

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第3774538号  
(P3774538)

(45) 発行日 平成18年5月17日(2006.5.17)

(24) 登録日 平成18年2月24日(2006.2.24)

(51) Int. Cl. F I  
**G06F 9/38 (2006.01)** G O 6 F 9/38 3 7 0 C  
**G06F 9/50 (2006.01)** G O 6 F 9/46 4 6 5 Z

請求項の数 9 (全 48 頁)

(21) 出願番号	特願平9-84210	(73) 特許権者	590000879
(22) 出願日	平成9年4月2日(1997.4.2)		テキサス インスツルメンツ インコーポ レイテッド
(65) 公開番号	特開平10-83304		アメリカ合衆国テキサス州ダラス, ノース セントラルエクスプレスウェイ 135 00
(43) 公開日	平成10年3月31日(1998.3.31)		
審査請求日	平成16年4月2日(2004.4.2)	(74) 代理人	100066692
(31) 優先権主張番号	014734		弁理士 浅村 皓
(32) 優先日	平成8年4月2日(1996.4.2)	(74) 代理人	100072040
(33) 優先権主張国	米国 (US)		弁理士 浅村 肇
		(74) 代理人	100094673
			弁理士 林 拓三
		(74) 代理人	100091339
			弁理士 清水 邦明

最終頁に続く

(54) 【発明の名称】 パーソナルコンピュータ回路、コンピュータシステム、及びその動作方法

(57) 【特許請求の範囲】

【請求項1】

計算システムであって、  
 主中央処理装置(CPU)マイクロプロセッサ、  
 前記主CPUマイクロプロセッサとは異なる命令集合を有するデジタル信号処理装置(DSP)マイクロプロセッサ、  
 前記主CPUマイクロプロセッサと前記DSPマイクロプロセッサとに結合された記憶装置、

前記記憶装置内のファイルベースオペレーティングシステムであって、前記主CPUが他に係わって占有されている時間間隔中前記DSPが前記主CPUの動作を実行し、それによって前記計算システムの性能を高めるように配置され、前記計算システムはソフトウェアアプリケーションの少なくともソフトウェア部分を実行することができ、前記ファイルベースオペレーティングシステムは仮想メモリ内の何処でそのようなソフトウェア部分が開始し終了するかを示すハンドルを含む前記ファイルベースオペレーティングシステム

、  
 前記DSPマイクロプロセッサに主CPUマイクロプロセッサ機能を行わせるプログラムを含むDSPカーネルソフトウェアを備え、

何処に送り手ハンドルと受け手ハンドルとが存在するかを特定するために前記主CPUマイクロプロセッサが前記ファイルベースオペレーティングシステムを実行し、

10

20

前記主CPUマイクロプロセッサを制御して送り手ハンドルによって示される位置から受け手ハンドルによって示される位置へ情報を送るプログラムを前記ファイルベースオペレーティングシステムが含む、

前記DSPマイクロプロセッサに前記主CPUマイクロプロセッサの代わりに機能を実行させるために前記記憶装置内の何処に前記送り手ハンドルと前記受け手ハンドルとが存在するかに基づいて動作する前記DSPマイクロプロセッサを制御するプログラムを前記DSPカーネルソフトウェアが含む

計算システム。

【請求項2】

請求項1に記載の計算システムにおいて、前記DSPマイクロプロセッサと前記主CPUマイクロプロセッサとに結合されたビデオ集積回路を更に含み、前記記憶装置がディスクとダイナミックランダムアクセスメモリ(DRAM)を更に含み、前記DRAMが前記DSPマイクロプロセッサと、前記主CPUマイクロプロセッサと、前記ビデオ集積回路とに対して統一メモリアーキテクチャ内で結合される、計算システム。

10

【請求項3】

請求項1に記載の計算システムにおいて、前記計算システムのアプリケーションをサポートする少なくとも1つのアプリケーション装置を更に含み、前記アプリケーション装置が物理層のみに減少させられかつ前記DSPマイクロプロセッサに結合されたハードウェアを更に含み、その結果、前記DSPマイクロプロセッサが前記物理層によって媒介される信号を利用する前記アプリケーションの残りを仮想化しかつ遂行する、計算システム。

20

【請求項4】

請求項1に記載の計算システムにおいて、前記DSPマイクロプロセッサによる実行用に前記記憶装置内に前記DSPカーネルソフトウェアが存在する、計算システム。

【請求項5】

請求項4に記載の計算システムにおいて、前記DSPマイクロプロセッサに結合されたI/Oポートを更に有し、前記DSPカーネルソフトウェアが前記ファイルベースオペレーティングシステムと協働するDSPマイクロプロセッサ動作を制御し、前記主CPUマイクロプロセッサが占有されすぎて仮想ハードウェアを表す所与の機能を実行できない場合は、前記DSPマイクロプロセッサが前記機能を実行し、前記主CPUマイクロプロセッサと前記DSPマイクロプロセッサとの両方が自由である場合は、前記ファイルベースオペレーティングシステムによって決定されるに従って前記主CPUマイクロプロセッサ又は前記DSPマイクロプロセッサのどちらかが前記機能を実行するために選択され、それによって前記主CPUマイクロプロセッサ上か又は前記I/Oポート上かで動作する仮想ハードウェアを制御するプログラムを前記ファイルベースオペレーティングシステムが含む、計算システム。

30

【請求項6】

請求項4に記載の計算システムにおいて、前記DSPカーネルソフトウェアが前記ファイルベースオペレーティングシステムを用いて割り込みベースでDSPマイクロプロセッサ動作を制御し、所与の機能に対する優先権が実時間に計算され、かつ機能がダイナミックに実行され、かつ前記DSPカーネルソフトウェアが前記ファイルベースオペレーティングシステムを供給し、それによって実時間優先権を計算することができる、計算システム。

40

【請求項7】

請求項1に記載の計算システムにおいて、前記ファイルベースオペレーティングシステムが前記DSPマイクロプロセッサ用プログラム空間について、かつ更に前記主CPUマイクロプロセッサと前記DSPマイクロプロセッサとの共用メモリモデルとしての読み出しデータ送り手空間と書き込みデータ受け手空間とについて前記記憶装置に対するポインタアドレスを前記DSPマイクロプロセッサに供給するソフトウェアを含み、それによって前記配置が前記ファイルベースオペレーティングシステムを使用して前記DSPマイクロプロセッサを前記主CPUマイクロプロセッサに緊密に結合する、計算システム。

50

## 【請求項 8】

請求項 1 に記載の計算システムにおいて、前記ファイルベースオペレーティングシステムが、16ビットアプリケーション用にコールバック機能を通して、かつ32ビット用にセマフォを通してアプリケーションと通信する仮想デバイスドライバ(VxD)を制御するソフトウェアを含む、計算システム。

## 【請求項 9】

請求項 1 に記載の計算システムにおいて、前記ファイルベースオペレーティングシステムが前記DSPマイクロプロセッサと前記主CPUマイクロプロセッサとの間を通信するために使用される割り込みを仮想化するプログラムを含む、計算システム。

## 【発明の詳細な説明】

10

## 【0001】

## 【発明の属する技術分野】

本発明は、一般に改善されたパーソナルコンピュータ(以下、PCと称する)回路、コンピュータシステム、及びこれらの動作方法に関する。

## 【0002】

## 【発明が解決しようとする課題】

初期のコンピュータは、大きな空間を要し、室全体を占めた。時来、ミニコンピュータ及びデスクトップコンピュータが市場に出回ってきた。

## 【0003】

普及しているデスクトップコンピュータには、「Apple(商標)」(Motorola(商標)680x0マイクロプロセッサベース)及び「IBM(商標)-互換機」(Intel(商標)又はその他のx86マイクロプロセッサベース)等多種のものがあり、これらはPCとして知られ、オフィス及び家庭での使用に非常に普及してきている。また、いくつかのスーパースカラ及びSuperSPARC(商標)のような他の非常に高性能マイクロプロセッサに基づくワークステーションと呼ばれるハイエンド(high end)デスクトップコンピュータも導入されている。

20

## 【0004】

更に発展して、ノートブック寸法又はパームトップコンピュータが、携帯ユーザ応用のためにオプションに電池式を採っている。このようなノートブック及び小形コンピュータは、ミニチュア化、絶えずより高速、より高性能、より融通性及び充電間電池の長寿命化と云う相容れない目標を迫られる技術に挑戦している。また、ドッキングステーションと呼ばれるデスクトップ密閉容器はドッキングステーション内へはめ込まれる携帯用コンピュータを有し、かつこのような携帯用コンピュータ/ドッキングステーションシステムの改善が望まれる。しかしながら、これら全てのシステムは、一般に、中央処理装置(以下、CPUと称する)の選択がシステムの処理能力を決定しかつアドインカード(add-in-card)、すなわち、組込みカードがCPUに付加されて、モデム、マルチメディアのような特定応用又は機能を付加すると云う意味でCPU中心である。回路、集積回路デバイス、あらゆる型式のコンピュータシステムにおける改善、かつ、なかでも、いま上に挙げた挑戦に取り組む方法が、本明細書に説明するように望まれる。

30

## 【0005】

40

## 【課題を解決するための手段】

全体的に、かつ本発明の1つの形では、PCシステムは、主CPUマイクロプロセッサ、ファイルベースオペレーティングシステム、及びデジタル信号処理装置(以下、DSPと称する)マイクロプロセッサを含み、これらは主CPUが他に係わって占有されている時間間隔中DSPが主CPU動作を実行することができるように配置され、それによってこのPCシステムの帯域幅を広げる。このPCシステムは、多数のCPU及び(又は)多数のDSPを含むこともできる。

## 【0006】

本発明の他の形では、DSPマイクロプロセッサ、ビデオ/オーディオ制御論理回路、及び圧縮/伸張電子回路を含む組合わせビデオ/作像システムが、共に、DSPマイクロ

50

ロセッサ、主CPUマイクロプロセッサ、これら両マイクロプロセッサに結合されたメモリ管理回路、及びメモリ回路をメモリ管理路に結合するメモリ回路とローカルバスに結合され、それであるから、メモリ回路がDSP、主CPU、ビデオ/オーディオ制御論理回路、及び圧縮/伸張電子回路に対する統一メモリアーキテクチャとして働き、かつDSPがビデオ/オーディオ制御論理回路及び圧縮/伸張電子回路の両方に関する処理機能を遂行する。加えて、ビデオ/作像機能の部分を仮想化するソフトウェアがこのシステムに付加されることがある。

**【0007】**

本発明は、PCを総合的にソフト高度化するシステムを提供する。

**【0008】**

本発明は、アドインカードを通してアドイン特徴、すなわち、組込み特徴をまとめて取り扱うことによって性能を向上するシステムを提供する。

**【0009】**

本発明は、マザーボード実現のためにコア論理内の特徴を集積化する方法論を提供する。

**【0010】**

本発明は、CPUチップ上にシステムの全機能集積化する。

**【0011】**

本発明は、主CPU、DSP、及び実質的に物理層のみに減少したアプリケーションハードウェアの少なくとも1つの片を有し、それであるからそのDSPがこの物理層によって媒介される信号に関連したアプリケーションの残りを仮想化しかつ遂行するシステムを提供する。このシステムは、多数のアプリケーション及び層を採用することがある。DSPは、例えば、ローカルエリアネットワーク(以下、LANと称する)、ビデオコントローラ、画像圧縮/伸張電子回路、ファックス、及びモデムを仮想化する。

**【0012】**

本発明は、DSPコア、マスタ/スレーブバスインタフェース、このインタフェースに結合された先入れ先出し(以下、FIFOと称する)機能を含むメモリ回路、及びDSPコアに結合されたRAM機能を有する集積回路を提供する。

**【0013】**

本発明は、DSPコア、マスタ/スレーブバスインタフェース、及びこのインタフェースに結合されたFIFO機能を含むメモリ回路、及びこのメモリをDSPコアに結合する単一命令/多重データ制御回路を含む集積回路を提供する。

**【0014】**

本発明は、DSPコア、マスタ/スレーブバスインタフェースと翻訳回路を含むインタフェース回路、マスタスレーブバスインタフェースに結合されたFIFO機能と翻訳回路に結合されたRAM機能を含むメモリ回路を有し、前記DSPコアが前記メモリ回路とインタフェースに結合される集積回路を提供する。

**【0015】**

本発明は、第1バスインタフェース回路、ディスプレイインタフェースを有するディスプレイコントローラ回路、及び第2バスインタフェース回路を有し、前記ディスプレイコントローラ回路が前記第2バスインタフェース回路に結合されたバスマスタリング回路を含むビデオコントローラ集積回路を提供する。

**【0016】**

本発明は、次のようなソフトウェアシステムを提供する。すなわち、オペレーティングシステム、少なくとも1つのマルチメディアドライバ、x86オブジェクトコードアプリケーション、ハードウェアアプリケーションを仮想化する非x86DSPコードアプリケーション、及びDSPカーネルソフトウェアを有するソフトウェアシステムであって、このカーネルソフトウェアが、オペレーティングシステム及び(又は)マルチメディア実時間事象と関連して、実時間割り込み、及び(又は)直接メモリアクセス(以下、DMAと称する)仮想化及び(又は)DSPコア内でのマルチスレッドマルチタスキングオペレーションを遂行するためにDSPコア上で動作可能であり、及び(又は)DSPコアに、そう

10

20

30

40

50

でなければ×86によって遂行されるメモリトランザクション機能及び(又は)入出力(以下、I/Oと称する)トランザクション機能を遂行させる。

【0017】

本発明は、処理素子、前記処理素子に接続された相互接続回路、前記処理素子(又は前記相互接続回路)に接続されたメモリ電子回路、前記相互接続回路に接続された多重化バス、及び予め選択されたアプリケーション機能を遂行するために前記多重化バスに接続された仮想化回路を有す計算システムを提供する。

【0018】

本発明は、第1多重化バスインタフェース回路、第2多重化バスインタフェース回路、及び前記第1多重化バスインタフェース回路と第2多重化バスインタフェース回路に接続された少なくとも1つの信号処理素子を有する予め選択されたアプリケーション機能を遂行する仮想化回路を提供する。

10

【0019】

他の改善されたPC装置、システム、及びこれらの動作方法もまた特許請求の範囲で請求される。

【0020】

本発明は、添付図面と関連して次の詳細な説明を参照することによって理解される。

【0021】

これらの添付図面において、もし他に指示がなければ、同一機能を有する部品は同一符号で示してある。

20

【0022】

【発明の実施の形態】

図27を初めに参照すると、本発明の改善されたPC100のブロック図が示されている。図27で、マイクロプロセッサユニット(Micro Processor Unit、以下、MPUと称する)ブロック2702は、486(又はP5)CPU、又は×86CPUの他のいずれかの型式、ダイナミックRAM(以下、DRAMと称する)コントローラ回路、及び周辺デバイス相互接続(以下、PCIと称する)ブリッジ回路を含む。PC100は、マルチメディアアプリケーション能力を有する。

【0023】

ローカルCPUバス2706は、MPUブロック2702をDRAM2714に接続する。このDRAMは、好適には、少なくとも4Mの容量を有する。もっとも、多かれ少なかれメモリがこのような採用されるのは明白である。ローカルバス2706は、種々のCPUが同じバス上で使用されるようにCPUを独立させて維持するように設計される。

30

【0024】

PCIバス2710は、CPUブロック2702内のPCIブリッジを経由してブロック2702内のCPUに接続される。このようにして、PCI2710バスは、広い帯域幅バスを提供し、かつまた、そうでなければローカルバス2706を不利にロードするおそれのある多種の周辺デバイスに対してコネクタを提供する。

【0025】

周辺処理装置(以下、PPUと称する)2718は、PCIバス2710をISA/ATバス2734に接続するシステムブリッジとして働く。また、PCMCIAカード又はPCMCIAカードバス用PCMCIAブリッジ2726がPCIバスに接続する。PCIバスに接続されたネットワークブリッジ2730は、LANブリッジと同じように複雑でないものでよく、又は広域ネットワーク(以下、WANと称する)ブリッジ、無線周波数(以下、RFと称する)ブリッジ、非同時伝送モード(以下、ATMと称する)ブリッジ又は総合サービスデジタル網(以下、ISDNと称する)ブリッジであってよい。ブリッジのこれらの型式の各々は、PCIバス上に直接かなり多量のトラフィックを発生し、及びローカルバス2706上に間接にかなりのトラフィックを発生する。PC100は、ユーザ入力用にこれに接続されたキーボード及びマウス(図27には示されていない)、及びCPU出力のユーザ観察用ディスプレイ又はCRT(図27には示されていない)を有

40

50

する。

【0026】

図27において、バス2706、2710、及び2734、は、相互接続されていない。ISA/ATバス2734に接続されて、例えば、ゲーム2739、ハードディスク装置2742、プリンタ2746、ファックスデータモデム2750、電話代理応答機(DTADとも称する)2754、業務用オーディオブロック2758、及びマルチメディアPC用CD(コンパクトディスク)装置2762のような、しかしこれらに限定されないオプションアクセサリ周辺デバイスの集合が表示されている。これらのISAブロックの各々は、パートタイム機能を遂行するアクセサリ周辺デバイスであり、かつ多くのもの(星印を付されたもの)はDSPを含む。

10

【0027】

一般に、図27でPCIバス2710に接続された他のブロックは、ISA/ATバス2734に接続された全ての周辺デバイスとほぼ同じ位のコストがかかる。例えば、業務用オーディオシステム2758の総合システムコストは、10ドル未満、すなわち総合PCシステムコストの約0.5%未満のはずである。

【0028】

コスト低減の構図(roadmap)、将来の高度化へと、及び現存する(「遺産」)ソフトウェア目標との互換性は、いずれの改善されたPCシステムにとっても重要な目標である。本発明のシステム実施例は、システムレベルでの改善がこれらの目標を達成し得ることを確認しかつ証明する。特に、システム実施例は、PCマザーボードに含まれるそのいずれもが、好適には、低コスト、ソフトウェア無用、及び本質的に無視可能なマザーボードエリアしか占めないことを確認する。

20

【0029】

DOS及びWindows(商標)3.11との後方互換性(backward compatibility)が、本発明の「仮想」ハードウェアによって提供されると云える。仮想ハードウェアは、図27に示された全ての機能が固定されかつ専用され、及びたとえこれらの機能が分散されても更新可能でないことを意味する。ソフトウェアは、それらを仮想化するのを助援することができる。もし1チップが図27に示された機能の全てを実現するならば、このチップは非常にプログラマブルなものであろう。したがって、本発明の種々の実施例は、図27に示した全ての冗長なかつ相容れないハードウェアを組み合わせて図27に示された機能の全てを「仮想化」するのに十分にプログラマブルであるいくつかのチップを作成し、かつ、最終的に、これらの機能を遂行する単一チップを提供する。

30

【0030】

Windowsは、最近及び将来のPC用オペレーティングシステム(以下、OSと称する)であって、かつライブラリにダイナミックにリンクし、それであるからソフトウェアは実行時間前にコンパイルされかつリンクされなくてよく、例えば、ダイナミックリンクローディング(以下、DLLと称する)はラン時間までリンクされず、かつ仮想ソフトウェアを提供する。これが、ソフトウェア寸法を縮小する。

【0031】

一般に、従来のCPUシステムは、種々のアプリケーションを実現するために固定CPUハードウェアを使用するが、しかしそのCPUはATバス又は図27に示された全てのI/Oポートに係わることでがきない。本明細書に後に説明される本発明の一実施例では、ハードウェアがこれを正にできるようにする。

40

【0032】

マルチメディアアプリケーション用に、システムブリッジ2718内のDMA2719及び割り込みコントローラ2720が、固定ハードウェア及び固定機能を有する。本発明の一実施例では、DMA及び割り込み制御は仮想化される。マルチメディアをサービスするに当たって2つの鍵になる挑戦が、マルチメディアデータ用広帯域幅に及び実時間割り込みをサービスする能力に必要である。これらの挑戦は、障害、及び実時間事象をサービス

50

する能力に制約又は限界を持ち込む割り込み可能性を生じることがある。これの全てを、本発明の実施例では、ハードウェアを仮想化して、例えば、ハードウェアを多数の個性を帯びるようにプログラマブルにすることによって克服する。ハードウェアが割り込みハンドラ ( handler ) の個性を帯びるとき、それはCPUの部分又はそうでなければ実時間能力のあるCPU及びOSの拡張であって、コプロセッサ又は付属装置ではない。

#### 【0033】

仮想ハードウェアは、同時にCPU内とOS内とに係わる移動性を有する。これを物理的に達成する方法は知られていないので、本発明の教示に従って仮想化が実現される。プログラマブルであるDSP解決及びコアは、仮想ハードウェアの基礎として働き、これは、本明細書に説明されるようにカスタム化されかつ改善される。

10

#### 【0034】

DSPコアは、本発明の実施例に必要な種々の仮想化をランするために高度に有利な基礎である。しかしながら、DSPはメモリ周辺デバイス及びローカルDSPバスを利用し、これが更に加わる挑戦を導入する。本発明の教示を使用すると、正反対に異なるDSPが、Pentium ( R ) のような、x86CPUのように「見える」ように作られ、又はこの逆にも作られる。

#### 【0035】

ソフトウェアリンクが、DSPとその周辺デバイスをx86CPUに緊密に結合するために使用され、かつ、これを行うのに、例えば、Windows 95のような、しかしこれに限定されない、普及したOSを使用する。このソフトウェアリンクは、DSPとその周辺デバイスが同じチップ上にある場合よりもこれら2つを緊密に結合する。それゆえ、ことごとくがソフトウェアに作られ、ハードウェアとソフトウェアとの間の線がはっきりしておらず、或る著しい利点を達成する。結局、電話線によって又は制御情報転送によって遠隔的に高度化可能なソフトウェアを有する単一チップが得られる。

20

#### 【0036】

図19aは、Windows 3.11内で支配的な層構造アーキテクチャを示す。アプリケーションブロック1902は、音声アプリケーションのような、マルチメディアアプリケーションである。ブロック1902の下の普通ブロックは、独立系ソフトウェアベンダー ( 以下、ISV ) にトランスペアレントである。Windowsアプリケーションは、クライアント/サーバモデルを使用し、この場合アプリケーションはクライアントであり、かつサーバはそのリクエストを取り扱うものは何でもそうである。そのリクエストの発する方 ( issue ) が、多くの場合、正しいサーバが見つけている。ブロック1906は、ブロック1902の下の第1層MMSYSTEM ( マルチメディアシステム ) であり、マルチメディアハードウェアと通信し、かつ音声 . WAV ファイルを再生する。Windows . WAV ドライバブロック1910は、このシステム内の第2層又は第1ドライバ層であり、アプリケーション1902からのリクエストを取り扱い、かつ圧縮を伴わず . WAV フォーマットで信号を再生する。CPU環境内の最下層は、DSPドライバ1914であり、これはハードウェア適合層 ( 以下、HAL と称する ) 内のハードウェアを仮想化する仮想デバイスドライバである。オーディオ圧縮マネージャ ( 以下、AVM と称する ) ドライバ1918は、システム内に圧縮/伸張機能性を提供し、かつブロック1914と通信する。これは、CPU 2702 側からの画像マッピングを完全にする。

30

40

#### 【0037】

これは、クライアント/サーバアーキテクチャへプラグインする方法をWindowsアーキテクチャ内に提供する。それゆえ、Windowsを通してDSPサーバ又はオーディオ圧縮サーバをプラグインすることができ、それらの機構はどれも同じである。図19aは、所望のどれかのアーキテクチャをWindowsにプラグインする方法を示す。また、これは、後方互換性に対する鍵であると共に、将来への優れた ( bland ) 新たな非侵入性の ( nonintrusive ) 構図を確立する。

#### 【0038】

次に、図19bで、仮想ハードウェア環境1922が、既に利用可能な最大第3者設置ソ

50

ソフトウェアベース（例えば、モデム、オーディオ機器等）に適合する特別DSPカーネルソフトウェアを備えたTI TMS320C5x DSPで以て、しかしこれに限定されることなく、実現されている。ことごとくがプリエンティブ割り込み（preemptive）ベースでランし、及び優先権が実時間に計算されかつダイナミックに実行される。しかしながら、OSが実時間に優先権を計算することができるように或る何かがOSを供給しなければならない。その或る何かが実時間カーネルソフトウェア1922である。これはWindows OSを補完し、かつこれと相容れないようなことはなく、これを拡張するように働く。これによって、マルチスレッドかつマルチタスキングシステムが提供される。

【0039】

次に、オーディオコンバタブロック1926がDSP側上でオーディオ圧縮/伸張を遂行するのに対して、ACMドライバブロック1918がCPU側で同じ機能を果たすこともできる。しかしながら、もしそのCPUが占有されており、この機構を遂行する時間を持たないならば、DSPがそれを取り扱うことがある。もしCPUもDSPも自由ならば、CPUとDSPの間の線をはっきりさせないソフトウェアの能力のために、これらのいずれか又は両方がこの機能を行うことができる。それゆえ、全て、OSを通しての有利な結果として、CPUはDSPを体現でき、かつDSPはCPUを体現できる。従来、CPUはデータ移動を遂行するが、これはCPUが32又は64ビットチャンクで移動を遂行できるからである。しかしながら、本発明では、DSPに対するアプリケーション呼び出し内のアルゴリズム性の強い(algorithmic intensive)ブロックがそれを行う。このようにして、アプリケーションが加速されると云える。

【0040】

DSPコーデックドライバ1930は、ステレオコーデック1934に結合する。好適には、このシステム実施例の場合、コーデック更新を考慮に入れるために、DSPに対して1チップを外部ステレオコーデック1934と共に適当に使用する。

【0041】

図1は、本発明の改善されたPCシステム100のブロック図を示す。図1で、PCシステム100はCPU102を含み、このCPUはキャッシュ104及びホストブリッジ108に接続される。ホストブリッジ108は、主メモリバス106（ときには、CPUバス又はローカルバスと称する）に接続され、これによって主メモリ112に、かつまたPCIバス116に接続される。PCシステム100は、ユーザ入力用にこれに接続された典型的にキーボード及び（又は）マウス（図1には示されていない）、及びCPU出力のユーザ観察用ディスプレイ又はCRT（図1には示されていない）を有する。仮想DSP回路200は、PCIバス116に接続される。システム100用に低速から高速直列キャプチャポートを実現するP1394/USB（上側波帯）ブロック120は、PCIバス116に接続される。I/Oブロック124は、図1に示されており、PCIバス116に接続される。このI/Oブロック124は、ATM/LAN、ISDN又はRFリンクであってよい。DSPブロック200は、これらの種々の全てのI/Oシステムにオプションに接続する。

【0042】

DSP200は、ブロック124の上掲の機能又はアプリケーションを遂行するために必要なハードウェアを「仮想化」するDSP200の能力があるので、I/Oブロック124を包含する又は置換することができる。

【0043】

図1で、DSPブロック200に適当なモデムソフトウェアを備えることによって過去のモデムが仮想化されている。仮想化されることがある他の類似のアプリケーションは、スピーカホーン、スピーチ、DSVD(digital simultaneous voice and data modem)、T.120トランスポート層、VDSVD(video digital simultaneous voice and data)である。図1の「長円形ブロック」は、これらアプリケーションの或るものを示す。

10

20

30

40

50

## 【 0 0 4 4 】

他の仮想化可能アプリケーションには、ビデオ圧縮／伸張用ベクトル量子化（以下、VQと称する）、ビデオ圧縮／伸張用MPEG、ISDNに基づく室内会議H.320、ATN及びLANに基づくH.321、H.322、及び電話線によるテレビ会議がある。

## 【 0 0 4 5 】

更に他の仮想化可能アプリケーションには、3次元グラフィック表示アプリケーション、及び移送指向性オーディオ用3次元音声がある。

## 【 0 0 4 6 】

このようなアプリケーションの多くは、業務用オーディオ、ゲーム、ハードディスク装置、プリンタ、ファックスデータモデム、電話代理応答機(DTAD)、マルチメディア用CD装置、及びデータ／ファイル圧縮があり、後者はシステムトラフィック及びCPU負担を軽減するためにDSPによって実行され、またフォーマット変換、及びデジタルフィルタとデジタル変換がある。

10

## 【 0 0 4 7 】

図1に、P1394/USBブロック120を含めてあるが、これは国内市場はPCを表向き家庭器具として供給しているが、常時運転しているとは限らないと認めるためである。例えば、PCは、簡単にオン、オフスイッチされ、ユーザはこれにインタフェースしない。1つはUSB用、1つはP1394用に2つのケーブルがこのPCに接続され、かつRFインタフェースが同様に接続されることがある。ユーザは電話代理応答機を欲するとき、低コスト電話機を持つ、すなわち、その電話機はこれからPCのUSBジャックへ延びるUSBケーブルを持つ、また同様に通常の壁取り付け電話ジャックへの線を持つ。もしPCがその家の他の部屋内にあるならば、RFが使用されるか、又はその家がP1394に対して配線される。

20

## 【 0 0 4 8 】

セルラ電話からコストを借り出すために、ハードウェアをセルラ電話の設計から除去し、PCへ移す。それゆえ、セルラ電話は、もはや孤立したシステムでない。

## 【 0 0 4 9 】

それゆえ、各家庭アプリケーション毎に、現在知られているようにハードウェアをそのアプリケーションから除去して、このハードウェアを仮想化するPCへ移す。

## 【 0 0 5 0 】

I/Oブロック124のRFハードウェアも同様にその物理層だけになるように仮想化される。それゆえ、PCシステムは、真に多機能化かつマルチタスキングである。

30

## 【 0 0 5 1 】

図2は、本発明の改善されたPCシステム100の仮想DSPブロック200の更に詳細なブロック図を示す。図2で、ブロック200は、PCIバス116及びISAバス128に接続されている。他の実施例では、ブロック200がなおまたローカルバス106にも接続される。

## 【 0 0 5 2 】

図2で、PCIバス116はハードウェアインタフェース電子回路210に接続され、このインタフェースが第1層に接続され、この層が次にインタフェース電子回路214に接続され、この回路が次に2つのテキサスインスツルメンツTMS320C5x DSPコア218及び222に結合される。電子回路210は、スピーチトランスコーディング用PCISレーブインタフェース(Truespeech Slave)、Windowsベースモデムデータポンプ用PCISレーブインタフェース(WinModem Slave)、DMA散乱-収集機能性(scatter-gather)用PCISーパーバス・マスタインタフェース、I/Oポートに似たサウンドブラスタ(R,Creative Labs)、及びDMAインタフェースを含む。電子回路210及びPCIバス116によって、PCI/ISAブリッジがまたマルチメディア用に確立される。

40

## 【 0 0 5 3 】

ブロック200の心臓は、1つ以上のDSPコア218、222である。電子回路214

50

は F I F O R A M である、すなわち、電子回路 2 1 4 はハードウェア F I F O でなくて F I F O 機能を遂行する主メモリ 1 1 2、すなわち、R A M の片である。D S P コア 2 1 8、2 2 2 にとって R A M 1 1 2 は正規 R A M のように見えるのに対して、このインタフェースの P C I バス側にとって R A M 1 1 2 は F I F O のように見える。電子回路 2 1 4 内で沢山の翻訳が行われる。データ及びオペランドは、C P U 1 0 2 とブロック 2 0 0 との間に流れる。C P U 1 0 2 は、バイト又はビット単位で動作しない。たとえ C P U 1 0 2 が 3 2 又は 6 4 ビット単位で動作しても、それはキャッシュ線に基づくキャッシュ 1 0 4 及び主メモリ 1 1 2 を経由するバーストを通して動作する。このデータ流の非互換性が、翻訳に当たって多くのサイクルを消費する。

【 0 0 5 4 】

インタフェースハードウェア 2 1 4 は、3 2 又は 6 4 ビット幅データ路を取り、そのどれかの部分を抽出してバイトにし、次いで 1 6 ビット語を D S P コアに供給するように設計される。それゆえ、単一命令多重データ（以下、S I M D と称する）アーキテクチャに対する複雑命令セットコンピュータ（以下、C I S C と称する）又は縮小命令セットコンピュータ（以下、R I S C と称する）がインタフェース挑戦を解決するために実現される。

【 0 0 5 5 】

それゆえ、有利上、インタフェース電子回路 2 1 4 は、次の機能を行う。すなわち、1 ) F I F O 互換性バーストとして 6 4 又は 1 2 8 ビット幅バーストを流し込む。2 ) それらを予め選択された D S P フォーマットに翻訳する。3 ) D S P コアに対する R A M として作用する。

【 0 0 5 6 】

異なるクロックが異なるチップ上でランするので、D S P コア動作を P C I 動作から減結合することによってクロッキング挑戦をブロック 2 1 0 内で解決する。C P U 側からは、このシステムは、十分に P C I 動作と同期させられる。D S P 側からは、このシステムは P C I 動作と非同期である。このようにして多数待機状態を回避し、D S P 動作速度を上げても、C P U にインタフェースするとき苛酷な停止 ( g r i n d i n g h a l t ) に立ち至ることはない。

【 0 0 5 7 】

D M A トランザクションは、好適には、流れ - I / O に基づく。C P U は仮想空間内で動作するが、D M A は物理空間で働く。翻訳は、C P U 1 0 2 内の古い又は新しいハードウェアに頼ることなく、好適には、W i n d o w s 9 5 を使用するソフトウェア制御の下に D S P によって遂行される。D S P は、主 D M A のエンジンになる。他の実施例では、C P U 1 0 2 内の D M A エンジンを D S P 2 1 8 から減結合することができ、かつ D S P コア 2 1 8 及び D S P コア 2 2 2 が D M A エンジンと関連させられる。D S P は、C P U 1 0 2 又は W i n d o w s が望む方法で D M A を組み立てかつ実行する。例えば、1 0 0 M b y t e / s の D M A 処理能力が、現行技術の D M A コアを用いて容易に達成可能である。

【 0 0 5 8 】

ブロック 2 1 0 自体は、P C I エージェントであり、かつ動作中以外は、ハードウェア内の単一マスタ、マルチスレーブ R C I エージェントであると言う意味においてコストを低減している。ブロック 2 1 0 のバス支配能力によって、全てのスレーブが動作中従動させられるが、しかしアプリケーションによって P C I バスを支配するように呼び出されるとそれを行う能力を有する。ソフトウェアは、適当な構成データによってスレーブをマスタであるように呼び出す。この多重体現インターフェースは、C P U が他に係わって占有されていないならば C P U がポーリングを行う場合に、働く。もしバスマスタリングが要求されるが、しかし C P U が他に係わって占有されているならば、同じことを実現するためにブロック 2 1 0 内のマスタを通して D S P をバスマスタにならせる。それゆえ、同じアプリケーションのために、D S P は、C P U 1 0 2 自体の代わりにホストの資格で働く。

【 0 0 5 9 】

W i n d o w s のソフトウェアアーキテクチャは、W i n d o w s 下でファイルランザ

10

20

30

40

50

クションのように見えることごとくを要求する。ファイルランザクションは、メモリトランザクションであり、かつCPU又はDSPのどれがそれらを遂行しようとかまわない。CPUは一時に1つのことを遂行することしかできず、それでDSPがシステム休止時間に作業を行い、休止時間を満たすので、ハードウェアは必要ない。チェリーピック(cherry pick)するかつ休止時間に作業を行うこの能力は、CPUでは遂行することができないこの調査研究の重要な利点である。ビデオチップ、DSP、CPU、...等、これら全てのチップは、ことごとくのランザクションがメモリトランザクション又はファイルランザクション(又はこの目的では余り重要でないI/Oランザクション)でなければならないと云う同じ拘束によって統制される。全ては、OSの下で同等に扱われる。

10

## 【0060】

図3は、図1の改善されたコンピュータシステムの部分の一好適実施例の更に詳細な電気回路図(部分的に概略図、部分的にブロック図)である。図3は、図2のハードウェアインタフェース電子回路210及びインタフェース電子回路214を概略的に示す。特に、PCIマスタ/スレーブインタフェース回路304がPCIバス116に接続されていることが判る。PCIマスタ/スレーブインタフェース回路304は、マスタ動作及びスレーブ動作電子回路を含み、かつハードウェア層305に接続され、この層はPCI構成制御及び状態レジスタ電子回路306、PCI I/O空間レジスタ電子回路308、2重ポート読み出し書き込みFIFO電子回路310、及びDSP I/O空間レジスタ電子回路312を含む。ハードウェア層305は、次にインタフェース/コーデックDMA制御回路316に接続され、後者は次にDSP(図3に示されていない)に接続される。

20

## 【0061】

図4aは、図1の改善されたコンピュータシステムの部分の好適実施例の更に詳細な電気回路図(部分的に概略図、部分的にブロック図)である。図4aで、PCIバス116は、マスタ/スレーブインタフェース424へのゲートウェイを経由してマザーボード上又はアドインカード上の単一チップ420に接続される。図4aは、また、PCIバス116に接続される第2チップ460を示す。2つのチップ420、460は、本発明の教示による種々の機能性を区分する2つの異なる方法を示す。

## 【0062】

チップ420は、オンチップ加速バス343を有しかつ好適にはデスクトップPCである。汎用バスGPI401及びGPI402が、チップ420又は460のどちらか用に備わる(図4aには示されていない)。チップ420内で、IDSP論理ブロック428が図2のブロック200の全ての論理を少なくとも含む。同様に、チップ420のIDSP論理ブロック428が図2のブロック200の全ての論理を少なくとも含む。チップ460は、2つのズームビデオ(以下、ZVと称する)バスを含み、かつ好適には携帯用PC内に使用される。グラフィック/ビデオコントローラ432は、2次元、3次元グラフィック能力ばかりでなく、ビデオ捕獲/圧縮/伸張能力を有する。コントローラ432は、バスマスタであることを要するが、しかし慣例的には常にスレーブである。提案された新しい統一メモリアーキテクチャ(UMAと称する)は、バスマスタを有利にする。

30

## 【0063】

典型的又は従来のメモリサイクルは長過ぎるので、メモリも変化しなければならない。メモリは機能性の点では商品であり得ない。それでメモリは追加機能を要し、かつこれを新ブロックにインタフェースさせなければならない。本発明は、この追加機能性を有するメモリの利点を活かす。

40

## 【0064】

図4bで、PCIバス116が図4aのDSP/ビデオチップ420に接続されて、PCシステムを形成する。チップ420はメモリコントローラ484に接続され、このコントローラはCPU102との他の結合を有する。メモリコントローラ484及びCPU102は、更に、データバッファ488に結合され、これを制御し、かつこのバッファはローカルバス106上のUMAメモリアクセスする。データバッファ488を使用すること

50

によって、CPUは、また、メモリアクセスを制御するのではなくメモリアクセスをリクエストしなければならない。有利には、ビデオ/グラフィックブロック432の機能が、PCシステム内に合体されている。チップ420は、おそらくパッドバウンド(すなわち、多くのピンを有しかつ要求される全てのパッドに適合するのに必要なシリコンの全てを回路が占めない)であるので、ビデオ/グラフィック機能は有利な空間内に存在する。ここで初めに挙げた種々のアプリケーションの全ては、IDSP加速バスに持ち込まれる。

【0065】

UMAブロック492は、ローエンド(low-end)市場用拡張データ出力DRAM(Extended Data Output DRAM、以下、EDOと称する)を含む。EDOは、不変ピンカウントを有し、486レベルCPUを支持し、かつそのデータ出力は予充電を回避するためにより長く活性に維持される。他の適当なUMAは、キャッシュ線を満たすバースト能力を備える高速EDOである。高レベルUMAは、マルチバンクバーストEDOである。熟練作業員は、特定アプリケーションシステムに望まれる性能及び市場価格点考慮に従ってメモリ型式を選択する。

10

【0066】

UMAブロック492から、一実施例は、バス106上に最高11Gbyte/sの帯域幅の非圧縮出力を有する。これは、広帯域バス又は500MHz以上で動作するRAMバスのような高速バスのどちらかによって達成されると云える。図4bで、コントローラ432は、UMA492をアクセスしかつ非圧縮ビデオをディスプレイ436(図4bには示されていない)へ送る。広帯域メモリ及びPCIバス116を通してのメモリへのアクセスは、広帯域及び短待ち時間(例えば、2ms)を維持する。

20

【0067】

図4aの加速バス434は、コントローラ432の加速を考慮するために外部アクセス又は内部アクセスできるようにDSP428に接続される。同様に、図4bも外部又は内部アクセスをできるように加速バスを含む。

【0068】

図5は、ホスト依存性である非対称多重処理に対する本発明の改善されたコンピュータシステムの一実施例のブロック図を示す。図5で、P5プロセッサはCPUである。図5で、PC内のどのCPU/IDSPインテリジェンスも、これらをアプリケーションを支持するのに必要な量だけ本発明の仮想技術を介して一括集中することができ、そのアプリケーションが停止した後これを解散することができる。ユーザには、いかなるときにも所与のアプリケーションに要求される計算リソースの所与の量さえあればよい。

30

【0069】

どれか特定のアプリケーションの必要を満足する多数のIDSPが図5に示されている。これらのチップは、ISAバス及びPCIバスに適当にインターフェースするために異なって設計されている。いくつかの例が図5に示されている。IDSPブロックは、これらのバスに取り付けられて示されているが、これらをノース(north)ブリッジ又はサウス(south)ブリッジのどちらかに取り付けてもよく、及びそのブロックの機能内に集積化してもよい。

【0070】

図6aは、本発明の改善されたコンピュータシステムに対してスーパスカラ拡張を使用するCPUモデルの実施例の概略ブロック図を示す。図6bで、新チップが3つのパイプラインを有し、これらのパイプラインは2つの従来スーパスカラパイプライン及び第3のDSPオペレーションパイプラインである。すなわち、スーパスカラCISC/RISC及びDSP CPUアーキテクチャが単一チップ上にディスパッチされた3つのオペレーションを有する。CISC内にマイクロコード記憶がある。DSPコアはDSPハードウェアマイクロ記憶であり、これはDSPオペレーションをディスパッチする。DSPコアは、ユーザがWindowsアイコンをクリックオンし、次いで、DSPコードが実行のためにハードディスクから主メモリ、ローカルメモリへキャッシュされるまで、「空」でいられる。このようにして、x86CPUは、非標準オペレーション及びディスパッチDS

40

50

Pオペレーションを行わなくて済む。このような組み合わせアーキテクチャは、Windows 95と互換性である。

【0071】

図6bは、1つのこのようなメモリキャッシュ階層構造を示す。ディスク記憶装置2742は、図6bに定義されたスペクトルの1極にある。次は、主メモリ112である。外部単一アクセスメモリは、ISAバス、PCIバス上のどれか他のメモリ、又はチップ200内の又は図6bの第3レベルのPCシステム内の他の何処かにあるメモリ214である。第2レベルに、構成可能(configurable)DSPコア・アドオンメモリ内のように、単一アクセスオンチップメモリが備わっている。第1レベルに、テキサスインツルメントTMS320CxコアのB0、B1、B2メモリを有する2重アクセスオンチップDSPコアメモリが、例示として、しかしこれに限られる訳でなく、備わっている。

10

【0072】

これは、キャッシュ動作するブロック210及びキャッシュ動作するCPU、OSを備えるソフトウェアキャッシュである。それゆえ、DSPはソフトウェア内でキャッシュ動作サイクルを組み立て、かつ制御が階層構造を上へ過ぎる。

【0073】

例として、1つのV.34モデムは、モデム速度交渉がこのモデムにどれか他のモデムと話すことができるように要求するので、従来のようにコードとデータ間隔の64Kを使用する。75ms期間中、このモデムは他のモデムが何を話しているかを知り、次いで、交渉の結果として求められたそのモデムに相当する特定のアプリケーションを維持するためにコードの断片のみの間中スイッチを入れる。このようにして、先行技術に比較して必要とするコード及びデータは少なくて済む。

20

【0074】

132Mbyte/sの転送速度を持つことによって、多くのアプリケーションが、要求されるコードをメモリからそれを実行するDSPローカルメモリへ容易に移動させることができる。

【0075】

図7は、DSPとCPUを結合するためにいかに共用メモリモデルを使用するかを示す。Windowsのアーキテクチャに基づく共用メモリモデルは、その下にあるどんな対称ソフトウェアアーキテクチャ上にも非対称にDSPとCPUを緊密に結合する。

30

【0076】

図7で、Windowsは、仮想メモリ内の何処でソフトウェアが開始し、終了するかを告げるファイル引出し状のハンドルを呼び出すアーキテクチャを有する。それらのハンドルは、仮想メモリ空間内のメモリリソースの位置を示す機構を提供する。これらのハンドルによって、DSPはホストCPUが行うことができる。ホストは、アプリケーションが或るメモリ内容をマニピュレートすることを必要とすること、及び送り手(source)ハンドルが何処にあるか、また受け手(destination)ハンドルが何処にあるか発言する。内容をベースにするWindowsは、情報を受け手ハンドルによって定義された位置へ送る。1)何処に送り手ハンドルがあるか、2)何処に受け手ハンドルがあるかを知って、DSPは動作に入り、CPUに代わって転送を取り扱う。アプリケーションに従って、OSの仮想メモリマネージャが何処に物理アドレスがあるかをCPUに告げる。また、CPUは、物理アドレスの決定を助援するオンチップハードウェアを有する。DSPは、スーパーバスマスタ(例えば、このマスタは単なるバスマスタと異なりページ境界と交差することができる)としてのOSの仮想メモリマネージャに質問する。ハンドルを仮想アドレスとして取り扱うCPUと異なり、DSPはアプリケーションをランしている間、ハンドルをロックしてこれらを実際の物理アドレスにならせる。DSPロックは、HAL層内の主DSPドライバに書き込まれたユーティリティであり、OS仮想メモリマネージャ内の既存のユーティリティを活性化して物理アドレスを復帰させる。

40

【0077】

図8は、本発明の改善されたコンピュータシステムに対するマルチメディア拡張モデルの

50

実施例の概略線図である。特に、本発明のIDSPによって提供された実時間サービスが、また、マルチスレッドかつマルチタスキングであるWindows OSの割り込みスケジューラに実時間優先権を供給することが判る。図32及び33は、後に論じられるが、実時間性を与えるためにいかに割り込みを取り扱うかの詳細を示す。図35は、或る実時間サービスの詳細を示す。

**【0078】**

図9は、本発明の改善されたコンピュータシステムに対するシステム/キャッシュ/仮想メモリモデルの実施例の概略線図である。DSPコアは、データ及び命令を非常に効率的に往き、戻り移動させるキャッシュである。CPUは、データをハードディスクからこのキャッシュへ移動させる。本発明の改善されたPC用キャッシュ動作方式を説明する。

10

**【0079】**

図9で、ラン時間中、コードをハードディスク装置(HDDとも称する)から主メモリ112へ取り出す。ラン時間中、コードの断片を主メモリ112から図2のブロック214へ取り出す。ブロック210がDSP及びCPUと独立に、コードを更に図9のDSPメモリ918及びC5xメモリへ移動させる。キャッシュ動作方式は図6bに示されている。図2で、ブロック210はキャッシュ動作方式の部分を行う。図19bで、CPUの休止時間に相当するコードの断片を、いまやブロック300内のDSP側上のメモリを含むように拡張されているCPU内の仮想空間を通して、ページする。このようにして、OSを使用して仮想メモリを物理メモリに翻訳する。図9で、DARAMは2重アクセススタティックRAM、SARAMは単一アクセススタティックRAMである。

20

**【0080】**

図10は、本発明の改善されたコンピュータシステムに採用されることがあるMPEG再生フィルタグラフィックモデルの実施例の概略ブロック図である。図10でユーザがソフトウェアオブジェクトをクリックオンすると、Windowsは、ダイナミックにメモリを配分しかつオブジェクトリンキング埋込み(object linking embedding、以下、OLEと称する)をランする。MPEGは、層から層式の代わりにオブジェクトからオブジェクト式で働く。送り手は、伸張するために画像捕獲データを含むことがある。

**【0081】**

図11は、本発明の改善されたコンピュータシステムに対する仮想I/Oハードウェア-PCIDMA及びマルチメディア実時間割り込みハンドラモデルの実施例の概略ブロック図である。図11で、システムは、実時間に割り込みを遮る。DSP空間は、主メモリ112内に実際目的のために在駐する外部スタティックRAM(以下、SRAMと称する)である。それゆえ、送り手空間、受け手空間、及びDSP空間は、全て同じ空間内に包含される。このようにして、そうでなければWindowsシステムに配分することのできないリソースをダイナミックに得る。遺産コードは、システム100上で互換的にWindowsの下で有利にランする。アプリケーション(APPとも称する)は主メモリの部分を使用し、これらをPCIバス散乱-収集DMAコントローラを示されたブロックへ効果的に分解する。

30

**【0082】**

図12は、本発明の改善されたコンピュータシステムに対するフレーム内のCPU及びIDSPによる並列処理の概略ブロック図である。図12で、フレームが与えられると、ホストCPUは、処理を全て実行しかつ時間帯1204内の多数のスロットを独占しなければならない。有利上、DSPとCPUの組み合わせが2つの時間帯1210と1212を提供し、ここでDSPは時間帯1210内のフレームを処理し、かつ離散余弦変換器(以下、DCTと称する)がCPU時間帯1212内の2つの短いタイムスロットになる。

40

**【0083】**

それゆえ、比較的中位の性能のDSPが主要な画像処理を行う充分なリソースを有するが、これはこのDSPがその信号処理タスクに専用されるためである。LAN、モデム、及び他のアプリケーションは画像処理と時分割されるが、しかし図12の分析点と無関係な

50

他の時間間隔中においてである。

【 0 0 8 4 】

図 1 3 は、本発明の改善されたコンピュータシステムが採用することがある M P E G エンコーダ 1 3 0 0 の単純化ブロック図である。特に、入来ビデオ画像をブロック 1 3 0 1 内でリオーダ ( r e o r d e r ) し、次いで運動推定ブロック 1 3 0 3 に供給して画像のどのエリアが変化しているかを判定する。運動推定ブロック 1 3 0 3 からの出力を画像 / 記憶予測ブロック 1 3 0 5、出力マルチプレクサ 1 3 0 7、及び加算器 1 3 0 9 に供給する。画像 / 記憶予測ブロック 1 3 0 5 からの出力を加算器 1 3 1 1 及び加算器 1 3 0 9 に供給する。加算器 1 3 0 9 は D C T ブロック 1 3 1 3 の入力へ供給し、ブロック 1 3 1 3 は出力を量子化ブロック 1 3 1 5 へ供給する。量子化ブロック 1 3 1 5 は、出力を可変長エンコーダ 1 3 1 7 及び逆量子化ブロック 1 3 1 9 へ供給する。逆量子化ブロック 1 3 1 9 は、出力を逆 D C T ブロック 1 3 2 1 に供給し、後者は出力を加算器 1 3 1 1 に供給する。可変長エンコーダブロック 1 3 1 7 は、出力をマルチプレクサ 1 3 0 7 に供給し、このマルチプレクサは出力をバッファ 1 3 2 3 に供給し、このバッファはコード化ビデオデータを出力する。このようなエンコーダは、ハードウェア内又はソフトウェア内で実現される。

10

【 0 0 8 5 】

図 1 4 は、本発明の改善されたコンピュータシステムが採用することがある M P E G デコーダ 1 4 0 0 の単純化ブロック図である。入来コード化ビデオを入力バッファ 1 4 0 1 に記憶し、このバッファはデマルチプレクサ 1 4 0 3 へ出力し、このデマルチプレクサは画像記憶 / 予測ブロック 1 4 0 5 及び可変長デコーダ 1 4 0 9 へ出力する。画像記憶 / 予測ブロック 1 4 0 5 は加算器 1 4 0 7 へ出力する。可変長デコーダ 1 4 0 9 は逆量子化ブロック 1 4 1 1 へ出力し、このブロック 1 4 1 1 は量子化ブロックステップをデマルチプレクサ 1 4 0 3 から供給される。逆量子化ブロック 1 4 1 1 は逆 D C T ブロック 1 4 1 3 へ出力し、後者は加算器 1 4 0 7 へ出力する。加算器 1 4 0 7 は画像記憶 / 予測ブロック 1 4 0 5 及び画像リオーダブロック 1 4 1 5 へ出力し、ブロック 1 4 1 5 はデコードしたビデオ画像を出力する。

20

【 0 0 8 6 】

図 1 5 は、C P U として P 5 プロセッサを採用することがある本発明のノートブックコンピュータに対するビデオ解決の実施例の単純化ブロック図である。図 1 5 で、システムは、図 4 a に示されたのと同じように、P C M C I A 標準、改正 2 . 1 を用いて働く。

30

【 0 0 8 7 】

Z V は、エキストラフレームバッファを要せずフレームバッファ内に収容されるビデオ入力リソースに適合する。P C I は、等時的ではなく、バーストされかつ割り込まれ、したがってビデオには極めて不利である。本発明によれば、フレームバッファに接続された専用バックドア私設高速バス Z V によって等時性能が付けられる。ブロックを図 4 のチップ 4 2 0 又は 4 6 0 内に一括にリンクするのが、加速バスである。

【 0 0 8 8 】

図 1 6 は、C P U として P 5 プロセッサを採用することがある本発明のデスクトップコンピュータに対するビデオ解決の実施例の単純化ブロック図である。図 1 6 は、P C M C I A 又はカードバスを用いないデスクトップ P C を示す。それゆえ、M P E G ビデオを再生中、C P U はビデオデコード機能を遂行するのに対して、D S P はオーディオデコード機能を遂行するか、又はこれの反対である。もしデータが C D R O M から到来するのであるならば、システム同期は C P U によって遂行される。もしデータが外部カメラ 1 3 9 4 又は他の外部画像捕獲システムから来るならば、D S P がシステム同期を遂行するが、それはこのデータがまず D S P に到来するためである。このようにして、このシステム及び方法は、どのプロセッサがシステム同期を遂行するべきかの問題を有利に解決する。

40

【 0 0 8 9 】

図 1 7 は、本発明の改善されたコンピュータシステムに対するアプリケーションパイプラインの単純化ブロック図である。図 1 7 で、単一プロセッサは、たとえスーパースカラブ

50

ロセッサであっても、アプリケーションレベルでパイプラインすることはできない。これは、パイプラインプロセッサのハードウェアパイプラインと明確に異なる。意味する所は、アプリケーションをパイプラインするために同じ時間にパイプラインの異なる部分を実行すると云うことである。

【0090】

ステージが、アプリケーションの性質によって確立される。例えば、DSPがビデオのNフレームを伸張するのに対して、x86CPUが同時にフレームN-1についてプロセッサデータをスクリーンへ出力し、次いで、このサイクルをパイプラインに関係した仕方で繰り返す。DSPは、また、フィルタリング、スケーリング、及びカラー変換を遂行するのに対して、CPUはデータを入力又はデータを出力する。

10

【0091】

図18は、本発明による改善されたコンピュータシステムに対するハードウェアアーキテクチャの他の実施例のシステムレベルを示すブロック図である。図18は、図1の単純化ブロック図の代替配置を示す。図18は、PCIバスを使用しかつプラグインPCIカード1820用スロット1810を有するPDCを示す。プラグインPCIカード1820は、図3又は図4aに示されたように、PCIインタフェース(I/F)1822、DSP1824、及び関連コーデック1826を含む。その他はいずれも標準PCハードウェアである。ハードディスク装置1830は、標準PCハードウェアの部分である。コーデック1826は、オーディオ機能を取り扱う。

【0092】

PCIカード1820は、PCIマスタ及びPCIスレーブの両方である。ホストCPUは、スレーブモードでPCI構成レジスタ及び主メモリポインタレジスタにアクセスすることができる。マスタモードで、DSPは、散乱-収集DMAを使用してプログラムコード及びデータを取り出し、かつ主メモリに記憶することができる。

20

【0093】

図19bは、図1の改善されたコンピュータシステムの好適実施例の更に詳細な電気回路図(部分的に概略図、部分的にブロック図)である。図19bは、図3に類似しているが、しかし更にDSP1950及び関連メモリ1952、1954の詳細を示す。加えて、コーデックチップ1956ばかりでなく、チップ300(図3に示されている)、スピーカ1962、及びマイクロホン1960へのその接続を示す。図38から図44は、この実施例における一配置の詳細を示す。

30

【0094】

DSP1950は、CPUと独立にランする。CPUは、PCIカードがスレーブのとき、リセットの後初期化コードをDSPへダウンロードし、そこで、DSPは独立する。CPUは、PCI I/O空間レジスタ308内のSTARTビットをセットすることによってDSPを始動させることができる。STARTビットは、DSPにとって割り込みであり、DSPにコード実行を開始させる。DSPは、コード実行を終了するまでPCIマスタとして独立に実施する。DSPアルゴリズムが終了したとき、DONEビットがPCI状態語(Status Word)308内にセットされかつCPUへの割り込みがPCIバス上に発生される。

40

【0095】

CPUは、主メモリにポインタアドレスを供給する。必要な3つのアドレスは、プログラム空間についてのDSPの128Kバイトに対するベースアドレス、読み出し空間(送り手)についての128Kバイトに対するアドレス、及び書き込み空間(受け手)についての128Kバイトに対するアドレスである。アプリケーションに従って、これらはメモリの異なるエリアを指摘する、又はメモリの同じエリアを指摘する。CPUは、DSPに対するSTARTビットを有するレジスタ308に書き込むことによってDSPを制御する。DSPは、このビットを割り込みとして使用する。この割り込みは、DSPにロードされたアルゴリズムを開始させるか、又はホストがコマンドをDSPに送ったことを表示する。

50

## 【 0 0 9 6 】

レジスタ308内の他のビットは、DSPリセットビットである。このビットは、DSPのリセットピンへ送られる。このビットは、現行DSPタスクを打ち切らせて、DSPをブートロード順序で開始させる。新タスクをDSPにロードするために、DSPをリセットするか（これはブートロードするために追加の時間を要する）又はMASTER ABORTビットをホスト制御されるレジスタ308にセットするかのどちらかができる。MASTER ABORTビットがセットされると、割り込みがDSPに発生されて、これが初めにロードされたブートコードを実行させる。このブートコードは、RCI I/O空間メモリプログラム空間ポインタによって現在指摘されているホスト主メモリからDSPタスクをロードする。DSPは、PCI I/O空間CMD/STATUSレジスタ308内にEVENTビットをセットすることによってそのタスクが終了したことを表示し、かつPCI割り込みを発生する（タスクを完了する前にビットがまたセットされて、DSPが働いていることを表示する）。

10

## 【 0 0 9 7 】

PCIアプリケーションカードDSPアーキテクチャは、初期的には無メモリである。これは、コストを可能な限り低く押さえる必要のあるシステム製造業者の関心を引く。しかしながら、このアーキテクチャは、メモリ1952、1954で示されたような、外部プログラム空間メモリ又は外部データ空間メモリを追加することを妨げない。もしDSPアプリケーションソフトウェアが逐次8ビットアクセスを行う時間及びオーバーヘッドを提供できるならば、データメモリを実現するために16ビット2RAMチップメモリシステムの代わりに1つの8ビット幅外部RAMを使用してもよい。無メモリシステムを達成するために、DSPの内部メモリをプログラム空間とデータ空間とに分割することになる。もしプログラムが内部メモリに適合するには大き過ぎるならば、コードを必要に応じて主メモリからPCIバス上へ取り出さなければならない。生データが主メモリから取り出され、かつDSP動作の結果のデータを主メモリに帰されることになる。

20

## 【 0 0 9 8 】

外部メモリがない場合は、DSP内へのコードを実行化可能にするただ1つの方法は、DSPをリセットし、かつROMをベースにしたブートローダを使用して主メモリからの小さい初期化プログラムをロードすることである。DSPのブートローダ版（「BDSP」）を以下の説明に使用する。このDSPは、どの型式のブートローディングが起こるかを決定するために、リセットされたグローバルメモリアドレスFFFFを読み出す。4つの最下位ビット内の1100の値は、ハンドシェーキング用XF信号及びベーシックI/O（以下、BIOと称する）信号を使用する16ビット並列I/Oロードを表示する。BIOが低のとき、データバスが駆動される。ブートロードが完了した後、BIOはDSPデータバス上のバス衝突を防止するために高のままでなければならない。

30

## 【 0 0 9 9 】

16ビット語をI/Oブートモードを介してロードすることによって、CPUがベーシック主メモリ初期化プログラムをDSPに敏速にロードすることができる。DSPはリセット状態になり、それであるからDSPはスレーブとしてPCIバストランザクションに回答することができるだけである。PCIホストは、DSPにロードされている初期化プログラム値を制御することになる。これが、初期化プログラムをソフトウェアで修正できるようにする。このルーチンがロードされた後、DSPはリセットを解かれ、それであるからDSPは初期化プログラムを実行することができる（DSPのBIO信号をFIFO状態信号で駆動できるようにするためBIO選択ビットをまたセットしなければならない）。初期化プログラムは、PCIバスマスタとして働き、かつ主メモリに入って実際DSPアプリケーションソフトウェアを取り出す。初期化プログラムのみをブートロードすることによって、DSP PCIバスマスタが実際DSPアプリケーションコードのローディングを制御することができる。

40

## 【 0 1 0 0 】

もしDSPアプリケーションソフトウェアがDSP内部メモリ内に全体として適合できな

50

いならば、より多くのコードをスワップインする必要がある。このコードは、主メモリから取り出される。多くのコードを取り出すために、DSPアプリケーションはBLDP (Block Load Program to Data) 命令を遂行することがあり、これはグローバル (Global) 空間データを取り出し、それをプログラム空間へ移動させる。もしグローバルデータ (Global Data) 空間が外部メモリであると指定されかつプログラム空間が内部メモリであると指定されるならば、コードを移動させるために必要な外部アクセスをBDSPが行う。DSPは外部世界へのグローバルデータ空間取り出しを行いつつあるので、そのアプリケーションソフトウェアは、PCIインタフェースにDSPコードがバス上に取り出されつつあり、データではないことを告げる機構を必要とする。I/Oポートレジスタビットが、取り出されつつあるのはプログラム空間コードか、送り手データか、受け手データか、又はDSP決定したデータかどうかを表示するために使用されることになる。I/Oポートレジスタビットは、リセットされているプログラム空間をアクセスするようにセットされる。

10

#### 【0101】

アプリケーションコード内に持ち込まれて使用された最初にブートロードされたコードは、好適には、メモリ内に維持されているものとする。これは、CPUにこのアプリケーションをリロードするように割り込みを強制することによってDSP動作を打ち切らせる。もしこのコードが喪失するならば、CPUはDSPを制御しなくなり、かつプログラムコードをリロードするためにDSPをリセットしなければならない。DSPアプリケーションソフトウェアは、データを外部メモリから内部メモリへ転送しかつBLDD (Block Load Data to Data) コマンドによってこれを復帰させることができる。

20

#### 【0102】

PCIインタフェースは、PCIマスタ及びPCISレーブの両方である。このインタフェースは、CPUが構成空間にアクセスできるようにスレーブであることを要する。CPUは、DSPへの或る制御を実行しかつ主メモリポインタを組み立てるためにメモリ空間又はI/O空間のどちらかにアクセスすることができる。DSPは、コード及びデータについて主メモリをアクセスするためにPCIマスタであることになる。また、PCIインタフェースに含まれてFIFO310がある。このFIFOは、好適には、データの64バイトを保持する。BDSPは16ビット外部データバスを有するので、PCIバスとBDSPバスとの間のFIFOは、DSPプログラムとデータを32ビットから16ビットに変換しなければならない。各主メモリ場所に2つの命令又は2つのデータ語を詰めることによって、所与のアプリケーションに要求されるPCIバスの数を2分の1に減少することができる。

30

#### 【0103】

PCI標準によって要求される構成レジスタは、PCI I/O空間及びメモリアドレス空間内のエリアを0及び1で以て指定するために使用される。要求される限りのPCI空間は5つの倍語である。アドレス空間は2のべきで要求されるので、8アドレスを確保しなければならない。必要とされるこれら5つの倍語より上のレジスタは使用されず、もし読み出されるならば、0を返すことになる。これら5つの倍語は、プログラム空間ポインタ、データ空間ポインタ、DSPコマンド、DSP状態語、及びBDSPをブートロードするためのI/Oポートに使用される。

40

#### 【0104】

PCIホストは、PCI I/O空間レジスタ308を書き込む。DSPは、これらのレジスタによって制御される。構成レジスタ内のベースアドレス0及び1は、このチップが応答するPCI空間のエリアを指摘する。両ベースアドレスが同じ空間を指摘するが、しかしベースアドレス0はI/O空間として構成される。ベースアドレス1は、メモリ空間として構成される。これによって、メモリベースシステム又はI/Oベースシステムのいずれかを設計するシステム設計者にとって融通性が増すようになる。DSPと通信しかつこれを制御するために、20バイトが必要である。32バイトがPCI構成レジスタ内に

50

確保されている。レジスタ  $0 \times 64$  から  $0 \times 1F$  が確保されることになりかつ  $0$  だけを返す。これらのアドレスは、ベースアドレス +  $0 \times 0$  からベースアドレス +  $0 \times 13$  である。これらのレジスタによって、CPU が、主メモリのどの部分を DSP メモリに対して使用するべきかを決定できるばかりでなく、DSP の状態を開始、停止、及び監視できるようになる。主メモリプログラム（及びデータ）空間レジスタの下位 12 ビットが確保される。これによって、プログラム及びデータ空間が 4 K 境界内に再配置される。

#### 【0105】

DSP アルゴリズムが終了すると、状態語内の EVENT INTERRUPT ビットがセットされる。このビットがセットされると、割り込みが PCI バスを通してアプリケーションソフトウェアに対して発生される。この割り込みは DSP ソフトウェアによって制御されるので、割り込み許可 (enable) はハードウェア内で実現されない。

10

#### 【0106】

DSP ハードウェア制御語は、DSP ハードウェアの動作を直接制御する。その最下位 4 ビットは、リトライカウンタ (Retry Counter) ビットである。リトライカウンタビットは、もし PCI マクロ機能が PCI トランザクションを初期的に失敗するならば、この機能にそれをリトライさせる。この値は、上掲の機能にいつまでもリトライをさせることになる  $0000$  にリセットされる。この値を、1 から 15 リトライの間に変化させることができる。もしトランザクションが遂行されないためにリトライカウンタがランしなくなるならば、MISC CTRL レジスタ内のリトライカウンタ期限切れ (Retry Counter Expired) ビットがセットされる。この状態は、

20

#### 【0107】

DSP インタフェース ASIC は、PCI インタフェース、PCI FIFO、DSP (接続デバイス) インタフェース、及び CODEC インタフェースからなる。この調査研究は、BDSP が離散チップであることを必要とする。

#### 【0108】

DSP は、利用可能な 5 ユーザ割り込みを有する。これらの割り込みの 1 つは、マスク不可能であり、4 つはマスク可能である。CODEC インタフェースは、3 割り込みを必要とし、PCI インタフェースレジスタは 2 割り込み及び RESET ピンを必要とする。これらの割り込みは、次のように使用される。すなわち、NMI - マスタ打ち切り (PCI ホストから DSP へ)、INT1 - CDRQ (コーデックから)、INT2 - PDRQ (コーデックから)、INT3 - IRQ / IRQ2 (コーデック 1 とコーデック 2 から)、及び INT4 - コマンド (PCI ホストから DSP へ)。コマンド割り込みは最低優先権であるが、これは RESET ピンから出ると DSP アルゴリズムに応答するのとの間になされる十分なハウスキーピングがあるから問題はない。マスタ打ち切り割り込みは、最高優先権であり、DSP にその初めの DSP コードをリロードさせるのに使用される。割り込み 1、2、及び 3 は、コーデックによって発生される。もし第 2 コーデックがユーザによって必要とされるならば、IRQ 割り込みを共用することができる。IRQ 信号と IRQ2 信号が一緒にゲートされて DSP に対する 1 割り込みを生じる。IRQ 信号、CDRQ 信号、及び PDRQ 信号は、ASIC によって反転される。ユーザによって供給される IRQ2 信号は、低で活性でなければならない。

30

40

#### 【0109】

DSP は、メモリマップ I/O ポートにアクセスし、このポートは動作マスタリング用 PCI マクロを制御するために使用される。

#### 【0110】

CODEC レジスタは、DSP によって PCI アプリケーションボード上の離散 CODEC と通信するために使用される。FIFO 状態は、I/O ポートで DSP によって利用可能である。この ASIC 又は将来の ASIC の改正によっては、FIFO が異なった寸法を持つこともある。性能を同調させるために、どんな寸法の FIFO と共に DSP が働かなくてはならないかを知ることは、DSP を助援するであろう。FIFO 寸法レジスタ

50

は、F I F Oが保持することのできる語の最大数を表示する。転送するD S P語の数は、他のI / Oポートに書き込まれる。この値は、転送するバイトの数を取得するためにハードウェアでは2倍される。転送することのできる語の最大数は、3 2 7 6 8 ( 6 5 5 3 6 バイト)である。これは、レジスタ0 x 5 9の最上位ビットを使用することができないことを意味する。もし1がそのビットに書き込まれるならば、それは無視される。

**【 0 1 1 1 】**

P C Iアドレスオフセット( 0 x 5 8 )はD S P空間内のアドレスであって、P C I空間内に転送されて、データをホストへ又はこれから転送するのに使用される。D S Pは、D S PポインタをP C I加算( A d d e r )ポインタレジスタに書き込むことによってホスト主メモリをアクセスし、P C I加算ポインタレジスタは2 D S P語を取り上げかつ3 2 ビットP C Iポインタを作る。これは、もしD S Pが散乱 - 収集テーブルにデータベースを有する主メモリ内の位置を計算することを必要とするならば、有効である。P C I制御レジスタ内のP C Iアドレス空間選択ビットは、P C Iバストランザクションが開始されたとき、P C I I / O空間に書き込まれたD S Pポインタ又はP C Iポインタを使用するかどうかを決定する。

10

**【 0 1 1 2 】**

コーデック状態 / モードレジスタ( 0 x 5 F )は、全1 6ビットについて読み出し可能である。その下位8ビットだけが書き込み可能である(上位8ビットは読み出されるだけ)、B I O選択は、D S P B I O入力に接続されるF I F O状態又はユーザ信号を融通性選択を可能にする。状態信号の全て、したがって、B I Oは、高で活性である。

20

**【 0 1 1 3 】**

D S Pは、P C I A S I C上の6 4バイトF I F Oを通してP C Iバスからの又はこれへの読み出し及び書き込みを遂行する。P C IのF I F Oは、P C Iクロックに同期する2並列1 6語F I F Oとして構成される。D S P制御信号は、D S PのC L K O U T 1信号に同期する。プロセッサをクロックするのに2つのオプションが用意されている。第1オプションは、D S PをP C Iクロック速度(最高3 3 M H z)で内部的にランさせるためにP C Iクロックを2で分周しかつD S PのC L K I N 2入力を使用する(これはC L K I N 2信号を2で倍周することになる)。この結果、P C I C L K信号がC L K O U T 1信号と同じ周波数になる状況を生じる。しかしながら、C L K O U T 1信号が最大約2 7 0度移相されることがある。クロック分周電子回路を通しての遅延及びD S Pの位相ロッキング(以下、P L Lと称する)を通しての移相は、最長2 5 n sの総合遅延を起こすこともある。第2オプションは、P C I C L K信号を1で分周しかつD S Pの1による分周入力オプションを使用する(D S Pはこのモードを支持しない)ことである。最終結果は、P C I C L K信号が依然C L K O U T 1信号と同じ周波数であるが、しかしその位相は異なると云うことである。

30

**【 0 1 1 4 】**

D S PのR D z (読み出し許可 - - 低で活性)信号及びW E z (書き込み許可 - - 低で活性)信号が、F I F O読み出しパルス及び書き込みパルスを発生するために使用される。D S Pによる読み出しを、クロック遅延を考慮することを保証するために、少なくとも3ソフトウェア待機状態で以て遂行しなければならない。

40

**【 0 1 1 5 】**

クロック遅延を考慮に入れる保証をするために、D S Pによる書き込みを少なくとも2ソフトウェア待機状態で以て遂行しなければならない。

**【 0 1 1 6 】**

D S Pは、F I F Oから到来する及びこれへ行くデータを管理する。このデータ流は、F I F Oをオーバーランかつアンダラン( u n d e r u n )させない限りD S Pにトランスペアレントであり得ない[ D S PはP C Iバスをアウトラン( o u t r u n )しかつF I F Oをオーバーランすることもあり得る]。D S Pにリソースを最善に使用させるために、いつF I F Oが読み出すデータを有する又はF I F Oが書き込み中いつ空になったかをD S Pに告げるためにフラグとしてB I O信号を使用することになっている。D S Pからの

50

書き込みを遂行する2つに方法がある。第1の方法は、F I F Oが(読み出し又は書き込み前に)空になるまで待機し、次いでF I F Oを満たす数の語を読み出す又は書き込むことである。これは、現行のハードウェア構成では32語を扱うことになり、F I F Oをオーバーラン又はアンダランしないように保証する。32語を読み出し又は書き込んだ後、D S PはB I O信号をループに載せ(これはF I F Oが他の12語を受ける用意を整えたことを表示する)かつ他の32語を読み出し又は書き込みすることができる。

#### 【0117】

第2の方法は、12語より大きいデータのブロックを転送させることである。この方法は、B I O信号を使用するが、しかしD S PはF I F Oへの毎読み出し又は書き込みの前にB I O信号をループに載せる。これらの動作に関してB I O信号を介してF I F O状態を表示するために、半空及び半満フラグを使用することができる。D S Pは、そのI/O空間を使用して、P C Iトランザクションを開始させるために制御レジスタ内にビットをセットする。その制御レジスタパラメータは、P C Iアドレス空間選択ビット、F I F Oリセットビット、P C Iマクロビット、D S P状態ビット、開始ビット、及びB I O選択ビットである。トランザクションが開始する前にF I F Oの状態を検査するために状態レジスタがまた必要である。P C Iバストランザクションを開始させるのに必要なパラメータは、D S Pアドレス(これはP C I主メモリアドレスに翻訳される)、転送する語の数(ハードウェアは語の数をバイトの数へ変換する)、及び方向である。

#### 【0118】

2語だけがP C Iバスを通して転送されるとき、信号データ位相転送が起こる。これらの2語は1つの32ビットトランザクションを満たすだけである。P C Iマクロは、P C Iプロトコルを正しく取り扱うためにこの特別な状況を知っている必要がある。D S Pによって書き込まれた値(語の数)は、バイトの数を得るために2倍され、それであるからバイトの最少数は2である。実際に転送すべきバイトの数は4ある。もし4バイト未満が転送されるならば、D S PはP C Iバスバイト許可ビットを変化させなければならないか、又はP C Iデータ位相の4バイトのうち2つが有効でないことになる。方向ビットは、これがF I F Oの入力の送り手内の決定因子であるので、重要である。方向ビットは、D S PがF I F Oに書き込むとき自動的にセットされる(これは、ビットをセットし、F I F Oを書込み、次いでトランザクションを開始させなければならないことを防止する)。P C Iバスからの読み出しの際、方向ビットは、トランザクションを開始させるレジスタへの書き込みによってセットされる。B I O信号は、B I O選択レジスタの値によって選択される。この値を、読み出し動作と書き込み動作との間に変化がある毎に変化させなければならない。

#### 【0119】

これらのパラメータがインタフェースA S I Cに書き込まれた後に、開始ビットがD S Pによって書き込まれたときP C I転送が開始される。D S P順序の例は次のとおりである。すなわち、

#### 【0120】

D S P読み出し順序

1. 転送されて来るデータのアドレスを書き込め。
2. 転送する語の数( 2 )を書き込め。
3. 終了(D O N E)ビット(それをクリアせよ)、ワーキングビット(それをセットせよ)、転送の方向(0 = 読み出し)、B I O選択、及び開始(S T A R T)ビットを書き込め。

#### 【0121】

いったん開始ビットがセットされると、ハードウェアは、すなわち、

- 3.1 P C Iバスが自由になる(P 2 C D G N T C U P L信号が不活性へ移行する)のを待機せよ。
- 3.2 C D 2 P R E Q C U L P L信号及びC D 2 P G N T C U P L信号をセットするためにレジスタに書き込め。

10

20

30

40

50

- 3.3 P2CDGNTCUP L 信号が活性へ移行するのを待機せよ。
- 3.4 (他のトランザクションを防止するために) CD2PMSTCUP L 信号をセットするためにレジスタに書き込め。
- 3.5 新アドレス、バイトの数、方向、及び多相状態をマクロヘストローブせよ。
- 3.6 FIFO を満す動作を開始せよ。
4. FIFO が空でないことをマクロが表示するのを待機せよ。(この状態を BIO として DSP にもたせ)。
5. FIFO から読み出せ。読み出しを 2 サイクル / 1 待機状態命令を以て FIFO から行うことができる。これらの待機状態は、ソフトウェア待機状態である。FIFO 読み出しパルスは、DSP 読み出し信号によって発生される。もし FIFO がデータを持たないならば、BIO はセットされず、動作は延期される(これについて、内部時間切れを使用しなければならないことがある)。BIO 信号を使用することによって、DSP はデータを待機する。3. にループバックして、完成まで繰り返せ。PCI 割り込みの発生が完了するとき、DONE ビットをオプションにセットすることができる。

10

## 【0122】

もし PCI バススレーブが最後の 2 データ位相中待機状態なしでランできるならば、PCI マクロは FIFO を満たすことになる。もし最後のデータ位相になんらの待機状態がないならば、FIFO は最後の語を書き込まれないことになる。FIFO は、FIFO Full - 1 倍語を内部に有することになる。

## 【0123】

20

## DSP 書き込み順序

1. FIFO が空であることをマクロが表示するのを待機せよ。(この状態を BIO として DSP 内に持たせ。)
2. FIFO に書き込め。書き込みを 3 サイクル / 1 待機状態命令を以て FIFO に行うことができる。これらの待機状態は、ソフトウェア待機状態である。FIFO 書き込みパルスは、DSP 書き込み信号によって発生される。
3. 転送されるデータのアドレスを書き込め。
4. 転送する語の数(この場合、32)を書き込め。
5. 終了(DONE)ビット(それをリセットせよ)、ワーキングビット(それをセットせよ)、BIO 選択、及び開始ビットを書き込め。

30

## 【0124】

いったん開始ビットがセットされると、ハードウェアは、すなわち、

- 5.1 PCI バスが自由になる(P2CDGNTCUP L 信号が不活性へ移行する)のを待機せよ。
- 5.2 CD2PREQCUP L 信号及び CD2PGNTCUP L 信号をセットするためにレジスタに書き込め。
- 5.3 P2CDGNTCUP L 信号が活性へ移行するのを待機せよ。
- 5.4 (他のトランザクションを防止するために) CD2PMSTCUP L 信号をセットするためにレジスタに書き込め。
- 5.5 新アドレス、バイトの数、方向、及び多相状態をマクロヘストローブせよ。
- 5.6 FIFO が空であることを DSP に警告せよ(BIO 信号を使用せよ)。3. にループバックして完成まで繰り返せ。

40

## 【0125】

DSP が主メモリにアクセスするとき、DSP の 16 ビット語指向アドレスを 32 ビットバイト指向アドレスに翻訳しなければならない。この翻訳の部分は、DSP の 16 ビットを 1 ビット左へシフトさせる(2 を乗じる)ことを要求する。これは、メモリ空間の 64 K 語をアドレス指定する代わりに、DSP が PCI ホスト主メモリの 128 K バイトをアクセスすることが実際にできることを意味する。PCI 仕様は、直線アドレス指定(アドレスがバースト転送中直線的に増分する)を使用することを望む PCI イニシエータがアドレスの 2 最下位ビットを 00 として残すことを要求する。これは、主メモリからの全て

50

の取り出しを奇数アドレス（最下位ビットが0）で以て開始しなければならないと云う制約をDSPソフトウェアに課する。主メモリ内のアプリケーションに融通性を持たせるために、DSPデータ空間及び命令空間は4K境界上で再配置可能であることになる。

#### 【0126】

PCIバスは、選択可能幅バスである。どんな4バイト許可信号も、このバス上の4バイトのどれか又は全てが有効データであることを、活性で以て表示することができる。DSPは16ビットバスを有し、このバスは毎転送の際有効データの16ビットを有するものと仮定する。DSPがPCIバス上のイニシエータである間、全てのバイト許可信号は活性である（全てのデータは32ビット）。PCIバスとDSPバスとの間で転送するために、並列な2つの16ビットがあってPCIデータを捕獲する。DSPは、いくつかのFIFOの出力を多重化（mux）して16ビット語を得る。多重化はDSPの最下位ビットによって制御される。アドレスビット0上の0は、最下位語FIFOを選択しかつアドレスビット0上の1は最上位語FIFOを選択する。

10

#### 【0127】

ステレオオーディオコーデック1956は、ISAバスインタフェースを備える8ビットデバイスである。このコーデックは、DSPのデータバスから分離されているデータバスに接続される。DSPは、I/O空間レジスタを経由してこのコーデックにアクセスする。DSPが低速ISAインタフェースを待機するのに拘束されるのを防止するために、ASIC316がDSPからの及びこれへの実際読み出し及び書き出しを制御するステートマシンを実現する。

20

#### 【0128】

コーデックを統制するこのステートマシンは、DSP I/Oアドレス空間におけるコーデック状態/モード（CODEC State/Mode）レジスタを介して制御される。ステートマシンは、1が開始ビットに書き込まれるとき開始する。読み出し又は書き込みが要求されるかどうか、及びPIOアクセス又はDMAアクセスが要求されるか否かに従って、ステートマシンは適当なプロトコルに従い、かつそれらのバイトをI/O空間レジスタに書き込むか又はコーデックの内容を読み出してそれらのレジスタ内に入れるかする。BYTE XFER COUNTビットは、DMA動作中いくつかのバイトが転送されるかを決定する（ビット1から4）。ADDR SELビットは、PIOアクセス中どのアドレスに最下位バイトが書き込まれるかを決定する（ビット0から3）。コーデックステートマシンが完了すると、コーデック状態ビットDONEがセットされる。このビットは、ステートマシンが開始するときクリアされる。このビットは、DSPのみに対する情報のために提供される。

30

#### 【0129】

PCIカードユーザは、DSPのプログラム空間内及びデータ空間内の外部メモリを含むオプションを有する。このメモリ（容易な設置及び除去のためにソケット接続される）によって、DSPアルゴリズムは、ホスト主メモリへのアクセスの利点を活かしてコードを再書き込みできるようになるまで、その現行形式を維持することができるようになる。データ空間は、ローカルデータをホスト主メモリから分離して維持するために利用可能である。グローバルメモリは、ホストシステムへのインタフェースとして指定される。このインタフェースは、グローバルメモリ空間内に在駐するFIFOを経由することになる。このメモリは、READY信号の代わりにソフトウェア待機状態を使用してアクセスされる。

40

#### 【0130】

ブートロードプロセスは、グローバルメモリ位置0xFFFF及びメモリマップI/Oポート50hを使用する。この位置は、もしPCI I/O空間レジスタ内のBIOビット（DSPコマンド語 - - ビット3）が高にセットされるならば、グローバルメモリ内で利用可能である。もしBIOが低にセットされるならば、位置0xFFFFからの読み出しの結果、ブートロードデータとグローバルメモリデータとの間にバス衝突が起こる。もしユーザがグローバルメモリを使用しなければならないならば、DSPからASICへ

50

の B R # 信号がユーザアドレスデコード信号で以てゲートされて、ユーザのグローバルメモリアクセスの際に A S I C の F I F O が活性になるのを防止する。

【 0 1 3 1 】

パワーオンの際に実行される P C B I O システム（以下、B I O S と称する）ソフトウェアは、どの種の P C I カードがシステムに設置されているかを発見する責務を負う。このソフトウェアは、云々ゆるパワーオンセルフテスト（以下、P O S T と称する）コードであり、かつ P C I アプリケーションの総称的初期化を遂行する責務を負う。初期化を完了した後、カードを主メモリに挿入し、自己のプログラムコードを取り出す。

【 0 1 3 2 】

いったん P C がパワーアップされると、D S P は、ホスト D S P ドライバが D S P のリセットを解くまで、リセット状態に保持される。この点で、ドライバは、P C I I / O 空間コマンド語（C O M M A N D W O R D）レジスタに書き込みして D S P のリセットを解く。そこで、C P U が B D S P をブートロードする。このデバイスドライバは、X F をセットするために B D S P を捜すことによって P C I I / O 空間ブートロードデータ（B O O T L O A D D A T A）レジスタを経由して 1 6 ビット語を D S P にダウンロードし、次いで、有効データがブートロードデータレジスタに書き込まれた後、（コマンド語レジスタ内に）B I O をセットする。これは、P C I A S I C 内に又はアプリケーションカード上に追加のハードウェアを必要としない。これは、ホスト D S P ドライバに関していくらかのオーバーヘッドを起こす。これは、また、ブートロードコードをソフトウェア内で変化させることができるので、最も融通性に富む調査研究である。いったん D S P が初期化されると、D S P は C P U がレジスタに書き込みかつ C O M M A N D ビットをセットするのを待機する。C O M M A N D ビットは、D S P への割り込みを発生しかつ D S P に実行を開始させる。この初期 D S P コードのなすべき第 1 のことの 1 つは、データメモリの存在を検査することである。もしデータメモリが発見されるならば、そのメモリが 8 ビット幅か 1 6 ビット幅かどうか判定しなければならない。D S P グローバルメモリの存在は、D S P アプリケーションが P C I カードにアクセスする P C I バスを経由することを要せず P C I カード上の高速メモリにデータの最大 6 4 K 語（又はバイト）までを記憶する機会を得ることを意味する。

【 0 1 3 3 】

図 2 0 は、本発明の改善されたコンピュータシステムに対するメモリ分配及びロッキングモデルの概略ブロック図である。図 2 0 に示されたメモリ配分及びロッキングは、図 1 1 に示されたものに相当する。D S P 空間に対するポイントがあることが D S P ソフトウェアを通して知られている。トランザクション毎に、何処に送り手及び受け手テーブルあるかがまた知られている。図 2 0 は、メモリアーキテクチャに対するディレクトリを提供する。図 1 1 は、単一送り手空間及び連続受け手空間を表す。W i n d o w s 9 5 では、これは仮想空間について云えるが、物理アドレスは全空間にわたって散乱している。それゆえ、D S P による散乱 - 収集バスマスタリングなる用語がある。

【 0 1 3 4 】

図 2 1 は、本発明の改善されたコンピュータシステムに対する送り手及び受け手データ D M A 転送テーブルモデルの概観ブロック図である。図 2 1 で、各ブロック 2 1 1 0、2 1 2 0、2 1 3 0、2 1 4 0 は、ラン時間にそのアプリケーションに利用されるメモリ 1 1 2 又はメモリ 1 1 2 の部分内の（散乱ロックされた）領域である。その瞬間にディレクトリが組み立てられ、これが領域内の多数の片に分裂する。リンクリストによって、アプリケーションは仮想空間内を飛び回ることができるようになる。ピンポンバッファ 2 1 5 0、2 1 6 0 調査研究が使用される。この C P U に巧く継がる C P U にとって、受け手が送り手になる。

【 0 1 3 5 】

図 2 2 は、本発明の改善されたコンピュータシステムに対する送り手データ D M A 転送テーブルモデルの内部構造の概略ブロック図である。図 2 2 で、図 2 0 と比較して送り手 D M A 転送テーブルの更に詳細な構造が示されている。

10

20

30

40

50

## 【 0 1 3 6 】

図 2 3 は、図 2 2 の送り手 D M A 転送テーブルに対する領域リストの詳細である。図 2 4 は、本発明の改善されたコンピュータシステムに対する受け手データ D M A 転送テーブルモデルの内部構造の概略ブロック図である。図 2 5 は、図 2 0 のプログラマ/データ空間の部分テーブルの更に詳細なブロック図である。図 2 4 は、受け手転送テーブル構造の詳細を示す。図 2 5 は、図 2 0 のプログラマ/データ空間ブロックの詳細を示す。図 2 3 は、図 2 2、2 4、及び 2 5 のいずれか 1 つにおけるリスト領域の詳細を示す。これらの図は、いかに D S P 及びホスト C P U がローカルメモリ 1 1 2 のような同じメモリ空間を使用する際効に共用しかつ共存するかを示す。

## 【 0 1 3 7 】

図 2 6 は、本発明による改善されたコンピュータシステムの他の実施例の電気回路ブロック図である。この実施例の場合、メモリ管理論理、メモリコントローラ、及びメモリ、及びキャッシュは 1 つのブロックに組み合わせられ、かつローカルバスを經由して C P U に接続される。ローカルバスは、次に、P C I ブリッジに接続され、このブリッジは P C I バスに接続される。更に、ビデオ/グラフィックチップ 2 6 1 4 及び画像捕獲/伸張ブロック 2 6 1 8 が、2 次元及び 3 次元グラフィック及び本明細書で先に挙げた種々のビデオ画像機能を提供する。

## 【 0 1 3 8 】

図 2 8 は、本発明による改善されたコンピュータシステムの他の実施例の電気回路ブロック図である。この実施例の場合、C P U は、そのチップ上に集積されたメモリコントローラ及び P C I ブリッジを有し、それゆえメモリ及び P C I バスに直接接続される。集積マルチメディアグラフィックコントローラが P C I バス及びディスプレイに接続される。加えて、2 つのオプションチップが提供され P C I バスに接続される。1 つのチップは D S P を備える周辺 I / O C O M B O 用である。他のチップは、D S P を備える P C M C I A コントローラ用である。キーボード、マウス、及び並列ポート（以下、P P と称する）が接続されているように、P C の B I O S が I / O ブロックに接続される。ハードディスク装置（以下、H D D と称する）及びフロッピディスク装置（以下、F D D と称する）ばかりでなく、直列ポート（以下、S P と称する）が D S P 部分に接続される。

## 【 0 1 3 9 】

図 2 9 は、本発明による改善されたコンピュータシステムの他の実施例の電気回路ブロック図である。図 2 9 は、図 2 8 に類似しているが、しかし図 2 9 の無線 L A N は P C M C I A コントローラブロックと共に集積化され、1 3 9 4 データ流を含むように高度化されている。

## 【 0 1 4 0 】

図 2 9 は、1 チップで以て全ての機能が提供されかつ裸ボンド物理層で以て I / O を提供される実施例を示す。物理層以外のことごとくが P C の部分である。この実施例は、多重周辺電子回路を回避しており、米連邦通信委員会（F C C）の承認を必要とする物理層に重点を置くことができる。回線保護装置（以下、D A A と称する）は、ソフトウェアであり、スピーチコーデックで以て音声注釈する能力を備えている。L A N は物理層に重点を置き、D S P は他の L A N 機能を仮想化する。コスト低減の終局的デジタル革新において、物理層以外のことごとくが仮想化され、D S P 2 0 0 はこの革新の中核をなしている。

## 【 0 1 4 1 】

図 3 0 は、本発明の改善されたコンピュータシステムに対するアプリケーション/ドライバモデルの実施例の電気回路ブロック図である。V x D 環境（以下、V X D E と称する）は、Windows アプリケーション用 D S P 関連低レベルサービス、例えば、マルチメディアアプリケーションを提供する。図 3 0 に示された D S P ドライバの Windows 3 . 1 モデルは、Windows V x D 及び Windows D L L を含む。Windows アプリケーションとの又はこれへの D L L インタフェースは、アプリケーションコールバック及びファイル I / O を遂行し、割り込みを取り扱い、及び V x D とインタフェース

10

20

30

40

50

する。VxDは、DSPによって使用されることがある物理メモリをロックダウンし及び自由にする責務を負うだけである。

【0142】

DSPドライバDLLが通信する必要があるWindowsドライバは16ビット又は32ビットであるので、Windows95の下では、上述のモデルはもはや適当でない。16ビットWindowsドライバ及び32ビットWindowsドライバの両方がDSPドライバと通信できるためには、上述のモデルを図31に示したように修正する。

【0143】

図31は、本発明の改善されたコンピュータシステムに対するアプリケーション/ドライバモデルの他の実施例の電気回路ブロック図である。図31に示されたモデルは、Windows VxD、16ビットWindows DLL、及び32ビットWindows DLLを有する。これらのDLLは、WindowsドライバとDSP VxDとの間の通信のためにのみ必要である。しかしながら、このモデルにとって、DSP VxDは、ことごとくに責務を負う、すなわち、DSPとインタフェースし、アプリケーションコールバック及びファイルI/O等を取り扱い、割り込みを遂行し、かつDSP VxD環境と云う名称を与えられている。

10

【0144】

VXDEは、Windowsアプリケーション用DSP関連低レベルサービス、例えば、マルチメディアアプリケーションを提供する。マルチメディアアプリケーションの場合、これらのサービスには、次が挙げられる。NodeAdvise、NodeAllocate、NodeDestroy、NodeGetAttr、NodeGetData、NodeGetPosition、NodePause、NodePutData、NodeResetStream、NodeRun、NodeSetAttr、NodeSignalEvent、NodeCovertData、NodeWaitSemaphore、NodeCreateSemaphore、及びNodeDestroySemaphore。

20

【0145】

Windowsマルチメディアアプリケーションは16ビットDLL又は32ビットDLLを通してVXDEと通信し、VXDEは16ビットの場合はコールバック機能を通して又は32ビットの場合は或る事象信号機構[セマフォ(Semaphore)]を通してWindowsアプリケーションと通信する。VXDEは、DSPのメモリマップポートに書き込むことによってDSPと通信し、DSPはハードウェア割り込みを発生することによってVXDEと通信する。ただし、このハードウェアはVXDEによってVPICDを通して仮想化される。

30

【0146】

VXDEとの通信を望むどのWindowsアプリケーションにとっても、行う必要のある全ては、16ビットDLL又は32ビットDLL(DSAPI.DLL又はDSPAP32.DLLと称する)と一緒に上に挙げたAPIの集合を使用することである。好適には、16ビットDLLは、呼出し構造にパラメータを入れ、この構造を16ビット側からの呼としてマークし、次いで入口(entry)点を通してVXDEを呼び出す。入口点は、Int 2F機能1684hを使用して位置検出される。他方、32ビットDLLは、DEVICEIOCONTROLインタフェースを通してVXDEを呼び出す。VXDE側では、呼を入口点から受信すると、呼び出し構造のポインタが直線アドレスに翻訳されかつその呼が相当する機能へディスパッチされる。この機能は、どんな型式のポインタでもSELECTOR/OFFSETアドレスから直線アドレスに翻訳し、次いでこの呼を処理する。もし呼がDEVICEIOCONTROLインタフェースから受信されるならば、この呼はなんら翻訳を要せず相当する機能へ直接ディスパッチされる。

40

【0147】

いつDSPハードウェアが特定タスクを終了したかの通知を望む12ビットWindowsアプリケーションは、それを次によって行うことができる、すなわち、OVERLAP

50

P E D 構造内に有効事象ハンドルを入れかつポインタを V X D E へ送る。したがって、これは、事象に際して W a i t F o r S i g l e O b j e c t を呼び出す必要がある。通知を望まないアプリケーションの場合、V X D E を呼び出すとき、O V E R L A P P E D 構造に対するポインタを含むパラメータは零であるとする。

【0148】

いつ D S P ハードウェアが特定タスクを仕上げたかの報せを望む 16 ビット W i n d o w s アプリケーションは、コールバック機能を使用することができる。

【0149】

図 3 2 及び 3 3 は、本発明の改善されたコンピュータシステムに対する図 3 1 及び 3 2 のアプリケーション/ドライバモデルにおける種々の作用を示す概略ブロック図である。V X D E は、V P I C D を通して D S P と通信するために使用される I R Q を仮想化する。指定された I R Q へのどんな割り込みも V X D E へディスパッチされる。割り込みが発生すると、V X D E はまず当該割り込みをクリアしてどんな将来の割り込みも使用許可し、次いで当該割り込みへのサービスを開始する。もし当該割り込みが 16 ビットアプリケーションからの呼び出しの結果であるならば、コールバック機能がスケジュール事象 ( C a l l \_ P r i o r i t y \_ E v e n t ) を通して取り扱われ、この間コールバック機能は V m m サービス S i m u l a t e \_ F a r \_ C a l l を通して呼び出される。これは、V x D がハードウェア割り込みを処理するとき、それがその呼び出す番号サービスによって制約されるためである。もし割り込みが 32 ビットアプリケーションからの呼び出しの結果ならば、アプリケーションの通告がやはりスケジューリング事象を通して取り扱われ、ここで、通告は E B X = o v e r l a p p e d .

I n t e r n a l を用いて W i n 3 2 サービス v W i n 3 2 \_ D I O C C o m p l e t i o n R o u t i n e を呼び出すことによって実現される。

【0150】

図 3 4 は、本発明の改善されたコンピュータシステムに対する仮想メモリモデルの実施例の概略ブロック図である。図 3 4 のモデルは、いかに全ての通信を直列ポートを通して D S P ハードウェアへ再指向させるかを示す。

【0151】

直列ポートを通して通信を D S P ハードウェアへ再指向させるために、D S P V x D がサービスの集合をポートドライバ及びポート仮想化 V x D に提供する。ポート仮想化 V x D がロードされるシステムブート時間に際し、この仮想化 V x D は、その実行環境を初期化し、I/Oトラッピングハンドラを設置し、ポートI/Oコンテンションハンドラを設置する、等々、それゆえコンピュータ出力マイクロフィルム(以下、COMと称する)デバイスを仮想化する。

【0152】

ディスクオペレーションシステム(以下、DOSと称する)内のDOSアプリケーションがCOMポートを獲得しようとするとき、そのポートの所有権が実モード仮想メモリ(以下、仮想メモリをVMと称する)に与えられ、そのCOMに関連したI R Q が V P I C D を通して仮想化され、それだからそのI R G からの割り込みを正しいVMへ反映することができる。COMポートを通ずどんな通信も設置されたポートI/Oトラッピングハンドラを通してトラップされ、かつD S P V x D によって提供されるサービスの集合を使用してD S P ハンドラへ再指向させられる。

【0153】

システムVMz上でラン中のWindowsアプリケーションがCOMポートを獲得しようとするとき、ポートドライバが、ポート仮想化V x D によって設置されたコンテンションハンドラを呼び出してそのCOMポートの所有権をシステムVMにセットし、かつI/Oトラッピングをターンオフする。COMポートを通るどんな通信もD S P V x D によって提供されるサービスの集合を使用するポートドライバによってD S P ハードウェアへリダイレクトさせられる。

【0154】

10

20

30

40

50

図35は、本発明の改善されたコンピュータシステムに対する仮想メモリモデルの他の実施例の概略ブロック図である。ハードウェアとのインタフェースがポートドライバ内のみ起こるように図34のモデルを図35へ変化させることができる。これを行う利点は、ハードウェアを変化させるとき、ポートドライバのみを修正すればよいと云うことである。欠点は、DOSベースアプリケーションに関する追加の遅延である。

【0155】

図36は、本発明による改善されたコンピュータシステムの部分の実施例の電気回路ブロック図である。図36は、グラフィックコントローラのメモリコントローラとの集積化及びこれらのメモリデータバスを通してのメモリの共用を示す。CPUは、メモリコントローラにアドレスを供給し、これを制御し、かつそのホストバス上のデータをデータバッファを経由して受信する。

10

【0156】

図37は、本発明による改善されたコンピュータシステムの部分の実施例の電気回路ブロック図である。図37は、PCカードソケットとビデオグラフィックアレイ（以下、VGAと称する）コントローラとの間の点对点単方向ビデオバスとしてのVZ（ズームビデオ）の使用を示す。この図は、いかに窓内のTVを低コストPCカードを備える携帯用コンピュータ内で達成することができるかを示す。MPMG又はテレビ会議カードをまたPCMCIAスロットにプラグインすることもできる。

【0157】

図38から44は、本発明による改善されたコンピュータシステムの部分のブレッドボードを示す実施例の電気回路ブロック図である。この実施例は、PCIショートカード定義に適合する多層PCボードであるDC/PC質問カードに変形される。このカードは、デジタルエリア、アナログエリア、及びブランクエリアを有し、ブランクエリアに子カードを背の低い姿勢で接続することが可能である。アナログエリア用アナログ接地面及びデジタルエリア用デジタル接地面をコーデックチップに近くして接続するものとする。このボードの端板は、C5xエミュレーションヘッダ（マザーボードに最も近い）、3つのステレオホンジャック（3.5mm）、2つのRCA844ジャック、及びRJ11ホンジャック（マザーボードから最も遠い）を収容する。

20

【0158】

子カード用ヘッダはマザーボード上の2重90度ピンである。子カードは、全体的に背の低いボード用に作られる2重ヘッダソケットを有する。子カード上のI/Oバスを経由して利用可能なアドレス範囲は、ソフトウェア待機状態を異なるアドレス範囲に割り当てるのに融通性を持たせるために分解される。

30

【0159】

DSPのC5x系列は、どの型式の入力クロック動作方式を使用するか決定する2つの入力ピンを有する。CLKMD1信号及びCLKD2信号は、4つの異なるロック動作を使用できるようにする（これらの1つは試験用に確保される）。このボード設計は、2つのクロック動作オプションを有する。

【0160】

図45は、本発明の改善されたコンピュータシステムに採用されるMPMG再生フィルタグラフモデルの実施例の概略ブロック図である。図10で、ユーザがソフトウェアオブジェクトをクリックオンすると、Windowsアプリケーションは、ダイナミックにメモリを配分しかつOLEをランする。図45の「COMインタフェース」円は、アプリケーションとフィルタグラフマネージャとの間のインタフェースであり、及びMCIBロックはメディアインタフェースを制御する。MPMGは、層から層式の代わりにオブジェクトからオブジェクト式に働く。送り手は、伸張するために画像捕獲データを含むことがある。

40

【0161】

図46は、本発明による改善されたコンピュータシステムの他の実施例の電気回路ブロック図である。図46は、図29に類似しているが、しかしPCカードバスブロックが集積

50

化 I D S P、及びビデオ用 Z V バスを有する。サウスブリッジは、また、集積化 I D S P 機能を含む。グラフィック及びビデオ機能は P C I バスに接続された単一チップ内に集積されている。

【 0 1 6 2 】

図 4 7 は、本発明の改善されたコンピュータシステムに使用されるソフトウェアアプリケーション及びそれらの関連アプリケーションのブロック図である。図 4 7 は、図 2 9 の下側部分に類似している。

【 0 1 6 3 】

図 4 8 は、本発明の改善されたコンピュータシステムに使用されるオーディオデコーダの単純化ブロック図である。入来コード化オーディオは多重分離され、誤り検査され、かつ  
10  
どんな付随データも出力される。種々のオーディオ副帯域が逆量子化ブロックへ出力され、この逆量子化ブロックは量子化ブロックステップをデータの部分として供給される。逆量子化ブロック 1 4 1 1 は逆フィルタバンクブロックへ出力し、後者はデコードしたオーディオ信号を出力する。

【 0 1 6 4 】

図 4 9 は、直接 D S P ( d i r e c t D S P ) 構成要素及びいかにこの要素をそのドライバ及びエミュレーションブロックとインタフェースするかの概略ブロック図である。このような構成要素は本発明の改善されたコンピュータシステムに採用される。

【 0 1 6 5 】

図 5 0 は、本発明の改善されたコンピュータシステムに対する仮想メモリモデルの他の実施例の概略ブロック図である。図 5 0 に示されたモデルは、図 1 2 のそれに類似しているが、しかし W i n d o w s 直接 D S P H A L、1 6 ビット W i n d o w s D L L、及び  
20  
3 2 ビット W i n d o w s D L L を有する。これらの D L L は、W i n d o w s ドライバと直接 D S P との間の通信にのみ必要である。しかしながら、このモデルにとって、直接 D S P は、ことごとくの責務を負う、すなわち、D S P とインタフェースし、割り込みを取り扱い、アプリケーションコールバック及びファイル I / O を遂行する、等々、かつ直接 D S P 環境の名称を与えられている。

【 0 1 6 6 】

図 5 1 は、本発明の改善されたコンピュータシステムを実現する W i n d o w s アプリケーションと基礎をなす P C ハードウェアとの間のソフトウェアの種々の層を示す概略ブ  
30  
ロック図である。

【 0 1 6 7 】

図 5 2 は、本発明の改善されたコンピュータシステムに対する図 5 0 の種々のモデル内の種々の作用を示す概略ブロック図である。図 5 2 は、図 3 2 に類似しているが、しかしまた A P I レベルブロックを含む。

【 0 1 6 8 】

図 5 3 は、本発明の改善されたコンピュータシステムに対する図 5 0 の種々の仮想モデル内の種々の作用を示す概略ブロック図である。図 5 2 は、図 3 3 に類似しているが、しかしまた A P I レベルブロックを含む。

【 0 1 6 9 】

図 5 4 は、本発明の改善されたコンピュータシステムに対する仮想モデルの他の実施例の概略ブロック図である。図 5 4 は、図 3 4 に類似しているが、しかしまた直接 H A L ブ  
40  
ロックを含む。

【 0 1 7 0 】

図 5 5 は、本発明の改善されたコンピュータシステムに対する仮想モデルの他の実施例の概略ブロック図である。図 5 5 は、図 3 5 に類似しているが、しかしまた直接 H A L ブ  
40  
ロックを含む。

【 0 1 7 1 】

図 5 6 は、本発明の改善されたコンピュータシステム上で使用可能なマルチメディアに対する仮想モデルの他の実施例の概略ブロック図である。このモデルは d i r e t x n o  
50

menclatureを使用しかつアプリケーションとドライバ構成要素との間の種々のインタフェースを示す。

【0172】

図57は、本発明の改善されたコンピュータシステムを実現するWindowsアプリケーションとその基礎をなすPCハードウェアとの間のソフトウェアの種々の層を示す概略ブロック図である。このモデルは、アプリケーション、ドライバ構成要素、及び種々のハードウェアブロックの間の種々のインタフェースを示す。

【0173】

図58は、本発明の技術に従いDSPを含むブロックを提供するように選択されたアプリケーション機能性が一括組み合わせられる種々の方法を示す概略ブロック図である。この図は、PCIバスに典型的に付属する機能性を使用する種々の集積化を示す。

10

【0174】

図59は、本発明の技術に従いDSPを含むブロックを提供するように選択されたアプリケーション機能性が一括組み合わせられる代替方法を示す概略ブロック図である。この図は、PCIバスに典型的に付属する機能性を使用する種々の集積化及びいかにこの機能性の或るものがCPUの動作を加速するためにノースブリッジと共に含まれるかを示す。加えて、第2PCIバスが種々のネットワーキング機能性又は高速アクセス用に示されている。

【0175】

図60は、本発明の改善されたコンピュータシステムに採用される他の実施例の概略ブロック図である。図60は、CPU及びメモリへのアクセスを提供するインタフェース5550の使用を示す。インタフェース5550に接続されて、複数のブロック5510、5520、5530があり、これらは各々DSPを含みかつ本発明の教示に従って1つ以上のアプリケーションを仮想化するために使用される。これらのブロックは、次にインタフェース5560に接続され、後者はPCMCIA、PCI、及び(又は)ISAバスに適当に接続される。加えて、高速ZVバスがブロック5510、5520、5530の適当な1つと相互接続するようにまた示されている。

20

【0176】

図61は、本発明の改善されたコンピュータシステムに採用される他の実施例の高レベル概略ブロック図である。図61は、いかに図60の総合ブロック5500が、メモリ及び適当なAD変換ブロック、DA変換ブロックと接続している処理素子として分類されるかを示す。追加のI/Oもまた示されている。

30

【0177】

本発明は、図示の実施例を参照して説明されたが、この説明を限定的に解釈することを意図しているのではない。図示の実施の種々の変形ばかりでなく、本発明の他の実施例は、本説明を参照するならば当技術の習熟者に実施可能でありかつ明白である。したがって、添付の特許請求の範囲は、本発明の真の範囲に属するあらゆるこのような変形及び実施例を包含すると考える。

【0178】

以上の説明に関して更に以下の項を開示する。

40

【0179】

(1) 計算システムであって、

主CPUマイクロプロセッサ、

前記主CPUマイクロプロセッサとは異なる命令集合を有するDSPマイクロプロセッサ、

前記主CPUマイクロプロセッサと前記DSPマイクロプロセッサとに結合された記憶装置、及び

主CPUが他に係わって占有されている時間間隔中DSPが前記CPUの動作を実行することによって前記システムの性能を高めるように配置された前記記憶装置内のファイルベースオペレーティングシステム

50

を含む計算システム。

【0180】

(2) 第1項記載の計算システムであって、前記DSPマイクロプロセッサと前記主CPUマイクロプロセッサとに結合されたビデオ集積回路を更に含み、前記記憶がディスクとDRAMを更に含み、前記DRAMが前記マイクロプロセッサと、前記主CPUマイクロプロセッサと、前記ビデオ集積回路とに対して統一メモリアーキテクチャ内で結合される、計算システム。

【0181】

(3) 第1項記載の計算システムであって、前記システムのアプリケーションを支持する少なくとも1つのアプリケーション装置を更に含み、前記アプリケーション装置が物理層のみに実質的に減少させられたハードウェアを更に含み、それであるから、前記DSPマイクロプロセッサが前記物理層によって媒介される信号を利用して前記アプリケーションの残りを仮想化しかつ遂行する、計算システム。

10

【0182】

(4) 第1項記載の計算システムであって、前記DSPマイクロプロセッサによる実行用に前記記憶装置内にカーネルソフトウェアを更に含む計算システム。

【0183】

(5) 第4項記載の計算システムであって、前記DSPマイクロプロセッサに結合されたI/Oポートを更に有し、前記カーネルソフトウェアが前記ファイルベースオペレーティングシステムと適切に調整するDSPマイクロプロセッサ動作を定義し、もし前記主CPUマイクロプロセッサが仮想ハードウェアを表す所与の機能を実行するのに占有されるならば、前記DSPマイクロプロセッサが前記機能を実行し、もし前記主CPUと前記DSPの両方が自由であるならば、前記ファイルベースオペレーティングシステムによって決定されるに従って前記主CPUマイクロプロセッサ又は前記DSPマイクロプロセッサのどちらかが前記機能を実行することができ、前記仮想ハードウェアが前記CPUと前記I/Oとに係わる移動度を有する、計算システム。

20

【0184】

(6) 第4項記載の計算システムにおいて、前記カーネルソフトウェアが前記ファイルベースオペレーティングシステムを用いて割り込みベースでDSPマイクロプロセッサ動作を定義し、所与の機能に対する優先権が実時間に計算され、かつ機能がダイナミックに実行され、かつ前記カーネルソフトウェアが前記ファイルベースオペレーティングシステムを供給し、それによって実時間優先権を計算することができる、計算システム。

30

【0185】

(7) 第1項記載の計算システムにおいて、前記ファイルベースオペレーティングシステムが前記DSPマイクロプロセッサ用プログラム空間について、かつ更に前記主CPUマイクロプロセッサと前記DSPマイクロプロセッサとの共用メモリモデルとしての読み出しデータ送り手空間と書込みデータ受け手空間とについて前記記憶装置に対するポインタアドレスを前記DSPマイクロプロセッサに供給するソフトウェアを含み、それによって前記配置が前記ファイルベースオペレーティングシステムを使用して前記DSPマイクロプロセッサを前記主CPUマイクロプロセッサに緊密に結合する、計算システム。

40

【0186】

(8) ソフトウェアアプリケーションの少なくとも部分を実行する能力のある第1項記載の計算システムであって、前記ファイルベースオペレーティングシステムが、仮想メモリ内の何処にソフトウェアの部分が開始しかつ終了するか告げるハンドルを定義する前記ソフトウェアを含み、前記システムが前記DSPマイクロプロセッサに主CPU機能を実行させる動作を定義するDSPカーネルを更に含み、前記主CPUマイクロプロセッサは何処に送り手ハンドルと受け手ハンドルがあるかを定義し、前記ファイルベースオペレーティングシステムは情報を受け手ハンドルによって定義された位置へ送る動作を定義し、前記DSPカーネルは何処で前記送り手ハンドルと前記受け手ハンドルとが前記DSPマイクロプロセッサに前記主CPUの代わりに機能を実行させるべきかに基づき動作を定義

50

する、計算システム。

【0187】

(9) 第1項記載の計算システムにおいて、前記主CPUマイクロプロセッサが仮想アドレスを利用し、かつ前記ファイルベースオペレーティングシステムが前記仮想アドレスに相当する物理アドレスを提供する仮想メモリマネージャを含み、かつ前記計算システムがDSPマイクロプロセッサ機能用物理アドレスを利用する動作を定義するDSPカーネルを更に含む、計算システム。

【0188】

(10) 第1項記載の計算システムにおいて、前記ファイルベースオペレーティングシステムが、16ビットアプリケーション用にコールバック機能を通して、かつ32ビット用にセマフォを通してアプリケーションと通信する仮想デバイスドライバ(VxD)を定義するソフトウェアを含む、計算システム。

10

【0189】

(11) 第1項記載の計算システムにおいて、前記ファイルベースオペレーティングシステムが前記DSPマイクロプロセッサと前記主CPUマイクロプロセッサとの間を通信するために使用される割り込みを仮想化する動作を定義する、計算システム。

【0190】

(12) 幅を有するデータ入力、前記データ入力と異なる幅のデータ出力、アドレス入力、アドレス出力、及びアドレスとデータ幅と第1バスクロック周波数とを有する第1バスと前記第1バスと異なるデータ幅の第2バスであってかつアドレスを有する前記第2バスとの間のインタフェースとして使用されるようなデバイスとを有する集積回路デバイスであって、

20

前記第1バスクロック周波数に同期するFIFOの動作クロック入力を有する少なくとも2つの並列複数語FIFO、

制御入力を有するマルチプレクサ電子回路であって、前記FIFOを経由して前記データ入力と前記異なる幅のデータ出力との間のデータのバイトを多重化するように結合された前記マルチプレクサ電子回路、及び

前記アドレス入力と前記アドレス出力との間でアドレスを翻訳するアドレス翻訳回路であって、前記アドレス入力前記マルチプレクサ電子回路の前記制御入力に結合された最下位ビットを有し、それであるからデータバイトが前記アドレス入力最下位ビットに従って異なって多重化される、前記アドレス翻訳回路を含む集積回路デバイス。

30

【0191】

(13) 第12項記載の集積回路デバイスであって、命令集合を有しかつ前記集積回路デバイス上に集積されたプロセッサを更に含み、前記プロセッサが前記アドレス入力と前記アドレス出力とに結合された、集積回路デバイス。

【0192】

(14) 第13項記載の集積回路デバイスであって、バイト使用許可出力を有し、かつ前記バイト使用許可出力をセットしかつ前記FIFOからのデータを前記データ出力に結合するために前記プロセッサに更に応答性の集積回路デバイス。

40

【0193】

(15) 第13項記載の集積回路デバイスであって、DRAMメモリコントローラとPCIバスインタフェースとを含むノースブリッジ電子回路を更に有する集積回路デバイス。

【0194】

(16) 第13項記載の集積回路デバイスであって、カードバスコントローラ電子回路を更に有する集積回路デバイス。

【0195】

(17) 第13項記載の集積回路デバイスであって、汎用直列バス(USB)とIEEE1394-準拠直列バスとを含む群から選択される直列バスコントローラ電子回路を更

50

に有する集積回路デバイス。

【0196】

(18) 第12項記載の集積回路デバイスであって、前記アドレス出力に結合された散乱-収集DMA電子回路を更に有する集積回路デバイス。

【0197】

(19) 第12項記載の集積回路デバイスであって、前記データ出力に結合されたマスタ回路とスレーブ回路との両方を含むバス制御ブロック更に含む集積回路デバイス。

【0198】

(20) 主CPUマイクロプロセッサ、  
前記主CPUマイクロプロセッサに結合された第1バスであって、アドレス線と、データ幅と、第1バスクロック周波数とを有する前記第1バス、  
前記主CPUマイクロプロセッサと異なる命令集合を有する第2マイクロプロセッサ、  
前記第1バスと異なるデータ幅の第2バスであって、アドレス線を有しかつ前記第2マイクロプロセッサに結合された前記第2バス、及び  
幅を有するデータ入力と、前記データ入力と異なる幅のデータ出力と、アドレス入力とアドレス出力とを有し、前記第1バスと第2バスとへのインタフェースとして結合された集積回路デバイスであって、  
前記第1バスクロック周波数に同期するFIFOの動作クロック入力を有する少なくとも2つの並列複数語FIFO、  
制御入力を有するマルチプレクサ電子回路であって、前記FIFOを経由して前記データ入力と前記異なる幅のデータ出力との間のデータのバイトを多重化するように結合された前記マルチプレクサ電子回路、及び  
前記アドレス入力と前記アドレス出力との間でアドレスを翻訳するアドレス翻訳回路であって、前記アドレス入力の前記マルチプレクサ電子回路の前記制御入力に結合された最下位ビットを有し、それであるからデータバイトが前記アドレス入力最下位ビットに従って異なって多重化される前記アドレス翻訳回路  
を有する前記集積回路デバイス  
を含み、  
、前記第2マイクロプロセッサが前記第2バスを経由して前記アドレス入力と前記データ入力とに結合されている  
計算システム。

【0199】

(21) 改善されたPCシステム(100)であって、主CPUマイクロプロセッサ(102)と、ファイルベースオペレーティングシステム(図19a)と、DSPマイクロプロセッサ(200)とを含み、主CPU(102)が他に係わって忙殺されている時間間隔中前記DSP(200)が前記主CPUの動作を実行するように配置され、それによって前記PCシステムの帯域幅の増大が達成される。このPCシステムは多重CPU及び(又は)多重DSPを含むこともできる。

【図面の簡単な説明】

【図1】本発明による改善されたコンピュータシステムの実施例の電気回路ブロック図。 40

【図2】図1の改善されたコンピュータシステムの部分の好適実施例の詳細な電気回路ブロック図(部分的に概略図、部分的にブロック図)。

【図3】図1の改善されたコンピュータシステムの部分の一好適実施例の更に詳細な電気回路図(部分的に概略図、部分的にブロック図)である。

【図4】図1の改善されたコンピュータシステムの部分の好適実施例の詳細な電気回路ブロック図(部分的に概略図、部分的にブロック図)であって、aはその含むチップ内の構成図、bは全体の接続図。

【図5】ホスト依存性である非対称多重処理のための本発明の改善されたコンピュータシステムの一実施例のブロック図。

【図6】本発明の改善されたコンピュータシステムに対してスーパスカラ拡張を使用する 50

実施例の概略ブロック図であって、aはCPUモデルの図、bはメモリキャッシュ階層構造図。

【図7】本発明の改善されたコンピュータシステムに対する共用メモリモデルの実施例の概略線図。

【図8】本発明の改善されたコンピュータシステムに対するマルチメディア拡張モデルの実施例の概略線図。

【図9】本発明の改善されたコンピュータシステムに対するシステム/キャッシュ/仮想メモリモデルの実施例の概略線図。

【図10】本発明の改善されたコンピュータシステムに採用されることがあるMPEG再生フィルタグラフモデルの実施例の概略ブロック図。

10

【図11】本発明の改善されたコンピュータシステムに対する仮想I/Oハードウェア-PCIDMA及びマルチメディア実時間割り込みハンドラモデルの実施例の概略ブロック図。

【図12】本発明の改善されたコンピュータシステムに対するフレーム内の並列処理の概略ブロック図。

【図13】本発明の改善されたコンピュータシステムが採用されることがあるMPEGエンコーダの単純化ブロック図。

【図14】本発明の改善されたコンピュータシステムが採用されることがあるMPEGデコーダ1400の単純化ブロック図。

【図15】本発明のノートブックコンピュータに対するビデオ解決の実施例の単純化ブロック図。

20

【図16】本発明のデスクトップコンピュータに対するビデオ解決の実施例の単純化ブロック図。

【図17】本発明の改善されたコンピュータシステムに対するDSPアルゴリズム用アプリケーションパイプラインの単純化ブロック図。

【図18】本発明による改善されたコンピュータシステムの他の実施例の電気回路ブロック図。

【図19】本発明の改善されたコンピュータシステム流れ図と電気回路図であって、aはシステム流れ図、bは図1の改善されたコンピュータシステムの好適実施例の更に詳細な電気回路図(部分的に概略図、部分的にブロック図)。

30

【図20】本発明の改善されたコンピュータシステムに対するメモリ分配及び物理ロッキングモデルの概略ブロック図。

【図21】本発明の改善されたコンピュータシステムに対する送り手及び受け手データDMA転送テーブルモデルの概観ブロック図。

【図22】本発明の改善されたコンピュータシステムに対する送り手データDMA転送テーブルモデルの内部構造の概略ブロック図。

【図23】図22の送り手DMA転送テーブルに対する領域リストの詳細を示すブロック図。

【図24】本発明の改善されたコンピュータシステムに対する受け手データDMA転送テーブルモデルの内部構造の概略ブロック図。

40

【図25】図20のプログラマ/データ空間の部分テーブルの更に詳細なブロック図。

【図26】本発明による改善されたコンピュータシステムの他の実施例の電気回路ブロック図。

【図27】本発明による改善されたコンピュータシステムの他の実施例の電気回路ブロック図。

【図28】本発明による改善されたコンピュータシステムの更に他の実施例の電気回路ブロック図。

【図29】本発明による改善されたコンピュータシステムのなお更に他の実施例の電気回路ブロック図。

【図30】本発明の改善されたコンピュータシステムに対する仮想メモリモデルの実施例

50

の電気回路ブロック図。

【図31】本発明の改善されたコンピュータシステムに対する仮想メモリモデルの他の実施例の電気回路ブロック図。

【図32】本発明の改善されたコンピュータシステムに対する図31及び32の仮想メモリ内の種々の作用を示す概略ブロック図。

【図33】本発明の改善されたコンピュータシステムに対する図31及び32の仮想メモリ内のなおまた種々の作用を示す概略ブロック図。

【図34】本発明の改善されたコンピュータシステムに対する仮想メモリモデルの他の実施例の概略ブロック図。

【図35】本発明の改善されたコンピュータシステムに対する仮想メモリモデルの更に他の実施例の概略ブロック図。 10

【図36】本発明による改善されたコンピュータシステムの部分の実施例の電気回路ブロック図。

【図37】本発明による改善されたコンピュータシステムの部分の実施例の電気回路ブロック図。

【図38】本発明の改善されたコンピュータシステムの部分の実施例の概略ブロック図。

【図39】本発明の改善されたコンピュータシステムの部分の実施例の概略ブロック図。

【図40】本発明の改善されたコンピュータシステムの部分の実施例の概略ブロック図。

【図41】本発明の改善されたコンピュータシステムの部分の実施例の概略ブロック図。

【図42】本発明の改善されたコンピュータシステムの部分の実施例の概略ブロック図。 20

【図43】本発明の改善されたコンピュータシステムの部分の実施例の概略ブロック図。

【図44】本発明の改善されたコンピュータシステムの部分の実施例の概略ブロック図。

【図45】本発明の改善されたコンピュータシステムに対するMPE再生フィルタグラフモデルの実施例の概略ブロック図。

【図46】本発明による改善されたコンピュータシステムの他の実施例の電気回路ブロック図。

【図47】本発明の改善されたコンピュータシステムに使用されるソフトウェアアプリケーション及びそれらの関連アプリケーションのブロック図。

【図48】本発明の改善されたコンピュータシステムに使用される音声デコーダの単純化ブロック図。 30

【図49】直接DSP構成要素及びいかにこの要素をドライバ及びエミュレーションブロックとインタフェースさせるかの概略ブロック図。

【図50】本発明の改善されたコンピュータシステムに対する仮想メモリモデルの他の実施例の概略ブロック図。

【図51】本発明の改善されたコンピュータシステムを実現するWindowsアプリケーションと基礎をなすPCハードウェアとの間のソフトウェアの種々の層を示す概略ブロック図。

【図52】本発明の改善されたコンピュータシステムに対する図50の種々のモデル内の種々の作用を示す概略ブロック図。

【図53】本発明の改善されたコンピュータシステムに対する図50の種々の仮想モデル内の種々の作用を示す概略ブロック図。 40

【図54】本発明の改善されたコンピュータシステムに対する仮想モデルの他の実施例の概略ブロック図。

【図55】本発明の改善されたコンピュータシステムに対する仮想モデルの他の実施例の概略ブロック図。

【図56】本発明の改善されたコンピュータシステム上で使用可能なマルチメディアに対する仮想モデルの他の実施例の概略ブロック図。

【図57】本発明の改善されたコンピュータシステムを実現するWindowsアプリケーションとその基礎をなすPCハードウェアとの間のソフトウェアの種々の層を示す概略ブロック図。 50

【図58】本発明の技術に従いDSPを含むブロックを提供するように選択されたアプリケーション機能性が一括組み合わせられる種々の方法を示す概略ブロック図。

【図59】本発明の技術に従いDSPを含むブロックを提供するように選択されたアプリケーション機能性が一括組み合わせられる代替方法を示す概略ブロック図。

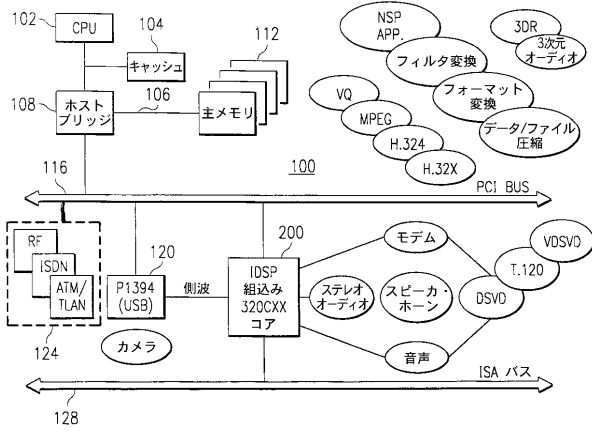
【図60】本発明の改善されたコンピュータシステムに採用される他の実施例の概略ブロック図。

【図61】本発明の改善されたコンピュータシステムに採用される他の実施例の高水レベル略ブロック図。

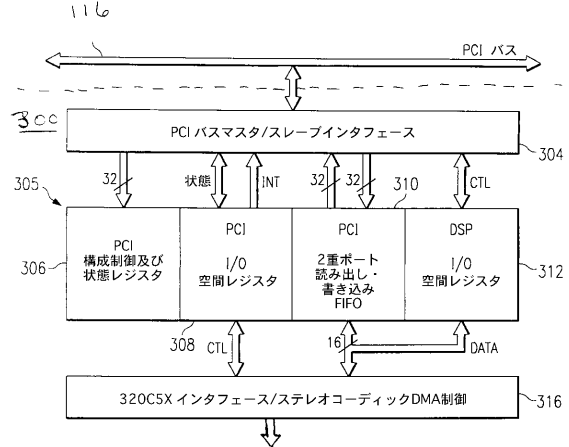
【符号の説明】

100	PCシステム	10
102	CPU	
104	キャッシュ	
106	主メモリバス	
108	ホストブリッジ	
112	主メモリ	
116	PCIバス	
120	P1394/USBブロック	
124	I/Oブロック	
128	ISAバス	
200	DSPブロック	20
210	ハードウェアインタフェース電子回路	
214	インタフェース電子回路、メモリ	
218	DSPコア	
222	DSPコア	
304	PCIマスタ/スレーブインタフェース回路	
305	ハードウェア層	
306	PCI構成制御及び状態レジスタ電子回路	
308	PCI I/O空間レジスタ電子回路	
310	2重ポート読み出し書き込みFIFO電子回路	
312	DSP I/O空間レジスタ電子回路	30
316	インタフェース/コーデックDMA制御回路	
420	チップ	
424	マスタ/スレーブインタフェース	
428	DSP	
432	カードバスコントローラ	
460	チップ	
488	データバッファ	
APP	アプリケーション	
FFD	フロッピディスク装置	
HDD	ハードディスク装置	40
MUX	マルチプレクサ	
PCI	周辺デバイス相互接続	
PP	並列ポート	
PPU	周辺処理装置	
SP	直列ポート	
UMA	統一メモリアーキテクチャ	
VxD	ビデオドライバ	
WAN	広域ネットワーク	

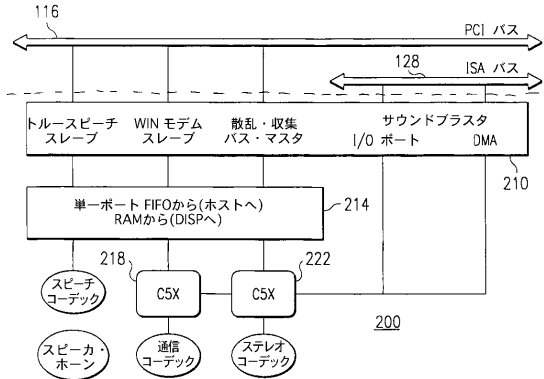
【図1】



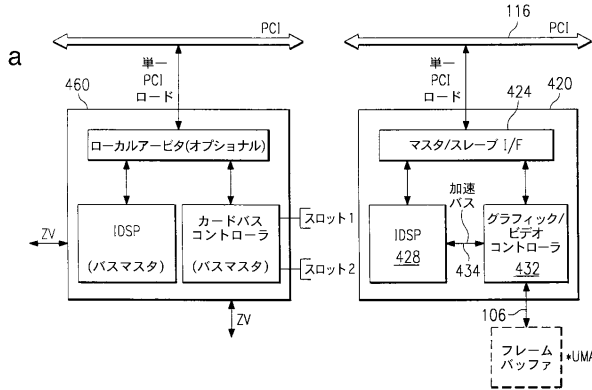
【図3】



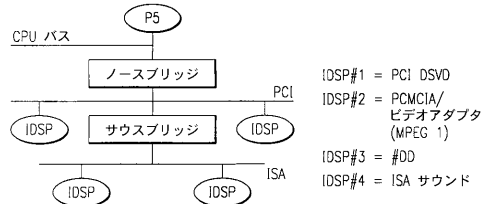
【図2】



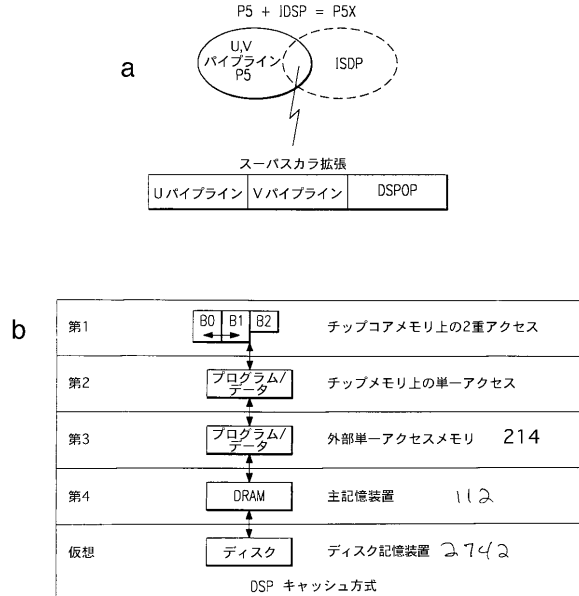
【図4】



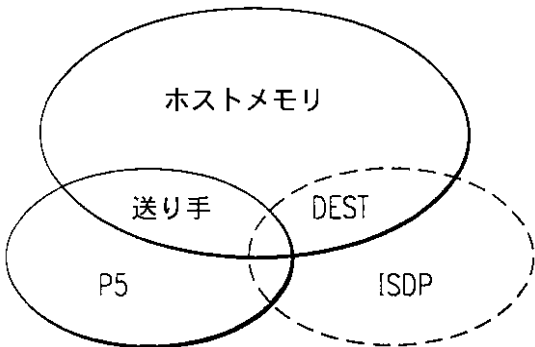
【図5】



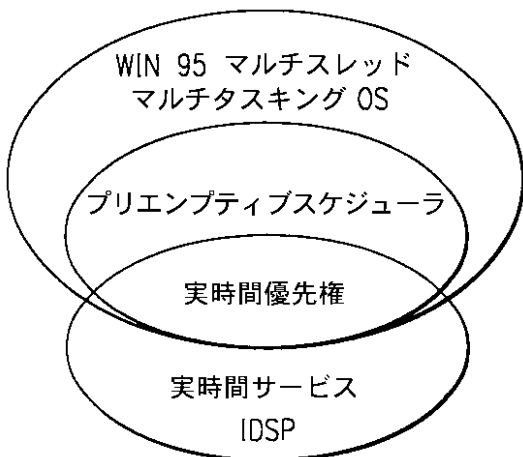
【図6】



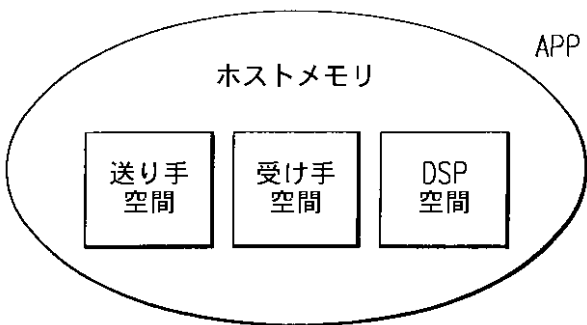
【図7】



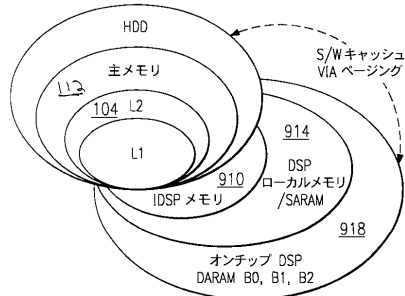
【図8】



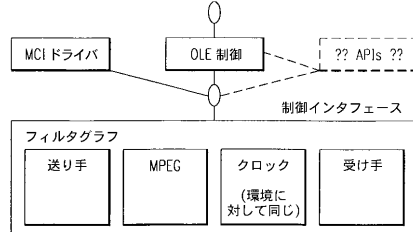
【図11】



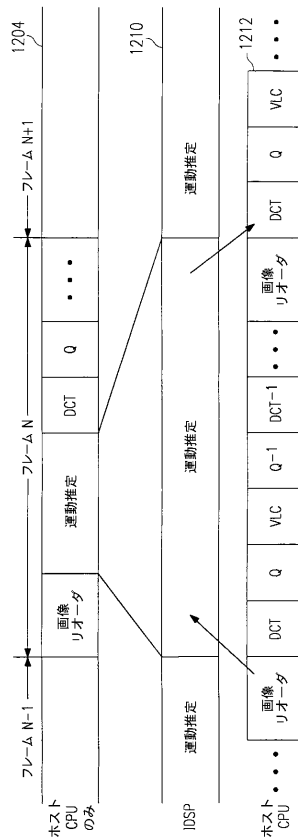
【図9】



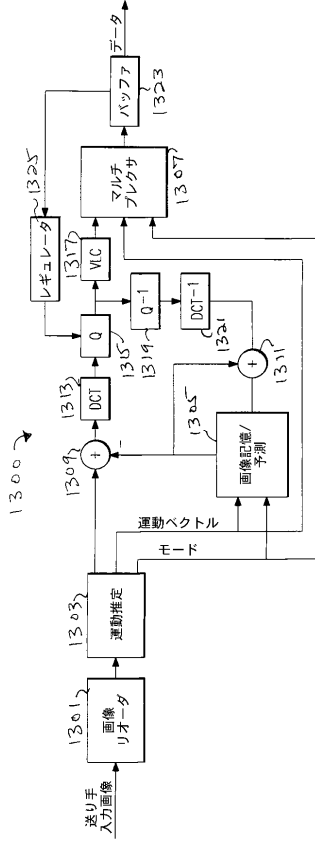
【図10】



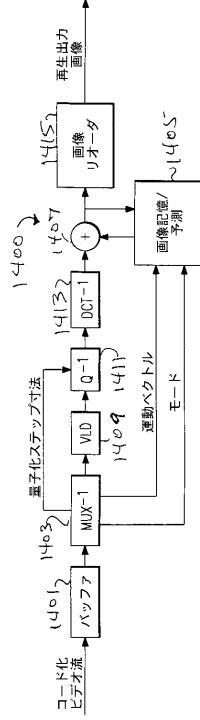
【図12】



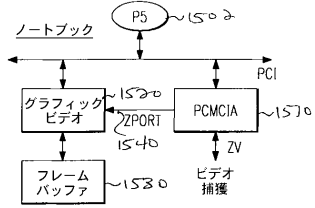
【図 13】



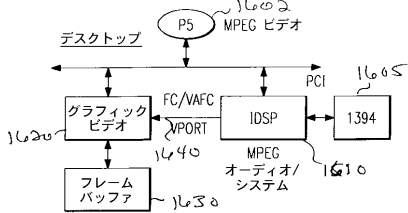
【図 14】



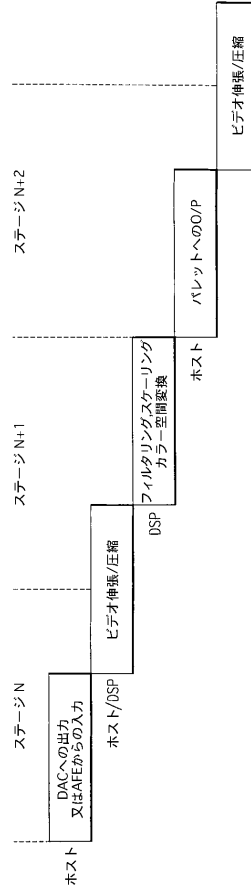
【図 15】



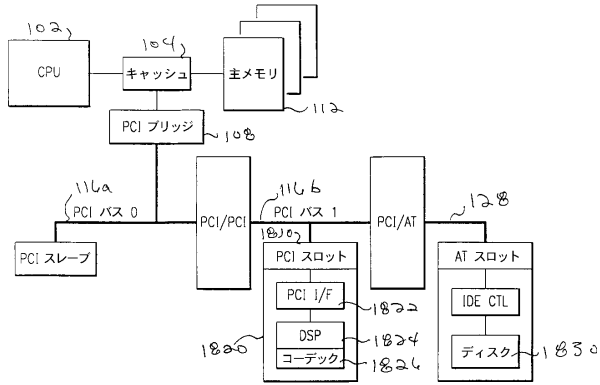
【図 16】



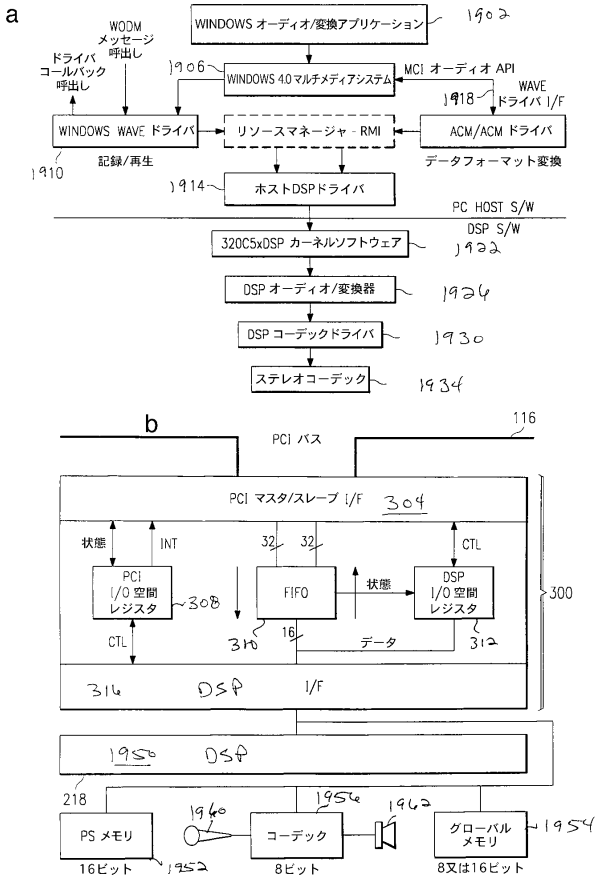
【図 17】



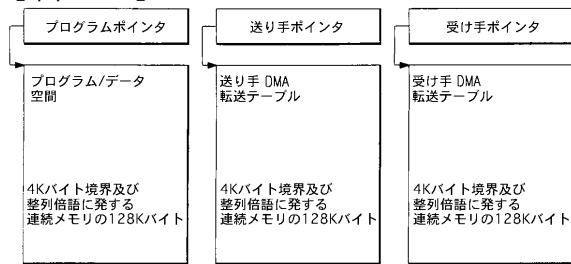
【図18】



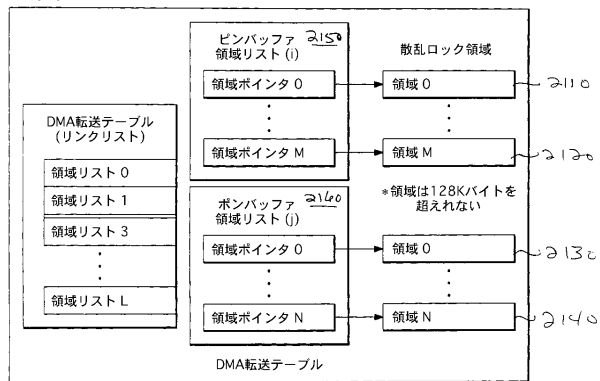
【図19】



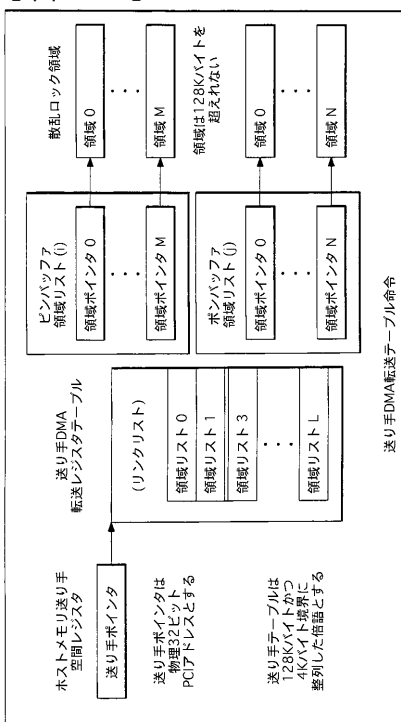
【図20】



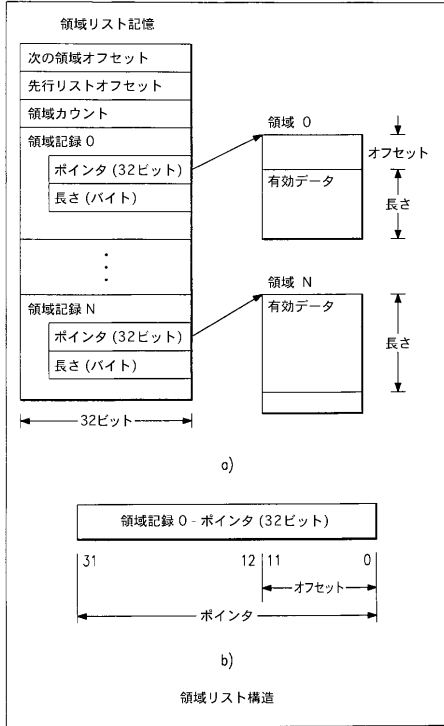
【図21】



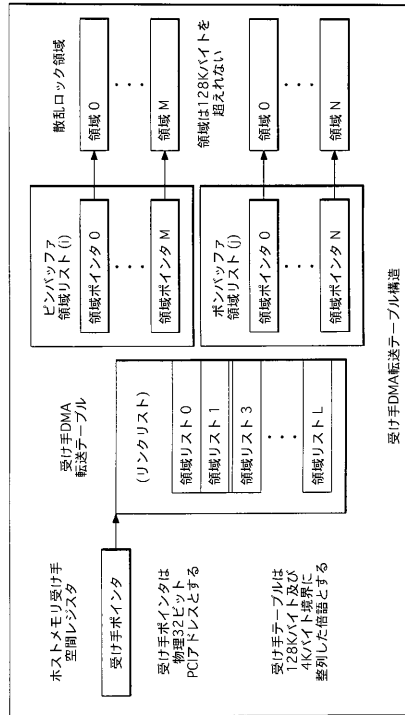
【図22】



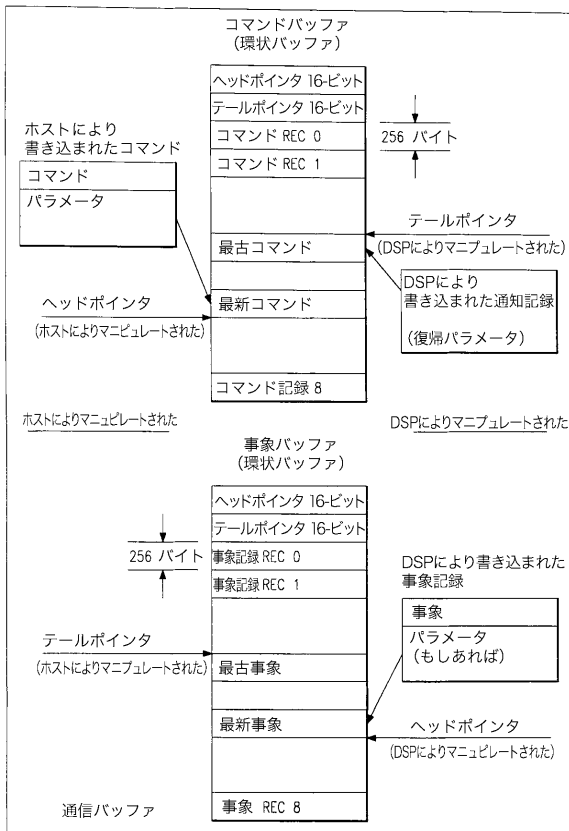
【 図 2 3 】



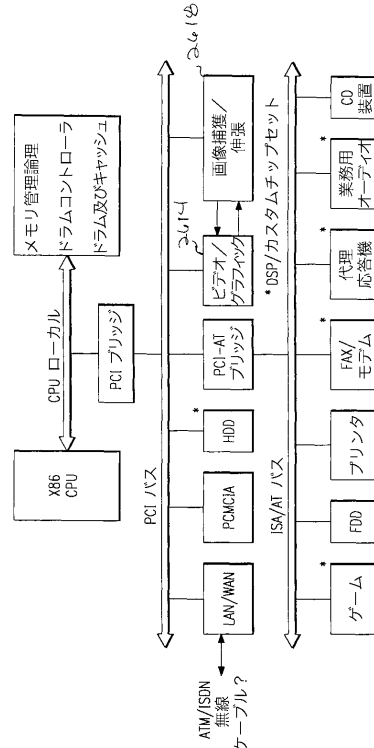
【 図 2 4 】



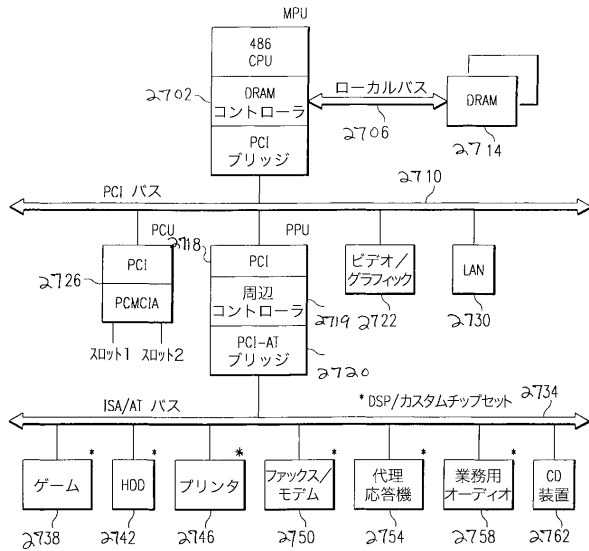
【 図 2 5 】



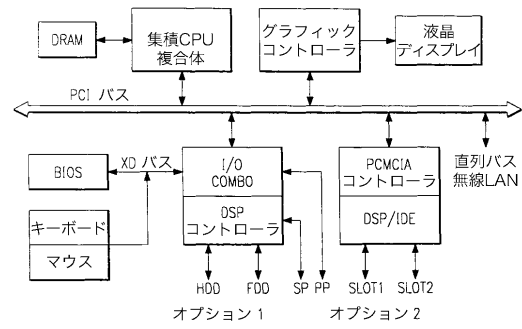
【 図 2 6 】



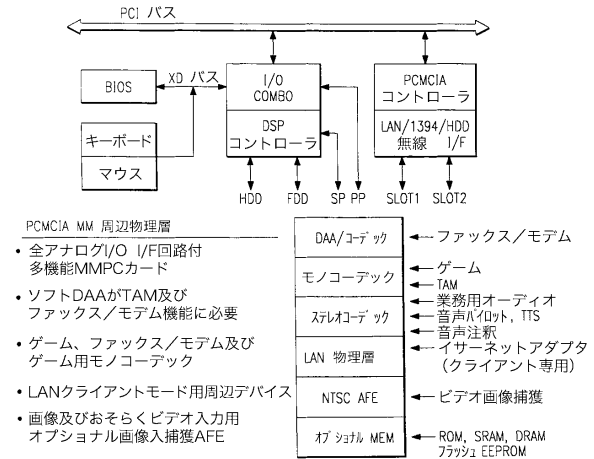
【図 27】



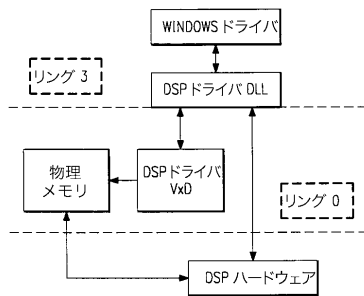
【図 28】



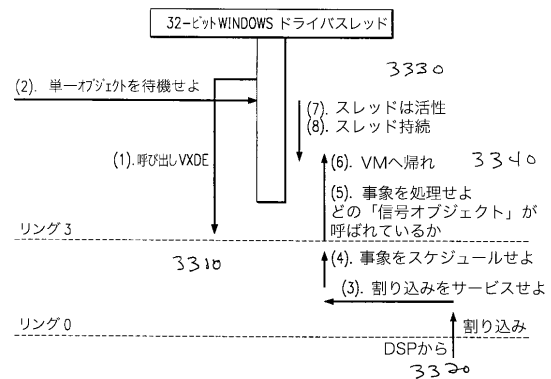
【図 29】



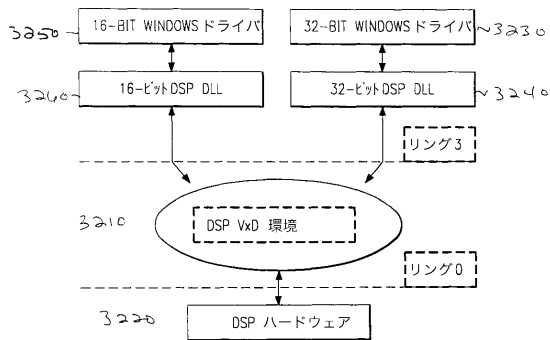
【図 30】



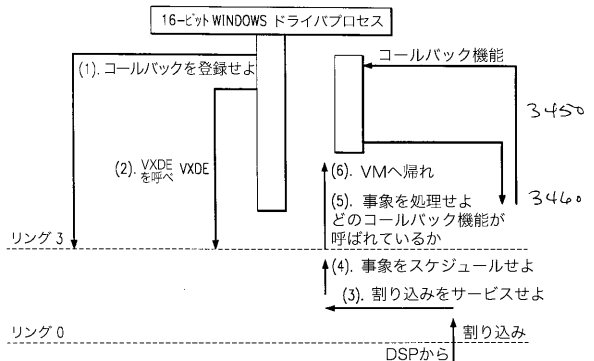
【図 32】



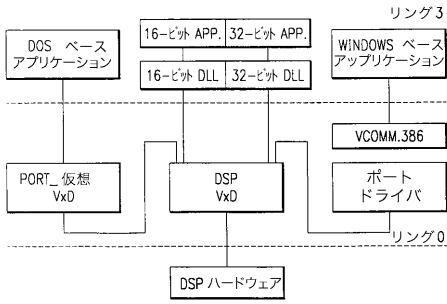
【図 31】



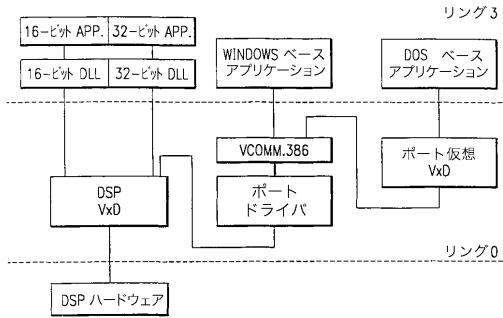
【図 33】



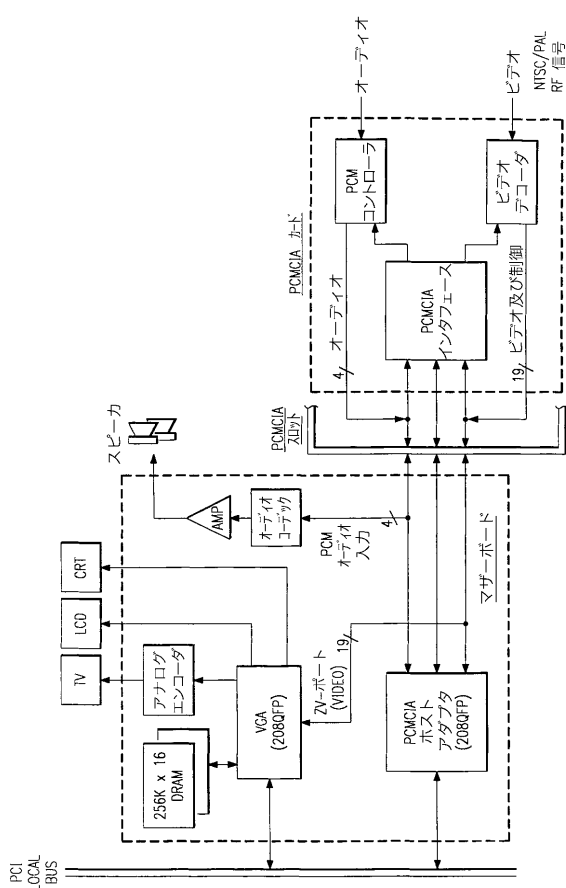
【 図 3 4 】



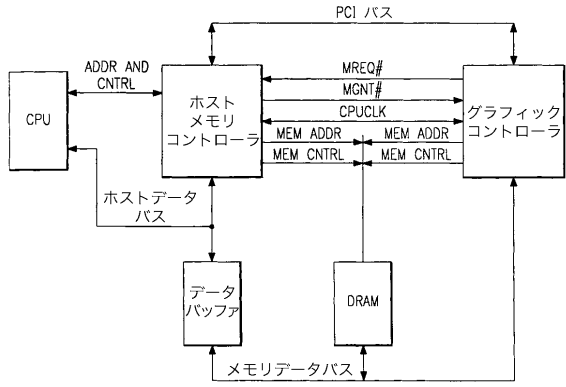
【 図 3 5 】



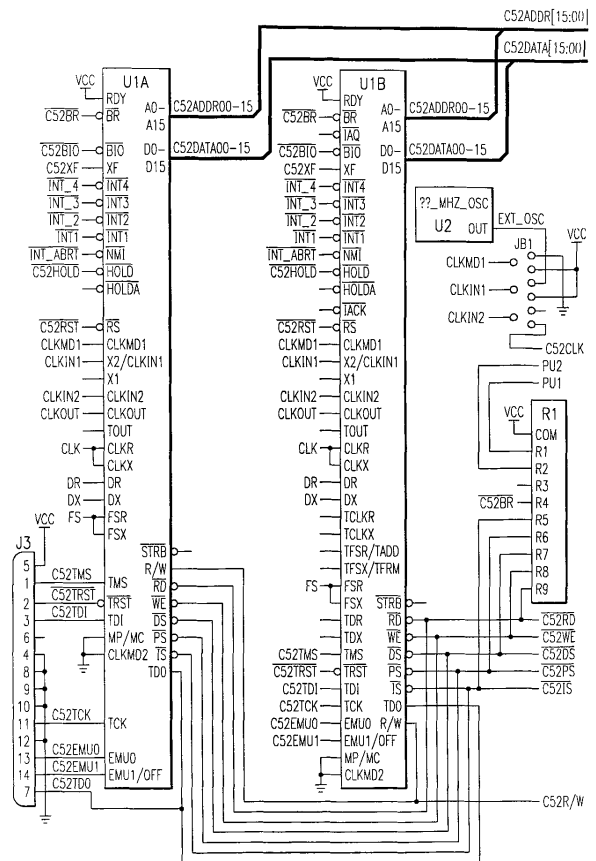
【 図 3 7 】



【 図 3 6 】

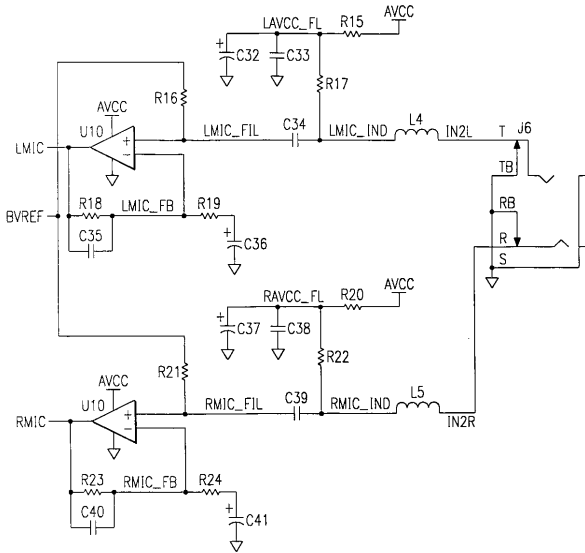


【 図 3 8 】

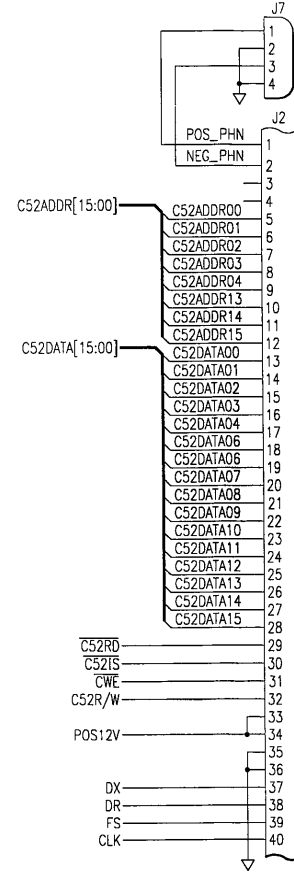




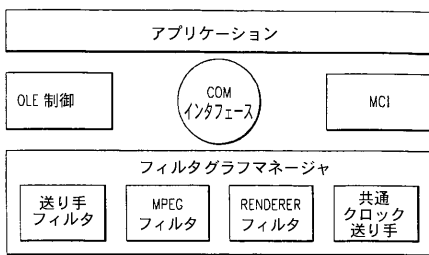
【 図 4 3 】



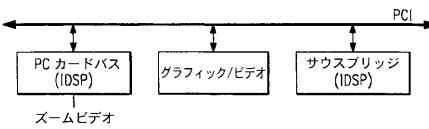
【 図 4 4 】



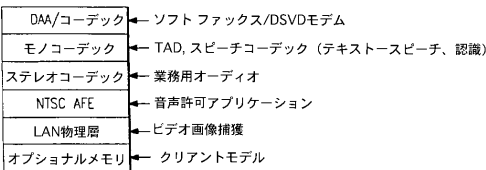
【 図 4 5 】



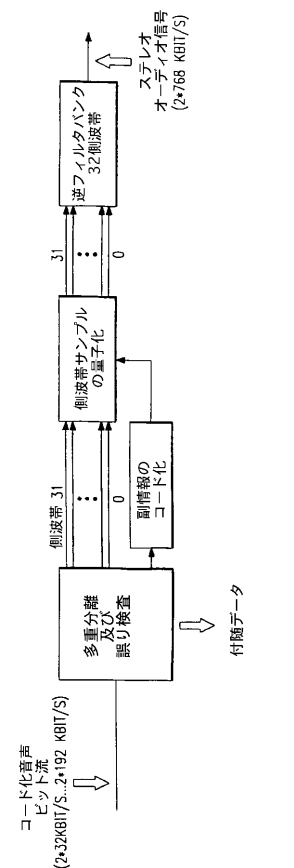
【 図 4 6 】



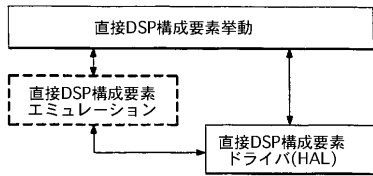
【 図 4 7 】



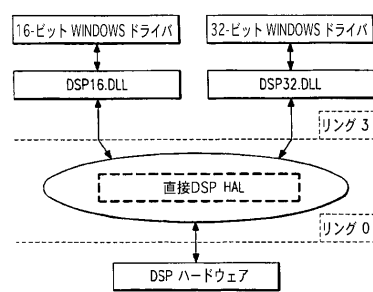
【 図 4 8 】



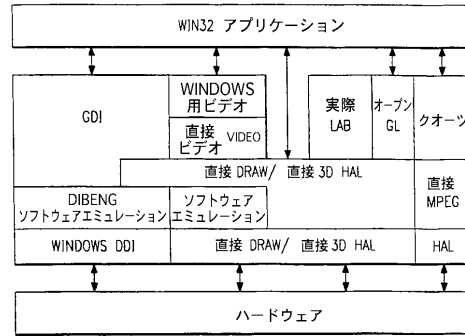
【 図 4 9 】



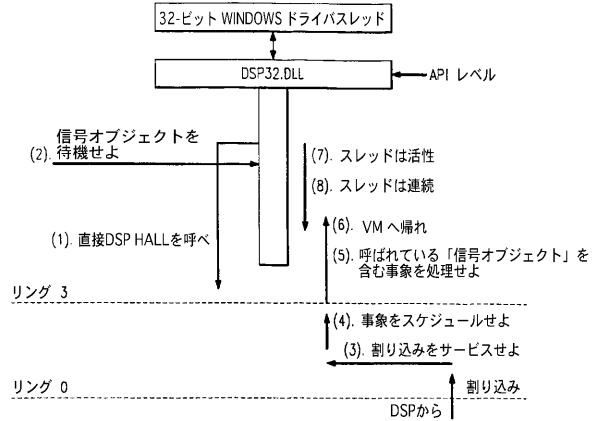
【 図 5 0 】



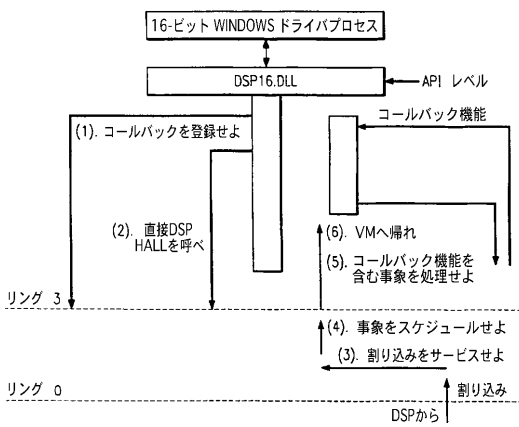
【 図 5 1 】



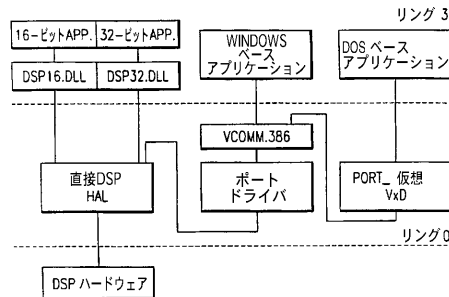
【 図 5 2 】



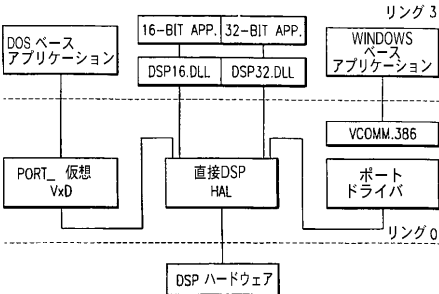
【 図 5 3 】



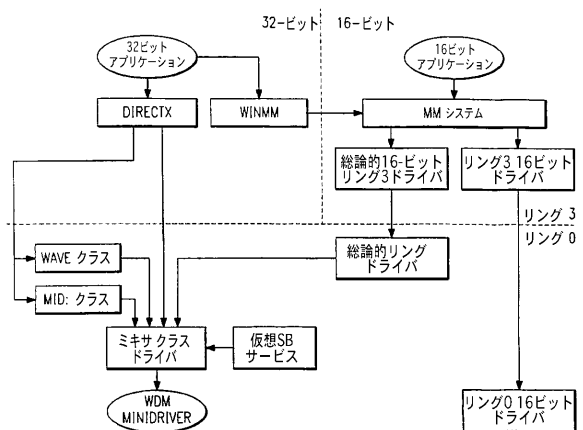
【 図 5 5 】



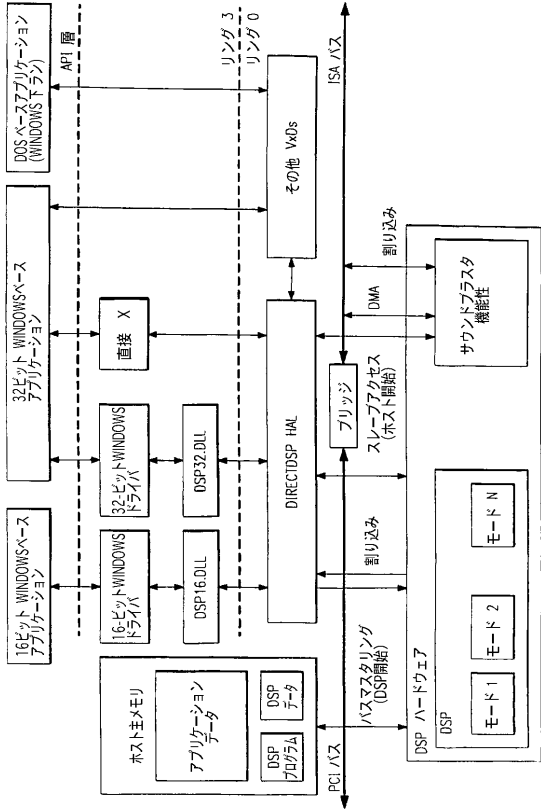
【 図 5 4 】



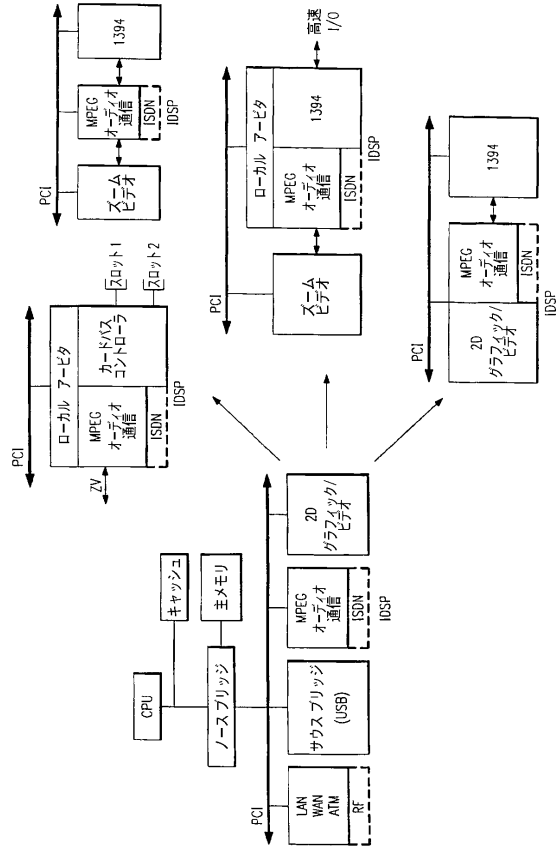
【 図 5 6 】



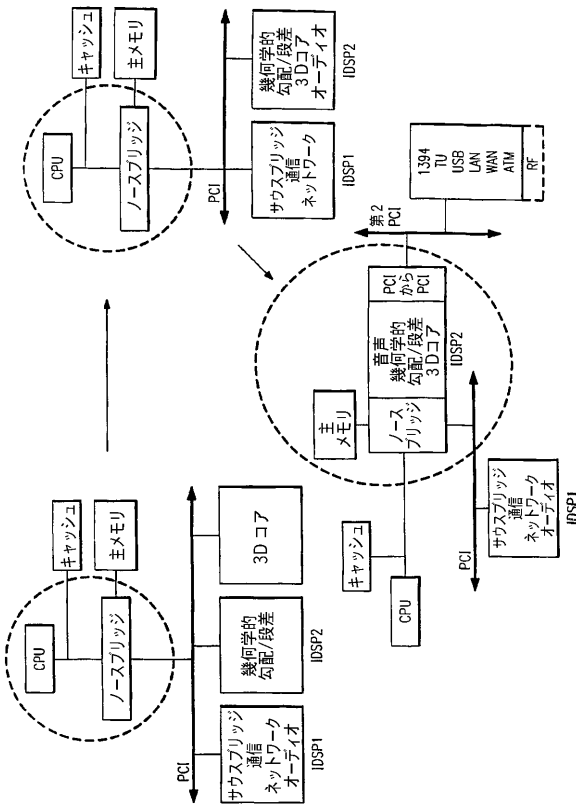
【 図 5 7 】



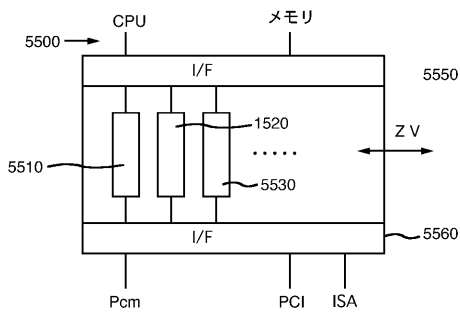
【 図 5 8 】



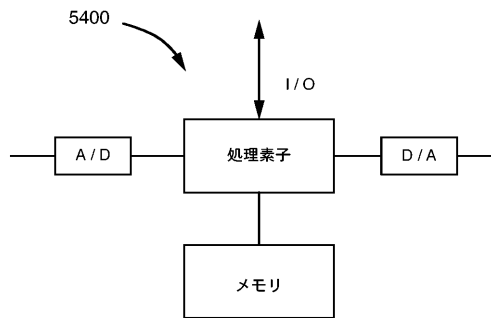
【 図 5 9 】



【 図 6 0 】



【 図 6 1 】



---

フロントページの続き

(72)発明者 ジョン リン ウィン ソー  
アメリカ合衆国テキサス州コリン カウンティ, プラノ, マウント バーノン ウェイ 3701

審査官 後藤 彰

(56)参考文献 特開平5 - 274276 (JP, A)  
特表平7 - 504054 (JP, A)

(58)調査した分野(Int.Cl., DB名)

G06F 9/38

G06F 9/46 - 9/54