(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7548421号
(P7548421)

(45)発行日　令和6年9月10日(2024.9.10)

(24)登録日　令和6年9月2日(2024.9.2)

(51)国際特許分類
Ｇ０６Ｆ　9/50 (2006.01)

ＦＩ
Ｇ０６Ｆ　9/50　１５０Ｃ

請求項の数　8　（全17頁）

(21)出願番号　特願2023-515978(P2023-515978)
(86)(22)出願日　令和3年4月22日(2021.4.22)
(86)国際出願番号　PCT/JP2021/016313
(87)国際公開番号　WO2022/224409
(87)国際公開日　令和4年10月27日(2022.10.27)
審査請求日　令和5年8月18日(2023.8.18)

(73)特許権者　000004226
日本電信電話株式会社
東京都千代田区大手町一丁目５番１号
(74)代理人　110002147
弁理士法人酒井国際特許事務所
(72)発明者　横野 智也
東京都千代田区大手町一丁目５番１号
日本電信電話株式会社内
(72)発明者　山部 芳朗
東京都千代田区大手町一丁目５番１号
日本電信電話株式会社内
(72)発明者　石崎 晃朗
東京都千代田区大手町一丁目５番１号
日本電信電話株式会社内
審査官　田中 幸雄

最終頁に続く

(54)【発明の名称】　アクセラレータ制御システム、アクセラレータ制御方法およびアクセラレータ制御プログラム

(57)【特許請求の範囲】

【請求項１】

アクセラレータ制御装置と、複数のアクセラレータとを有し、

前記アクセラレータ制御装置は、

演算処理の対象のデータの所在と、演算処理を行うアクセラレータの情報と、該アクセラレータの演算処理の内容を指定する情報とを含む制御用データを記憶する記憶部と、

前記アクセラレータにより演算処理済みの制御用データが前記記憶部に格納されている場合に、該アクセラレータによる演算処理の完了を判定する判定部と、を有し、

前記アクセラレータは、

前記記憶部から前記制御用データを取得する取得部と、

取得された前記制御用データに含まれる前記演算処理の対象のデータの所在とアクセラレータの演算処理の内容を指定する情報に従って、前記演算処理の対象のデータに対する演算処理を行う演算部と、

前記演算処理が完了した場合であって、前記制御用データに、転送先のアクセラレータの情報が含まれていない場合に、前記制御用データを前記記憶部に格納する格納部と、

前記制御用データに、転送先のアクセラレータの情報が含まれている場合に、該アクセラレータに該制御用データを転送する転送部と、

他のアクセラレータから前記制御用データを受信した場合に、該制御用データに含まれる前記演算処理の対象のデータの所在と、該演算処理の内容を指定する情報に従って、前記演算処理の対象のデータに対する演算処理を行う第２の演算部と、を有する

ことを特徴とするアクセラレータ制御システム。
【請求項２】
他のアクセラレータからの前記制御用データの受信可否を確認する確認部を、さらに有することを特徴とする請求項１に記載のアクセラレータ制御システム。
【請求項３】
前記転送部は、所定の時間内に転送先の前記アクセラレータから前記制御用データの受信可否が通知されない場合に、前記制御用データを再送することを特徴とする請求項２に記載のアクセラレータ制御システム。
【請求項４】
前記格納部は、転送先の前記アクセラレータに前記制御用データが転送された場合に、前記演算処理対象のデータに対する前記演算処理の結果を前記記憶部に格納し、
転送先の前記アクセラレータにおいて、前記第２の演算部が、前記記憶部に格納された前記演算処理の結果を読み込んで前記演算処理の対象とする、
ことを特徴とする請求項１に記載のアクセラレータ制御システム。
【請求項５】
前記記憶部に前記制御用データが有るか否かを監視する監視部をさらに有することを特徴とする請求項１に記載のアクセラレータ制御システム。
【請求項６】
前記判定部は、前記記憶部に前記演算処理済みの制御用データが有るか否かを監視することを特徴とする請求項１に記載のアクセラレータ制御システム。
【請求項７】
アクセラレータ制御装置と、複数のアクセラレータとを有するアクセラレータ制御システムが実行するアクセラレータ制御方法であって、
前記アクセラレータ制御システムは、演算処理の対象のデータの所在と、演算処理を行うアクセラレータの情報と、該アクセラレータの演算処理の内容を指定する情報とを含む制御用データを記憶する記憶部を有し、
前記記憶部から前記制御用データを取得する取得工程と、
取得された前記制御用データに含まれる前記演算処理の対象のデータの所在とアクセラレータの動作を指定する情報に従って、前記演算処理の対象のデータに対する演算処理を行う演算工程と、
前記演算処理が完了した場合であって、前記制御用データに、転送先のアクセラレータの情報が含まれていない場合に、前記制御用データを前記記憶部に格納する格納工程と、
演算処理済みの制御用データが前記記憶部に格納されている場合に、演算処理の完了を判定する判定工程と、
前記制御用データに、転送先のアクセラレータの情報が含まれている場合に、該アクセラレータに該制御用データを転送する転送工程と、
他のアクセラレータから前記制御用データを受信した場合に、該制御用データに含まれる前記演算処理の対象のデータの所在と、該演算処理の内容を指定する情報に従って、前記演算処理の対象のデータに対する演算処理を行う第２の演算工程と、
を含むことを特徴とするアクセラレータ制御方法。
【請求項８】
コンピュータを請求項１～６のいずれか１項に記載のアクセラレータ制御システムとして機能させるためのアクセラレータ制御プログラム。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、アクセラレータ制御システム、アクセラレータ制御方法およびアクセラレータ制御プログラムに関する。
【背景技術】
【０００２】

従来、演算処理システムに複数の演算加速器（アクセラレータ）を組み込んで処理を高速化する技術が知られている。半導体の微細化技術の限界により、プロセッサの動作周波数の向上や演算機の高集積化が困難となってきたことから、この技術により、特定の処理に特化してアクセラレータに処理をオフロードして、高性能な計算を実現する。さらに、複数台のアクセラレータを協調して動作させることにより、複数の演算で構成される処理の高速化を実現する（非特許文献１～３参照）。

【０００３】

具体的には、あるアクセラレータの演算結果を別のアクセラレータに送信するアクセラレータ間通信を制御することが必要となる。そこで、転送元アクセラレータにおける演算処理の終了後に、ＣＰＵがアクセラレータ間のデータ通信の制御を行う。ＣＰＵは、転送先アクセラレータにデータ受付の準備を要求する制御信号を送信した後に、転送元アクセラレータに転送開始の制御信号を送信することで、アクセラレータ間のデータ通信を駆動する。アクセラレータ間のデータ通信の後、転送先アクセラレータがデータに対する演算処理を行う。

【先行技術文献】

【非特許文献】

【０００４】

【文献】J. Yang, D. B. Minturn, F. Hady, “When Poll is Better than Interrupt”, FAST, vol.12, pp.3-3, [online], 2012年, ［2021年3月29日検索］, インターネット〈URL：https://www.usenix.org/system/files/conference/fast12/yang.pdf〉

【文献】“CUDA Toolkit Documentation v11.2.2”, [online], NVIDIA, ［2021年3月29日検索］, インターネット〈URL：https://docs.nvidia.com/cuda/〉

【文献】K. Gulati, S. P. Khatri, “GPU Architecture and the CUDA Programming Model”, Hardware acceleration of EDA algorithms, pp.23-30, [online], Springer, ［2021年3月29日検索］, インターネット〈URL：https://link.springer.com/content/pdf/10.1007%2F978-1-4419-0944-2_3.pdf〉

【発明の概要】

【発明が解決しようとする課題】

【０００５】

しかしながら、従来技術では、複数のアクセラレータを用いてリアルタイム処理を高速化することが困難な場合がある。例えば、アクセラレータはＣＰＵによって制御されるため、アクセラレータ間のデータ転送の処理が発生するたびに、転送元アクセラレータ・転送先アクセラレータの排他制御や処理完了の割込み通知、同期制御が発生する。これらの処理がオーバヘッドとなり、遅延時間が増大する。また、多数のアクセラレータを用いる場合には、上記のようなオーバヘッドに加え、バスの仕様によりアクセラレータ間で直接データ転送が不可能な場合が存在するため、スケーラビリティが低下する。したがって、アクセラレータをリアルタイム処理や多様な処理の領域で用いることが困難である。

【０００６】

本発明は、上記に鑑みてなされたものであって、複数のアクセラレータを用いてリアルタイム処理を高速化することを目的とする。

【課題を解決するための手段】

【０００７】

上述した課題を解決し、目的を達成するために、本発明に係るアクセラレータ制御システムは、アクセラレータ制御装置と、複数のアクセラレータとを有し、前記アクセラレータ制御装置は、演算処理の対象のデータの所在と、演算処理を行うアクセラレータの情報と、該アクセラレータの演算処理の内容を指定する情報とを含む制御用データを記憶する記憶部と、前記アクセラレータにより演算処理済みの制御用データが前記記憶部に格納されている場合に、該アクセラレータによる演算処理の完了を判定する判定部と、を有し、前記アクセラレータは、前記記憶部から前記制御用データを取得する取得部と、取得され

た前記制御用データに含まれる前記演算処理の対象のデータの所在とアクセラレータの演算処理の内容を指定する情報に従って、前記演算処理の対象のデータに対する演算処理を行う演算部と、前記演算処理が完了した場合であって、前記制御用データに、転送先のアクセラレータの情報が含まれていない場合に、前記制御用データを前記記憶部に格納する格納部と、前記制御用データに、転送先のアクセラレータの情報が含まれている場合に、該アクセラレータに該制御用データを転送する転送部と、他のアクセラレータから前記制御用データを受信した場合に、該制御用データに含まれる前記演算処理の対象のデータの所在と、該演算処理の内容を指定する情報に従って、前記演算処理の対象のデータに対する演算処理を行う第２の演算部と、を有することを特徴とする。

【発明の効果】

【０００８】

本発明によれば、複数のアクセラレータを用いてリアルタイム処理を高速化することが可能となる。

【図面の簡単な説明】

【０００９】

【図１】図１は、本実施形態のアクセラレータ制御システムの概略構成を例示する模式図である。

【図２】図２は、本実施形態のアクセラレータ制御システムを説明するための図である。

【図３】図３は、メタデータのデータ構成を例示する図である。

【図４】図４は、メタデータバッファの構造を例示する図である。

【図５】図５は、データバッファを説明するための図である。

【図６】図６は、アクセラレータ間通信を説明するための図である。

【図７】図７は、アクセラレータにおけるアクセラレータ制御処理手順を示すフローチャートである。

【図８】図８は、アクセラレータ制御プログラムを実行するコンピュータの一例を示す図である。

【発明を実施するための形態】

【００１０】

以下、図面を参照して、本発明の一実施形態を詳細に説明する。なお、この実施形態により本発明が限定されるものではない。また、図面の記載において、同一部分には同一の符号を付して示している。

【００１１】

［アクセラレータ制御システムの構成］

図１は、本実施形態のアクセラレータ制御システムの概略構成を例示する模式図である。図１に示すように、アクセラレータ制御システム１は、アクセラレータ制御装置１０と複数のアクセラレータ２０とを有する。アクセラレータ制御装置１０とアクセラレータ２０とは、バス配線等の通信回線を介して相互にデータ通信が可能である。

【００１２】

アクセラレータ制御装置１０は、ＣＰＵ（Central Processing Unit）等の汎用プロセッサを用いて実現される情報処理装置である。アクセラレータ制御装置１０において、ユーザ空間上で、ユーザが定義した処理であるアプリケーションが実行される。またカーネル空間上で、アクセラレータ２０等の物理デバイスの操作や管理、ユーザ空間処理における物理デバイスの抽象化処理が行われる。

【００１３】

ユーザメモリ空間、カーネルメモリ空間は、ＲＡＭ（Random Access Memory）、フラッシュメモリ（Flash Memory）等の半導体メモリ素子、または、ハードディスク、光ディスク等の記憶装置によって実現され、記憶部として機能する。記憶部には、アクセラレータ制御装置１０を動作させる処理プログラムや、処理プログラムの実行中に使用されるデータなどが予め記憶され、あるいは処理の都度一時的に記憶される。

【００１４】

汎用プロセッサを用いて実現される制御部は、メモリに記憶された処理プログラムを実行することにより、図１に例示するように、デバイスドライバのメモリ管理機能１２、メタデータ制御機能１３、後述するＡＰＩ１４等の各機能部として機能する。なお、制御部は、不図示の他の機能部を備えてもよい。

【００１５】

アクセラレータ制御装置１０では、演算処理の対象のデータ（以下、処理データとも記す）が、ユーザ空間上のユーザメモリ空間のデータバッファ１１ａに配置される。また、演算処理の対象のデータの所在と、アクセラレータの演算処理の内容を指定する情報とを含む制御用データ（以下、メタデータとも記す）が、カーネル空間上のカーネルメモリ空間のメタデータバッファ１１ｂに配置される。

【００１６】

アクセラレータ２０は、ＧＰＵ（Graphics Processing Unit）、ＦＰＧＡ（Field Programmable Gate Array）、ＤＰＵ（Data Processing Unit）等で実現される演算加速器であり、図１に例示するように、メタデータアクセス機構２１、機能制御ブロック２２、データ転送機能２３、演算機能２４等の各機能部として機能する。

【００１７】

アクセラレータ２０は、処理データとメタデータとを読み込んで、演算処理のオフロードを実現する。具体的には、アクセラレータ２０は、メタデータバッファ１１ｂを自律的なＰｏｌｌｉｎｇにより、メタデータが配置されているか否かを監視している。アクセラレータ２０は、メタデータが配置されている場合に、配置されているメタデータを読み込んで、解析してアクセラレータ２０を駆動する。また、アクセラレータ２０は、処理データに対する演算処理の結果をデータバッファ１１ａに格納する。

【００１８】

なお、アクセラレータ制御装置１０は、アクセラレータ２０とは非同期に、次の処理に進めることが可能である。アクセラレータ２０は、演算処理が完了した場合にメタデータをメタデータバッファ１１ｂに配置する。これにより、アクセラレータ制御装置１０は、演算処理が完了したとみなす。また、アクセラレータ制御装置１０は、データバッファ１１ａから処理データに対する演算処理の結果を取得して、次の処理に進める。

【００１９】

具体的には、アクセラレータ制御装置１０は、デバイスドライバに実装されたメタデータ制御機能１３を介して、カーネルメモリ空間上のメタデータバッファ１１ｂに処理データの所在や演算処理の内容を含むメタデータを配置することで、アクセラレータ２０の動作を決定する。

【００２０】

アクセラレータ制御装置１０は、アクセラレータ制御装置１０とアクセラレータ２０とが参照可能なデータバッファ１１ａを、メモリ管理機能１２を介してユーザメモリ空間に作成し、処理データを配置する。

【００２１】

アクセラレータ２０は、メタデータアクセス機構２１を有する。メタデータアクセス機構２１は、カーネルメモリ空間上のメタデータバッファ１１ｂをＰｏｌｌｉｎｇする機能、メタデータの読込機能、メタデータバッファへのメタデータの書込機能を備え、アクセラレータ制御装置１０からの明示的な制御信号によらず自律的にメタデータを読み込むことが可能である。

【００２２】

メタデータバッファ１１ｂのアドレス等の情報は、デバイスドライバのロードの際に、アクセラレータ初期化機能によってアクセラレータ２０に通知される。

【００２３】

アクセラレータ２０は、メタデータバッファ１１ｂからメタデータを読み込み、メタデータ解析機能２２ａがデータを解析し、その記述に従って動作する。アクセラレータ２０は、メタデータに含まれる処理データのアドレスや長さの情報を元に、データ転送機能２

３におけるデータ読込機能２３ａを駆動し、処理データをデータバッファ１１ａから読みだして、演算機能２４での演算に用いる。演算機能２４では、メタデータに記述されている演算情報を元に、演算処理が実行される。メタデータ解析機能２２ａは、解析結果を演算機能２４に渡し、演算機能２４による演算処理内容を設定する。

【００２４】

演算処理の終了後には、演算処理の結果が演算機能２４からデータ書込機能２３ｂに転送され、データ転送制御機能２２ｂにより指定されたデータバッファに書き込まれる。その後、データ転送が終了した旨の通知がデータ転送機能２３から完了通知機能２２ｃに送信され、メタデータアクセス機構２１の書込機能によりメタデータがメタデータバッファ１１ｂに書き込まれ、アクセラレータ２０の動作が完了する。

【００２５】

このように、アクセラレータ制御システム１では、データ転送や演算処理を制御するためのメタデータをメインメモリに配置して、アクセラレータ２０が自律的にＰｏｌｌｉｎｇおよび読み込みを行う。これにより、割込み等のアクセラレータ制御のオーバヘッドを低減し、リアルタイム処理が可能となる。

【００２６】

また、アクセラレータ２０が処理データをアクセラレータ制御装置１０のユーザ空間に直接転送可能であり、アクセラレータ制御装置１０とアクセラレータ２０とが非同期に駆動可能であるため、遅延もＣＰＵ使用率も削減される。

【００２７】

さらに、アクセラレータ２０がアクセラレータ制御装置１０のメインメモリにアクセスして動作可能であるため、複数のアクセラレータ２０への拡張が可能であり、スケーラビリティが大きく向上する。

【００２８】

また、複数のアクセラレータ２０を用いて演算処理が行われる場合には、メタデータには、さらに、演算を行うアクセラレータの情報、アクセラレータでのオペレーション、読み込む処理データの所在等の情報が含まれる。これらの情報を元に、転送元アクセラレータ２０ａが転送先アクセラレータ２０ｂの設定や制御情報（以下、メタデータ、またはアクセラレータ間メタデータと記す）を作成し、アクセラレータ制御装置１０を介さずにアクセラレータ間で直接、メタデータのデータ通信を行う。

【００２９】

転送元アクセラレータ２０ａは、演算処理の終了後、演算処理結果をカーネルメモリ空間上の中間データバッファ１１ｃに出力し、転送先アクセラレータ２０ｂにメタデータを送信する。

【００３０】

転送先アクセラレータ２０ｂは、現在のアクセラレータ２０ｂの状態から、送信されたメタデータの受信が可能か否かを判定する。メタデータを受信可能な場合には、転送先アクセラレータ２０ｂは、転送元アクセラレータ２０ａに受信完了を通知する。転送先アクセラレータ２０ｂから受信完了を通知されるまで、転送元アクセラレータ２０ａは待機状態であり、所定の時間以内に受信完了の通知がない場合には、再度、メタデータを転送先アクセラレータ２０ｂに送信する。

【００３１】

転送先アクセラレータ２０ｂは、受信したメタデータを解析することにより、中間データバッファ１１ｃの中間データを読み込んで、演算処理を実行する。

【００３２】

具体的には、複数のアクセラレータ２０で連携する場合には、メタデータに複数のアクセラレータ２０で動作するための情報が記述されている。アクセラレータ２０は、メタデータ解析機能２２ａで読み込んだメタデータを解析することで、演算処理終了後にアクセラレータ間のデータ通信を行う。

【００３３】

まず、転送元アクセラレータ２０ａでは、演算処理終了後に、カーネルメモリ空間上の中間データバッファ１１ｃに、転送先アクセラレータ２０ｂに転送するための処理データの演算結果を出力する。

【００３４】

その後、転送元アクセラレータ２０ａは、転送先アクセラレータ通知機能２２ｄにより、バスを介して転送先アクセラレータ２０ｂに、演算処理結果の位置情報や演算内容、アクセラレータの連携に関する情報を含むメタデータを送信する。

【００３５】

転送先アクセラレータ２０ｂは、アクセラレータ間メタデータ通信機構２１ｄの受信機能により、メタデータを受け取る。メタデータは、解析機能に送信され、アクセラレータの状態に基づいて、メタデータを受信するか否かを判断する。その際に、アクセラレータ間メタデータ通信機構２１ｄの受信通知機能が、受信したか否かの情報を生成し、転送元アクセラレータ２０ａに返送する。

【００３６】

待機状態の転送元アクセラレータ２０ａは、転送先アクセラレータ２０ｂから受信完了の通知を受け取ったことにより、転送が完了したとみなして、次の処理に移行する。所定の時間以内に受信完了の通知がない場合に、転送元アクセラレータ２０ａは、再度、メタデータを送信する。

【００３７】

転送先アクセラレータ２０ｂは、転送されたメタデータを受信可能であれば、アクセラレータ間メタデータ通信機構２１ｄの解析機能から読み込んだメタデータを基に、演算処理を実行する。

【００３８】

このように、アクセラレータ制御システム１は、複数のアクセラレータ２０を用いた処理において、アクセラレータ制御装置１０を介さずに実行されるため、転送元アクセラレート／転送先アクセラレートの排他処理や処理完了の割込み通知、同期制御等のオーバヘッドを低減して、スケーラビリティを低下させない。また、通信処理においても、アクセラレータ制御装置１０は、各アクセラレータ２０と非同期に駆動可能であるため、それぞれの演算リソースを有効に活用することが可能である。また、処理データの受け渡しにメインメモリを使用することにより、アクセラレータ制御装置１０の制御がなくても複数のアクセラレータ２０からデータソースを参照可能である。そのためバスの仕様に依らずにデータ転送が可能であり、スケーラブルかつ幅広いワークロードに対応したシステムを構築することが可能である。

【００３９】

［アクセラレータ制御装置］

図２は、本実施形態のアクセラレータ制御システムを説明するための図である。図２に例示するように、アクセラレータ制御装置１０において、ＡＰＩ１４は、アプリケーションから呼び出し可能な機能群であり、アクセラレータ２０の動作に必要な機能を提供する。例えば、ＡＰＩは、データバッファ確保機能、データバッファ解放機能、メタデータ作成機能、メタデータ送信機能、演算完了感知機能、通信用メタデータ作成機能として機能する。

【００４０】

データバッファ確保機能は、アクセラレータ２０とデータの送受信を行うための記憶領域であるデータバッファ１１ａを生成する。データバッファ確保機能は、メモリ管理機能１２を呼び出して、メモリ管理機能１２からデータバッファ１１ａの領域情報を取得する。

【００４１】

データバッファ解放機能は、データバッファ１１ａを削除する。データバッファ解放機能は、メモリ管理機能１２を呼び出して、解放するデータバッファ１１ａの領域情報を通知して、当該領域を解放する。

【００４２】

メタデータ作成機能は、アクセラレータ２０の制御情報が付加された制御用データであるメタデータを作成する。メタデータ送信機能は、メタデータ作成機能が作成したメタデータをメタデータ制御機能１３に送信する。演算完了感知機能は、メタデータ制御機能１３を呼び出して、メタデータ制御機能１３の返り値により、任意のアクセラレータ２０の処理が終了しているかを確認する。通信用メタデータ作成機能は、アクセラレータ間通信を行うための通信路情報を作成する。

【００４３】

デバイスドライバは、カーネル空間上に存在し、特定のデバイスを扱うためのメモリ管理機能１２、メタデータ制御機能１３等の機能群を提供する。

【００４４】

メモリ管理機能１２は、ユーザ空間上のＡＰＩ１４から呼び出され、データバッファ１１ａの領域の確保、解放、管理を行う。

【００４５】

メタデータバッファ１１ｂは、アクセラレータ２０とメタデータの送受信を行うための記憶領域であり、アクセラレータ初期化機能により、アクセラレータ２０の数に応じて生成される。例えば、メタデータバッファ１１ｂは、演算処理の対象のデータの所在と、演算処理を行うアクセラレータ２０の情報と、該アクセラレータ２０の演算処理の内容を指定する情報とを含む制御用データを記憶する。

【００４６】

メタデータバッファ１１ｂは、アクセラレータ２０にメタデータを送信するＲＱ（Request Queue）バッファと、アクセラレータ２０からメタデータを受信するＣＱ（Completion Queue）バッファからなる。

【００４７】

メタデータ制御機能１３は、ユーザ空間上のＡＰＩ１４から呼び出され、メタデータバッファ１１ｂのＲＱバッファ／ＣＱバッファへの読み込み、書き込み、管理を行う。メタデータ制御機能１３は、Ｐｏｌｌｉｎｇ機能１３ａと書込機能１３ｂとを有する。

【００４８】

Ｐｏｌｌｉｎｇ機能１３ａは、ＣＱバッファをＰｏｌｌｉｎｇし、演算の完了を確認する。すなわち、Ｐｏｌｌｉｎｇ機能１３ａは、判定部として機能して、アクセラレータ２０により演算処理済みのメタデータがメタデータバッファ１１ｂに格納されているか否かを監視する。そして、Ｐｏｌｌｉｎｇ機能１３ａは、アクセラレータ２０により演算処理済みのメタデータがメタデータバッファ１１ｂに格納されている場合に、該アクセラレータ２０による演算処理の完了を判定する。また、書込機能１３ｂは、ＲＱバッファに書き込みを行う。

【００４９】

［アクセラレータ］

アクセラレータ２０において、メタデータアクセス機構２１は、カーネル空間上のメタデータバッファ１１ｂにアクセスする機能として、Ｐｏｌｌｉｎｇ機能２１ａ、読込機能２１ｂ、書込機能２１ｃ、アクセラレータ間メタデータ通信機構２１ｄを含む。

【００５０】

Ｐｏｌｌｉｎｇ機能２１ａは、メタデータバッファ１１ｂのＲＱバッファをＰｏｌｌｉｎｇして、ＲＱバッファにメタデータが有るか否かを確認する。すなわち、Ｐｏｌｌｉｎｇ機能２１ａは、監視部として機能して、メタデータバッファ１１ｂにメタデータが有るか否かを監視する。

【００５１】

読込機能２１ｂは、取得部として機能して、メタデータバッファ１１ｂからメタデータを取得する。すなわち、読込機能２１ｂは、Ｐｏｌｌｉｎｇ機能２１ａによりＲＱバッファにメタデータが有ることが確認された場合に、メタデータを読み込み、機能制御ブロック２２に転送する。

【００５２】

書込機能２１ｃは、メタデータバッファ１１ｂのＣＱバッファにメタデータを書き込む。すなわち、書込機能２１ｃは、格納部として機能して、演算処理が完了した場合であって、メタデータに、転送先のアクセラレータ２０ｂの情報が含まれていない場合に、メタデータをメタデータバッファ１１ｂに格納する。

【００５３】

機能制御ブロック２２は、アクセラレータ２０の各機能部に対する必要なデータの共有、駆動・休止タイミングを制御する。機能制御ブロック２２は、メタデータ解析機能２２ａ、データ転送制御機能２２ｂ、完了通知機能２２ｃ、転送先アクセラレータ通知機能２２ｄを含む。

【００５４】

メタデータ解析機能２２ａは、メタデータアクセス機構２１から送信されたメタデータから、オペレーション、データバッファ１１ａのアドレス・長さの情報を抽出する。データ転送制御機能２２ｂは、データ転送機能２３におけるデータの読み込み、書き込みを制御する機能を提供する。完了通知機能２２ｃは、メタデータアクセス機構２１にＣＱバッファに書き込むメタデータを転送する。

【００５５】

演算機能２４は、演算部として機能して、取得されたメタデータに含まれる演算処理の対象の処理データの所在とアクセラレータ２０の演算処理の内容を指定する情報に従って、演算処理の対象の処理データに対する演算処理を行う。

【００５６】

具体的には、演算機能２４は、入力制御機能、演算回路、出力制機能を含む。演算回路は複数の独立した演算回路を有し、それぞれに対して演算を定義することが可能である。入力制御機能は、機能制御ブロック２２から送信されるメタデータの情報を基に、データ転送機能２３のデータ読込機能２３ａから送信されたしデータを適切な演算回路に入力する。

【００５７】

出力制御機能は、演算回路から出力された処理データに対する演算処理の結果を、適切なタイミングでデータ転送機能２３のデータ書込機能２３ｂに転送する。

【００５８】

データ転送機能２３のデータ書込機能２３ｂは、格納部として機能して、処理データに対する演算処理の結果をデータバッファ１１ａに格納する。

【００５９】

次に、図３は、メタデータのデータ構成を例示する図である。図３に示すように、メタデータは、３２ｂｉｔのタスクＩＤ、３２ｂｉｔのオペレーション、３２ｂｉｔの読み出しデータの長さ、６４ｂｉｔの読み出しデータのアドレス、３２ｂｉｔの書込みデータの長さ、６４ｂｉｔの書込みデータのアドレスの計２５６ｂｉｔのデータで構成される。

【００６０】

また、図４は、メタデータバッファの構造を例示する図である。図４に示すように、図３に示したメタデータが、ＲＱバッファ／ＣＱバッファに格納される。図４に示すＲＱバッファのｈｅａｄ情報は、メタデータ制御機能１３により制御される。メタデータ制御機能１３は、ＡＰＩ１４のメタデータ送信機能から送信されたメタデータを、図３に示した構成に成形し、ＲＱバッファに格納した後、ｈｅａｄの情報を次に位置にずらす。

【００６１】

同様に、メタデータ制御機能１３は、ＣＱバッファにおいても、ｈｅａｄの情報を用いて、ＡＰＩ１４の演算完了感知機能に呼び出された場合に、ＣＱバッファからメタデータを読みだして、ｈｅａｄを次の位置にずらす。

【００６２】

また、図５は、データバッファを説明するための図である。メモリ管理機能１２は、図５に例示するテーブルで、データバッファ１１ａのアドレスと状態とを管理する。図５に示すように、テーブルでは、データバッファ１１ａの物理アドレスとそのアドレスの状態

とが管理されている。アクセラレータ２０に対するメタデータをセットする際に、ｋｅｙを参照することにより、所望のデータバッファ１１ａの物理アドレスを取得して、読み出しデータアドレスと書き出しデータアドレスとをセットする。これにより、カーネル空間およびユーザ空間における仮想アドレスの違いを吸収して管理できる。

【００６３】

図２の説明に戻る。上記した書込機能２１ｃは、転送部として機能して、メタデータに、転送先のアクセラレータ２０ｂの情報が含まれている場合に、該アクセラレータ２０ｂに該メタデータを転送する。

【００６４】

アクセラレータ間メタデータ通信機構２１ｄは、受信機能、解析機能、受信通知機能を含み、アクセラレータ間通信に必要な機能を提供する。受信機能は、確認部として機能して、他のアクセラレータ２０ａからのメタデータの受信可否を確認する。すなわち、受信機能は、他のアクセラレータ２０（転送元アクセラレータ２０ａ）から送られてきたメタデータの受信可否を通知するためのデータを生成し、転送元アクセラレータ２０ａに送信する。

【００６５】

機能制御ブロック２２の転送先アクセラレータ通知機能２２ｄは、転送先アクセラレータ２０ｂに、メタデータを通知する機能を提供する。メタデータ解析機能２２ａは、メタデータアクセス機構２１から送られたメタデータから、オペレーション、データバッファのアドレス・長さの情報を抽出する。加えて、メタデータ解析機能２２ａは、メタデータから転送先アクセラレータ２０ｂの情報を抽出する。また、メタデータ解析機能２２ａは、転送先アクセラレータ２０ｂからの受信完了の通知がない場合に、転送先アクセラレータ通知機能２２ｄおよび書込機能２１ｃを介してメタデータを再送する。書込機能２１ｃが、所定の時間内に転送先のアクセラレータ２０ｂからメタデータの受信可否が通知されない場合に、メタデータを再送する。

【００６６】

また、転送先のアクセラレータ２０ｂにおいて、演算機能２４は、第２の演算部として機能して、後述するように、他のアクセラレータ２０ａからメタデータを受信した場合に、演算機能２４が、メタデータに含まれる演算処理の対象の中間データの所在と、該演算処理の内容を指定する情報に従って、演算処理の対象の中間データに対する演算処理を行う。

【００６７】

データ転送機能２３のデータ書込機能２３ｂは、転送先のアクセラレータ２０ｂにメタデータが転送された場合に、演算処理対象の処理データに対する演算処理の結果の中間データを中間データバッファ１１ｃに格納する。転送先のアクセラレータ２０ｂでは、演算機能２４が、中間データバッファ１１ｃに格納された演算処理の結果を読み込んで演算処理の対象とする。

【００６８】

ここで、図６は、アクセラレータ間通信を説明するための図である。アクセラレータ制御システム１では、メタデータを用いて、アクセラレータ間通信を実現する。図６（ａ）に示すように、アクセラレータ間通信の通信路は、各アクセラレータ２０を頂点V_0、V_1、V_2、…とするグラフで表現する。このグラフを隣接行列として表現することにより、データとして取り扱う。例えば、図６（ａ）に例示したグラフは、図６（ｂ）に示すように、隣接行列として表現できる。

【００６９】

これらの情報は、図６（ｃ）に示すデータ構造にすることで、アクセラレータ２０で扱うことができる。図６（ｃ）に示すデータ構造では、先頭の８ｂｉｔは行列の大きさを表し、アクセラレータ２０が行列本体のデータの大きさを知ることができる。続く２５６ｂｉｔは行列本体のデータであり、アクセラレータ間の接続情報が含まれる。続くフィールドには、転送先アクセラレータ２０ｂを駆動するための演算内容や処理データの所在等の

情報が記述される。
【００７０】
［アクセラレータ制御処理］
次に、図７は、アクセラレータ２０におけるアクセラレータ制御処理手順を示すフローチャートである。図７のフローチャートは、例えば、アプリケーションにより開始が指示されたタイミングで開始される。
【００７１】
図７のフローチャートは、例えば、所定の間隔で開始される。アクセラレータ２０は、メタデータバッファ１１ｂのＲＱバッファをＰｏｌｌｉｎｇにより監視している（ステップＳ１）。また、アクセラレータ２０は、他のアクセラレータ２０からメタデータが転送されてきているか否かを確認する（ステップＳ２）。
【００７２】
他のアクセラレータ２０からメタデータが転送されていない場合には（ステップＳ２、Ｎｏ）、アクセラレータ２０は、ステップＳ３に処理を進める。一方、他のアクセラレータ２０からメタデータが転送されている場合には（ステップＳ２、Ｙｅｓ）、アクセラレータ２０（転送先アクセラレータ２０ｂ）は、転送元アクセラレータ２０ａに、メタデータの受信完了の通知を送信するとともに、メタデータを解析し（ステップＳ４）、ステップＳ５に処理を進める。
【００７３】
ステップＳ３において、ＲＱバッファにメタデータがセットされていない場合には（ステップＳ３、Ｎｏ）、アクセラレータ２０は、ステップＳ１に処理を戻す。一方、ＲＱバッファにメタデータがセットされている場合には（ステップＳ３、Ｙｅｓ）、アクセラレータ２０は、メタデータを読み込んで解析する（ステップＳ５）。
【００７４】
また、アクセラレータ２０は、解析結果を基に、データバッファ１１ａの処理データを取得して（ステップＳ６）、演算処理を行う（ステップＳ７）。
【００７５】
また、転送先のアクセラレータ２０が存在しない場合には（ステップＳ８、Ｎｏ）、アクセラレータ２０は、演算処理済みの処理データをデータバッファ１１ａに書き込み（ステップＳ９）、ＣＱバッファにメタデータを送信して（ステップＳ１０）、処理を完了する。その後、アクセラレータ２０は、ステップＳ１に処理を戻す。
【００７６】
一方、転送先のアクセラレータ２０が存在する場合には（ステップＳ８、Ｙｅｓ）、アクセラレータ２０は、転送元アクセラレータ２０ａとして、処理データの演算処理結果を中間データバッファ１１ｃに書き出し（ステップＳ１１）、転送先アクセラレータ２０ｂにメタデータを送信する（ステップＳ１２）。そして、アクセラレータ２０は、転送先アクセラレータ２０ｂから受信完了の通知を受け取った場合に、メタデータの転送が完了したものとみなし、ステップＳ１に処理を戻す。
【００７７】
［効果］
以上、説明したように、本実施形態のアクセラレータ制御システム１において、アクセラレータ制御装置１０では、メタデータバッファ１１ｂが、演算処理の対象のデータの所在と、演算処理を行うアクセラレータ２０の情報と、該アクセラレータ２０の演算処理の内容を指定する情報とを含む制御用データを記憶する。Ｐｏｌｌｉｎｇ機能１３ａが、アクセラレータ２０により演算処理済みのメタデータがメタデータバッファ１１ｂに格納されている場合に、該アクセラレータ２０による演算処理の完了を判定する。
【００７８】
また、アクセラレータ２０では、読込機能２１ｂが、メタデータバッファ１１ｂからメタデータを取得する。演算機能２４が、取得されたメタデータに含まれる演算処理の対象の処理データの所在とアクセラレータ２０の演算処理の内容を指定する情報に従って、演

算処理の対象の処理データに対する演算処理を行う。演算処理が完了した場合であって、メタデータに、転送先のアクセラレータの情報が含まれていない場合に、書込機能２１ｃが、メタデータをメタデータバッファ１１ｂに格納する。メタデータに、転送先のアクセラレータ２０ｂの情報が含まれている場合に、書込機能２１ｃが、該アクセラレータ２０ｂに該制御用データを転送する。他のアクセラレータ２０ａからメタデータを受信した場合に、演算機能２４が、メタデータに含まれる演算処理の対象の中間データの所在と、該演算処理の内容を指定する情報に従って、演算処理の対象の中間データに対する演算処理を行う。

【００７９】

これにより、アクセラレータ制御システム１は、複数のアクセラレータ２０を用いた処理において、アクセラレータ制御装置１０を介さずに実行されるため、転送元アクセラレート／転送先アクセラレートの排他処理や処理完了の割込み通知、同期制御等のオーバヘッドを低減して、スケーラビリティを低下させず、リアルタイム処理を高速に行うことが可能となる。

【００８０】

また、アクセラレータ間メタデータ通信機構２１ｄが、他のアクセラレータ２０ａからのメタデータの受信可否を確認する。これにより、アクセラレータ間でのデータ転送が確実に実行される。

【００８１】

また、書込機能２１ｃが、所定の時間内に転送先のアクセラレータ２０ｂからメタデータの受信可否が通知されない場合に、メタデータを再送する。これにより、アクセラレータ間でのデータ転送が確実に実行される。

【００８２】

また、データ書込機能２３ｂは、転送先のアクセラレータ２０ｂにメタデータが転送された場合に、演算処理対象の処理データに対する演算処理の結果の中間データを中間データバッファ１１ｃに格納する。転送先のアクセラレータ２０ｂにおいて、演算機能２４が、中間データバッファ１１ｃに格納された演算処理の結果を読み込んで演算処理の対象とする。これにより、アクセラレータ間でのデータ転送が効率よく完了する。

【００８３】

また、Ｐｏｌｌｉｎｇ機能２１ａが、メタデータバッファ１１ｂにメタデータが有るか否かを監視する。これにより、アクセラレータ２０が自律的に処理を行うことが可能となる。

【００８４】

また、Ｐｏｌｌｉｎｇ機能１３ａが、は、メタデータバッファ１１ｂに演算処理済みのメタデータが有るか否かを監視する。これにより、アクセラレータ制御装置１０は、アクセラレータ２０と非同期に処理を行うことが可能となる。

【００８５】

［プログラム］

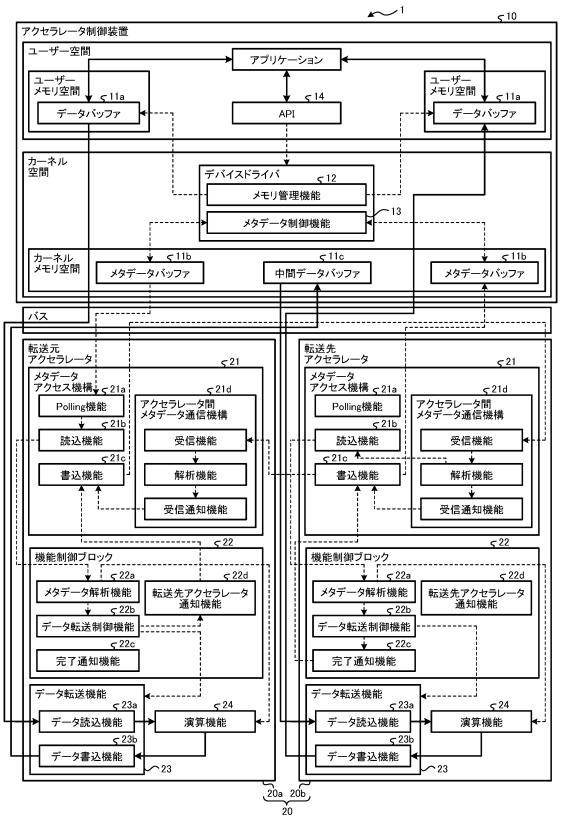
上記実施形態に係るアクセラレータ制御装置１０が実行する処理をコンピュータが実行可能な言語で記述したプログラムを作成することもできる。一実施形態として、アクセラレータ制御装置１０は、パッケージソフトウェアやオンラインソフトウェアとして上記のアクセラレータ制御処理を実行するアクセラレータ制御プログラムを所望のコンピュータにインストールさせることによって実装できる。例えば、上記のアクセラレータ制御プログラムを情報処理装置に実行させることにより、情報処理装置をアクセラレータ制御装置１０として機能させることができる。ここで言う情報処理装置には、デスクトップ型またはノート型のパーソナルコンピュータが含まれる。また、その他にも、情報処理装置にはスマートフォン、携帯電話機やＰＨＳ（Personal Handyphone System）などの移動体通信端末、さらには、ＰＤＡ（Personal Digital Assistant）などのスレート端末などがその範疇に含まれる。また、アクセラレータ制御装置１０の機能を、クラウドサーバに実装してもよい。

【００８６】
図８は、アクセラレータ制御プログラムを実行するコンピュータの一例を示す図である。コンピュータ１０００は、例えば、メモリ１０１０と、ＣＰＵ１０２０と、ハードディスクドライブインタフェース１０３０と、ディスクドライブインタフェース１０４０と、シリアルポートインタフェース１０５０と、ビデオアダプタ１０６０と、ネットワークインタフェース１０７０とを有する。これらの各部は、バス１０８０によって接続される。
【００８７】
メモリ１０１０は、ＲＯＭ（Read Only Memory）１０１１およびＲＡＭ１０１２を含む。ＲＯＭ１０１１は、例えば、ＢＩＯＳ（Basic Input Output System）等のブートプログラムを記憶する。ハードディスクドライブインタフェース１０３０は、ハードディスクドライブ１０３１に接続される。ディスクドライブインタフェース１０４０は、ディスクドライブ１０４１に接続される。ディスクドライブ１０４１には、例えば、磁気ディスクや光ディスク等の着脱可能な記憶媒体が挿入される。シリアルポートインタフェース１０５０には、例えば、マウス１０５１およびキーボード１０５２が接続される。ビデオアダプタ１０６０には、例えば、ディスプレイ１０６１が接続される。
【００８８】
ここで、ハードディスクドライブ１０３１は、例えば、ＯＳ１０９１、アプリケーションプログラム１０９２、プログラムモジュール１０９３およびプログラムデータ１０９４を記憶する。上記実施形態で説明した各情報は、例えばハードディスクドライブ１０３１やメモリ１０１０に記憶される。
【００８９】
また、アクセラレータ制御プログラムは、例えば、コンピュータ１０００によって実行される指令が記述されたプログラムモジュール１０９３として、ハードディスクドライブ１０３１に記憶される。具体的には、上記実施形態で説明したアクセラレータ制御装置１０が実行する各処理が記述されたプログラムモジュール１０９３が、ハードディスクドライブ１０３１に記憶される。
【００９０】
また、アクセラレータ制御プログラムによる情報処理に用いられるデータは、プログラムデータ１０９４として、例えば、ハードディスクドライブ１０３１に記憶される。そして、ＣＰＵ１０２０が、ハードディスクドライブ１０３１に記憶されたプログラムモジュール１０９３やプログラムデータ１０９４を必要に応じてＲＡＭ１０１２に読み出して、上述した各手順を実行する。
【００９１】
なお、アクセラレータ制御プログラムに係るプログラムモジュール１０９３やプログラムデータ１０９４は、ハードディスクドライブ１０３１に記憶される場合に限られず、例えば、着脱可能な記憶媒体に記憶されて、ディスクドライブ１０４１等を介してＣＰＵ１０２０によって読み出されてもよい。あるいは、アクセラレータ制御プログラムに係るプログラムモジュール１０９３やプログラムデータ１０９４は、ＬＡＮやＷＡＮ（Wide Area Network）等のネットワークを介して接続された他のコンピュータに記憶され、ネットワークインタフェース１０７０を介してＣＰＵ１０２０によって読み出されてもよい。
【００９２】
以上、本発明者によってなされた発明を適用した実施形態について説明したが、本実施形態による本発明の開示の一部をなす記述および図面により本発明は限定されることはない。すなわち、本実施形態に基づいて当業者等によりなされる他の実施形態、実施例および運用技術等は全て本発明の範疇に含まれる。
【符号の説明】
【００９３】
１０　アクセラレータ制御装置
１１ａ　データバッファ（記憶部）
１１ｂ　メタデータバッファ（記憶部）

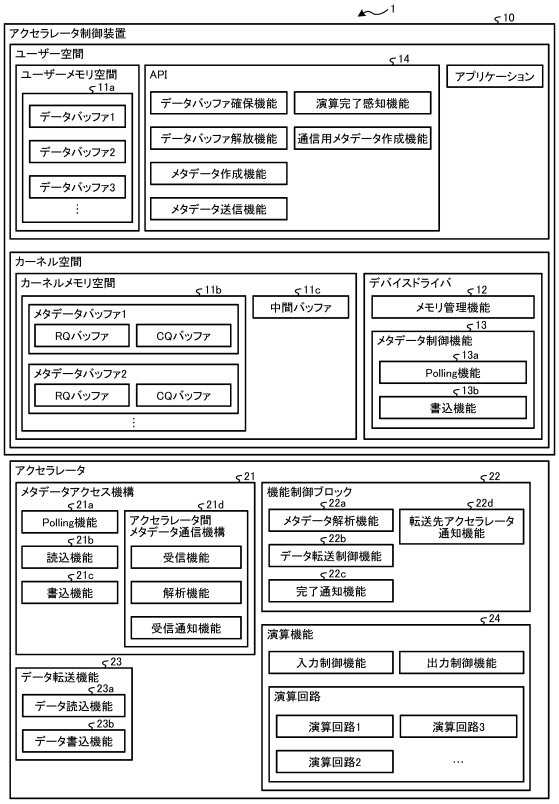
１１ｃ　中間データバッファ（記憶部）
１２　メモリ管理機能
１３　メタデータ制御機能
１３ａ　Ｐｏｌｌｉｎｇ機能（判定部）
１３ｂ　書込機能
１４　ＡＰＩ
２０　アクセラレータ
２０ａ　転送元アクセラレータ
２０ｂ　転送先アクセラレータ
２１　メタデータアクセス機構
２１ａ　Ｐｏｌｌｉｎｇ機能（取得部）
２１ｂ　読込機能
２１ｃ　書込機能（格納部、転送部）
２１ｄ　アクセラレータ間メタデータ通信機構（確認部）
２２　機能制御ブロック
２２ａ　メタデータ解析機能
２２ｂ　データ転送制御機能
２２ｃ　完了通知機能
２２ｄ　転送先アクセラレータ通知機能
２３　データ転送機能
２３ａ　データ読込機能
２３ｂ　データ書込機能
２４　演算機能（演算部、第２の転送部）

【図面】

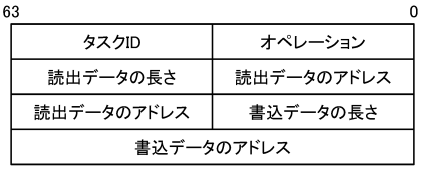
【図１】



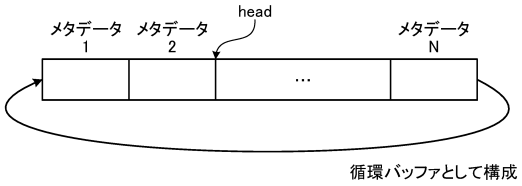
【図２】



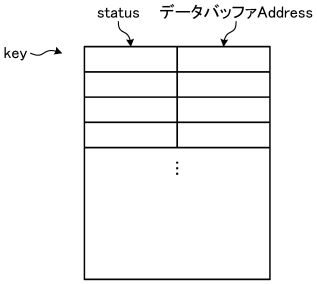
【図３】


63
0
タスクID
オペレーション
読出データの長さ
読出データのアドレス
読出データのアドレス
書込データのアドレス
書込データのアドレス

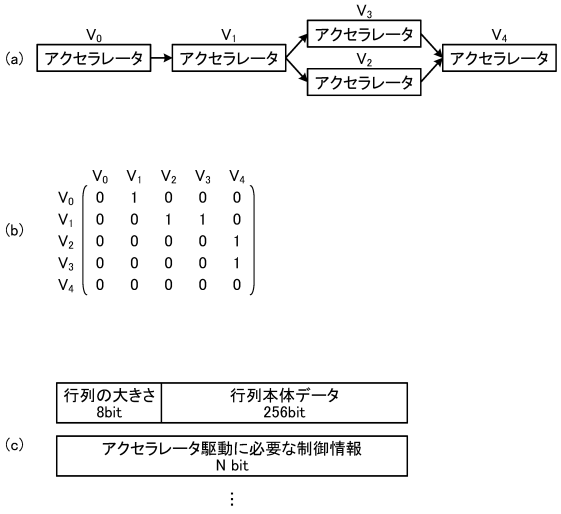
【図４】


head
メタデータ 1
メタデータ 2
…
メタデータ N
循環バッファとして構成

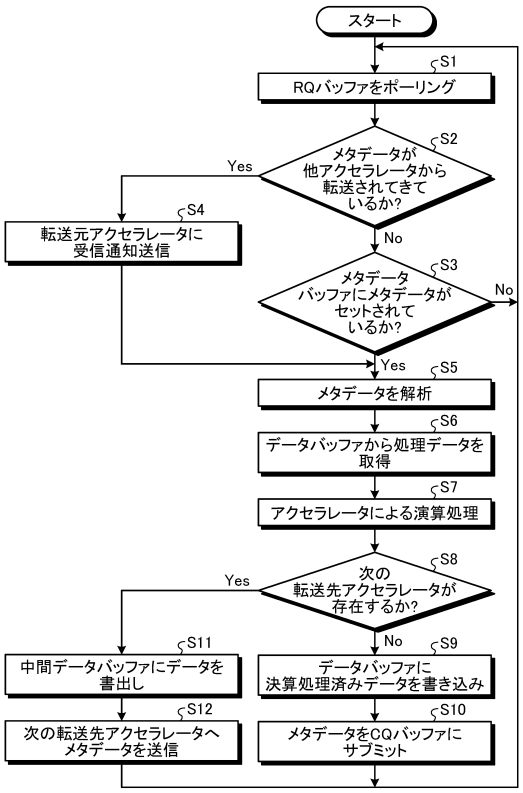
【図５】


status
データバッファAddress
key
⋮

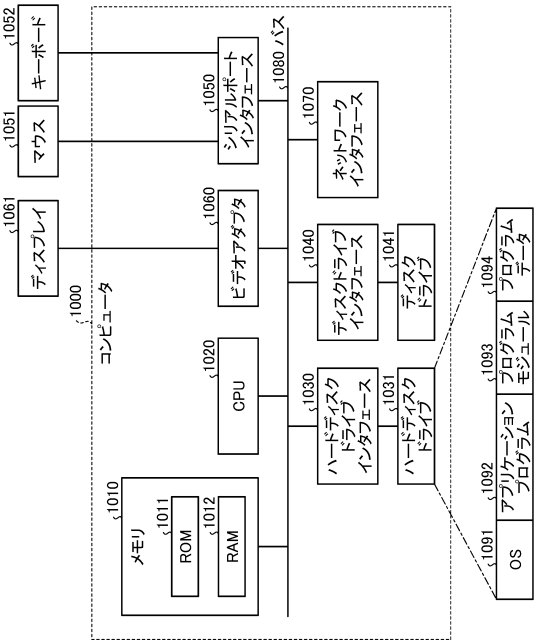
【図６】


(a)
V0
アクセラレータ
V1
アクセラレータ
V3
アクセラレータ
V2
アクセラレータ
V4
アクセラレータ
(b)
V0 V1 V2 V3 V4
V0 0 1 0 0 0
V1 0 0 1 1 0
V2 0 0 0 0 1
V3 0 0 0 0 1
V4 0 0 0 0 0
(c)
行列の大きさ 8bit
行列本体データ 256bit
アクセラレータ駆動に必要な制御情報 N bit
⋮

【図７】


スタート
S1
RQバッファをポーリング
S2
メタデータが
他アクセラレータから
転送されてきて
いるか?
Yes
No
S4
転送元アクセラレータに
受信通知送信
S3
メタデータ
バッファにメタデータが
セットされて
いるか?
No
Yes
S5
メタデータを解析
S6
データバッファから処理データを
取得
S7
アクセラレータによる演算処理
S8
次の
転送先アクセラレータが
存在するか?
Yes
No
S11
中間データバッファにデータを
書出し
S9
データバッファに
決算処理済みデータを書き込み
S12
次の転送先アクセラレータへ
メタデータを送信
S10
メタデータをCQバッファに
サブミット

【図８】


1052
キーボード
1051
マウス
1061
ディスプレイ
1000
コンピュータ
1050
シリアルポート
インタフェース
1060
ビデオアダプタ
1020
CPU
1010
メモリ
1011
ROM
1012
RAM
1080 バス
1070
ネットワーク
インタフェース
1040
ディスクドライブ
インタフェース
1041
ディスク
ドライブ
1030
ハードディスク
ドライブ
インタフェース
1031
ハードディスク
ドライブ
1094
プログラム
データ
1093
プログラム
モジュール
1092
アプリケーション
プログラム
1091
OS

フロントページの続き

(56)参考文献 特開２０１９－１４９０８６（ＪＰ，Ａ）
特表２０２０－５３７２３５（ＪＰ，Ａ）

(58)調査した分野 (Int.Cl.，ＤＢ名)
Ｇ０６Ｆ ９／５０