＭ３４３中のどの設計値を比較する必要があるかどうかを判定するために使用される。

【００７０】

メモリ・マップ・アクセスＣＡＭ３４６は、メモリ・アクセス命令およびＩ／Ｏ命令の目標アドレスおよびデータを記憶するためのいくつかのＣＡＭエントリを含む。各ＣＡＭエントリは、アクセス（たとえば、ロード・タイプまたはストア・タイプの）命令の目標アドレス、および目標アドレスによって識別される記憶位置またはリソースに書き込まれまたはそこから読み出されるデータを記憶するための目標アドレス・フィールド３４８およびデータ・フィールド３５２を有する。このＣＡＭエントリはさらに、Ｌ／Ｓ（load／store、ロード／ストア）フィールド３４９、およびＩ／Ｏフィールド３５０を含み、これらはそれぞれ、関連するメモリ・アクセス命令がロード・タイプ命令か、ストア・タイプ命令か、また関連アクセス命令がＩ／Ｏ装置に割り付けられたアドレスを対象としたも

のかどうかを示す。メモリ・マップ・アクセスＣＡＭ３４６内の各ＣＡＭエントリはさらに、スヌープ・キル・フィールド３４７を含み、このフィールドは、その目標アドレスを対象とするストア・オペレーションまたはその他の無効オペレーションが、Ｉ／Ｏチャネル１５０、ローカル・プロセッサ・コア１０８またはスイッチング機構１０６からスヌープされたかどうかを示す。したがって、命令ストリーム・バッファ３４０の内容は、命令ストリーム・バッファ３４０に記録される命令シーケンスによって実施される作業が命令シーケンスを最後に実行してから、修正されたかどうかを示す。

【００７１】

図９は、１つの命令シーケンスに関連したバイパスＣＡＭ３２４内のリソースを示すが、任意数の多分繰返し可能な命令シーケンスの記憶を実現するためにかかるリソースを複製できることを理解されたい。

【００７２】

次に図１０を参照すると、本発明による、プログラムの実行中に反復的コード・シーケンスをバイパスする方法の一例の高レベル論理フローチャートが示されている。図に示すように、プロセスはブロック３６０から開始し、このブロックでは、プロセッサ・コア１０８が、あるプロセス（たとえば、アプリケーション・プロセス、ミドルウェア・プロセスまたはオペレーティング・システム・プロセス）内の任意の点で命令を実行している。図８に示すプロセッサ・コアの実施形態では、命令バイパス回路３２０中の取込みロジック３２２は、ＩＳＵ２７０によって生成される命令アドレスを受け取り、任意選択でまたは追加として、ＩＳＵ２７０によってフェッチまたはディスパッチあるいはその両方が行われる命令を受け取るように結合される。たとえば、一実施形態では、取込みロジック３２２は、ＩＳＵ２７０のＩＡＲ（instruction address register、命令アドレス・レジスタ）２７１に含まれる次の命令フェッチ・アドレスを受け取るように結合することができる。図９のブロック３５２に示すように、取込みロジック３２２は、繰り返し実行されるコード・シーケンスの始めに一般に見出される、ＯＳ　ＡＰＩコールなどの命令があるかどうか、ＩＳＵ２７０内の命令アドレスまたはオペレーション・コードあるいはその両方を監視する。取込みロジック３２２が反復的コード・シーケンスを開始するものと認識する、１つまたは複数の命令アドレスまたは命令オペレーション・コード（opcode）あるいはその両方に基づいて、取込みロジック３２２は、多分反復的コード・シーケンスが検出されたことを命令バイパスＣＡＭ３２４に知らせる「コード・シーケンス開始」指示を命令バイパスＣＡＭ３２４に伝送する。他の実施形態では、各命令アドレスをバイパスＣＡＭ３２４に単に提供するだけでよい。

【００７３】

ブロック３６４に示すように、「コード・シーケンス開始」信号または命令アドレスに応答して、命令バイパスＣＡＭ３２４は、多分反復的なコード・シーケンスをバイパスすべきか否かを判定する。この判定を行う際に、好ましい一実施形態では、バイパスＣＡＭ３２４は４つの要因を考慮に入れる。第１に、バイパスＣＡＭ３２４は、命令ストリーム・バッファ３４０を参照して、検出された命令アドレスが命令ストリーム・バッファ３４０内に記録された開始命令アドレスと一致するかどうかを判定する。第２に、バイパスＣＡＭ３２４は、ユーザ・レベルの設計状態ＣＡＭ３４３を参照して、使用フィールド３４４がセットされた始めの各ユーザ・レベル設計状態レジスタの値が、検出命令の実行後のプロセッサ・コア１０８中の対応するレジスタの値と一致するかどうかを判定する。この比較をする際に、使用フィールド３４４がリセットされたレジスタ（すなわち、命令シーケンス中で使用されないレジスタ、または読み出される前に書き込まれたレジスタ）は考慮に入れない。第３に、バイパスＣＡＭ３２４は、命令ストリーム・バッファ３４０のスヌープ・キル・フィールド３４１を参照して、命令シーケンス中の命令のどれかがスヌープ・キル・オペレーションによって修正または無効化されたかどうかを判定する。第４に、バイパスＣＡＭ３２４は、メモリ・マップ・アクセスＣＡＭ３４６のスヌープ・キル・フィールド３４７を参照して、命令シーケンス中のアクセス命令の目標アドレスのどれかがスヌープ・キル・オペレーションの対象になったかどうかを判定する。

【００７４】
一実施形態では、バイパスＣＡＭ３２４が、４つの条件すべてが満たされたと判断する場合、すなわち検出された命令アドレスが記憶されたコード・シーケンスの最初の命令アドレスと一致し、ユーザ・レベル設計状態が一致し、命令シーケンスの命令アドレスまたは目標アドレスがスヌープ・キルを受けていない場合には、検出済みのコード・シーケンスをバイパスすることができる。より好ましい一実施形態では、第４の条件は、たとえスヌープ・キル・フィールド３４７により、ストア・タイプの命令の目標アドレスに対して（ただしロード・タイプではそうではない）、１つまたは複数のスヌープ・キルが指示される場合でも、バイパスＣＡＭ３２４はコードをバイパスさせるように修正される。これが可能なのは、以下にさらに論ずるように、コードのバイパスをサポートするために、スヌープ・キルの影響を受けるメモリ記憶動作を実施できるからである。

【００７５】
バイパスＣＡＭ３２４が、検出された命令で始まるコード・シーケンスをバイパスすることができないと判定する場合、プロセスは以下に説明するブロック３８０に進む。しかしながら、バイパスＣＡＭ３２４が、検出されたコード・シーケンスをバイパスすることができると判定する場合には、プロセスは、処理中のコア１０８が反復的コード・シーケンスをバイパスするブロック３６８に進む。

【００７６】
反復的コード・シーケンスのバイパスは、ＩＳＵ２７０が、処理中のコア１０８の命令パイプライン中にある反復的コード・シーケンスに属する任意の命令をキャンセルし、その反復的コード・シーケンス中の命令を追加してフェッチしないようにするものであることが好ましい。さらに、バイパスＣＡＭ３２４は、最後のユーザ・レベル設計状態をユーザ・レベル設計状態ＣＡＭ３４３からプロセッサ・コア１０８のユーザ・レベル設計レジスタにロードし、Ｉ／Ｏフィールド３５０によりＩ／Ｏリソースを対象とすると指示される命令シーケンス中の各アクセス命令を実施する。Ｉ／Ｏストア・タイプのオペレーションでは、データ・フィールド３５２からのデータが使用される。最後に、ストア・タイプ・オペレーションの目標アドレスに対するスヌープ・キルが存在するときに、コード・バイパスがサポートされる場合には、バイパスＣＡＭ３２４は、データ・フィールド３５２に含まれるデータを利用して、少なくともスヌープ・キルの影響を受けるメモリ・ストア・オペレーションがあれば、そのそれぞれのオペレーション（および任意選択で命令シーケンス中のあらゆるメモリ・ストア・オペレーション）を実施する。したがって、バイパスＣＡＭ３２４が反復的コード・シーケンスをバイパスすることを選択した場合には、バイパスＣＡＭ３２４は、プロセッサ・コア１０８のユーザ・レベル設計状態、メモリ・イメージ、およびプロセッサ・コア１０８のＩ／Ｏリソースが、まるでプロセッサ・コア１０８の実行ユニット２８２～２９０内で反復的コード・シーケンスを実際に実行するように見えるようにするために必要なオペレーションをすべて実施する。その後、ブロック３６８からブロック３９０に進むプロセスで示すように、プロセッサ・コア１０８は、反復的コード・シーケンスに続く命令で始まるプロセス中で、命令の通常のフェッチと実行を再開し、それによって、反復的コード・シーケンスを含む１つまたは複数の（および任意数までの）非ノーオペレーション命令を実行する必要が完全になくなる。

【００７７】
次に図１０のブロック３８０を参照すると、命令バイパスＣＡＭ３２４が多分反復的なコード・シーケンスをバイパスできないと判定する場合には、命令バイパスＣＡＭ３２４は、検出されたコード・シーケンスの始めのユーザ・レベル設計状態をユーザ・レベル設計状態ＣＡＭ３４３内に記録し、検出されたコード・シーケンス中の命令の命令アドレスを、命令ストリーム・バッファ３４０の命令アドレス・フィールド３４２内に記録することを開始し、メモリ・アクセス命令に属する目標アドレス、データ結果、およびその他の情報をメモリ・マップ・アクセスＣＡＭ３４６中に記録することを開始する。判断ブロック３８４で示すように、命令バイパスＣＡＭ３２４は、取込みロジック３２２が反復的コード・シーケンスの終りを検出するまで、検出されたコード・シーケンスに属する情報を

記録し続ける。命令バイパスＣＡＭ３２４が一杯になるか、または取込みロジック３２２が反復的コード・シーケンスの終りを検出するのに応答して、たとえば、１つまたは複数の命令アドレスおよびオペレーション・コードまたは割込みイベントの発生に基づいて、取込みロジック３２２は、「コード・シーケンス終了」信号をバイパスＣＡＭ３２４に送る。ブロック３８６に示すように、「コード・シーケンス終了」信号の受取りに応答して、バイパスＣＡＭ３２４は、プロセッサ・コア１０８の終了ユーザ・レベル設計状態をユーザ・レベル設計状態ＣＡＭ３４３中に記録し、その後記録を継続する。その後、ブロック３９０で命令の実行が続行され、コード・シーケンスを次にそれが検出されたときにバイパスするのに必要な情報がバイパスＣＡＭ３２４にロードされる。

【００７８】

本明細書中に記載の命令バイパスは、推論的プロセッサ、非推論的プロセッサ、および順不同実行プロセッサ中で実施することができることに留意されたい。あらゆる場合に、コード・シーケンスをバイパスすべきか否かの判断はバイパスＣＡＭ３２４内に記憶される非推論的な情報に基づいており、プロセッサ・コア１０８の設計状態にまだコミットしていない推論的な情報に基づいてはいない。

【００７９】

本発明の命令バイパス回路３２０は、任意長の反復的コードをバイパス可能にすることも理解されたい。この場合、最大可能コード・バイパス長は、少なくとも部分的にはバイパスＣＡＭ３２４の容量によって決まる。したがって、長いコード・シーケンスのバイパスをサポートすることが望ましい実施形態では、メモリ１０４など、一部または全部がオフチップ・メモリの形でバイパスＣＡＭ３２４を実施するのが望ましいこともある。一部の実施形態では、バイパスＣＡＭ３２４を命令トレース・ログ２６０に書き込むべき命令のオンチップ「キャッシュ」として使用すること、および、たとえば命令シーケンスがバイパスＣＡＭ３２４からの情報で置き換えられたときには、バイパスＣＡＭ３２４からの情報をメモリ１０４に繰り返し書き込むことが好ましいこともある。かかる実施形態では、命令トレース・ログ２６０に書き込まれる情報は、たとえばリンクしたリスト・データ構造を利用して、ストア・オペレーションの順序が維持されるように構造化されていることが好ましい。

【００８０】

図９～図１０は、理解を容易にするために、ユーザ・レベル設計状態だけに基づいたコード・バイパスを示しているが、コード・シーケンスをバイパスするか否かを判定する際に、追加のレイヤの状態情報を含めて追加の状態情報を考慮に入れることができることを理解されたい。たとえば、命令シーケンスをバイパスすべきかを判定するために、プロセッサ・コア１０８の現在のスーパーバイザ・レベルの設計状態と比較するためスーパーバイザ・レベルの設計状態を状態ＣＡＭ３４３中に記録しておくこともできるはずである。かかる実施形態では、状態ＣＡＭ３４３中に記録されるスーパーバイザ・レベル設計状態は、必ずしも命令シーケンスの始めではなく、命令シーケンスの内部でＯＳコールがなされた時点での「スナップ・ショット」とすることが好ましい。記憶されたユーザ・レベル設計状態と現在のユーザ・レベル設計状態が一致し、記憶されたスーパーバイザ・レベル状態と現在のスーパーバイザ・レベル状態が一致しない場合には、命令シーケンスの部分的なバイパスを依然として実施することができ、このバイパスは、命令シーケンスがスーパーバイザ・レベル設計状態に入る前に（たとえば、ＯＳコールの前に）終わる。

【００８１】

以上に説明してきたように、本発明は、データ処理の改善された方法、機器、およびシステムを提供する。一態様では、集積回路が、プロセッサ・コアと、入出力通信リンクを介して入出力通信をサポートする外部通信アダプタの少なくとも一部分との両方を含んでいる。プロセッサ・コアと同じ集積回路内にＩ／Ｏ通信アダプタを一体化することにより、一般的にデータ処理に対するいくつかの機能改善がサポートされ、具体的にはＩ／Ｏ通信がサポートされる。たとえば、同じ集積回路にＩ／Ｏ通信アダプタおよびプロセッサ・コアを一体化することにより、ロック取得の待ち時間、プロセッサ・コアとＩ／Ｏ通信ア

ダプタとの間の通信の待ち時間、およびＩ／Ｏアドレス変換の待ち時間を含めて、Ｉ／Ｏ通信の待ち時間をもたらす複数の原因を減らしまたは取り除くことが容易になる。さらに、プロセッサ・コアおよびその関連するキャッシュと同じ集積回路内にＩ／Ｏ通信アダプタを集積化することにより、修正されたキャッシュ・コヒーレンシ状態および排他的なキャッシュ・コヒーレンシ状態をＩ／Ｏデータに割り当てることを含めて、完全にキャッシュ・コヒーレントなＩ／Ｏ通信が容易になる。

【００８２】

他の一態様では、データ処理性能が、Ｉ／Ｏ通信プロセス中で一般に見られるような反復的コード・シーケンスの実行をバイパスすることによって改善される。

【００８３】

さらに他の一態様では、関連する低レベルのメモリのプロセッサ・メモリ域内で各プロセッサ・コアの命令トレースを作成することにより、データ処理の振る舞いのテスト、検証、性能評価および監視が容易になる。

【００８４】

本発明を、好ましい一実施形態に関して具体的に示し説明してきたが、本発明の趣旨および範囲を逸脱することなく、形態および細部の様々な変更を行うことができることが、当業者には理解されよう。

【００８５】

まとめとして、本発明の構成に関して以下の事項を開示する。

【００８６】

（１）基板と、前記基板中に形成された集積回路要素とを備える集積回路であって、前記集積回路要素が、

命令を実行するプロセッサ・コアと、

前記プロセッサ・コアに結合され、前記プロセッサ・コアと前記集積回路の外部のシステム相互接続との間の通信をサポートする相互接続インターフェースと、

前記プロセッサ・コアに結合され、入出力通信リンクを介して入出力通信をサポートする外部通信アダプタの少なくとも一部分と

を含む集積回路。

（２）前記集積回路要素がさらに、

前記プロセッサ・コアに結合されたキャッシュ階層構造と、

前記キャッシュ階層構造に結合された集積化メモリ制御装置と

を備える、上記（１）に記載の集積回路。

（３）前記外部通信アダプタを含めて複数の外部通信アダプタの少なくとも一部分を含み、前記複数の外部通信アダプタのうちの少なくとも２つがそれぞれ異なる入出力通信プロトコルを実施する、上記（１）に記載の集積回路。

（４）前記集積回路要素が、通信要求で指定されるアドレスに基づいて、前記集積回路要素内の通信コマンドを前記相互接続インターフェースおよび前記外部通信アダプタに経路指定するためのメモリ・マップを含む、上記（１）に記載の集積回路。

（５）前記外部通信アダプタに結合された変換索引バッファをさらに備え、前記変換索引バッファが、入出力（Ｉ／Ｏ）コマンドで指定される実効アドレスを実アドレスに変換する、上記（１）に記載の集積回路。

（６）前記外部通信アダプタの前記少なくとも一部分が、Ｉ／Ｏデータにアクセスするためにメモリにアクセスする入出力（Ｉ／Ｏ）メモリ制御装置を含む、上記（１）に記載の集積回路。

（７）前記外部通信アダプタの前記少なくとも一部分が、

前記プロセッサ・コアおよびリンク・レイヤ制御装置に結合され、前記プロセッサ・コアによるＩ／Ｏコマンドに応答して、入出力（Ｉ／Ｏ）データ転送を制御するデータ転送ロジックを備える、上記（１）に記載の集積回路。

（８）前記外部通信アダプタの前記少なくとも一部分が、リンク・レイヤ制御装置をさらに備える、上記（１）に記載の集積回路。

（９）前記外部通信アダプタが、Ｉ／Ｏデータ転送の完了を示すソフトウェア・アクセス可能な標識をセットする手段を含む、上記（１）に記載の集積回路。
（１０）前記集積回路内の前記外部通信アダプタの前記部分が、着信Ｉ／Ｏデータと発信Ｉ／Ｏデータの少なくとも一方をバッファするバッファを含む、上記（１）に記載の集積回路。
（１１）上記（１）に記載の少なくとも１つの集積回路と、
前記相互接続インターフェースに結合されたシステム相互接続と、
前記少なくとも１つの集積回路に結合されたメモリ・システムと
を備えるデータ処理システム。
（１２）前記集積回路が、第１の集積回路を備え、
前記外部通信アダプタの前記少なくとも一部分が、第１の部分を備え、
前記データ処理システムがさらに、前記第１の集積回路に接続された第２の集積回路を含み、前記外部通信アダプタが、前記第２の集積回路内で実施される第２の部分を含む、上記（１１）に記載のデータ処理システム。
（１３）前記第２の部分が、リンク・レイヤ制御装置を備える、上記（１２）に記載のデータ処理システム。
（１４）基板と、前記基板中に形成された集積回路要素とを含む第１の集積回路であって、前記集積回路要素が
命令を実行するプロセッサ・コアと、
前記プロセッサ・コアに結合された相互接続インターフェースであって、前記プロセッサ・コアが前記少なくとも１つのプロセッサ・コアと前記第１の集積回路の外部のシステム相互接続との間での通信をサポートする相互接続インターフェースと、
前記プロセッサ・コアに結合され、入出力通信リンクを介して入出力通信をサポートする外部通信アダプタの第１の部分とを含む、第１の集積回路と、
前記第１の集積回路のピンに接続された第２の集積回路とを備え、前記外部通信アダプタが前記第２の集積回路内に実施される第２の部分を含むシステム。
（１５）前記外部通信アダプタの前記第１の部分が、前記プロセッサ・コアに結合され、プロセッサ・コアによるＩ／Ｏコマンドに応答して入出力（Ｉ／Ｏ）データ転送を制御するデータ転送ロジックを備え、
前記外部通信アダプタの前記第２の部分が、リンク・レイヤ制御装置を備える、上記（１４）に記載のシステム。
（１６）プロセッサ・コアを含む集積回路を動作させる方法であって、
前記集積回路内の相互接続インターフェースを利用して、プロセッサ・コアと前記集積回路の外部のシステム相互接続との間での通信をサポートするステップと、
前記集積回路内の外部通信アダプタを利用して、入出力（Ｉ／Ｏ）通信リンクを介して入出力通信をサポートするステップであって、集積回路内の通信だけを利用して、前記プロセッサ・コアから前記外部通信アダプタにＩ／Ｏ通信コマンドを伝送することを含むステップと
を含む方法。
（１７）前記プロセッサ・コアがアクセスする可能性が高いデータおよび命令を前記集積回路内のキャッシュ階層構造にキャッシュするステップと、
前記集積回路内の集積化メモリ制御装置を利用して外部メモリにアクセスするステップと
をさらに含む、上記（１６）に記載の方法。
（１８）前記集積回路が、第１および第２の外部通信アダプタの少なくとも一部分を含み、前記Ｉ／Ｏ通信をサポートするステップが、
前記集積回路内の前記第１の外部通信アダプタを利用して、第１のリンク・レイヤ・プロトコルを使用するＩ／Ｏ通信をサポートするステップと、
前記集積回路内の前記第２の外部通信アダプタを利用して、異なる第２のリンク・レイヤ・プロトコルを使用するＩ／Ｏ通信をサポートするステップと

を含む、上記（１６）に記載の方法。
（１９）前記通信要求で指定されるアドレスに基づいて、前記集積回路内の通信コマンドを前記相互接続インターフェースおよび前記外部通信アダプタに経路指定するステップをさらに含む、上記（１６）に記載の方法。
（２０）Ｉ／Ｏ通信をサポートする前記ステップが、前記Ｉ／Ｏコマンド内で指定される実効アドレスを、メモリ位置を識別するために利用される実アドレスに変換するステップをさらに含む、上記（１６）に記載の方法。
（２１）Ｉ／Ｏ通信をサポートする前記ステップが、前記外部通信アダプタ内の入出力（Ｉ／Ｏ）メモリ制御装置を利用して、Ｉ／Ｏデータにアクセスするためにメモリにアクセスするステップを含む、上記（１６）に記載の方法。
（２２）Ｉ／Ｏ通信をサポートする前記ステップが、Ｉ／Ｏデータ転送の完了を示すソフトウェア・アクセス可能な標識をセットするステップを含む、上記（１６）に記載の方法。
（２３）前記Ｉ／Ｏ通信をサポートするステップが、着信Ｉ／Ｏデータと発信Ｉ／Ｏデータの少なくとも一方を前記集積回路内にバッファするステップを含む、上記（１６）に記載の方法。

【図面の簡単な説明】

【００８７】

【図１】従来技術による対称型マルチプロセッサ（ＳＭＰ）データ処理システムを示す図である。
【図２】本発明を有利に利用することができるネットワーク・システムの一例を示す図である。
【図３】本発明によるマルチプロセッサ（ＭＰ）データ処理システムの実施形態の一例を示すブロック図である。
【図４】図３のデータ処理システム内の処理装置のより詳細なブロック図である。
【図５】本発明の好ましい一実施形態による、図３に示されたＭＰデータ処理システム内のシステム・メモリのＩ／Ｏデータ構造およびその他の内容を示すブロック図である。
【図６】図３のＭＰデータ処理システム内で実行されるソフトウェアの一例を示すレイヤ図である。
【図７】本発明によるＩ／Ｏ通信の方法の一例を示す高レベル論理フローチャートである。
【図８】本発明の好ましい一実施形態によるプロセッサ・コアのブロック図である。
【図９】本発明の好ましい一実施形態によるバイパスＣＡＭのより詳細な図である。
【図１０】本発明による反復的コード・シーケンスの実行をバイパスする方法の一例を示す高レベルの論理フローチャートである。

【符号の説明】

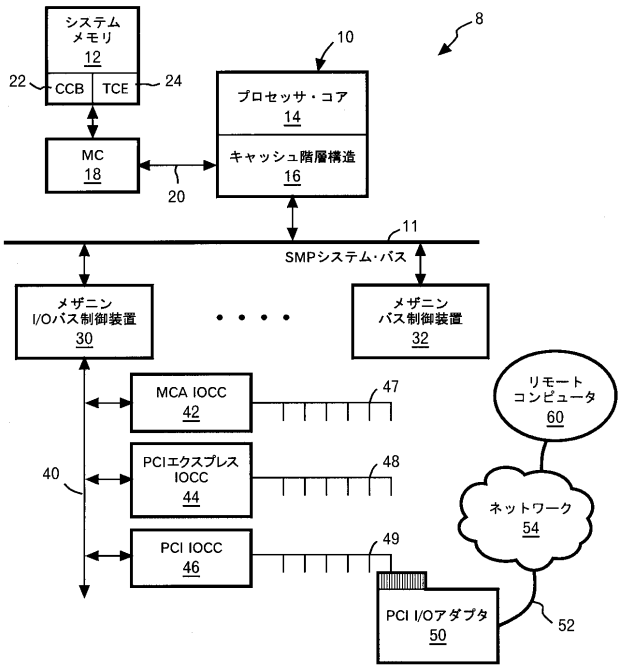
【００８８】

８　対称型マルチプロセッサ（ＳＭＰ）データ処理システム
１０　処理装置
１１　ＳＭＰシステム・バス
１２　外部システム・メモリ
１４　プロセッサ・コア
１６　キャッシュ階層構造
１８　メモリ制御装置
２０　専用メモリ・バス
２２　コマンド制御ブロック（ＣＣＢ）
２４　変換制御エントリ（ＴＣＥ）テーブル
３０　メザニンＩ／Ｏバス制御装置
３２　メザニン・バス制御装置
４０　メザニン・バス

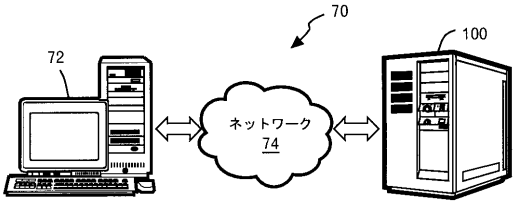
４２　マイクロチャネル・アーキテクチャＩＯＣＣ
４４　ＰＣＩエクスプレス（３ＧＩＯ）ＩＯＣＣ
４６　ＰＣＩ　ＩＯＣＣ
４７～４９　バス
５０　周辺装置相互接続（ＰＣＩ）Ｉ／Ｏアダプタ
５２　イーサネット（Ｒ）通信リンク、Ｉ／Ｏ通信リンク
５４　ネットワーク
６０　リモート・コンピュータ
７０　ネットワーク・システム
７２　ワークステーション・コンピュータ・システム
７４　ネットワーク
１００　サーバ・コンピュータ・システム
１０２　処理装置
１０４　メモリ
１０６　スイッチング機構
１０８　プロセッサ・コア
１１０　キャッシュ階層構造
１１２　集積化メモリ制御装置（ＩＭＣ）
１１４　集積化機構インターフェース（ＩＦＩ）
１１８　メモリ・バス
１２２　メモリ・マップ（ＭＭ）
１２４　変換索引バッファ（ＴＬＢ）
１３０　外部通信アダプタ（ＥＣＡ）
１３１　Ｉ／Ｏメモリ制御装置（Ｉ／Ｏ　ＭＣ）
１３２　バッファ記憶装置
１３３　データ転送ロジック（ＤＴＬ）
１３４　プロトコル・ロジック
１３５　データ・キュー
１３６　エントリ
１３８　リンク・レイヤ制御装置（ＬＬＣ）
１４０　並直列変換器／直並列変換器（ＳＥＲ／ＤＥＳ）
１５０　Ｉ／Ｏ通信リンク
１５８　ソフトウェア、ソフトウェア構成
１６０　システム・スーパバイザ（ハイパーバイザ）
１６２　オペレーティング・システム
１６３　ミドルウェア
１６４　ユーザ・レベル・プロセス
１６４　アプリケーション・プログラム
２４９　プロセッサ領域
２５０　Ｉ／Ｏ領域
２５２　共用領域
２５３　データ転送制御ブロック（ＤＴＣＢ）
２５４　仮想キュー
２５５　Ｉ／Ｏデータ・バッファ
２５６　制御状態バッファ
２６０　命令トレース・ログ
２６２　Ｉ／Ｏデータ
２６４　ページ・テーブル
２７０　命令シーケンス・ユニット（ＩＳＵ）
２７１　命令アドレス・レジスタ（ＩＡＲ）

２７２　命令メモリ管理ユニット（ＩＭＭＵ）
２７３　モード・セレクタ
２７４　Ｌ１　Ｉ－キャッシュ
２７６　Ｉ－キャッシュ再ロード・バス
２８０　命令バス
２８１　トレース・バス
２８２　条件レジスタ・ユニット（ＣＲＵ）
２８４　分岐実行ユニット（ＢＥＵ）
２８６　固定小数点ユニット（ＦＸＵ）
２８８　ロード・ストア・ユニット（ＬＳＵ）
２９０　浮動小数点ユニット（ＦＰＵ）
３００　制御レジスタ・ファイル（ＣＲＦ）
３０２　汎用レジスタ・ファイル（ＧＰＲＦ）
３０４　浮動小数点レジスタ・ファイル（ＦＰＲＦ）
３０８　Ｌ１　Ｄ－キャッシュ
３２０　命令バイパス回路
３２２　取込みロジック
３２４　連想記憶装置（ＣＡＭ）
３４０　命令ストリーム・バッファ
３４１　スヌープ・キル・フィールド
３４２　命令アドレス・フィールド
３４３　ユーザ・レベル設計状態ＣＡＭ
３４４　使用フラグ
３４５　レジスタ値フィールド
３４６　メモリ・マップ・アクセスＣＡＭ
３４７　スヌープ・キル・フィールド
３４８　目標アドレス・フィールド
３４９　ロード／ストア（Ｌ／Ｓ）フィールド
３５０　Ｉ／Ｏフィールド
３５２　データ・フィールド

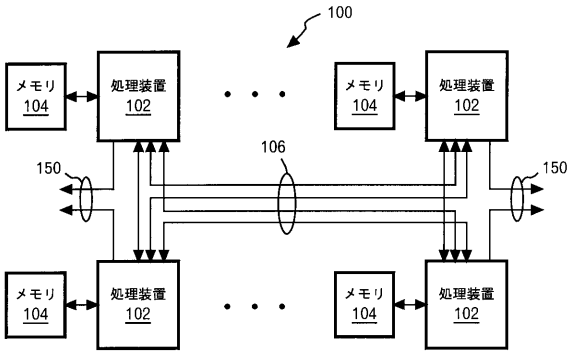
【図１】



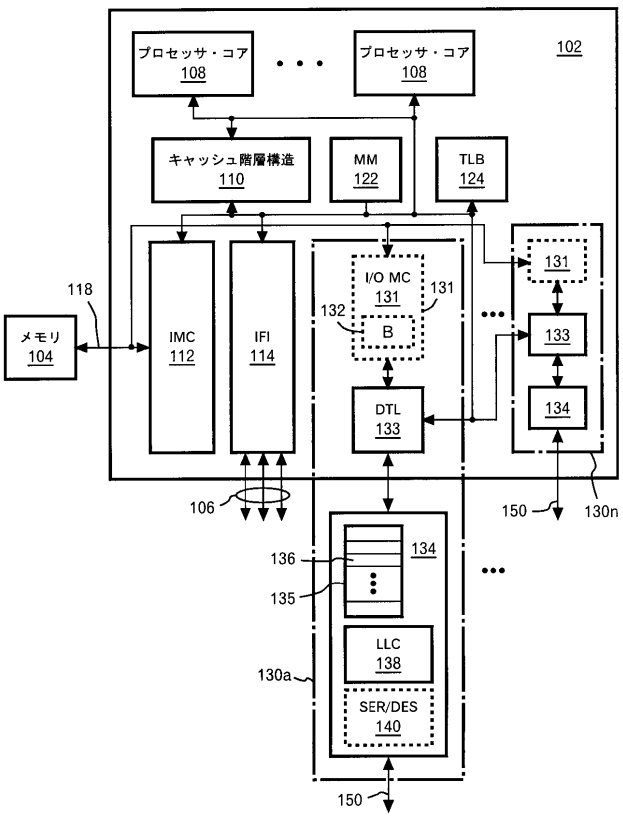
【図２】



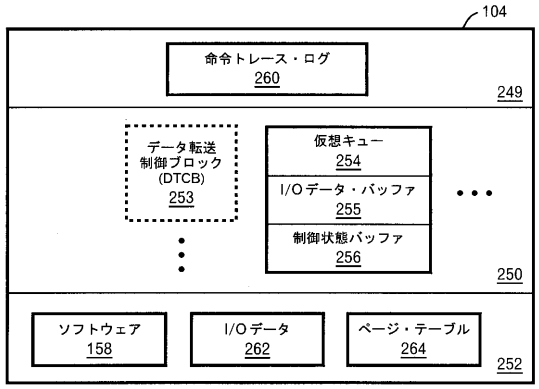
【図３】



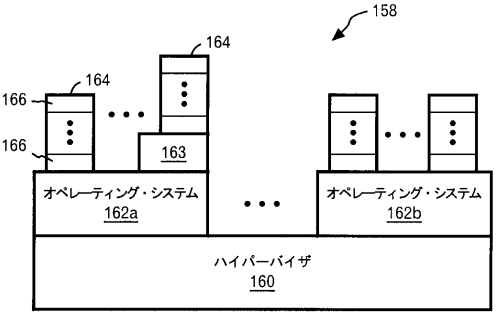
【図４】



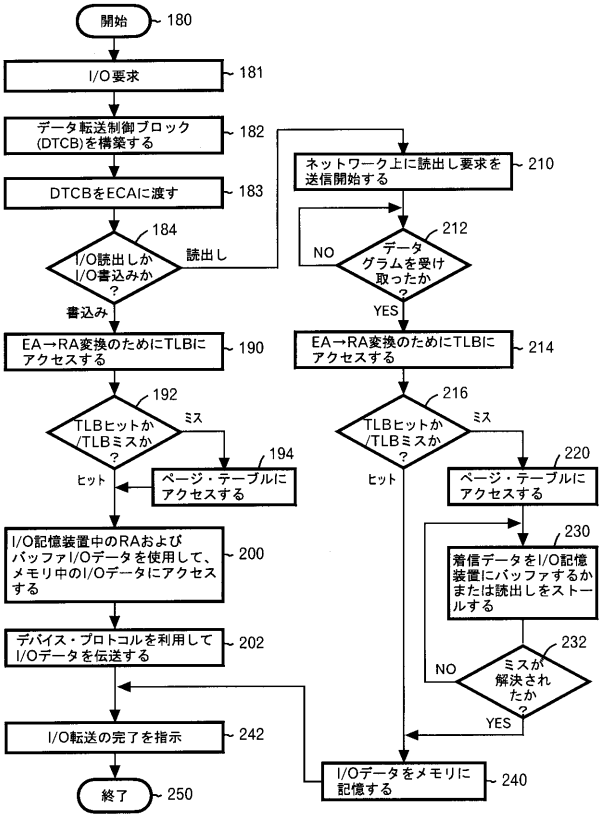
【図５】



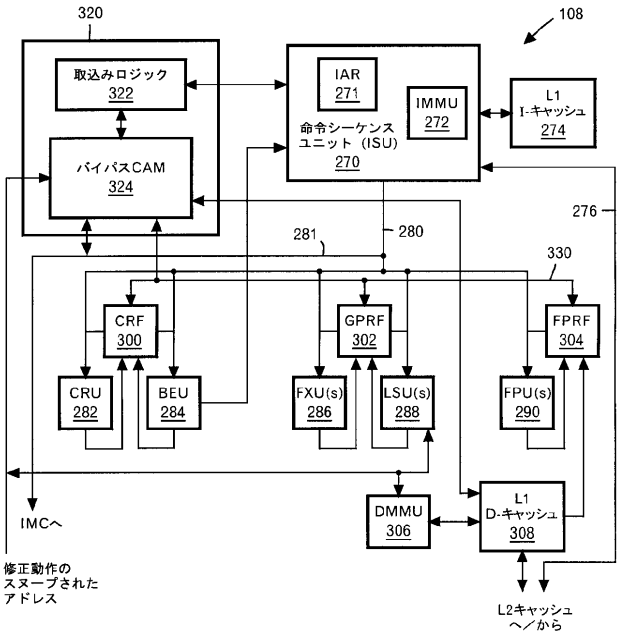
【図６】



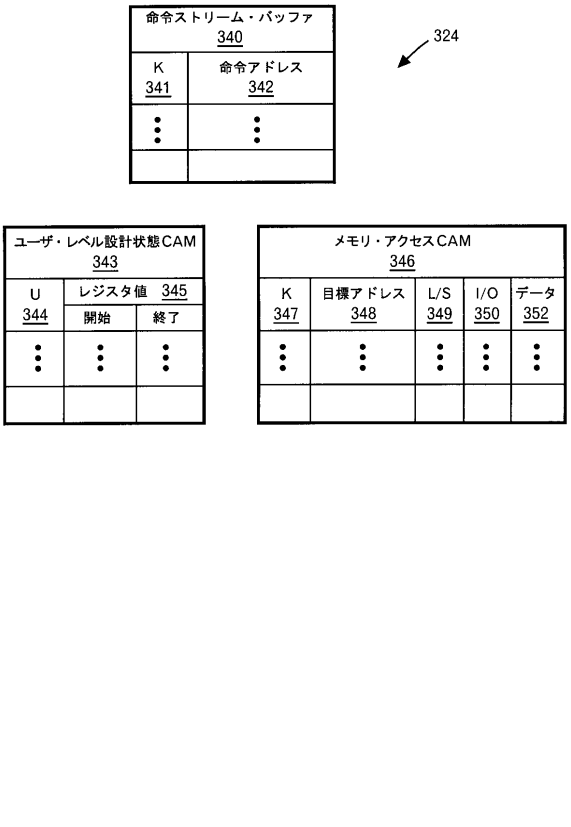
【図７】


開始
180
I/O要求
181
データ転送制御ブロック(DTCB)を構築する
182
DTCBをECAに渡す
183
I/O読出しかI/O書込みか？
184
読出し
書込み
EA→RA変換のためにTLBにアクセスする
190
TLBヒットか/TLBミスか？
192
ミス
ヒット
ページ・テーブルにアクセスする
194
I/O記憶装置中のRAおよびバッファI/Oデータを使用して、メモリ中のI/Oデータにアクセスする
200
デバイス・プロトコルを利用してI/Oデータを伝送する
202
I/O転送の完了を指示
242
終了
250
ネットワーク上に読出し要求を送信開始する
210
データグラムを受け取ったか？
212
NO
YES
EA→RA変換のためにTLBにアクセスする
214
TLBヒットか/TLBミスか？
216
ミス
ヒット
ページ・テーブルにアクセスする
220
着信データをI/O記憶装置にバッファするかまたは読出しをストールする
230
ミスが解決されたか？
232
NO
YES
I/Oデータをメモリに記憶する
240

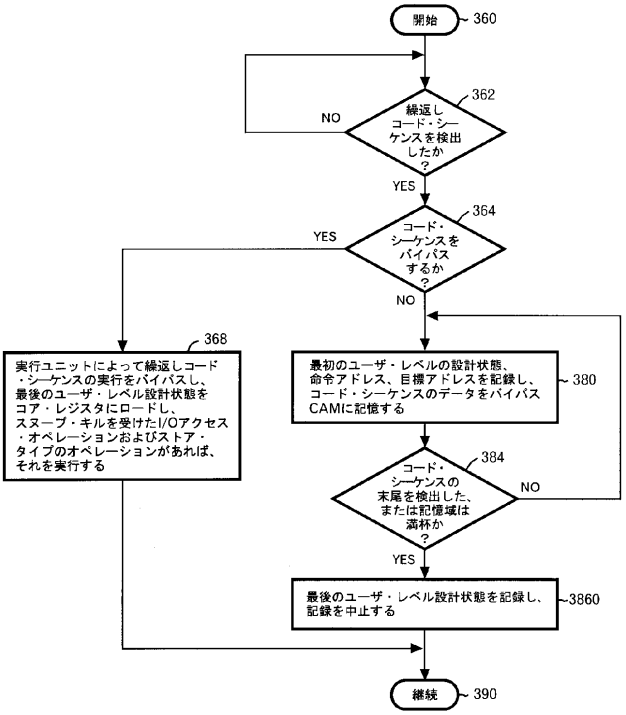
【図８】


320
108
取込みロジック
322
バイパスCAM
324
IAR
271
IMMU
272
命令シーケンスユニット（ISU）
270
L1
I-キャッシュ
274
276
281
280
330
CRF
300
GPRF
302
FPRF
304
CRU
282
BEU
284
FXU(s)
286
LSU(s)
288
FPU(s)
290
DMMU
306
L1
D-キャッシュ
308
IMCへ
修正動作のスヌープされたアドレス
L2キャッシュへ／から

【図９】


命令ストリーム・バッファ
340
324
K
341
命令アドレス
342
ユーザ・レベル設計状態CAM
343
U
344
レジスタ値
345
開始
終了
メモリ・アクセスCAM
346
K
347
目標アドレス
348
L/S
349
I/O
350
データ
352

【図１０】


開始
360
繰返しコード・シーケンスを検出したか？
362
NO
YES
コード・シーケンスをバイパスするか？
364
YES
NO
実行ユニットによって繰返しコード・シーケンスの実行をバイパスし、最後のユーザ・レベル設計状態をコア・レジスタにロードし、スヌープ・キルを受けたI/Oアクセス・オペレーションおよびストア・タイプのオペレーションがあれば、それを実行する
368
最初のユーザ・レベルの設計状態、命令アドレス、目標アドレスを記録し、コード・シーケンスのデータをバイパスCAMに記憶する
380
コード・シーケンスの末尾を検出した、または記憶域は満杯か？
384
NO
YES
最後のユーザ・レベル設計状態を記録し、記録を中止する
3860
継続
390

フロントページの続き

(51)Int.Cl.[7]	FI		テーマコード（参考）
	G06F 12/10	501Z	
	G06F 12/10	555	
	G06F 12/10	557	

(72)発明者 ラビ・クマール・アーミミリ
アメリカ合衆国78759 テキサス州オースティン スパイスブラッシュ・ドライブ 9221
(72)発明者 ロバート・アラン・カーグノニ
アメリカ合衆国78746 テキサス州オースティン キー・ウェスト・コーブ 1904
(72)発明者 ガイ・リン・ガスリー
アメリカ合衆国78726 テキサス州オースティン カラバール・ドライブ 11145
(72)発明者 ウィリアム・ジョン・スターク
アメリカ合衆国78681 テキサス州ラウンド・ロック シー・アッシュ・サークル 8612

Fターム(参考) 5B005 JJ12 KK16 MM01 MM51 NN12 PP03 PP05 PP21 RR01 UU15
5B014 GC31 HB26
5B060 AB26 AC20 KA02 KA06