

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第4789777号  
(P4789777)

(45) 発行日 平成23年10月12日(2011.10.12)

(24) 登録日 平成23年7月29日(2011.7.29)

(51) Int.Cl. F I  
**G 0 6 F 9/50 (2006.01)** G 0 6 F 9/46 4 6 5 E

請求項の数 7 (全 20 頁)

(21) 出願番号	特願2006-302057 (P2006-302057)	(73) 特許権者	310021766 株式会社ソニー・コンピュータエンタテインメント 東京都港区港南1丁目7番1号
(22) 出願日	平成18年11月7日(2006.11.7)	(74) 代理人	100105924 弁理士 森下 賢樹
(65) 公開番号	特開2008-117313 (P2008-117313A)	(74) 代理人	100109047 弁理士 村田 雄祐
(43) 公開日	平成20年5月22日(2008.5.22)	(74) 代理人	100109081 弁理士 三木 友由
審査請求日	平成19年9月12日(2007.9.12)	(74) 代理人	100134256 弁理士 青木 武司
審判番号	不服2009-5152 (P2009-5152/J1)		
審判請求日	平成21年3月9日(2009.3.9)		

最終頁に続く

(54) 【発明の名称】 タスクの分配方法および情報処理装置

(57) 【特許請求の範囲】

【請求項1】

複数のプロセッサを備える情報処理装置が、複数のタスクを含むアプリケーションプログラムを実行する際に、前記複数のプロセッサに対してタスクを分配する方法であって、前記複数のタスクは、同一のプログラムコードにもとづく処理であって、処理対象とするデータごとに実行されるタスクであり、

前記複数のプロセッサは、所定の処理単位ごとに、当該処理単位に含まれるデータに対応するタスクを実行し、前記処理単位に含まれるデータの個数はアプリケーションプログラムの実行中に処理対象に応じて所定の範囲で変動するものであり、

本方法は、前記所定の処理単位ごとに、

前記情報処理装置のハードウェアにより実現される機能が、前記アプリケーションプログラムのプログラムコード中に予め記述されている、前記複数のタスクのプロセッサへの分配規則であって、前記処理単位に含まれるデータの個数をパラメータとして、それぞれのデータを処理するタスクを前記複数のプロセッサのいずれに割り当てるかを記述する分配規則を検出する過程と、

前記情報処理装置のハードウェアにより実現される機能が、前記処理単位に含まれるデータの個数を判定する過程と、

前記情報処理装置のハードウェアにより実現される機能が、判定された前記処理単位に含まれるデータの個数に応じた分配規則にもとづき、前記複数のタスクを指定されたプロセッサへ分配する過程と、

を含むことを特徴とする分配方法。

【請求項 2】

前記プロセッサが  $N$  ( $N$  は 2 以上の整数) 個であり、判定されたデータが  $M$  ( $M$  は自然数) 個である場合に、

前記分配規則は、

$M$  が 2 以上  $N$  以下の整数  $L$  を約数として有しないとき、前記  $M$  個のデータに対応した  $M$  個のタスクが、 $N$  個のプロセッサに順に均等となるよう分配され、

$M$  が 2 以上  $N$  以下の整数  $L$  を約数として有するとき、前記  $M$  個のデータに対応した  $M$  個のタスクが、 $L$  個のプロセッサに  $(M/L)$  個ずつ分配されるように規定されることを特徴とする請求項 1 に記載の分配方法。

10

【請求項 3】

前記分配規則は、

処理すべきタスクを、各プロセッサの占有時間が短くなるように、プロセッサの個数方向に優先して分配する第 1 の規則と、

処理すべきタスクを、占有されるプロセッサの個数が少なくなるように、時間軸方向に優先して分配する第 2 の規則と、

を含み、前記情報処理装置のハードウェアにより実現される機能が、前記第 1、第 2 の規則を適宜選択して、切り替えられるように規定されていることを特徴とする請求項 1 または 2 に記載の分配方法。

20

【請求項 4】

前記第 1、第 2 の規則はアプリケーションプログラムより下位のレイヤで規定されており、そのレイヤ上で実行されるアプリケーションプログラムは、そのプログラムに含まれる所定のタスクを複数処理すべきときに、各タスクを、前記第 1、第 2 の規則いずれにより分配すべきかを指定するプログラムコードを含むことを特徴とする請求項 3 に記載の分配方法。

【請求項 5】

所定の処理単位ごとに、当該処理単位に含まれるデータに対して、所定の処理を実行する複数のプロセッサと、

前記処理単位に含まれる、アプリケーションプログラムの実行中に処理対象に応じて所定の範囲で変動する前記データの個数を、前記処理単位ごとに判定するデータ数検出部と

30

、判定した前記データの個数に応じた分配規則にもとづき各データを前記複数のプロセッサに分配する分配部と、

を備え、

前記分配規則は、前記アプリケーションプログラムに予め記述されており、かつ前記分配規則は、前記処理単位に含まれる前記データの個数をパラメータとして、それぞれのデータを前記複数のプロセッサのいずれに割り当てるかを規定することを特徴とする情報処理装置。

【請求項 6】

前記プロセッサが  $N$  ( $N$  は 2 以上の整数) 個であり、前記所定の処理単位に含まれるデータが  $M$  ( $M$  は自然数) 個である場合に、

前記分配規則は、

$M$  が 2 以上  $N$  以下の整数  $L$  を約数として有しないとき、前記  $M$  個のデータに対応した  $M$  個のタスクを、 $N$  個のプロセッサに順に均等となるよう分配し、

$M$  が 2 以上  $N$  以下の整数  $L$  を約数として有するとき、前記  $M$  個のデータに対応した  $M$  個のタスクを、 $L$  個のプロセッサに並列に分配するように規定されることを特徴とする請求項 5 に記載の情報処理装置。

40

【請求項 7】

前記分配部は、

前記データを、各プロセッサの占有時間が短くなるように、プロセッサの個数方向に優

50

先して分配する第1の方式と、

前記データを、占有されるプロセッサの個数が少なくなるように、時間軸方向に優先して分配する第2の方式と、

のうち、選択されたいずれかの方式にもとづき、前記データを分配することを特徴とする請求項5に記載の情報処理装置。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、複数のプロセッサを備えたマルチプロセッサシステムにおいて、タスクをプロセッサに分配する技術に関する。

10

【背景技術】

【0002】

近年のコンピュータゲームやデジタル放送などの分野に利用されるコンピュータグラフィックス技術や画像処理技術の著しい進歩に伴い、コンピュータ、ゲーム機器、テレビなどの情報処理装置は高精細の画像データをより高速に処理する能力を求められている。これらの情報処理装置において高性能な演算処理を実現するためには、複数のプロセッサを備えた情報処理装置において、それぞれのプロセッサにタスクを割り当てることにより、複数のタスクを並列処理したりパイプライン処理する方法が採用される。

【発明の開示】

【発明が解決しようとする課題】

20

【0003】

マルチプロセッサシステムにおいて、たとえば、音楽再生や、動画再生、ゲームの画面の生成など、リアルタイム性が要求される処理を実行する場合、各プロセッサを有効に利用しながら、各タスクを所定の時間内に完了させる必要がある。したがって、マルチプロセッサシステムにおいては、各プロセッサにどのようにタスクを分配するかが重要な課題となる。

【0004】

本発明は、このような課題に鑑みてなされたものであり、その目的のひとつは、マルチプロセッサシステムにおける効率的なタスクの分配技術を提供することにある。

【課題を解決するための手段】

30

【0005】

本発明のある態様によれば、複数のプロセッサにタスクを分配する方法が提供される。この分配方法では、複数のタスクを含んで構成されるアプリケーションプログラムのプログラムコード中に、複数のタスクのプロセッサへの分配規則を予め記述しておく。そして、アプリケーションプログラムの実行時において、分配規則を読み取り、その規則に従って複数のタスクを指定されたプロセッサへ分配する。

本明細書において、「タスク」とは、ある目的を達成するためにプログラムされたアプリケーションプログラムまたはそれに含まれる情報処理の内容をいい、アプリケーションプログラムと対応してもよいし、入出力制御やユーザが指定したコマンドなどアプリケーションプログラムよりも小さい単位と対応してもよく、何らかの処理または機能の単位に対応すればよい。タスクは、処理対象となるデータを単位としてもよい。また、「複数のプロセッサ」は、物理的に別個に構成されていてもよいし、ひとつのプロセッサ内に設けられた複数の演算コアユニットであってもよい。

40

【0006】

「分配規則」とは、各タスクをどのプロセッサに割り当てるかを具体的に記述したものであってもよいし、アプリケーションプログラムの外部で記述されたいくつかの分配規則から、いずれかの方式を選択するかを記述したものであってもよい。すなわち、「分配規則」は、タスクのプロセッサの分配に影響を及ぼす記述を含んでいけばよい。

この態様によれば、アプリケーションプログラムのプログラマが、あるタスクをどのプロセッサにおいて処理すべきかを指定できるため、効率的なタスクの分配が可能となる。

50

## 【 0 0 0 7 】

なお、本発明を装置、システム、ソフトウェアにより表現したものもまた、本発明の態様として有効である。

## 【 発明の効果 】

## 【 0 0 0 8 】

本発明によれば、マルチプロセッサシステムにおいて効率的にタスクを分配することができる。

## 【 発明を実施するための最良の形態 】

## 【 0 0 0 9 】

はじめに、本発明の実施の形態に係るタスクの分配方法およびそれを利用した情報処理装置の概要について説明する。本実施の形態に係るタスクの分配技術は、複数のプロセッサを備えるパーソナルコンピュータ、ゲーム機器、HDDレコーダ、DVDレコーダなどの情報処理装置において、各プロセッサにタスクを分配するために利用される。

## 【 0 0 1 0 】

本発明のある実施の形態によれば、複数のプロセッサにタスクを分配する方法が提供される。この分配方法は、複数のタスクを含んで構成されるアプリケーションプログラムのプログラムコード中に、複数のタスクのプロセッサへの分配規則を予め記述しておく過程と、アプリケーションプログラムの実行時において、分配規則を読み取り、その規則に従って複数のタスクを指定されたプロセッサへ分配する過程と、を含む。

## 【 0 0 1 1 】

この実施の形態によれば、アプリケーションプログラムのプログラマが、あるタスクをどのプロセッサにおいて処理すべきかを指定できるため、効率的なタスクの分配が可能となる。また、所定の処理が完了するまでの時間を見積もることができる。

## 【 0 0 1 2 】

本発明の別の実施の形態もまた、分配方法に関する。この方法は、複数のプロセッサにタスクを分配する方法であって、複数のタスクを含んで構成されるアプリケーションプログラムのプログラムコード中に予め記述されている、複数のタスクのプロセッサへの分配規則を検出する過程と、検出された分配規則に従って複数のタスクを指定されたプロセッサへ分配する過程と、を含む。

## 【 0 0 1 3 】

上述のいずれかの分配方法において、複数のタスクは、同一のプログラムコードにもとづく処理であって、処理対象とするデータごとに実行されるタスクであってもよい。

この場合、複数のデータに対して、ある所定の処理を実行する場合に、これらのデータごとのタスクが、分配規則にもとづいて各プロセッサに分配されることになる。その結果、複数のデータを処理する場合に、それらの処理に要する時間を見積もることができる。

また、複数のタスクは、その処理に略同一の時間を要する処理であってもよい。この場合、複数のタスクは、異なるプログラムコードにもとづく処理であってもよい。

## 【 0 0 1 4 】

ある実施の形態において、分配規則は、所定の処理単位に含まれるデータの個数をパラメータとして、それぞれのデータを処理するタスクを複数のプロセッサのいずれに割り当てるかを記述したものであってもよい。この場合に、本分配方法は、所定の処理単位に含まれる処理すべきデータの個数を検出する過程をさらに含み、検出したデータの個数に応じて、各タスクをプロセッサに分配してもよい。

「所定の処理単位」とは、一定時間を単位としてもよいし、複数のデータを含んで構成されるデータ群を単位としてもよく、所定の規則によってひとまとまりとされた処理の単位をいう。なお、本明細書において、「ひとつのデータ」は、所定のフォーマットにより区分されるデータ、所定のビット数、バイト数を有するデータ、など、予め決められた方法によって抽出可能なデータの単位をいう。

## 【 0 0 1 5 】

ある実施の形態において、プロセッサがN（Nは2以上の整数）個であり、処理すべき

10

20

30

40

50

データが $M$  ( $M$ は自然数)個である場合に、 $M$ が、 $2$ 以上 $N$ 以下の整数 $L$ を約数として有するとき、分配規則は、 $M$ 個のデータに対応した $M$ 個のタスクを、 $L$ 個のプロセッサに並列に分配するように規定されてもよい。

この場合、各プロセッサのタスクによる占有時間がばらつくのを防止することができるとともに、データの個数によっては、タスクに占有されない空きプロセッサを設けることができるため、その他の処理を、空きプロセッサに分配することができる。

【0016】

本発明のさらに別の態様もまた、複数のプロセッサにタスクを分配する方法に関する。この分配方法は、処理すべきタスクを、各プロセッサの占有時間が短くなるように、プロセッサの個数方向に優先して分配する第1の方式と、処理すべきタスクを、占有されるプロセッサの個数が少なくなるように、時間軸方向に優先して分配する第2の方式と、を備え、第1、第2の方式を適宜切り替えることを特徴とする。

10

【0017】

タスクの内容によっては、ある程度時間を要しても、なるべく少ない個数のプロセッサで処理を行いたい場合や、プロセッサを多く使用して、なるべく短時間で処理を完了させたい場合がある。この実施の形態によれば、プログラマは、処理内容に応じて2つの分配方式を選択できるため、ソフトウェアの設計の自由度を高めることができる。

【0018】

ある実施の形態において、第1、第2の方法はアプリケーションプログラムより下位のレイヤで規定されており、そのレイヤ上で実行されるアプリケーションプログラムは、そのプログラムに含まれる所定のタスクを複数処理すべき場合に、各タスクを、第1、第2のいずれにより分配すべきかを指定するプログラムコードを含んでもよい。

20

アプリケーションプログラムより下位のレイヤとは、OS (Operating System) レイヤ、ミドルウェア、あるいはカーネルなどを含み、そのレイヤを前提として、アプリケーションプログラムが動作するレイヤをいう。

【0019】

本発明のさらに別の態様は、情報処理装置である。この情報処理装置は、分配されたデータに対して、所定の処理を実行する複数のプロセッサと、所定の処理単位に含まれるデータの個数を検出するデータ数検出部と、所定の処理単位に含まれるデータの個数をパラメータとして定められた所定の分配規則に従って、各データを複数のプロセッサに分配する分配部と、を備える。

30

【0020】

この態様によると、プログラマは、プロセッサの占有状態をデータの個数をパラメータとして制御することができる。

【0021】

ある実施の形態において、プロセッサが $N$  ( $N$ は $2$ 以上の整数)個であり、所定の処理単位に含まれるデータが $M$  ( $M$ は自然数)個である場合に、 $M$ が、 $2$ 以上 $N$ 以下の整数 $L$ を約数として有する場合に、分配規則は、 $M$ 個のデータに対応した $M$ 個のタスクを、 $L$ 個のプロセッサに並列に分配するように規定されてもよい。

【0022】

40

分配部は、データを、各プロセッサの占有時間が短くなるように、プロセッサの個数方向に優先して分配する第1の方式と、データを、占有されるプロセッサの個数が少なくなるように、時間軸方向に優先して分配する第2の方式と、のうち選択されたいずれかの方式にもとづき、データを分配してもよい。

【0023】

以下、本発明を好適な実施の形態をもとに図面を参照しながら説明する。各図面に示される同一または同等の構成要素、部材、処理には、同一の符号を付するものとし、適宜重複した説明は省略する。また、実施の形態は、発明を限定するものではなく例示であって、実施の形態に記述されるすべての特徴やその組み合わせは、必ずしも発明の本質的なものであるとは限らない。

50

## 【 0 0 2 4 】

図 1 は、実施の形態に係る情報処理装置 2 0 0 の機能ブロック図である。情報処理装置 2 0 0 は、統括制御部 1 0 0、メインメモリ 1 0 2、メディアプロセッサ 1 1 0、サウスブリッジ 1 1 2、ディスク装置 1 1 4 を備える。情報処理装置 2 0 0 の外部には、表示装置 1 2 0 とスピーカやヘッドホンなどの音声出力部 1 2 2 が接続される。表示装置 1 2 0 および音声出力部 1 2 2 はそれぞれ、統括制御部 1 0 0 およびメディアプロセッサ 1 1 0 によるデータ処理の結果得られた映像および音声を出力する。

## 【 0 0 2 5 】

以降の図において、様々な処理を行う機能ブロックとして記載される各要素は、ハードウェア的には、CPU (Central Processing Unit)、メモリ、その他の L S I (Large Scale Integration) で構成することができ、ソフトウェア的には、メモリにロードされたプログラムなどによって実現される。したがって、これらの機能ブロックがハードウェアのみ、ソフトウェアのみ、またはそれらの組合せによっていろいろな形で実現できることは当業者には理解されるところであり、いずれかに限定されるものではない。

また、この情報処理装置 2 0 0 では、情報処理装置 2 0 0 を効率よく使用するための機能、環境を提供し、装置全体を統括的に制御するオペレーティングシステム (以下、単に「OS」とよぶ) が実行される。

## 【 0 0 2 6 】

情報処理装置 2 0 0 の全体構成の概略は以下の通りである。統括制御部 1 0 0 は、情報処理装置 2 0 0 において、アプリケーションプログラム (以下、単にアプリケーションともいう) によって規定された汎用的な演算処理を実行する演算ユニットである。一方、メディアプロセッサ 1 1 0 は、グラフィックスプロセッサとも呼ばれるユニットであり、グラフィックスに関連する演算処理に特化された演算ユニットである。さらに、メディアプロセッサ 1 1 0 は、情報処理装置 2 0 0 において生成されたオーディオ信号を、音声出力部 1 2 2 に出力する機能を有する。統括制御部 1 0 0 およびメディアプロセッサ 1 1 0 は、互いに協働しながら、表示装置 1 2 0 に表示すべき画像データやオーディオデータを生成する。

## 【 0 0 2 7 】

次に、統括制御部 1 0 0 の構成について説明する。

統括制御部 1 0 0 は、演算処理機能を有するユニットとして、1 つのメインプロセッサ 2 0 と複数のサブプロセッサ 1 0 a ~ 1 0 d を備えるマルチプロセッサシステムとして構成される。さらに、統括制御部 1 0 0 は、サブプロセッサ 1 0 a ~ 1 0 d ごとに設けられた複数のローカルメモリ 1 2 a ~ 1 2 d、メモリコントローラ 2 2、I/O 制御部 2 4 を含む。各ブロックは、バス 2 6 を介して接続されており、相互通信可能である。なお、図 1 では、サブプロセッサ 1 0 が 4 個示されるが、これに限定されるものではなく、6 個、8 個、1 0 個などであってもよく、その個数は任意である。また、メインプロセッサ 2 0 が複数設けられていてもよい。

## 【 0 0 2 8 】

メインプロセッサ 2 0 およびサブプロセッサ 1 0 は、それぞれに割り当てられたタスクを処理する。本実施の形態において、メインプロセッサ 2 0 およびサブプロセッサ 1 0 は、異なる特性を有するものとする。そこで、メインプロセッサ 2 0 およびサブプロセッサ 1 0 の特性について説明する。

## 1. メモリアクセスについて

メインプロセッサ 2 0 は、汎用的な処理を実行するプロセッサであり、キャッシュメモリを備えている。これに対して、サブプロセッサ 1 0 は、キャッシュメモリを備えていない。メインプロセッサ 2 0 は、キャッシュメモリを利用して投機的な命令実行を行うため、高速な処理が可能であるが、キャッシュヒット、キャッシュミスにより、ある処理もしくはタスクを実行するのに要するサイクル数がばらつく。これに対して、サブプロセッサ 1 0 はキャッシュメモリを有さないため、ある処理を実行するのに要する時間を確実に見積もることができる。

10

20

30

40

50

また、メインプロセッサ 20 は、メインメモリ 102 を利用してデータ処理が可能であるため大容量のデータ処理に適している。一方、サブプロセッサ 10 は、メインメモリ 102 を直接使用できず、ローカルメモリ 12 を使用しての処理となるため、大容量のデータ処理には適していない。

【0029】

## 2. 命令のサポートについて

メインプロセッサ 20 は、汎用命令セットをサポートする。これに対して、サブプロセッサ 10 は、特定の演算に特化した命令セットをサポートする。特定の演算は、たとえば算術演算、ビット演算であり、これらの演算処理に関して言えば、サブプロセッサ 10 の方が高速である。メインプロセッサ 20 は、汎用命令セットをサポートしていることにより、ある特定の演算に関して言えば、メインプロセッサ 20 の方が処理速度は遅くなる。

また、サブプロセッサ 10 は、SIMD (Single Instruction Multi-Data) 方式を採用している。これにより、データの並列演算処理に向いている。

【0030】

その他、メインプロセッサ 20 よりもサブプロセッサ 10 の方がパイプラインの段数が深く、サブプロセッサ 10 の方がリアルタイム処理に適した構成となっている。

【0031】

このような特性を有するサブプロセッサ 10 およびメインプロセッサ 20 に対するタスクの分配については後に詳述する。メインプロセッサ 20 およびサブプロセッサ 10 がそれぞれ割り当てられたタスクを実行することにより、複数のタスクが並列処理され、あるいはパイプライン処理される。

【0032】

なお、後述する各プロセッサに対するタスクの分配技術は、ある側面において、このような特性を前提として好適に利用されるものである。しかしながら、上述のメインプロセッサ 20 およびサブプロセッサ 10 の特性は以下で説明するタスクの分配技術についての理解を深めるための例示でもあり、本実施の形態に係るタスクの分配技術の適用範囲は、このような特性を有するプロセッサに限定されるものではない。

【0033】

メインプロセッサ 20 やサブプロセッサ 10 に含まれる図示しないDMAC (Direct Memory Access Controller) は、メインメモリ 102 やメディアプロセッサ 110 のグラフィックスメモリ (不図示) との間のデータ転送、データ退避などを、メインプロセッサ 20 またはサブプロセッサ 10 それぞれからの命令によって制御する。

【0034】

メインメモリ 102 は、主に統括制御部 100 によって使用される記憶領域である。メインメモリ 102 には、タスクの実行状態に関連するデータが格納される。たとえば、統括制御部 100 によりコンピュータグラフィックスに関するタスクが処理されて得られたモデリングデータなどが一時的に格納される。また、このメインメモリ 102 には、メディアプロセッサ 110 により生成されたデータが退避される場合もある。メモリコントローラ 22 は、統括制御部 100 内の各ユニットからメインメモリ 102 へのデータアクセスを制御する。

【0035】

上述したように、統括制御部 100 は、メディアプロセッサ 110 およびサウスブリッジ 112 と接続される。I/O 制御部 24 は、メディアプロセッサ 110 やサウスブリッジ 112 と、統括制御部 100 内部のブロックとのデータの送受信を制御するブロックである。サウスブリッジ 112 には、ディスク装置 114 や、ネットワークとの接続デバイス、USBメモリなどが接続される。サウスブリッジ 112 は、これらのデバイスと統括制御部 100 との間のデータ転送を制御するブロックである。

ディスク装置 114 は、BD (Blu-Ray Disc) や、DVD (Digital Versatile Disc)、CD (Compact Disc) などの光ディスクを読み込み可能な光ディスク装置である。また、ディスク装置 114 は、ハードディスク装置であってもよい。ディスク装置 114 には、

10

20

30

40

50

静止画および動画を含む画像データや、オーディオデータが所定のフォーマットで記録されている。

【 0 0 3 6 】

メディアプロセッサ 1 1 0 は、統括制御部 1 0 0 と協働して、主として画像処理を専用  
に実行するユニットであり、たとえば、レンダリング処理を実行する。メディアプロセッ  
サ 1 1 0 は、統括制御部 1 0 0 により処理されるそれぞれのタスクに関連する画像処理を  
行い、生成した画像、映像を表示装置 1 2 0 に出力する。メディアプロセッサ 1 1 0 は、  
複数の画像処理を時分割して並列的に実行してもよい。さらに、メディアプロセッサ 1 1  
0 は、統括制御部 1 0 0 により生成されたオーディオデータを、音声出力部 1 2 2 に出力  
する機能も有している。

10

【 0 0 3 7 】

メディアプロセッサ 1 1 0 は、たとえば、図示しないグラフィックスメモリ、メモリコ  
ントローラ、演算ユニット、制御ブロック、ディスプレイコントローラ等を備える。これ  
らのブロック同士は、バスを介して接続されており、各ブロック間でデータの送受信、共  
有が可能である。

【 0 0 3 8 】

制御ブロックは、メディアプロセッサ 1 1 0 全体を制御するブロックである。制御ブロ  
ックは、演算ユニット、グラフィックスメモリ、ディスプレイコントローラを統括的に制  
御し、各ブロック間のデータ転送の同期管理やタイマー管理等を行う。演算ユニットは、  
制御ブロックからの命令にしたがって、グラフィックに関する様々な演算処理を行う。そ  
の処理の一例としては、3次元モデリングデータをもとに座標変換、陰面消去、シェー  
ディングを行って画像フレームデータを作成し、フレームバッファに書き込む一連のレンダ  
リング処理などが挙げられる。

20

【 0 0 3 9 】

グラフィックスメモリは、メディアプロセッサ 1 1 0 により使用、管理されるグラフィ  
ックスデータを記憶するためのメモリ領域である。グラフィックスメモリには、画像フレ  
ームデータが格納されるフレームバッファやZバッファに加えて、画像フレームデータを  
描画する際に参照される基本データである頂点データ、テクスチャデータ、カラーlookup  
アップテーブルなどのデータに対応する領域が用意されている。

【 0 0 4 0 】

ディスプレイコントローラは、水平および垂直同期信号を生成し、表示装置 1 2 0 の表  
示タイミングにしたがって、グラフィックスメモリに格納されるフレームバッファから画  
像フレームデータのピクセルデータをライン状に順次読み込んでいく。さらにディスプレ  
イコントローラは、ライン状に読み込まれたピクセルデータを、R G B のカラー値を含む  
デジタルデータから表示装置 1 2 0 に対応したフォーマットに変換して出力する。

30

【 0 0 4 1 】

以上が、情報処理装置 2 0 0 の全体構成および動作の概要である。以下、統括制御部 1  
0 0 の内部のサブプロセッサ 1 0 およびメインプロセッサ 2 0 に対するタスクの分配に着  
目し、それを実現するための技術について詳細に説明する。

【 0 0 4 2 】

以下では、S A C D ( Super Audio Compact Disc ) の再生を例にして、本実施の形態に  
係るタスクの分配技術について説明する。S A C D の再生は、通常は専用の D S P ( Digi  
tal Signal Processor ) を用いて行われるものであり、情報処理装置 2 0 0 に搭載される  
ような、汎用的なマルチプロセッサシステムを利用して処理する場合には、そのリアルタ  
イム性から、各プロセッサに対して必要なタスクを効率的に分散する必要がある。また、  
S A C D 専用の再生装置であれば、その他の処理を実行する必要はないが、情報処理装置  
2 0 0 においては、S A C D の再生と並列して、別のアプリケーション(たとえば、テレ  
ビ番組の録画用のアプリケーション)が実行される場合があるため、それらの処理のため  
に、使用可能なハードウェア資源が限定されてしまうという事情がある。

40

【 0 0 4 3 】

50



図2は、本実施の形態に係る情報処理装置200をソフトウェアの観点から表現したブロック図である。情報処理装置200は、ハードウェアレイヤ50に、上述したサブプロセッサ10a~10d、メインプロセッサ20、ディスク装置114が存在する。

ソフトウェアレイヤ60は、コアレイヤ70およびアプリケーションプログラム(以下、単にアプリケーションという)90を含む。コアレイヤ70は、OS72、ライブラリ74、デバイスアクセスレイヤ76、バーチャルOS78、アプリケーションフレームワーク(以下、AFWという)80を含む。コアレイヤ70は、アプリケーション90の下位のレイヤであり、アプリケーションは、コアレイヤを前提としてプログラムされ、実行される。

#### 【0044】

OS72は、情報処理装置200を効率よく使用するための機能、環境を提供し、メインプロセッサ20、ディスク装置114等を統括的に制御するソフトウェアである。ライブラリ74は、サブプロセッサ10に実行させる命令などを規定する。OS72およびライブラリ74を含めて、OSとして認識してもよい。

OS72自体は、それより上位のレイヤから与えられたタスクを、OSレベルで規定された方法によって、メインプロセッサ20およびサブプロセッサ10a~10dに実行させる。OS72は、処理すべきタスクが複数ある場合、従来一般的な手法に従って、複数のサブプロセッサ10a~10dの占有状態を監視し、占有されていないサブプロセッサ10に対して順次タスクを投入する。特殊なタスク分配を必要としない一般的なアプリケーションであれば、以下で述べるバーチャルOS78およびAFW80を介さず、OS72上で実行されるため、アプリケーションのプログラマは、サブプロセッサ10やメインプロセッサ20の存在を意識せずにプログラムを記述できる。

#### 【0045】

バーチャルOS78は、OS72とライブラリ74の上位レイヤに設けられる。上述したように、OS72自体は、あるタスクを内部のカーネルに規定された規則にもとづいて、メインプロセッサ20およびサブプロセッサ10に対して分配する。これに対して、バーチャルOS78は、アプリケーション90がハードウェアに依存しないように設けられたレイヤである。このバーチャルOS78は、異なる特性を有するサブプロセッサ10とメインプロセッサ20のハードウェア的、あるいはソフトウェア的な相違点を隠蔽する。バーチャルOS78によって、上位のAFW80は、サブプロセッサ10およびメインプロセッサ20を、同種のプロセッサとしてアクセス可能となる。

#### 【0046】

AFW80は、バーチャルOS78上に設けられる。AFW80は、従来手法におけるOS72に代わって、メインプロセッサ20およびサブプロセッサ10a~10dの状態遷移を管理する。また、AFW80には、所定の処理単位が規定されている。AFW80は、所定の処理単位ごとに、所定の分配規則に従ってメインプロセッサ20やサブプロセッサ10に対してタスクを分配する。AFW80には、予め複数の分配規則が定められており、上位のアプリケーション90からの指示にもとづいて、いずれかの分配規則が選択される。また、AFW80は、内部で規定された分配規則ではなく、上位のアプリケーション90により指定された規則にもとづいて、タスクを分配してもよい。AFW80は、タスクの分配に際し、各タスクにともなうデータの入出力や、タイミング等の管理も行う。さらに、AFW80は、上位のレイヤに対して、各種のユーティリティを提供する。

#### 【0047】

SACDのディスクメディアに記録された情報は暗号化されている。デバイスアクセスレイヤ76は、ディスク装置114からのデータの読み出しと、暗号の復号処理を実行するレイヤである。

#### 【0048】

アプリケーション90は、本実施の形態では、SACDを再生する。アプリケーション90は、デバイスアクセスレイヤ76において復号されたビットストリームデータを受け、必要な処理を実行してPCM(Pulse Code Modulation)データに変換する。アプリケ

10

20

30

40

50

ーション90は、SACD再生のために必要なデータ読出モジュール92、フレームデコードモジュール94、フィルタモジュール96、エフェクタモジュール98、データ出力モジュール99を含む。各モジュールの機能については後述する。

#### 【0049】

従来のOSを利用した場合、アプリケーション90において処理すべきタスクが複数ある場合に、各タスクを複数のプロセッサのいずれに分配するかを指定することは行われていなかった。これに対して、図2に示すように、バーチャルOS78およびAFW80を設けたことにより、アプリケーション90において発生するタスクが、アプリケーション90からの指示にもとづいて、サブプロセッサ10およびメインプロセッサ20に分配することが可能となる。なお、本実施の形態では、従来のソフトウェア資源の承継の観点から、従来型のOSの上位レイヤにバーチャルOS78やAFW80を設けることにより、アプリケーション90からのタスク分配の指示を可能としているが、OSを新たに設計する場合には、あらかじめOS72およびAFW80の機能をOS自身に実装してもよい。

#### 【0050】

以下では、ハードウェア、ソフトウェア的に上述の構成を有する情報処理装置200において実行されるSACDの再生処理について説明する。

図3は、図1、図2に示す情報処理装置200により構成されるSACD再生装置300の構成を示すブロック図である。SACD再生装置300は、ディスク装置114、デバイスドライバ302、暗号復号器304、再生処理部310を含む。

ディスク装置114には、所定のフォーマットで圧縮されたオーディオデータが記録されたSACDのメディアが挿入される。デバイスドライバ302は、ディスク装置114の回転を制御し、メディアから所定数ごと、たとえば128個のセクタごとのデータを、順次読み出す。この機能は、図2のOS72のドライバと、図1のサウスブリッジ112、ディスク装置114の協働によって実現される。

#### 【0051】

デバイスドライバ302により読み出されたデータは、暗号化されている。暗号復号器304は、暗号化されたデータを復号する。この処理は、メインプロセッサ20、メモリコントローラ22、メインメモリ102、I/O制御部24等が図2のデバイスアクセスレイヤ76によって制御されることで実行される。暗号復号器304からは、DSD(Direct Stream Digital)方式で変調されたオーディオデータが出力される。以下の説明では、このオーディオデータは、DST(Direct Stream Transfer)方式により可逆圧縮されている場合について説明する。すなわち、暗号復号器304からの出力されるオーディオデータS1は、DST方式で圧縮されたDSDフォーマットのデータである。なお、DST方式で圧縮されていない場合には、伸張処理がスキップされる。

#### 【0052】

再生処理部310は、図2のアプリケーション90を含むソフトウェアレイヤ60と、図1の統括制御部100の内部ブロックの協働により実現される。再生処理部310は、DST方式で圧縮されたデータを伸張し、PCMデータに変換して出力する。以下、再生処理部310の構成について説明する。SACD再生の過程においては、さまざまなタスクが実行されるため、本実施の形態に係るSACD再生装置300は、各タスクに応じて、使用するプロセッサを選択し分配する。

#### 【0053】

データ読出部312は、暗号復号器304から出力されるDST形式で圧縮されたオーディオデータS1を読み出す。データ読出部312はディスク上の128セクタのデータをひとつのデータ単位として読み出し、読み出したデータを解析して、オーディオフィームに関するデータ(以下、フレームデータDfrmとよぶ)のみを抽出し、ある所定の処理単位ごとに、後段のオーディオフィームバッファ314に格納する。なお、一つのフレームデータDfrmには、2チャンネルから6チャンネル分のオーディオデータが含まれる。オーディオフィームバッファ314をはじめとするSACD再生装置300に設けられたバッファは、図1のメインメモリ102、ローカルメモリ12a~12dなどの記憶

10

20

30

40

50

領域に設けられる。

【 0 0 5 4 】

所定の処理単位に含まれるフレームデータ  $D f r m$  の個数は、再生すべきオーディオ信号の内容に応じて、所定の範囲で変動する。所定の処理単位自体は、S A C D再生装置 3 0 0 において任意に設定される。以下の説明では、ひとつの所定の処理単位に含まれるフレームデータ  $D f r m$  の個数は、1 ~ 4 の範囲で変動するものとする。たとえば、曲の始まりや終わりでは、フレームデータ  $D f r m$  は、1 個、2 個と少なく、通常の再生時では 3 個であり、早送りなどのトリックプレー時において最大で 4 個となる。なお、ひとつのフレームデータ  $D f r m$  は、再生時間に換算して、1 / 7 5 秒（以下、単にフレーム時間  $T f$  ともいう）の情報量を有している。データ読出部 3 1 2 によるデータ処理は、メイン  
10  
プロセッサ 2 0 に分配される。すなわち、データ読出部 3 1 2 は、主としてメインプロセッサ 2 0 と、アプリケーション 9 0 のデータ読出モジュール 9 2 により実現される。

【 0 0 5 5 】

フレームデコード部 3 1 6 は、オーディオフレームバッファ 3 1 4 から、オーディオフレームバッファ 3 1 4 から、所定の処理単位に含まれるフレームデータ  $D f r m$  を読み出す。つまり、フレームデコード部 3 1 6 に一度に入力されるフレームデータ  $D f r m$  の個数は 1 ~ 4 個となる。フレームデコード部 3 1 6 は、D S T 形式で圧縮された各フレームデータ  $D f r m$  を伸張し、D S D 形式のオーディオデータに変換する。

この伸張処理は、入力データ自体は圧縮されているためデータ量は小さいが、データの  
20  
処理内容は非常に多くなる。したがって、この伸張処理は、少ないデータを高速に処理可能な特性を有している複数のサブプロセッサ 1 0 を利用して実行する。本実施の形態に係る S A C D 再生装置 3 0 0 では、フレームデータ  $D f r m$  を、複数のサブプロセッサ 1 0 に分配する分配方式にひとつの特徴を有している。

【 0 0 5 6 】

前提として、ひとつのフレームデータ  $D f r m$  について伸張処理を完了するには、フレーム時間  $T f$  より長い時間を要するものとする。以下、伸張処理に要する時間を、実処理時間  $T p r c$  という。説明を簡略化するため、実処理時間  $T p r c$  は、フレーム時間  $T f$  の整数  $a$  倍であるとする。

実処理時間  $T p r c$  が、フレーム時間  $T f$  の  $a$  倍であるとき、リアルタイムに再生を行うためには、処理能力を  $a$  倍以上に高める必要があるため、ひとつのフレームデータ  $D f r m$  に対して、平均して  $a$  個以上のプロセッサを利用すればよい。以下では、具体的には  
30  
 $a = 2$  の場合のタスクの分配処理について説明する。

【 0 0 5 7 】

一度に処理すべきフレームデータ  $D f r m$  の個数は最大で 4 個であるため、もし仮に 4 個のサブプロセッサ 1 0 a ~ 1 0 d を利用できれば各サブプロセッサ 1 0 にそれぞれのフレームデータ  $D f r m$  に対応するタスクを割り当てればよいことになる。この場合、実時間にして 4  $T f$  相当のデータが、 $T p r c = 2 T f$  の時間で処理できるためリアルタイム性の観点から言えば望ましいが、サブプロセッサ 1 0 がすべて占有されるため、その他のタスクが実行できなくなるという問題がある。その他のタスクの一例としては、後述するエフェクト処理が挙げられる。  
40

【 0 0 5 8 】

そこで本実施の形態に係る情報処理装置 2 0 0 においては、その他のタスクを実行するために、伸張処理に、最大で 3 個のサブプロセッサ 1 0 a ~ 1 0 c を割り当てる。すなわち、ここで説明するタスク分配技術は、一度に処理すべきフレームデータの最大数  $b$ （この例では  $b = 4$  である）よりも、使用可能なサブプロセッサ 1 0 の個数  $c$ （この例では  $c = 3$ ）の方が少ないという前提がある。

【 0 0 5 9 】

上述したように、本実施の形態では、アプリケーション 9 0 が、処理すべきタスクが複数ある場合に、どのタスクをどのプロセッサに分配するかを指定できるようになっている。フレームデコード部 3 1 6 の機能を実現するためのプログラムモジュールであるフレー  
50

ムデコードモジュール94は、フレームデータDfrmを伸張するタスクを、3つのサブプロセッサ10a~10cに分配する。複数のタスク、すなわち、複数のフレームデータDfrmごとのデータ伸張のタスクを含んで構成されるフレームデコードモジュール94は、そのプログラムコード中に、複数のタスクの各プロセッサへの分配規則が予め記述されている。

【0060】

図4(a)~(d)は、フレームデコードモジュール94に記述されるタスクの分配規則を模式的に示す図である。タスクの分配処理は、OS72の直下で動作するメインプロセッサ20が行ってもよい。この場合、メインプロセッサ20は、フレームデコードモジュール94に関するタスクの実行時に、この分配規則を読み出し、この規則にもとづいて各サブプロセッサ10a~10cにタスクを分配する。

10

【0061】

ここでの圧縮データの伸張処理は、サブプロセッサ10に分配されるタスクは、同一のプログラムコードにもとづく処理であって、処理対象とするフレームデータDfrmごとに実行されるタスクである。

図4(a)~(d)に示される分配規則は、所定の処理単位に含まれるフレームデータDfrmの個数をパラメータとして、それぞれのフレームデータDfrmを伸張処理するタスクを、3つのサブプロセッサ10a~10cのいずれに割り当てるかを記述したものとなっている。

【0062】

20

図4(a)~(d)はそれぞれ、一度に処理すべきフレームデータ、すなわち、処理すべきタスクの個数Mが、1個~4個の場合の分配規則を示している。この分配規則では、M=1~3の範囲において、原則として、処理すべきタスクを、各プロセッサの占有時間が短くなるように、プロセッサの個数方向、言い換えれば処理の並列度が上がるように優先して分配している。すなわち、図4(a)~(c)に示すように、フレームデータDfrmの個数が増加するに従って、占有されるプロセッサの個数はそれにもなって増加する。M=1~3の場合、再生時間にしてM×Tf分のデータが、Tprc=2Tfの時間で処理される。なお、M=1の場合には、再生時間Tfのデータが、Tprc=2Tfの時間で処理されることになるが、処理バランスを考慮して最低でTf分のデータをバッファリングして処理することにより、再生に支障をきたすことはない。

30

【0063】

この分配規則では、例外として、プロセッサがN(Nは2以上の整数)個であり、処理すべきデータがM(Mは自然数)個である場合に、Mが、2以上N以下の整数Lを約数として有するとき、分配規則は、M個のデータに対応したM個のタスクを、L個のプロセッサに並列に分配するように規定される。

具体的にはサブプロセッサ10がN=3個であり、処理すべきフレームデータはM=4個である。このとき、M=4は、約数としてL=2を有している。そこで、図4(d)に示すM=4の場合には、2個のサブプロセッサ10a、10bに、それぞれ2個ずつのタスクを分配し、各サブプロセッサ10a、10bにバッチ処理させる。

【0064】

40

もし、この例外規則を設けない場合には、M=4個の場合、3つのサブプロセッサ10のうち、いずれかひとつが、たとえばサブプロセッサ10aに2つのタスクが投入され、残りのサブプロセッサ10b、10cに、1つにタスクが投入されることになる。したがって、各プロセッサが同期処理を行う場合には、サブプロセッサ10b、10cは、サブプロセッサ10aが2つめのタスク処理を実行する間、処理を行わない状況が発生し、プロセッサの利用効率は低下してしまう。すなわち、3個のサブプロセッサ10が2Tprc(=4Tf)の期間だけ占有されるから、処理コストは、3×2Tprcとなる。これに対して、実質的に処理を行っている時間は、4×Tprcであるから、プロセッサの使用率は、4Tprc/6Tprc=2/3となり、無駄が発生することになる。

【0065】

50

これに対して、図4(d)に示すタスク分配を行った場合、2個のサブプロセッサ10a、10bが2Tprcの期間占有されるから処理コストは $2 \times 2Tprc = 4Tprc$ となり、実際に処理されるデータ量は、 $4 \times Tprc$ であるから、 $4Tprc / 4Tprc = 1$ の高効率でデータ処理が実行できることになる。

さらに、3つのサブプロセッサ10a~10cのうち、1個のサブプロセッサは、占有されないため、フレームデコード部316以外による別のタスクを並列して処理することが可能となる。たとえば、空きプロセッサを利用して並列処理されるタスクとしては、再生中のオーディオデータにもとづいて映像を生成するタスクなど例示される。

#### 【0066】

図5は、図3のフレームデコード部316の機能ブロック図である。フレームデコード部316は、複数のサブプロセッサ10a~10cと、データ数検出部400と、タスク分配部402と、を含む。

サブプロセッサ10a~10cは、分配されたフレームデータDfrmに対して、所定のアルゴリズムにもとづく伸張処理を実行する。伸張処理は、アプリケーション90のフレームデコードモジュール94に記述される。データ数検出部400は、所定の処理単位に含まれるフレームデータDfrmの個数Mを検出する。タスク分配部402は、検出したフレームデータDfrmの個数Mをパラメータとして定められた所定の分配規則に従って、各データをサブプロセッサ10a~10cに分配する。アプリケーション90のフレームデコードモジュール94は、メインプロセッサ20にフレームデータDfrmの個数の検出を実行させ、データ数検出部400として機能させる。また、フレームデコードモジュール94は、メインプロセッサ20に、フレームデコードモジュール94の内部に記述された分配規則にもとづき、フレームデータごとのタスクの分配を実行させ、タスク分配部402として機能させる。

サブプロセッサ10a~10cは、DST方式で圧縮されたフレームデータDfrmを伸張する。伸張されたフレームデータは、DSD形式のビットストリームデータであり、サブプロセッサ10a~10cは、4704バイト単位のパケットデータDpktとして出力する。このパケットデータDpktは、図3に示すデータバッファ318に格納される。

#### 【0067】

このように、本実施の形態に係るSACD再生装置300において、フレームデコード部316は、そのプログラムコードの中に、DST方式で圧縮されたフレームデータDfrmごとの伸張処理をサブプロセッサに実行させるソースコードと、所定の処理単位に含まれるフレームデータDfrmの個数Mを検出するコードと、フレームデータDfrmをサブプロセッサに分配するための分配規則を、フレームデータDfrmの個数Mをパラメータとして定めたコードと、分配規則に応じてサブプロセッサにタスクを分配するコードと、を含んでいる。

#### 【0068】

従来のように、複数のタスクの分配をOSに任せただけの場合、必要なタスク処理が後回しにされる場合が起こりえるため、処理のリアルタイム性が損なわれる場合があった。一方、本実施の形態に係るSACD再生装置300では、データごとのタスクを、どのプロセッサにより実行するかをアプリケーションプログラムのプログラム時に指定できるため、アプリケーション90のプログラマは、タスクを完了するまでの時間を確実に見積もることができる。その結果、特にリアルタイム性が要求される処理に好適に利用可能である。

#### 【0069】

図3に戻り、続くデシメーションフィルタ320以降の処理について説明する。デシメーションフィルタ320は、パケットデータDpktをデータバッファ318から読み出す。デシメーションフィルタ320は、パケットデータDpktの高周波成分を除去し、ダウンサンプリングしてPCM形式のデータに変換する。デシメーションフィルタ320によるフィルタ処理は、畳み込み(convolution)演算、すなわちビットデータの遅延と四則演算が主な処理となるため、処理すべきデータは固定量であってそれほど多くはない

10

20

30

40

50

が、計算量が多いという性質を有している。したがって、このフィルタ処理は、メインプロセッサ20よりも、SIMD演算をサポートし、算術演算、ビット演算が高速なサブプロセッサ10が適しており、アプリケーション90は、フィルタ処理に関するタスクをサブプロセッサ10に実行させるようにコーディングされることが望ましい。デシメーションフィルタ320は、主としてアプリケーション90のフィルタモジュール96およびサブプロセッサ10によって実現される。

サブプロセッサ10dに設けられた専用のローカルメモリ12dの容量に制限がある場合には、パケットデータDpktを分割して、ソフトウェアパイプラインによってバッチ処理させてもよい。

#### 【0070】

図6(a)~(d)は、デシメーションフィルタ320に記述されるタスクの分配規則を模式的に示す図である。タスクの分配処理は、OS72の直下で動作するメインプロセッサ20が行ってもよい。この場合、メインプロセッサ20は、フィルタモジュール96に関するタスクの実行時に、この分配規則を読み出し、この規則にもとづいてサブプロセッサ10dにタスクを分配する。

#### 【0071】

サブプロセッサ10dに分配される処理は、同一のプログラムコードにもとづく処理であって、処理対象とするパケットデータDpktごとに実行されるタスクである。

図6(a)~(d)はそれぞれ、所定の処理単位に含まれる一度に処理すべきパケットデータDpkt、すなわち、処理すべきタスクの個数Pが、1個~4個の場合の分配規則を示している。この分配規則では、 $P = 1 \sim 4$ の範囲において、処理すべきタスクを、占有されるプロセッサの個数が少なくなるように、時間軸方向に優先して分配し、パイプライン方式にてバッチ処理を行う。

#### 【0072】

すなわち、図6(a)~(d)に示すように、処理されるパケットデータDpktの個数Pが増加するに従って、サブプロセッサ10dの占有時間が長くなる。バッチ処理可能なデータ数の上限は、リアルタイム性を損ねない範囲に制限することが望ましい。

たとえば、ひとつのパケットデータDpktの処理に要する時間を $T_{prc}$ とし、ひとつのパケットデータDpktに対応するオーディオデータの実再生時間が $T_{ply}$ であるとすると、 $d = T_{ply} / T_{prc}$ を満たすd個のパケットデータについてパイプライン方式によるバッチ処理を行うことができる。なお、仮に処理すべきデータ数がdを超えた場合には、別のサブプロセッサ10にもタスクを分配してもよい。

#### 【0073】

図6(a)~(d)に示す分配規則は、アプリケーション90のアプリケーションコードとして記述されていてもよい。また、図2のAFW80やバーチャルOS78のレイヤの中に、コードとして分配規則を記述しておき、アプリケーション90のフレームデコードモジュール94が、この分配規則にもとづくタスク処理を指示してもよい。

#### 【0074】

このように、デシメーションフィルタ320においては、占有されるプロセッサの個数が少なくなるように、時間軸方向に優先してタスクが分配される。その結果、同時並列的に実行されるフレームデコード部316の処理に、多くのサブプロセッサ10を割り当てることが可能となる。

#### 【0075】

デシメーションフィルタ320によるフィルタ処理の結果、PCM形式のデータ(以下PCMデータDpcmという)が生成される。PCMデータDpcmは、第1PCMバッファ322に格納される。

続いて、エフェクタ324は、PCMデータDpcmに対して、所定のエフェクト処理を実行する。エフェクト処理としては、フェードイン・フェードアウトや、ボリューム制御、LチャンネルとRチャンネルにデータを振り分けるパンニングなどを実行する。

エフェクト処理は、処理内容は単純であるが、扱うデータ量が多くなり一時的に使用す

10

20

30

40

50

るメモリ空間が大きくなる傾向がある。そこで、エフェクタ324は、上述した各プロセッサの特性を考慮して、エフェクト処理に関するタスクをメインプロセッサ20に分配する。この場合、エフェクタ324は、主としてメインプロセッサ20およびエフェクタモジュール98の協働によって構成される。

ただし、使用可能なサブプロセッサ10の個数に余裕がある場合には、SIMD方式の命令セットをサポートするサブプロセッサ10にタスクを割り当て、バッチ処理してもよい。この場合、エフェクタ324は、サブプロセッサ10およびエフェクタモジュール98で構成される。バッチ処理する場合には、図6(a)~(d)の分配規則を適用してもよい。

【0076】

エフェクタ324によりエフェクト処理されたPCMデータDpcmは、第2PCMバッファ326に格納される。出力部328は、第2PCMバッファ326からPCMデータDpcmを読み出し、音声出力部122に対して出力する。データ出力モジュール99は、データ出力に関するタスクを、サブプロセッサ10に分配する。したがって、出力部328は、主として、アプリケーション90のデータ出力モジュール99およびサブプロセッサ10により構成される。

【0077】

以上、実施の形態に係る情報処理装置200およびSACD再生装置300について説明した。SACD再生装置300を例として実現されるタスクの分配処理技術をまとめると以下の通りである。

【0078】

実施の形態では、アプリケーションプログラムより下位のレイヤが、アプリケーションプログラムに対して、アプリケーションプログラムで実行されるタスクをどのプロセッサにより分配するかを指定できる機能を提供している点がひとつの特徴である。アプリケーションプログラムのプログラムコード中には、複数のタスクのプロセッサへの分配規則を記述し、アプリケーションが各タスクの分配先を指定する構成となっている。

その結果、アプリケーションプログラムのプログラマは、各タスクをどのプロセッサに割り当てるかを、処理内容なプロセッサの特性に応じて、適切に設定可能となっている。実施の形態においては、アプリケーションプログラムにより処理されるタスクとして、「データ読出処理」、「フレームデコード処理」、「デジタルフィルタ処理」、「エフェクタ処理」、「データ出力処理」が例示されている。これらのタスクは、その内容に応じて、サブプロセッサ10もしくはメインプロセッサ20のいずれで実行すべきかが、アプリケーション90により指定されている。

【0079】

さらに、実施の形態では、複数のデータに対して同一の処理を行う場合、たとえば「フレームデコード処理」や、「デジタルフィルタ処理」においては、複数のデータごとのタスクの複数のサブプロセッサ10に対する分配規則が規定される。

【0080】

図7は、第1、第2のタスクの分配方式を模式的に示す図である。図7のマトリクスの横軸は、占有されるプロセッサの個数に対応し、縦軸は、各プロセッサの占有時間を示している。図7は、3個のプロセッサにタスクを割り当てる場合の規則である。第1の分配方式では、処理すべきタスクの数が増加すると、アラビア数字に示す規則にしたがって、横軸方向にタスクを分配する。つまり、第1の方式では、処理すべきデータごとのタスクを、各プロセッサの占有時間が短くなるように、プロセッサの個数方向、言い換えれば処理の並列度が上がるように優先して分配する。図4(a)~(d)に示すように、「フレームデコード処理」では、第1の方式に準拠してタスクの分配を行っている。第1の方式における横軸の長さは、使用可能なプロセッサの個数に応じて規定される。

【0081】

第2の分配方式では、処理すべきタスクの数が増加すると、ローマ数字に示す規則に従って時間軸方向にタスクを分配する。つまり、第2の方式では、処理すべきデータごとの

10

20

30

40

50

タスクを、占有されるプロセッサの個数が少なくなるように、時間軸方向に優先して分配している。図6(a)~(d)に示すように、「デジタルフィルタ処理」では、第2の方式に準拠してタスクの分配を行っている。第2の方式における時間軸(縦軸)の長さ、つまりひとつのプロセッサにいくつのタスクをバッチ処理させるかについては、処理時間に応じて決めればよい。図7の例ではひとつのプロセッサに3つのタスクをバッチ処理させる例が示されている。

#### 【0082】

実施の形態では、プロセッサの利用効率についても着目した。たとえば図7に示される第1の分配規則に従った場合、4個のタスクを処理する場合、3つのプロセッサのうち、1個に対して2つのタスクが分配され、残りの2つのプロセッサには、1つのタスクしか分配されない。したがって、残りの2つのプロセッサは、アラビア数字の5、6で示される領域が空き領域となってしまう。このようなプロセッサの無駄な占有を防止するため、「フレームデコード処理」では、処理すべきデータの個数Mが約数Lを有する場合に、分配規則は、M個のデータに対応したM個のタスクを、L個のプロセッサに並列に分配するように規定している。すなわち、図7において、アラビア数字1、2、3'、4'で示される形態でプロセッサが占有される。この場合、2つのプロセッサが等しい時間占有され、残りの1つのプロセッサは空きとなるため、他の処理を実行させることができる。

#### 【0083】

「フレームデコード処理」、「デジタルフィルタ処理」における分配規則は、所定の処理単位に含まれるデータの個数をパラメータとして、それぞれのデータを処理するタスクを複数のプロセッサのいずれに割り当てるかを記述したものとなっている。すなわちi番目のデータが、どのプロセッサに割り当てられるかが、1対1で対応づけられている。

ただし、タスクの分配規則は、図7に示される第1、第2の方式に限定されるものではなく、アプリケーションプログラムにおいて自由に規定すればよい。

#### 【0084】

また、第1、第2の分配方式を、アプリケーションレベルで規定するのではなく、システムレベル、つまりOSやVOS、あるいはAFWなどのアプリケーションプログラムより下位のレイヤで規定してもよい。つまり、アプリケーション90より下位のレイヤであるコアレイヤ70において、第1の方式と、第2の方式が切り替え可能となる。第1、第2の分配規則を、ライブラリや関数、オブジェクトとして記述し、OSやVOS、AFWにより提供してもよい。この場合、アプリケーションプログラムのコードには、タスクの種類ごとに、第1、第2の分配方式のいずれで処理すべきかを記述すればよいため、アプリケーションの開発効率を改善することができる。実施の形態に当てはめれば、「フレームデコード処理」を規定するフレームデコードモジュール94には、コアレイヤ70に対して第1の方式を指示するコードを記述すればよく、「デジタルフィルタ処理」を規定するフィルタモジュール96には、コアレイヤ70に対して第2の方式を指示するコードを記述すればよい。

この場合、アプリケーションのプログラマは、処理内容に応じて2つの分配方式を選択できるため、ソフトウェアの設計の自由度を高めることができる。

#### 【0085】

以上、本発明を実施の形態をもとに説明した。この実施の形態はあくまで例示であり、それらの各構成要素や各処理プロセスの組合せにいろいろな変形例が可能なこと、またそうした変形例も本発明の範囲にあることは当業者に理解されるところである。

#### 【0086】

実施の形態では、SACD再生を例として、タスクの分配技術について説明したが、タスクの分配技術は、コンピュータを用いたさまざまなシミュレーション、画像処理、その他さまざまな用途に利用することができる。

#### 【図面の簡単な説明】

#### 【0087】

【図1】実施の形態に係る情報処理装置の機能ブロック図である。



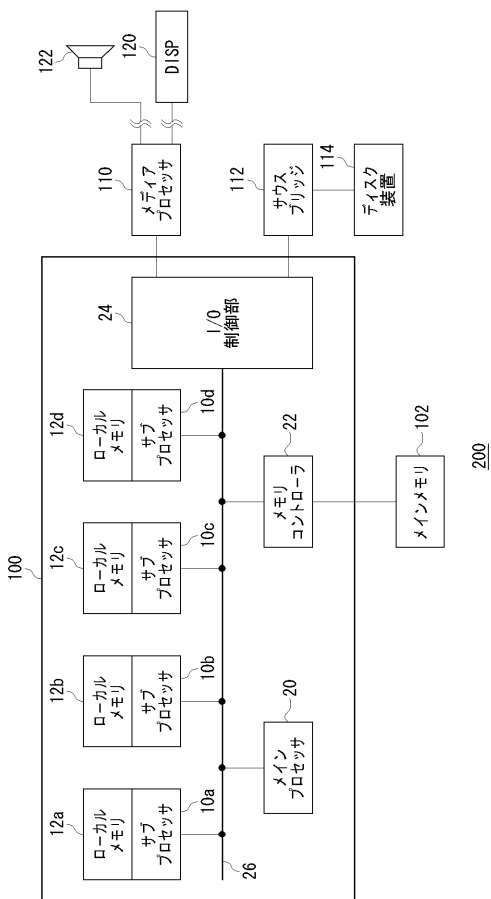
【図2】図1の情報処理装置をソフトウェアの観点から表現したブロック図である。  
 【図3】図1、図2に示す情報処理装置により実現されるSACD再生装置の構成を示すブロック図である。  
 【図4】図4(a)~(d)は、図2のフレームデコードモジュールに記述されるタスクの分配規則を模式的に示す図である。  
 【図5】図3のフレームデコード部の機能ブロック図である。  
 【図6】図6(a)~(d)は、図2のフィルタモジュールに記述されるタスクの分配規則を模式的に示す図である。  
 【図7】第1、第2のタスクの分配方式を模式的に示す図である。

【符号の説明】

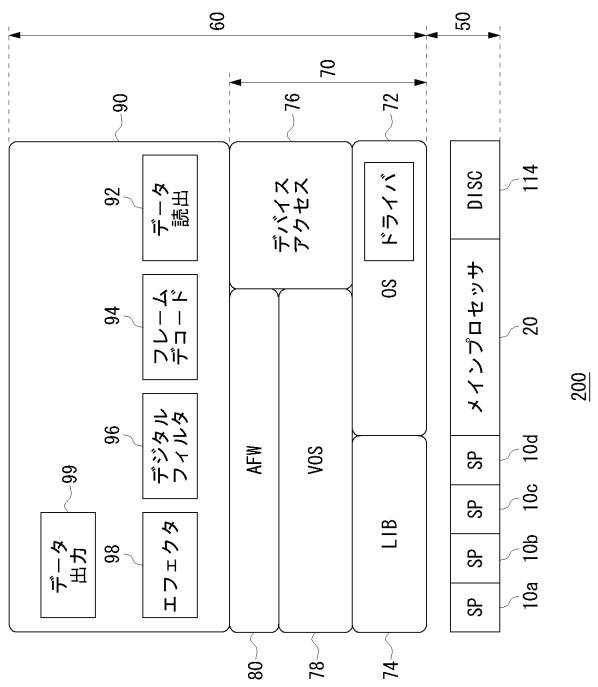
【0088】

200 情報処理装置、 100 統括制御部、 102 メインメモリ、 10 サブプロセッサ、 12 ローカルメモリ、 20 メインプロセッサ、 22 メモリコントローラ、 24 I/O制御部、 26 バス、 50 ハードウェアレイヤ、 60 ソフトウェアレイヤ、 70 コアレイヤ、 72 OS、 74 ライブラリ、 76 デバイスアクセスレイヤ、 78 バーチャルOS、 80 AFW、 90 アプリケーション、 92 データ読出モジュール、 94 フレームデコードモジュール、 96 フィルタモジュール、 98 エフェクタモジュール、 99 データ出力モジュール、 110 メディアプロセッサ、 112 サウスブリッジ、 114 ディスク装置、 120 表示装置、 122 音声出力部、 300 SACD再生装置、 302 デバイスドライバ、 304 暗号復号器、 310 再生処理部、 312 データ読出部、 314 オーディオフレームバッファ、 316 フレームデコード部、 318 データバッファ、 320 デシメーションフィルタ、 322 第1PCMバッファ、 324 エフェクタ、 326 第2PCMバッファ、 328 出力部。

【図1】



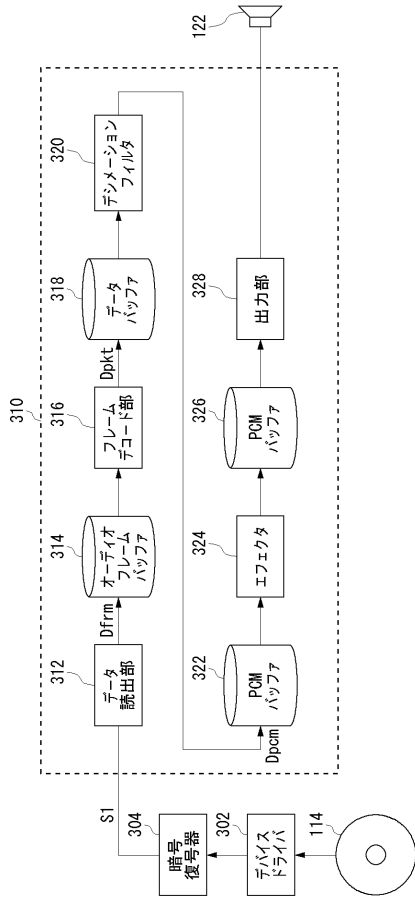
【図2】



10

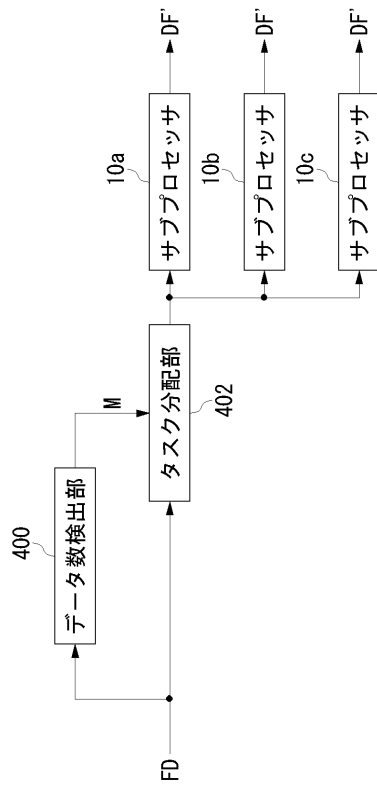
20

【図3】



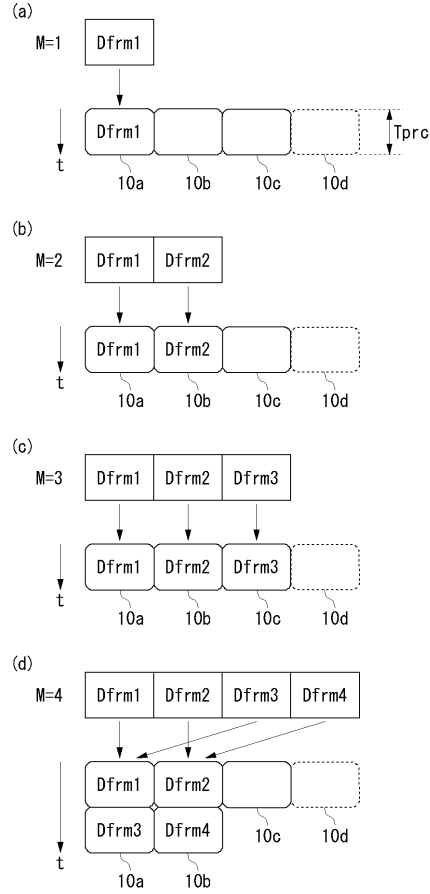
300

【図5】

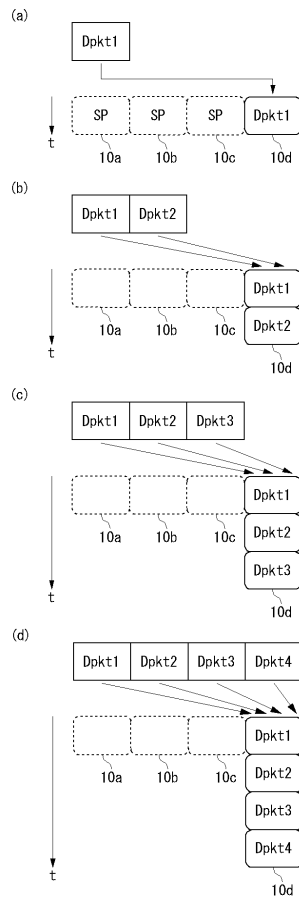


316

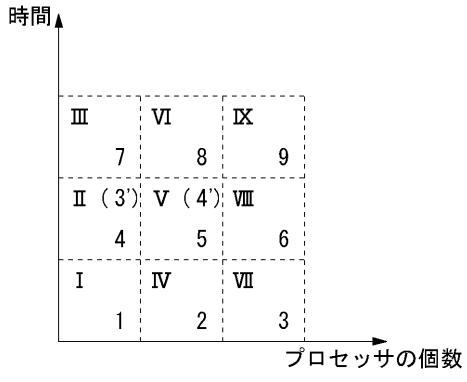
【図4】



【図6】



【図7】



---

フロントページの続き

(72)発明者 村田 乾

東京都港区南青山2丁目6番21号 株式会社ソニー・コンピュータエンタテインメント内

合議体

審判長 吉岡 浩

審判官 富吉 伸弥

審判官 山崎 達也

(56)参考文献 特開2002-358293(JP,A)

(58)調査した分野(Int.Cl., DB名)

G06F9/46