

(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7553176号
(P7553176)

(45)発行日 令和6年9月18日(2024.9.18)

(24)登録日 令和6年9月9日(2024.9.9)

(51)国際特許分類

F I

G 0 6 F 9/48 (2006.01)

G 0 6 F 9/48 3 0 0 A

G 0 6 F 9/48 3 0 0 C

請求項の数 13 (全28頁)

(21)出願番号	特願2022-559487(P2022-559487)	(73)特許権者	503433420
(86)(22)出願日	令和2年3月31日(2020.3.31)		華為技術有限公司
(65)公表番号	特表2023-519405(P2023-519405 A)		HUAWEI TECHNOLOGIES CO., LTD.
(43)公表日	令和5年5月10日(2023.5.10)		中華人民共和国 5 1 8 1 2 9 広東省深
(86)国際出願番号	PCT/CN2020/082395		▲チェン▼市龍崗区坂田 華為総部▲ベ
(87)国際公開番号	WO2021/195949		ン▼公樓
(87)国際公開日	令和3年10月7日(2021.10.7)		Huawei Administrat
審査請求日	令和4年11月1日(2022.11.1)		ion Building, Banti
前置審査			an, Longgang Distri
			ct, Shenzhen, Guang
			dong 5 1 8 1 2 9, P. R. C
			hina
		(74)代理人	100107766
			弁理士 伊東 忠重

最終頁に続く

(54)【発明の名称】 ハードウェアアクセラレータをスケジューリングするための方法およびタスクスケジューラ

(57)【特許請求の範囲】

【請求項1】

ハードウェアアクセラレータをスケジューリングするための方法であって：
タスクスケジューラによって、ターゲットタスクを取得するステップと；
依存関係に基づいて前記タスクスケジューラによって、前記ターゲットタスクに関連付けられた第1の関連付けられたタスクを決定するステップであって、前記依存関係は、タスクセット内のタスクの実行順序を示し、前記ターゲットタスクは前記タスクセット内のタスクであり、前記第1の関連付けられたタスクは前記ターゲットタスクが実行される前に前記依存関係に基づいて実行される必要があるタスクである、ステップと；
前記第1の関連付けられたタスクが実行されたとき、前記タスクスケジューラによって、前記ターゲットタスクを実行するように少なくとも1つのハードウェアアクセラレータをスケジューリングするステップと；
前記タスクスケジューラによって、前記タスクセット内の各タスクが実行された後に取得されたデータを格納するステップであって、前記タスクセット内のすべてのタスクがスケジューリングされたタスクを形成する、ステップと；を含み、
前記タスクセット内の前記タスクを形成するデータは、車両によって、前記車両に搭載されたカメラ装置を使用することによって取得され、
前記タスクセット内の各タスクが実行された後に取得された前記データはA Iモジュールによって計算されるフィードバックデータを含み、前記フィードバックデータは、車線境界線および／または停止線を検知するために使用されるデータ；または安全領域を検知

10

20

するために使用されるデータ；または障害物を感知するために使用されるデータ；を含み、前記フィードバックデータは、前記車両の操作を案内するために使用され、前記ＡＩモジュールは前記少なくとも１つのハードウェアアクセラレータに属する、
方法。

【請求項２】

前記タスクスケジューラが、前記ターゲットタスクを実行するように前記少なくとも１つのハードウェアアクセラレータをスケジューリングするとき、前記少なくとも１つのハードウェアアクセラレータは前記ターゲットタスクを順に実行する、

請求項１に記載の方法。

【請求項３】

前記少なくとも１つのハードウェアアクセラレータのそれぞれは実行キューに対応し、前記ターゲットタスクの識別子が各前記ハードウェアアクセラレータに対応する前記実行キューに格納される、

請求項１又は２に記載の方法。

【請求項４】

前記タスクスケジューラが前記ターゲットタスクを実行するように前記少なくとも１つのハードウェアアクセラレータをスケジューリングした後、前記方法はさらに：

前記タスクスケジューラによって、前記少なくとも１つのハードウェアアクセラレータによって返された指示メッセージを受信するステップであって、前記指示メッセージは、前記少なくとも１つのハードウェアアクセラレータが前記ターゲットタスクを実行したことを示す、ステップと；

前記タスクスケジューラによって、前記少なくとも１つのハードウェアアクセラレータに対応する実行キューから前記ターゲットタスクの前記識別子を削除するステップと；を含む、

請求項３に記載の方法。

【請求項５】

ターゲットタスクを取得するように構成された取得モジュールと；

依存関係に基づいて、前記ターゲットタスクに関連付けられた第１の関連付けられたタスクを決定するように構成された決定モジュールであって、前記依存関係はタスクセット内のタスクの実行順序を示し、前記ターゲットタスクは前記タスクセット内のタスクであり、前記第１の関連付けられたタスクは前記ターゲットタスクが実行される前に前記依存関係に基づいて実行される必要があるタスクである、決定モジュールと；

スケジューリングモジュールであって：前記決定モジュールが前記第１の関連付けられたタスクが実行されたと決定するときに、前記ターゲットタスクを実行するよう少なくとも１つのハードウェアアクセラレータをスケジューリングするように構成された、スケジューリングモジュールと；を有し、

前記スケジューリングモジュールは、さらに：前記タスクセット内の各タスクが実行された後に取得されたデータを格納するように構成される、前記タスクセット内のすべてのタスクがスケジューリングされたタスクを形成し、

前記タスクセット内の前記タスクを形成するデータは、車両によって、前記車両に搭載されたカメラ装置を使用することによって取得され、

前記タスクセット内の各タスクが実行された後に取得された前記データはＡＩモジュールによって計算されるフィードバックデータを含み、前記フィードバックデータは：車線境界線および／または停止線を感知するために使用されるデータ；または安全領域を感知するために使用されるデータ；または障害物を感知するために使用されるデータ；を含み、前記フィードバックデータは、前記車両の操作を案内するために使用され、前記ＡＩモジュールは前記少なくとも１つのハードウェアアクセラレータに属する、
タスクスケジューラ。

【請求項６】

前記スケジューリングモジュールが、前記ターゲットタスクを実行するように前記少な

10

20

30

40

50

くとも 1 つのハードウェアアクセラレータをスケジューリングするとき、前記少なくとも 1 つのハードウェアアクセラレータは前記ターゲットタスクを順に実行する、

請求項 5 に記載のスケジューラ。

【請求項 7】

前記少なくとも 1 つのハードウェアアクセラレータのそれぞれが実行キューに対応し、前記ターゲットタスクの識別子が各前記ハードウェアアクセラレータに対応する前記実行キューに格納される、

請求項 5 又は 6 に記載のスケジューラ。

【請求項 8】

前記スケジューリングモジュールは、さらに：

前記スケジューリングモジュールが前記ターゲットタスクを実行するように前記少なくとも 1 つのハードウェアアクセラレータをスケジューリングした後、前記少なくとも 1 つのハードウェアアクセラレータによって返された指示メッセージを受信し、前記指示メッセージは、前記少なくとも 1 つのハードウェアアクセラレータが前記ターゲットタスクを実行したことを示し；

前記少なくとも 1 つのハードウェアアクセラレータに対応する実行キューから、前記ターゲットタスクの前記識別子を削除する；ように構成される、

請求項 7 に記載のスケジューラ。

【請求項 9】

タスクスケジューラであって、ターゲットタスクを取得し、依存関係に基づいて、前記ターゲットタスクに関連付けられた第 1 の関連付けられたタスクを決定し、前記依存関係はタスクセット内のタスクの実行順序を示し、前記ターゲットタスクは前記タスクセット内のタスクであり、前記第 1 の関連付けられたタスクは前記ターゲットタスクが実行される前に前記依存関係に基づいて実行される必要があるタスクであり；前記第 1 の関連付けられたタスクが実行されたとき、前記ターゲットタスクを実行するように少なくとも 1 つのハードウェアアクセラレータをスケジューリングする、ように構成される、タスクスケジューラと；

前記ターゲットタスクを実行するように構成された前記少なくとも 1 つのハードウェアアクセラレータと；を有し、

前記タスクスケジューラはさらに：前記タスクセット内の各タスクが実行された後に取得されたデータを格納するように構成され、前記タスクセット内のすべてのタスクがスケジューリングされたタスクを形成し、

前記タスクセット内の前記タスクを形成するデータは、車両によって、前記車両に搭載されたカメラ装置を使用することによって取得され、

前記タスクセット内の各タスクが実行された後に取得された前記データは A I モジュールによって計算されるフィードバックデータを含み、前記フィードバックデータは：車線境界線および / または停止線を検知するために使用されるデータ；または安全領域を検知するために使用されるデータ；または障害物を検知するために使用されるデータ；を含み、前記フィードバックデータは前記車両の操作を案内するために使用され、前記 A I モジュールは前記少なくとも 1 つのハードウェアアクセラレータに属する、

制御システム。

【請求項 10】

前記タスクスケジューラは少なくとも 1 つの実行キューを含み、前記ターゲットタスクの識別子が各実行キューに格納され；

前記少なくとも 1 つのハードウェアアクセラレータは、前記ターゲットタスクのものであって、前記少なくとも 1 つの実行キューに格納されている識別子を使用することによって前記ターゲットタスクを実行するように構成され、各ハードウェアアクセラレータは 1 つの実行キューに対応する、

請求項 9 に記載のシステム。

【請求項 11】

10

20

30

40

50

プロセッサおよびメモリを有するタスクスケジューラであって、前記プロセッサは前記メモリに結合され；

前記メモリはプログラムを格納するように構成され；

前記プロセッサは、前記タスクスケジューラが請求項 1 乃至 4 のいずれか 1 項に記載の方法を実行するように、前記メモリ内の前記プログラムを実行するように構成される、タスクスケジューラ。

【請求項 1 2】

プログラムを含むコンピュータ可読記憶媒体であって、前記プログラムがコンピュータで実行されるとき、前記コンピュータは請求項 1 乃至 4 のいずれか 1 項に記載の方法を実行することを可能にされる、コンピュータ可読記憶媒体。

10

【請求項 1 3】

コンピュータプログラムであって、前記コンピュータプログラムがコンピュータで実行されるとき、前記コンピュータは、請求項 1 乃至 4 のいずれか 1 項に記載の方法を実行することを可能にされる、コンピュータプログラム。

【発明の詳細な説明】

【技術分野】

【0001】

本出願は、ネットワーク通信技術の分野、特にハードウェアアクセラレータをスケジューリングするための方法およびタスクスケジューラに関する。

【背景技術】

20

【0002】

ハードウェアアクセラレータ (hardware accelerator、HAC) は、論理型言語を使用していくつかの特定の機能を実装するハードウェアモジュールである。データを同時に処理できるため、ハードウェアアクセラレータは、処理能力および遅延においてソフトウェア実装よりも大きな利点がある。現在、いくつかの制御システム、例えば、自動運転マルチドメインコントローラ (Multi-Domain Controller、MDC、インテリジェントドライビングドメインコントローラとも呼ばれる) では、異なる情報フロータスクを処理するためにシステムオンチップ (System on Chip、SoC) によって使用される対応するアクセラレータが、対応するタスクを完了するために一緒に動作するようにスケジューリングする必要がある。各タスクはデータフレームに基づいて処理される。各データフレームは複数の HAC によって並列または連続的に処理される。

30

【0003】

現在、HAC は、一般的に、ソフトウェアによって管理およびスケジューリングされ、HAC はソフトウェアを使用してスケジューリングされる。各 HAC は、1 つまたは複数のソフトウェアによってスケジューリングされる。タスクを処理する必要がある場合、ソフトウェアは対応する HAC にタスク要求を送信する。処理が完了すると、HAC はタスク完了応答を返す。図 1 に示すように、MDC シナリオの制御システムでは、CPU ソフトウェアスケジューリングレイヤ、ユーザモード空間レイヤ、カーネルモード空間レイヤ、HAC を上から下にソフトウェアでスケジューリングすることによってタスクが実行される。ソフトウェアによってタスクをスケジューリングするプロセスは、具体的には以下の通りである：CPU が前のレベルによって送信されたデータ (すなわち、前のタスク) を受信し、ハードウェアアクセラレータ 0 がデータ処理を実行する必要がある。この場合、アドレス空間が仮想メモリ 0 内のデータをコピーするために、ユーザモード空間レイヤに作成される必要があり、ハードウェアアクセラレータ 0 に対応しカーネルモード空間レイヤにあるエグゼキュータ 0 (エグゼキュータ (Executor)) が呼び出され、その後、ハードウェアアクセラレータ 0 は呼び出されてデータを実行し、対応するタスクの処理を完了する。タスクの実行後、ハードウェアアクセラレータ 0 は、メッセージ応答を生成し (例えば、割り込み又はステータスレジスタを生成する)、カーネルモード空間レイヤのドライバ 0 (ドライバ (Driver)) およびエグゼキュータ 0 を介してユーザモード空間レイヤにメッセージ応答を順次報告し、最後にメッセージ応答を CPU に報告する。ハードウ

40

50

エアアクセラレータ 0 によって前のデータが処理された後、ハードウェアアクセラレータ 1 がさらにデータ処理を実行し続ける必要がある場合、ハードウェアアクセラレータ 0 によって処理された後に出力されたデータはさらにコピーされる必要があり、その後、ハードウェアアクセラレータ 1 は、データが最終的に CPU に報告されるまで、前述のプロセスに従って処理を実行し続ける。複数のタスクが処理される必要があり、タスクが相互に依存している場合（例えば、タスク 2 は、タスク 1 が処理された後にのみ処理できる）、各タスクは前のステップに従って処理する必要がある。加えて、タスク 2 は、タスク 1 が前のステップに従って処理された後にのみ、前のステップに従って処理されることができる。

【0004】

10

前述のハードウェアアクセラレータをスケジューリングするための方法では、タスクがレベルごとに配送されるプロセスおよびタスクが処理後にレベルごとに報告されるプロセスにおいて、カーネルモードドライバとエグゼキュータが必要とされ、その後タスクはユーザモード空間レイヤに到達し、最終的に処理のために CPU に報告される。このプロセスには長い時間がかかる。複数のタスクを処理する必要がある、タスクが互いに依存している場合、タスクは先行するステップに従って順に処理される必要がある。この期間の間、待ち時間が長く、CPU リソースが占有される。

【発明の概要】

【0005】

20

本出願の実施形態は、ハードウェアアクセラレータをスケジューリングする方法およびタスクスケジューラを提供する。タスクスケジューラは、CPU と各ハードウェアアクセラレータとの間に接続される。タスクスケジューラは、ハードウェアスケジューリングを介して、タスク間の依存関係に基づいて、各タスクを実行する HAC をスケジューリングするように構成される。各タスクの配信およびタスク実行後のレポートは、どちらも直接ハードウェアアクセスによって実行される。これは、タスクが処理された後のタスク配信およびレポートの遅延を短縮し、システムの処理性能を間接的に向上させる。

【0006】

この観点から、本出願の実施形態は以下の技術的解決策を提供する。

【0007】

30

第 1 の態様によると、本出願はまずハードウェアアクセラレータをスケジューリングする方法を提供する。本方法は、次のものを含む：タスクスケジューラが、まずターゲットタスクを取得する。例えば、メモリ（例えば、DDR）からターゲットタスクを取得する。ターゲットタスクは、タスクセットにおいて実行されることになるタスクである。ターゲットタスクを取得した後、タスクスケジューラは、依存関係に基づいて、ターゲットタスクに関連付けられた第 1 の関連タスクを決定し得る。第 1 の関連付けられたタスクは、ターゲットタスクが実行される前に依存関係に基づいて実行される必要があるタスクである。第 1 の関連付けられたタスクは、タスクセット内の 1 つのタスクであり得る、またはタスクセット内の複数のタスクであり得る。第 1 の関連付けられたタスクの量は、依存関係に基づいて具体的に決定される。複数の第 1 の関連付けられたタスクがある場合、依存関係はさらに複数の第 1 の関連付けられたタスクの実行順序（execution sequence）を含む。タスクスケジューラが、依存関係に基づいて、ターゲットタスクに関連付けられた第 1 の関連付けられたタスクを決定し、さらに第 1 の関連付けられたタスクが実行されたことを決定した後、タスクスケジューラは、ターゲットタスクの実行に関連する少なくとも 1 つのハードウェアアクセラレータを決定し、ターゲットタスクを実行するように少なくとも 1 つのハードウェアアクセラレータをスケジューリングする。

40

【0008】

前述の本出願の実装では、タスクスケジューラはまずターゲットタスクを取得し（例えば、メモリからターゲットタスクを取得し）、ターゲットタスクと関連タスクとの間の依存関係を取得する。依存関係に基づいて、関連付けられたタスク内の第 1 の関連付けられたタスク（例えば、ターゲットタスクを実行するための前提条件は、タスク 1、タスク 2

50

などがすべて成功裏に実行されることである)が成功裏に実行されたと決定される場合は、ターゲットタスクは実行条件を満たしていることを示す。この場合、タスクスケジューラは、ターゲットタスクを実行するようにターゲットタスクの実行に関連するハードウェアアクセラレータ(1つまたは複数であり得る)をスケジューリングする。本出願のこの実施形態では、タスクスケジューラはCPUと各ハードウェアアクセラレータとの間に接続される。タスクスケジューラは、ハードウェアスケジューリングを介して、タスク間の依存関係に基づいて、各タスクを実行するようにHACをスケジューリングするように構成される。各タスクの配信は、直接ハードウェアアクセスによって実行される。これは、タスクの配信遅延を短縮し、間接的にシステムの処理性能を向上させる。

【0009】

10

本出願の実施形態の第1の態様を参照すると、本出願の実施形態の第1の態様の第1の実装では、ターゲットタスクを実行する1つまたは複数のハードウェアアクセラレータがあり得る。ターゲットタスクを実行する複数のハードウェアアクセラレータがある場合、複数のハードウェアアクセラレータは順にターゲットタスクを実行する。なお、本出願のいくつかの実装では、複数のハードウェアアクセラレータがターゲットタスクを順に実行することは、複数のハードウェアアクセラレータが同時にターゲットタスクを実行することを含み得る。例えば、タスクスケジューラが、ターゲットタスクを取得した後、ターゲットタスクが、ハードウェアアクセラレータ1、ハードウェアアクセラレータ2、ハードウェアアクセラレータ3によって個別に実行される必要があると決定する場合、ターゲットタスクは、ハードウェアアクセラレータ1~3によって同時に実行され得、ハードウェアアクセラレータ1~3は競合しない。

20

【0010】

前述の本出願の実装では、ターゲットタスクの実行に関連する複数のハードウェアアクセラレータがある場合、どのように複数のハードウェアアクセラレータがターゲットタスクを実行するために連携する(work together)かについて説明している。これは動作可能(operable)である。

【0011】

本出願の実施形態の第1の態様および第1の態様の第1の実装を参照して、本出願の実施形態の第1の態様の第2の実装では、タスクスケジューラがターゲットタスクを実行するために少なくとも1つのハードウェアアクセラレータをスケジューリングした後、タスクスケジューラはさらに、その後必要になったときにターゲットタスクが実行された後に生成されたデータが別のハードウェアアクセラレータによって直接呼び出されることができるよう、ターゲットタスクの実行された後に生成されたデータを格納する(例えば、メモリDDRにデータを格納する)ようハードウェアアクセラレータに指示する。

30

【0012】

前述の本出願の実装では、各ハードウェアアクセラレータは、ターゲットタスクを処理した後、処理されたデータを格納する(例えば、処理されたデータをメモリに格納する)。この方法では、別のハードウェアアクセラレータが処理されたデータをさらに処理する必要がある場合、処理されたデータはコピーされる必要はなく、直接呼び出されることができる。これは、データ処理量を削減し、時間を短縮し、間接的にシステム性能を向上させる。

40

【0013】

本出願の実施形態の第1の態様または第1の態様の第1から第2の実装を参照して、本出願の実施形態の第1の態様の第3の実装では、各ハードウェアアクセラレータは1つまたは複数のタスクを実行し得る。この場合、各ハードウェアアクセラレータは1つの実行キューに対応し、ハードウェアアクセラレータによって実行されることになるタスクは各実行キューに順に格納される。ターゲットタスクを実行する複数のハードウェアアクセラレータがある場合、ターゲットタスクの識別子が、ターゲットタスクのための各ハードウェアアクセラレータに対応する実行キューに格納される。

【0014】

50

前述の本出願の実装では、ハードウェアアクセラレータは、対応する実行キューに配置されたタスクの識別子に基づいて、各タスクが処理される順序を識別する。これは柔軟である。

【0015】

本出願の実施形態の第1の態様の第3の実装を参照して、本出願の実施形態の第1の態様の第4の実装では、対応するターゲットタスクを実行するようハードウェアアクセラレータをスケジューリングした後、タスクスケジューラは、ターゲットタスクを実行する対応するハードウェアアクセラレータによって返される指示(indication)メッセージを受信する。指示メッセージは、ハードウェアアクセラレータがターゲットタスクを実行したことを示す。この場合、タスクスケジューラは、ハードウェアアクセラレータに対応する実行キューからターゲットタスクの識別子を削除し得る。

10

【0016】

前述の本出願の実装では、叙述のように、タスクが実行された後、対応するタスクの識別子は実行キューから削除される。実行キューは動的に維持される。これは柔軟(flexible)である。

【0017】

本出願の実施形態の第1の態様または第1の態様の第1から第4の実装を参照して、本出願の実施形態の第1の態様の第5の実装では、タスクセット内の各タスクは、ターゲットタスクのものと同様の処理方法で処理される。タスクセット内の各タスクが実行された後、タスクスケジューラは、各タスクが実行された後にすべてのデータを取得し、そのデータを格納する(例えば、データをメモリに格納する)。タスクセット内のすべてのタスクは、スケジューリングされたタスクを形成する。元のスケジューリングされたタスクは、タスクセット内のすべてのタスクが実行された後にのみ完了する。1つのスケジューリングされたタスクは、センサ(例えば、カメラ)を使用することにより端末装置によって取得されたデータフロー(生データと呼ばれることがある)が一連の関連するハードウェアアクセラレータによって指定された機能を持つ出力データ(例えば、障害物を感知するために使用されるデータおよび別のデバイスに送信されるために使用されるデータ)を取得するために処理される、特定の実装プロセスである。

20

【0018】

前述の本出願の実装では、上述のように、MDCシナリオでは、1つのタスクセット内のすべてのタスクが処理された後にのみ、1つのスケジューリングされたタスクが完了する。これは実用的である。

30

【0019】

本出願の実施形態の第1の態様の第5の実装を参照して、本出願の実施形態の第1の態様の第6の実装では、タスクセットは複数のタスクを含み、複数のタスクを形成するデータは、複数のチャンネルを介して取得され得、端末装置(例えば、車両)が、搭載されたカメラ装置(例えば、車両に搭載されたカメラ)を使用することによってデータを取得することを含むが、これに限定されない。この場合、複数のタスクを形成する取得されたデータは画像データまたはビデオデータである。端末装置がカメラを使用することによってデータを取得した後、データはタスクに分割され(例えば、データフレームのように分割され)、分割されたタスクはタスクセットを形成する。

40

【0020】

前述の本出願の実装では、タスクセットを形成するデータを取得するためのチャンネルが記述されている。これは普遍的に適用可能である。

【0021】

本出願の実施形態の第1の態様の第5から第6の実装を参照して、本出願の実施形態の第1の態様の第7の実装では、1つのスケジューリングされたタスクは、具体的には、AIモジュール(例えば、AIプロセッサ)がフィードバックデータを取得するためにコンピューティングを実行することであり得る。フィードバックデータは、端末装置(例えば、車両)上でユーザの操作をガイドするために使用され得る。本出願のこの実施形態では

50

、A I モジュールもハードウェアアクセラレータの一種である。

【 0 0 2 2 】

前述の本出願の実施形態では、上述のように、スケジューリングされたタスクが実行された後に取得されるデータは、A I モジュールによって計算されたフィードバックデータであり得、そのフィードバックデータはさらに端末装置上でユーザの操作をガイドするために使用され得る。これは実用的である。

【 0 0 2 3 】

本出願の実施形態の第 1 の態様の第 7 の実装を参照して、本出願の実施形態の第 1 の態様の第 8 の実装では、端末装置が車両である場合、一部の適用シナリオでは、フィードバックデータは、車線境界線 (lane line) および / または停止線を検知するために使用されるデータであり得、指定された車線を走行するまたは指定された駐車エリアで停止するように車両所有者を案内する (guide) ために使用される。代替的には、フィードバックデータは、安全領域を検知するために使用されるデータであり得、この車両を他の車両から安全な距離に保つように車両所有者を案内するために使用される。代替的には、フィードバックデータは障害物 (例えば障害物からの距離) を検知するために使用されるデータであり得、事前に安全対策 (例えば、停止や方向転換) を講じるように車両所有者を案内するために使用される。フィードバックデータの種類および機能はここでは限定されない。対応するフィードバックデータは、実際の適用シナリオに基づいて取得され得る。したがって、車両所有者は、対応するフィードバックデータに基づいて車両を操作するように案内され、それによって運転の安全性および運転性能を向上させる。

【 0 0 2 4 】

前述の本出願の実施形態では、フィードバックデータの適用シナリオが具体的に記述されている。これは多様化している。

【 0 0 2 5 】

本出願の実施形態の第 2 の態様は、タスクスケジューラを提供する。タスクスケジューラは、第 1 の態様または第 1 の態様の可能な実装のうちのいずれか 1 つによる方法を実装する機能を有する。機能はハードウェアによって実装され得る、または対応するソフトウェアを実行するハードウェアによって実装され得る。ハードウェアまたはソフトウェアは、前述の機能に対応する 1 つまたは複数のモジュールを含む。

【 0 0 2 6 】

本出願の実施形態の第 3 の態様は、制御システムを提供する。制御システムは、特にタスクスケジューラおよびハードウェアアクセラレータを含み得るが、これに限定されない。タスクスケジューラは：ターゲットタスクを取得し (例えば、メモリからターゲットタスクを取得し)、依存関係に基づいて、ターゲットタスクに関連付けられた第 1 の関連付けられたタスクを決定し、依存関係はタスクセット内のタスクの実行順序を示し、ターゲットタスクはタスクセット内のタスクであり；第 1 の関連付けられたタスクが実行されたと決定されるとき、ターゲットタスクを実行するようにハードウェアアクセラレータをスケジューリングする；ように構成される。ハードウェアアクセラレータは、ターゲットタスクを実行するように構成される。

【 0 0 2 7 】

本出願の実施形態の第 3 の態様を参照して、本出願の実施形態の第 3 の態様の第 1 の実装では、タスクスケジューラは、少なくとも 1 つの実行キューを含み得るが、それに限定されない。ターゲットタスクの識別子が各実行キューに格納される。少なくとも 1 つのハードウェアアクセラレータは、少なくとも 1 つの実行キューに格納されているターゲットタスクの識別子を使用することによってターゲットタスクを実行するように構成される。各ハードウェアアクセラレータは、1 つの実行キューに対応する。

【 0 0 2 8 】

本出願の実施形態の第 3 の態様および本出願の実施形態の第 3 の態様の第 1 の実装を参照して、本出願の実施形態の第 3 の態様の第 2 の実装では、タスクスケジューラはさらに、タスクセット内の各タスクが実行された後に取得されたデータを格納するように構成さ

10

20

30

40

50

れる。タスクセット内のすべてのタスクは、スケジューリングされたタスクを形成する。

【 0 0 2 9 】

本出願の実施形態の第 3 の態様の第 2 の実装を参照して、本出願の実施形態の第 3 の態様の第 3 の実装では、タスクセット内のタスクを形成するデータは、端末装置によって、搭載されたカメラ装置を使用することによって取得される。制御システムは端末装置に配置され得る（例えば、車両上のカメラが画像データを取得し、その画像データを車両上の制御システムに送信する）。代替的には、制御システムは端末装置に配置されていない場合もある（例えば、ユーザの携帯電話のカメラが画像データを取得し、その画像データを、携帯電話と通信接続している車両に送信し、制御システムは車両に配置されている）。

【 0 0 3 0 】

本出願の実施形態の第 3 の態様の第 2 から第 3 の実装を参照して、本出願の実施形態の第 3 の態様の第 4 の実装では、タスクセット内の各タスクが実行された後に取得されたデータは、A I モジュール（例えば、A I プロセッサ）によって計算されたフィードバックデータであり得る。フィードバックデータは端末装置の操作を案内するために使用され、A I モジュールはハードウェアアクセラレータの一種でもある。

【 0 0 3 1 】

本出願の実施形態の第 3 の態様または第 3 の態様の第 1 から第 4 の実装を参照して、本出願の実施形態の第 3 の態様の第 5 の実装では、タスクスケジューラはさらにスケジューリングエンジンおよびエクゼキュータを含み得る。スケジューリングエンジンは、メモリからターゲットタスクおよび依存関係と呼び出すように構成される。エクゼキュータは、ハードウェアアクセラレータへのインターフェイスを提供するように構成される。実行キューは、ターゲットタスクの識別子を順にバッファするために使用される。具体的には、スケジューリングエンジン、エクゼキュータ、および実行キューを含むタスクスケジューラは、第 1 の態様または第 1 の態様の可能な実装のうちのいずれか 1 つによる方法を実装する機能を有する。

【 0 0 3 2 】

本出願の実施形態の第 4 の態様は、タスクスケジューラを提供する。タスクスケジューラは、メモリ、プロセッサ、およびバスシステムを含み得る。メモリはプログラムを格納するように構成される。プロセッサは、メモリに格納されたプログラムを呼び出し、本出願の実施形態の第 1 の態様または第 1 の態様の可能な実装のいずれか 1 つによる方法を実行するように構成される。

【 0 0 3 3 】

本出願の第 5 の態様は、コンピュータ可読記憶媒体を提供する。コンピュータ可読記憶媒体は命令を格納する。命令がコンピュータ上で実行されるとき、コンピュータは第 1 の態様または第 1 の態様の可能な実装のうちのいずれか 1 つによる方法を実行することを可能にされる。

【 0 0 3 4 】

本出願の実施形態の第 6 の態様は、コンピュータプログラムを提供する。コンピュータプログラムがコンピュータ上で実行されるとき、コンピュータは、第 1 の態様または第 1 の態様の可能な実装のいずれか 1 つによる方法を実行することを可能にされる。

【 図面の簡単な説明 】

【 0 0 3 5 】

【 図 1 】タスクのためにソフトウェアによって H A C をスケジューリングすることの概略図である。

【 0 0 3 6 】

【 図 2 】本出願の一実施形態による制御システムの概略図である。

【 0 0 3 7 】

【 図 3 】本出願の一実施形態によるハードウェアアクセラレータのスケジューリングのための方法の概略図である。

【 0 0 3 8 】

10

20

30

40

50

【図 4】本出願の一実施形態による各ハードウェアアクセラレータと実行キューとの間の関係の概略図である。

【0039】

【図 5】本出願の一実施形態によるタスクセット内のタスク間の関連関係の概略図である。

【0040】

【図 6】本出願の一実施形態によるイベント概念が導入された後のタスクとイベントとの間の関連関係の概略図である。

【0041】

【図 7】本出願の一実施形態によるタスク間の依存関係が 2 次元チャートである概略図である。

10

【0042】

【図 8】MDC シナリオにおける現在の制御システムの構造の概略図である。

【0043】

【図 9】本出願の一実施形態による MDC シナリオにおける制御システムの構造の概略図である。

【0044】

【図 10】本出願の一実施形態による MDC シナリオにおけるスケジューリングされたタスクの記述の概略図である。

【0045】

【図 11】本出願の一実施形態による MDC の実際の適用シナリオの概略図である。

20

【0046】

【図 12】本出願の一実施形態によるタスクスケジューラの概略図である。

【0047】

【図 13】本出願の一実施形態によるタスクスケジューラの別の概略図である。

【発明を実施するための形態】

【0048】

本出願の実施形態は、ハードウェアアクセラレータをスケジューリングする方法およびタスクスケジューラを提供する。タスクスケジューラは CPU と各ハードウェアアクセラレータの間に接続される。タスクスケジューラは、ハードウェアスケジューリングを介して、タスク間の依存関係に基づいて、各タスクを実行するために HAC をスケジューリングするように構成される。各タスクの配信およびタスク実行後のレポートは、どちらも直接ハードウェアアクセスによって実行される。これは、タスクの配信およびタスクが処理された後のレポートの遅延を短縮し、システムの処理性能を間接的に向上させる。

30

【0049】

以下は、本出願の実施形態を添付の図面を参照して説明する。当業者は、技術の発展および新たなシナリオの出現により、本出願の実施形態で提供される技術的解決策が同様の技術的問題にも適用可能であることを学ぶことができる。

【0050】

本出願の本明細書、特許請求の範囲および添付図面において、「第 1」、「第 2」等の用語は、類似の対象を区別することを意図しているが、必ずしも特定の順番又は順序を示すものではない。このような方法で使用する用語は、適切な状況では交換可能であることを理解すべきであるが、これは、同じ属性を持つ対象が本出願の実施形態で記述されている場合に使用される単なる識別方法である。加えて、用語「含む (include)」、「含む (contain)」およびその他の変形は、非排他的包含をカバーすることを意味するため、一連のユニットを含むプロセス、方法、システム、製品、または装置は、必ずしもそれらのユニットに限定されず、そのようなプロセス、方法、製品、または装置に明示的にリストされていないまたはそれらに固有の他のユニットを含む場合がある。

40

【0051】

本出願の実施形態を説明する前に、まず、制御システムにおける本出願で説明するタスクスケジューラの位置を説明する。図 2 に示すように、タスクスケジューラは CPU と各

50

ハードウェアアクセラレータの間に位置する。各ハードウェアアクセラレータは、タスクスケジューラに対応するインターフェイスがあり。このインターフェイスは、タスクスケジューラ上のタスクエグゼキュータ（簡単にエグゼキュータと呼ぶこともある）によって提供される。タスクスケジューラは複数の実行キューを有し得る。各実行キューは1つのハードウェアアクセラレータに対応し（いくつかの実施形態では、実行キューは、メモリまたは対応するハードウェアアクセラレータにも存在する場合があるが、これはここでは限定されず、図2は単なる例である）、実行順序内のタスクの識別子を格納するために使用される。実行されることになるタスク、タスクが実行された後に取得されるデータ、タスク間の依存関係などは、すべてメモリDDRに格納され得る（いくつかの実装では、別の位置に格納されることもあり、例えば、タスクスケジューラのストレージユニットに直接格納されるが、これはここでは限定されず、図2は単なる例である）。関連データがメモリに格納される場合、タスクスケジューラがターゲットタスクを実行するように少なくとも1つのハードウェアアクセラレータをスケジューリングした後、タスクスケジューラは、ハードウェアアクセラレータに、ターゲットタスクが実行された後に生成されたデータを格納するよう指示し得る。この方法では、ターゲットタスクが実行された後に生成されたデータは、その後必要になったときに別のハードウェアアクセラレータによって直接呼び出される可能性があり、コピーする必要はない。これは、データ処理量を削減し、時間を短縮し、間接的にシステム性能を向上させる。タスクスケジューラは、タスクスケジューラ上のスケジューリングエンジンを使用して、メモリおよび各ハードウェアアクセラレータと情報を交換し得る（例えば、ハードウェアアクセラレータから返された指示メッセージを受信し、メモリ内の関連データを照会し、ハードウェアアクセラレータに指示を送信する）。例えば、カメラから収集されたデータはメモリに格納される。データ（すなわち、ターゲットタスク）はハードウェアアクセラレータ0およびハードウェアアクセラレータ1によって処理される必要がある。タスクスケジューラは、カメラによって収集された前述のデータをメモリから取得し、ハードウェアアクセラレータ0に指示を配信して、ハードウェアアクセラレータ0にデータを処理するように指示する。データを処理した後、ハードウェアアクセラレータ0は、ハードウェア信号ケーブルを使用してタスクスケジューラに通知し、処理されたデータをメモリに格納し得る。次に、タスクスケジューラは、ハードウェアアクセラレータ1に指示を配信して、さらなる処理のためにハードウェアアクセラレータ0によって処理された前述のデータをメモリから呼び出すようハードウェアアクセラレータ1に指示する。データを処理した後、ハードウェアアクセラレータ1は、タスクスケジューラに通知し、処理されたデータをメモリに再び格納する。本出願のこの実施形態では、タスク配信およびタスクが処理された後のレポートの両方が、直接ハードウェアアクセスによって実行される。これは、タスク配信およびタスクが処理された後のレポートの処理遅延を大幅に減少させ、チップの処理性能を間接的に向上させる。データはメモリに格納されるため、データキャッシュ領域が共有され得、繰り返しコピーする必要がない。これは、データコピーの遅延を減少させ、データ処理性能を向上させる。

【0052】

本出願の本実施形態で提供されるハードウェアアクセラレータをスケジューリングする方法は、図2の制御システムに基づいて実行され得る。この方法では、タスクスケジューラは、タスク間の依存関係に基づいた純粋なハードウェアスケジューリングを直接実装し、CPUリソースを占有する必要がない。詳細は図3を参照されたい。

【0053】

301：タスクスケジューラはターゲットタスクを取得する。

【0054】

タスクスケジューラは、まずターゲットタスクを取得する、例えば、メモリ（例えば、DDR）からターゲットタスクを取得する。ターゲットタスクは、タスクセットにおいて実行されることになるタスクである。タスクセットは複数のタスクを含む。複数のタスクを形成するデータは、複数のチャンネルを通じて取得され得、端末装置（例えば、車両）が搭載されたカメラ装置（例えば、車両に搭載されたカメラ）を使用してデータを取得する

ことを含むが、これに限定されない。この場合、複数のタスクを形成する取得されたデータは画像データまたはビデオデータである。端末装置がカメラを使用してデータを取得した後、データはタスクに分割され（例えば、データフレームのように分割され）、分割されたタスクがタスクセットを形成する。

【 0 0 5 5 】

3 0 2 : タスクスケジューラは、依存関係に基づいて、ターゲットタスクに関連付けられた第 1 の関連付けられたタスクを決定する。

【 0 0 5 6 】

ターゲットタスクを取得した後、タスクスケジューラは、依存関係に基づいて、ターゲットタスクに関連付けられた第 1 の関連付けられたタスクを決定し得る。第 1 の関連付けられたタスクは、ターゲットタスクが実行される前に依存関係に基づいて実行される必要があるタスクである。第 1 の関連付けられたタスクは、タスクセット内の 1 つのタスクである場合もあれば、タスクセット内の複数のタスクである場合もある。第 1 の関連付けられたタスクの量は、依存関係に基づいて具体的に決定される。加えて、複数の第 1 の関連付けられたタスクがある場合、依存関係はさらに、複数の第 1 の関連付けられたタスクの実行順序を含む。例えば、それぞれタスク 1 およびタスク 2 である 2 つの第 1 の関連付けられたタスクがある場合、依存関係に基づいてタスク 1 およびタスク 2 を実行する順序は次のようになる：タスク 1 はタスク 2 の前に実行される。タスク 1 およびタスク 2 の両方が実行される場合、ターゲットタスクが実行条件を満たしていると決定される。本出願のこの実施形態で説明されている順序は、さらに、同時実行を含み得ることが留意されるべきである。前述の例は、引き続き例として使用される。タスク 1 およびタスク 2 が依存関係に基づいて同時に実行される必要がある場合、タスク 1 およびタスク 2 が同時に実行され、2 つのタスクが両方とも実行された後、ターゲットタスクは実行条件を満たしていると決定される。

【 0 0 5 7 】

3 0 3 : 第 1 の関連付けられたタスクが実行されたらタスクスケジューラが決定する場合、タスクスケジューラは、ターゲットタスクを実行するように少なくとも 1 つのハードウェアアクセラレータをスケジューリングする。

【 0 0 5 8 】

タスクスケジューラは、依存関係に基づいて、ターゲットタスクに関連付けられた第 1 の関連付けられたタスクを決定し、さらに第 1 の関連付けられたタスクが実行されたことを決定した後、タスクスケジューラは、ターゲットタスクの実行に関連する少なくとも 1 つのハードウェアアクセラレータを決定し、ターゲットタスクを実行するように少なくとも 1 つのハードウェアアクセラレータをスケジューリングする。ターゲットタスクを実行する 1 つまたは複数のハードウェアアクセラレータがあり得ることが留意されるべきである。ターゲットタスクを実行する複数のハードウェアアクセラレータがある場合、複数のハードウェアアクセラレータはターゲットタスクを順に実行する。わかりやすくするため、以下では説明のための例を用いる。タスクスケジューラが、ターゲットタスクを取得した後、ターゲットタスクがハードウェアアクセラレータ 1、ハードウェアアクセラレータ 2、およびハードウェアアクセラレータ 3 によって実行される必要があると決定し、実行順序が、ターゲットタスクがハードウェアアクセラレータ 1 によって最初に行われ、実行を通じて取得されたデータ 1 がハードウェアアクセラレータ 2 によって実行され、次に実行を通じて取得されたデータ 2 がハードウェアアクセラレータ 3 によって実行されることである場合、ハードウェアアクセラレータ 3 による実行を通じて取得されたデータ 3 は、すべてのハードウェアアクセラレータ 1 ~ 3 がターゲットタスクを実行した後に生成されたデータ（データ 3 はターゲットデータと呼ばれることもある）である。

【 0 0 5 9 】

本出願のいくつかの実装では、複数のハードウェアアクセラレータがターゲットタスクを順に実行することは、複数のハードウェアアクセラレータが同時にターゲットタスクを実行する場合を含み得ることが留意されるべきである。前述の例は、引き続き例として使

10

20

30

40

50

用される。タスクスケジューラが、ターゲットタスクを取得した後、ハードウェアアクセラレータ 1、ハードウェアアクセラレータ 2、およびハードウェアアクセラレータ 3 によってターゲットタスクが別々に実行される必要があると決定する場合、ターゲットタスクはハードウェアアクセラレータ 1 ~ 3 によって同時に実行されてもよく、ハードウェアアクセラレータ 1 ~ 3 は互いに競合しない。

【 0 0 6 0 】

本出願のいくつかの実装では、タスクスケジューラがターゲットタスクを実行するためのハードウェアアクセラレータの数、ハードウェアアクセラレータの種類、実行順序などの情報をどのように決定するかも依存関係に記録される場合があることがさらに留意されるべきである。このように、タスクスケジューラは、依存関係から、ターゲットタスクに

10

【 0 0 6 1 】

さらに、本出願のいくつかの実装では、タスクスケジューラがターゲットタスクを実行するように少なくとも 1 つのハードウェアアクセラレータをスケジューリングした後、タスクスケジューラはさらに、ターゲットタスクが実行された後に生成されたデータを格納する（例えば、メモリ D D R にデータを格納する）ようハードウェアアクセラレータに指示することが留意されるべきである。この方法では、ターゲットタスクが実行された後に生成されたデータは、その後必要になったときに別のハードウェアアクセラレータによって直接呼び出される可能性があり、コピーする必要はない。これは、データ処理量を削減し、時間を短縮し、間接的にシステム性能を向上させる。わかりやすくするため、前述の例は引き続き説明のための例として使用される。タスクスケジューラが、ターゲットタスクを取得した後、ハードウェアアクセラレータ 1、ハードウェアアクセラレータ 2、およびハードウェアアクセラレータ 3 によってターゲットタスクが順に実行される必要があると決定し、実行順序が、ターゲットタスクがハードウェアアクセラレータ 1 によって最初に実行され、実行を通じて取得されたデータ 1 がハードウェアアクセラレータ 2 によって実行され、その後実行を通じて取得されたデータ 2 がハードウェアアクセラレータ 3 によって実行されることである場合、データ 3 は、ハードウェアアクセラレータ 3 による実行を通じて取得される。この場合、タスクスケジューラは、ターゲットタスクが実行された後に取得されたデータ 1 を格納する（例えば、データ 1 をメモリ D D R に格納する）ようにハードウェアアクセラレータ 1 に指示し、その後、タスクスケジューラは、実行を継続するためにメモリからデータ 1 を呼び出すようにハードウェアアクセラレータ 2 に指示する。タスクスケジューラはさらに、データ 1 が実行された後に取得されたデータ 2 を格納する（例えば、データ 2 をメモリ D D R に引き続き格納する）ようにハードウェアアクセラレータ 2 に指示する。最後に、タスクスケジューラは、実行を継続するためにメモリからデータ 2 を呼び出すようハードウェアアクセラレータ 3 に指示し、タスクスケジューラは、さらに、データ 2 が実行された後に取得されたデータ 3 を格納するようハードウェアアクセラレータ 3 に指示する。毎回ターゲットタスクを実行するハードウェアアクセラレータを区別するために、ターゲットタスクの元の識別子が、ターゲットタスクを実行するハードウェアアクセラレータに関する対応する情報を伝達し（carry）得る（元の識別子と、その情報を伝達する元の識別子の両方はターゲットタスクの識別子と呼ばれることがある）、またはターゲットタスクを実行するハードウェアアクセラレータに関する情報は別々に格納され得ることが留意されるべきである。これは特にここでは限定されない。一例が説明のために使用される。ターゲットタスクがハードウェアアクセラレータ 1 ~ 3 によって順に実行され、ターゲットタスクの識別子が M 0 0 2 であると仮定する場合、識別子 M 0 0 2 - 1 は、取得されたデータ 1 がハードウェアアクセラレータ 1 によってターゲットタスク M 0 0 2 が実行された後に取得されたデータであることを示すために使用され得る。同様に、M 0 0 2 - 1 - 2 は、取得されたデータ 2 がハードウェアアクセラレータ 1 およびハードウェアアクセラレータ 2 によってターゲットタスクが順に実行された後に

20

30

40

50

取得されたことを示すために使用される。識別子 M 0 0 2 - 1 - 2 - 3 は、取得されたデータ 3 がハードウェアアクセラレータ 1 ~ 3 によってターゲットタスクが順に実行された後に取得されたことを示すために使用される。ターゲットタスクがハードウェアアクセラレータ 1 ~ 3 によって同時に実行される場合、同様に、識別子 M 0 0 2 - 1、M 0 0 2 - 2、および M 0 0 2 - 3 は、取得されたデータがハードウェアアクセラレータ 1、ハードウェアアクセラレータ 2、およびハードウェアアクセラレータ 3 によってターゲットタスクが別々に実行された後に取得されたことをそれぞれ示すために使用され得る。

【 0 0 6 2 】

本出願のいくつかの実装では、各ハードウェアアクセラレータが 1 つ以上のタスクを実行する場合があることが留意されるべきである。この場合、各ハードウェアアクセラレータは 1 つの実行キューに対応し、ハードウェアアクセラレータによって実行されることになるタスクは各実行キューに順に格納される。ターゲットタスクを実行する複数のハードウェアアクセラレータがある場合、ターゲットタスクの識別子（ターゲットタスクの元の識別子であり得る、またはハードウェアアクセラレータによってターゲットタスクが実行された後の対応する識別子であり得る）は、ターゲットタスクのための各ハードウェアアクセラレータに対応する実行キューに格納される。これはここでは限定されない。わかりやすくするため、例がここでは説明のために使用される。図 4 に示すように、ターゲットタスクに関連付けられた 3 つの第 1 の関連付けられたタスク：タスク 1、タスク 2、およびタスク 3 があることが想定され、タスク 1 の依存関係およびタスク 2 の依存関係に基づいて、タスク 1 およびタスク 2 の両方がハードウェアアクセラレータ 1 によって実行され、ハードウェアアクセラレータ 1 が最初にタスク 1 を実行し、その後タスク 2 を実行する必要があると決定され、ターゲットタスクの依存関係に基づいて、ターゲットタスクもハードウェアアクセラレータ 1 およびハードウェアアクセラレータ 3 によって順に実行されると決定され、タスク 3 の依存関係に基づいて、タスク 3 がハードウェアアクセラレータ 2 によって実行されると決定される。この場合、タスク 1 の識別子 Q 0 0 1、タスク 2 の識別子 Q 0 0 2、およびターゲットタスクの識別子 M 0 0 2 が、ハードウェアアクセラレータ 1 に対応する実行キュー 1 に順に配置される（タスク 1 もタスク 2 も実行されていないと仮定する）。タスク 3 の識別子 Q 0 0 3 がハードウェアアクセラレータ 2 に対応する実行キュー 2 に格納される（タスク 3 は実行されていないと仮定する）。ターゲットタスクがハードウェアアクセラレータ 1 によって実行された後に取得されたデータの識別子 M 0 0 2 - 1 はハードウェアアクセラレータ 3 に対応する実行キュー 3 に格納される。

【 0 0 6 3 】

図 4 に示すように、各ハードウェアアクセラレータに対応する 1 つの実行キューがあり、実行キューは、タスクセット内にあり、対応するハードウェアアクセラレータによって実行されることになるタスクの識別子を格納することが留意されるべきである。本出願のいくつかの実装では、同時に複数のタスクセットがある場合、例えば 3 つのタスクセットがあると仮定すると、各ハードウェアアクセラレータは 3 つの実行キューに対応する。図 2 に示すように、1 つのハードウェアアクセラレータに対応する 3 つの実行キューは、それぞれ 3 つのタスクセットに対応する。m 個のタスクセットがある場合、各ハードウェアアクセラレータに m 個の実行キューがある。

【 0 0 6 4 】

本出願のいくつかの実装では、実行キューはタスクスケジューラ内にあるストレージ領域であり得る、またはメモリ内にあるストレージ領域であり得る、または各ハードウェアアクセラレータから対応して分割されるストレージ領域であり得ることが留意されるべきである。これは特にここでは限定されない。

【 0 0 6 5 】

本出願のいくつかの実装では、対応するターゲットタスクを実行するようにハードウェアアクセラレータをスケジューリングした後、タスクスケジューラは、ターゲットタスクを実行する対応するハードウェアアクセラレータによって返される指示メッセージを受信することがさらに留意されるべきである。指示メッセージは、ハードウェアアクセラレー

10

20

30

40

50

タがターゲットタスクを実行したことを示す。この場合、タスクスケジューラは、ハードウェアアクセラレータに対応する実行キューからターゲットタスクの識別子を削除し得る。理解しやすいように、図4が依然として説明のための例として使用される。タスク1がハードウェアアクセラレータ1によって実行された後、ハードウェアアクセラレータ1は、タスク1が実行されたことを示すために、指示メッセージをタスクスケジューラに返す。この場合、タスクスケジューラは、実行キューからタスク1の識別子Q001を削除する。同様に、その後、タスク2が実行キュー1に示す順序で実行される。タスク2がハードウェアアクセラレータ1によって実行された後、タスク2の識別子Q002が実行キューから削除され、その後、元のターゲットタスクが実行キュー1に示された順序で実行される。ターゲットタスクがハードウェアアクセラレータ1によって実行された後、ターゲットタスクの識別子M002が実行キューから削除される。加えて、タスクを実行するたびに、ハードウェアアクセラレータ1は、実行されたタスクに対応するデータを格納する（例えば、メモリDDRにデータを格納する）。ターゲットタスクがハードウェアアクセラレータ1によって実行された後、取得されたデータ1が格納される。タスク1、タスク2、およびタスク3がすべて実行された後、タスクスケジューラは、ハードウェアアクセラレータ3にデータ1を実行するように指示する（実行キュー3のM002-1の前に他のタスクの識別子がないことが前提）。

【0066】

本出願のいくつかの実装では、ターゲットタスクが実行される前に最初に実行される必要があるタスクおよびこれらのタスクの実行順序（すなわち、第1の関連付けられたタスク）はターゲットタスクの依存関係に基づいて決定され得、ターゲットタスクが実行された後に実行条件を満たすタスク（第2の関連付けられたタスクと呼ばれる場合もある）が依存関係からさらに決定され得ることが留意されるべきである。例が説明のために使用される。第2の関連付けられたタスクがタスクmであると仮定する。タスクmを実行するための前提条件が、ターゲットタスクが実行されたことである場合、タスクmの依存関係において、ターゲットタスクはタスクmの第1の関連付けられたタスクである。タスクmが実行されるとき、タスクmは上記のように新しいターゲットタスクとして処理される。詳細はここでは再度説明しない。類推すると、タスクセット内の各タスクは独自の依存関係を有し、タスクセット内のすべてのタスクは、前述のスケジューリング方法で各タスクの依存関係に基づいて実行され得る。

【0067】

前述の解決策をよりよく理解するために、以下はタスクセット内のタスク間の特定の依存関係を説明するために図5を例として使用する。タスクセットは7つのタスクを含むことが仮定され、これらはそれぞれTS1～TS7と表記され、それらの実行順序および依存関係が図5に示される。TS7がTS1から実行された後にのみ、タスクセット全体が完了する。より良くかつより明確に理解するために、「イベント(event)」の概念が導入される。図6に示すように、b0～b5は6個のイベントである。したがって、タスクTS1～TS7のグループは「イベントタスクグループ」と呼ぶこともある。図6に示した意味については、表1のタスクの説明および表2のイベントの説明を参照されたい。

10

20

30

40

50

【表 1】

表 1：タスクの説明

タスク I D	エントリイベント	関連イベント
T S 1	タスク T S 1 はエントリタスクであり、イベント b 0 によってトリガされる。	タスク T S 1 が実行された後、イベント b 1 がトリガされる。
T S 2	タスク T S 2 はイベント b 1 によってトリガされる。	タスク T S 2 が実行された後、イベント b 2 および b 4 がトリガされる。
T S 3	タスク T S 3 はイベント b 1 によってトリガされ、タスク T S 3 および T S 2 は同時に実行されることがある。	タスク T S 3 が実行された後、イベント b 3 および b 4 がトリガされる。
T S 4	タスク T S 4 はイベント b 2 によってトリガされる。	タスク T S 4 が実行された後、イベント b 3 がトリガされる。
T S 5	タスク T S 5 はイベント b 3 によってトリガされる。	タスク T S 5 が実行された後、イベント b 5 がトリガされる。
T S 6	タスク T S 6 はイベント b 4 によってトリガされる。	タスク T S 6 が実行された後、イベント b 5 がトリガされる。
T S 7	タスク T S 7 は、イベント b 5 によってトリガされる。	タスク T S 7 は終了タスクである。タスク T S 7 が処理された後、イベントタスクグループは完了する。

10

20

30

40

50

【表 2】

表 2：イベントの説明		
イベント I D	依存	イベントに関連付けられたタスクのリスト
b 0	実行を開始するようにソフトウェアによってトリガされるイベントタスクグループの開始イベント。	T S 1
b 1	タスク T S 1 の完了のみに依存する。	T S 2 および T S 3
b 2	タスク T S 2 の完了のみに依存する。	T S 4
b 3	タスク T S 3 および T S 4 の完了に依存する。	T S 5
b 4	タスク T S 2 および T S 3 の完了のみに依存する。	T S 6
b 5	タスク T S 5 および T S 6 の完了のみに依存する。	T S 7

【 0 0 6 8 】

上記のタスク間の関連関係に基づいて、各タスクと他のタスク間の依存関係が取得される。各タスクは 1 つの依存関係に対応し得る、または、タスクセット内のすべてのタスク間の関連関係が依存関係（例えば、図 7 に示す 2 次元チャート）を形成し得ることが留意されるべきである。これはここでは限定されない。

【 0 0 6 9 】

加えて、イベントベースのタスクスケジューリングプロセスは、具体的には次のようになり得る：ソフトウェアは、最初に b 0 をトリガしてタスク T S 1 を開始および実行する。タスク T S 1 が実行された後、イベント b 1 がトリガされる。イベント b 1 は、同時に実行されるようにタスクリストの T S 2 および T S 3 をスケジューリングする。タスク T S 2 が実行された後、イベント b 2 および b 4 が通知される。イベント b 2 は、実行されるようにタスクリストの T S 4 をスケジューリングする。タスク T S 3 が実行された後、イベント b 3 および b 4 が通知される。イベント b 4 が条件を満たす場合、タスクリストの T S 6 が実行されるようにスケジューリングされる。タスク T S 6 が実行された後、イベント b 5 が通知される。タスク T S 4 が実行される場合、イベント b 3 が通知される。イベント b 3 が条件を満たす場合、タスクリストの T S 5 が実行されるようにスケジューリングされる。イベント b 5 が条件を満たす場合、タスクリストの T S 7 が実行されるようにスケジューリングされる。タスク T S 7 が実行された後、イベントタスクグループ全体のタスクが完了する。

【 0 0 7 0 】

前述の説明に基づいて、本出願のいくつかの実装では、各タスクの依存関係は 2 次元チャートの方法でメモリに格納され得る。図 7 は、1 つのタスクセット内の各タスクの 1 つの依存関係の 2 次元チャート、すなわち図 7 のタスクグラフ 0 である。タスクグラフ n は、n 番目のタスクセット内の各タスクの 1 つの依存関係の 2 次元チャート（図 7 には詳細には示されていない）を表す。図 7 では、タスクグラフ 0 は、タスクセットに 7 つのタスクがあることを示し、これらのタスクはそれぞれ t 0 ~ t 6 である。タスクグラフ 0 の上

部は、タスク $t_0 \sim t_6$ の実行順序（矢印）と関連関係を表している。例として t_4 を用いる。 t_4 を実行するための前提条件は、 t_1 および t_2 の両方が実行されたことである。タスクグラフ 0 の中央部分は実行キュー内のタスクの識別子の配列順序を表し、識別子はタスクの実行順序の「先頭」から「末尾」まで矢印方向に配列される。タスクグラフ 0 の下部は、各実行キューに対応するエグゼキュータが、タスクスケジューラと各ハードウェアアクセラレータとの間のインターフェイスを提供し、実行キュー内のタスクの実行順序において、対応するハードウェアアクセラレータに実行命令を配信することを表す。

【0071】

本出願の前述の実装では、タスクスケジューラはまずターゲットタスクを取得し（例えば、はメモリからターゲットタスクを取得し）、ターゲットタスクと関連付けられたタスクとの間の依存関係を取得する。依存関係に基づいて、関連付けられたタスク内の第 1 の関連付けられたタスク（例えば、ターゲットタスクを実行するための前提条件は、タスク 1、タスク 2 などすべて成功裏に実行されていることである）が成功裏に実行されたと決定される場合、ターゲットタスクが実行条件を満たしていることを示す。この場合、タスクスケジューラは、ターゲットタスクを実行するようにターゲットタスクの実行に関連するハードウェアアクセラレータ

10

（1 つまたは複数であり得る）をスケジューリングする。本出願のこの実施形態では、タスクスケジューラが CPU と各ハードウェアアクセラレータとの間に接続される。タスクスケジューラは、ハードウェアスケジューリングを介して、タスク間の依存関係に基づいて、各タスクを実行するよう H A C をスケジューリングするように構成される。各タスクの配信は、直接ハードウェアアクセスによって実行される。これは、タスクの配信の遅延を短縮し、間接的にシステムの処理性能を向上させる。

20

【0072】

以下は、制御システムが M D C シナリオにおける制御システムである例を使用して、ハードウェアアクセラレータをスケジューリングするための前述の方法を説明する。M D C シナリオでは、現在の制御システムにおけるハードウェアアクセラレータ（いくつかのハードウェアアクセラレータのみが表示されている、例えば、ビデオデコーディングアクセラレータ、ビジョン前処理アクセラレータ、画像デコーディングアクセラレータ、画像エンコーディングアクセラレータ、および A I モジュール）の位置が図 8 に示される。メインコントロール CPU は、対応するタスクを実行するように各ハードウェアアクセラレータをスケジューリングする。音声、画像、またはビデオが、外部センサまたは P C I e インターフェイスから入力され、ストレージのためにメモリに送信され得る。代替的には、音声、画像、またはビデオが、画像デコーディング（J P E G デコーディング、J P E G D）アクセラレータまたはビデオデコーディング（Video Decoding、V E D C）アクセラレータなどのハードウェアアクセラレータによって処理された後、ビジョン前処理がビジョン前処理（Vision Pre-Processing、V P C）アクセラレータによって実行され、その後さらなる処理のために A I モジュールに送信される。M D C シナリオでは、1 つのスケジューリングされたタスクは、センサ（例えば、カメラ）を使用することによって端末装置（例えば、車両）によって取得されたデータフロー（生データと呼ばれることもある）が一連の関連するハードウェアアクセラレータによって処理されて、指定された機能を持つ出力データ（例えば、障害物を感知するために使用されるデータおよび別のデバイスに送信されるために使用されるデータ）を取得する特定の実装プロセスである。本出願のいくつかの実装では、1 つのスケジューリングされたタスクは、フロントエンドタスクおよびバックエンドタスクに具体的に分けられ得る。フロントエンドタスクは、主にデータ入力を担当し、主に複数のタスク、例えば、マルチセンサソース（画像、ビデオなど）情報フローの収集、画像処理、クロッピング（cropping）、再構築（reassembling）、サイズ調整などのデータ処理、およびそのプロセスにおけるデータ格納とデータ転送を含む。異なるタスクを処理するハードウェアアクセラレータは、連携するようにスケジューリングされる必要がある。バックエンドタスクは、処理されたフロントエンドタスクによって取得されたデータを計算することを主に担当する。すなわち、フロントエンドタスク

30

40

50

によって処理されたデータが取得された後、A I モジュールの A I C P U は、データの計算、分析、トレーニング、予測、および調整などの後続タスクを実行するように通知される。本出願の他のいくつかの実装では、1つのスケジューリングされたタスクはフロントエンドタスクのみを含み得ることが留意されるべきである。例えば、タスクスケジューラが、入力データに対する画像処理、クロッピング、再構築、サイズ調整などの処理を実行するように各ハードウェアアクセラレータを呼び出した後、処理されたデータは、別の端末装置（携帯電話やパソコンなど）に送信され、別の端末装置は、A I コンピューティングを実行し、その後計算結果を端末装置に返す。

【0073】

図9はMDCシナリオにおける制御システムのタスクスケジューラの位置を示す。タスクスケジューラは、各ハードウェアアクセラレータ、例えば、図9のビデオデコーディングアクセラレータ、ビジョン前処理アクセラレータ、画像デコーディングアクセラレータ、または画像エンコーディングをスケジューリングするように、各ハードウェアアクセラレータへのインターフェイスを提供する。MDCシナリオでは、1つのスケジューリングされたタスクのタスクフロー（タスクセットとも呼ばれることがある）が、複数のタスクに分割され、並列処理または逐次処理が複数のハードウェアアクセラレータによって実行される。本出願の本実施形態におけるハードウェアアクセラレータをスケジューリングする方法は、スケジューリングされたタスクの処理性能を向上させるために使用される。タスクセット内のタスクが実行された後、タスクスケジューラは、タスクが実行された後に生成されたデータを格納する（例えば、データをメモリに格納する）。タスクセット内のタスクは、1つのスケジューリングされたタスクを形成する。元のスケジューリングされたタスクは、タスクセット内のすべてのタスクが実行された後にのみ完了する。前述のスケジューリング方法によれば、1つのスケジューリングされたタスクにおけるターゲットタスクは、具体的には、タスクスケジューラが、A I モジュールの計算後にフィードバックデータを取得するために、A I コンピューティングを実行するようにA I モジュール（例えば、A I C P U ）に指示することであり得る。フィードバックデータは、車両上のユーザの操作をガイドするために使用され得る。具体的には、いくつかの適用シナリオでは、フィードバックデータは、車線境界線および/または停止線を感知するために使用されるデータであり得、車両所有者が指定された車線を走行するまたは指定された駐車エリアで停止するように案内するために使用される。代替的には、フィードバックデータは、安全領域を感知するために使用されるデータであり得、この車両を他の車両から安全な距離に保つように車両所有者を案内するために使用される。代替的には、フィードバックデータは障害物（例えば障害物からの距離）を感知するために使用されるデータであり得、事前に安全対策（例えば、停止や方向転換）を講じるように車両所有者を案内するために使用される。フィードバックデータの種類および機能はここでは限定されない。対応するフィードバックデータは、実際の適用シナリオに基づいて取得され得る。したがって、車両所有者は、対応するフィードバックデータに基づいて車両を操作するように案内され、それによって運転の安全性および運転性能を向上させる。

【0074】

スケジューリングされたタスクをよりよく理解するために、以下はMACの適用シナリオを参照して説明を提供する。図10を参照されたい。車両に搭載された各種センサ（一般的なレーダー、カメラ、レーザーレーダー、および慣性測定ユニットなど）は、まず各種センサデータを取得する。加えて、マップが、ロケーションメッセージを取得するまたはローカルマップを取得するために使用される。データは生データと呼ばれることがある。次に、生データは、処理されたデータ（すなわち、図10のセンシング（sensing（感知））データ）を取得するために、処理のために対応するハードウェアアクセラレータにタスクスケジューラによって割り当てられる。取得されたセンシングデータの特定の目的は、異なるハードウェアアクセラレータによって異なる。例えば、レーダーセンシングのためのアクセラレータ（1つまたは複数であり得る）が生データを処理した後に出力されるセンシングデータは、障害物を感知するために使用され得、視覚的な車線センシングの

10

20

30

40

50

ためのアクセラレータ（１つまたは複数であり襟）が生データを処理した後に出力されるセンシングデータは、車線境界線を検知するために使用され得る。生データから、タスクスケジューラによって各ハードウェアアクセラレータをスケジューリングすることによって取得されたセンシングデータまでの各プロセスは、スケジューリングされたタスクの特定の実行プロセスである。最後に、ユーザが直感的に読み取ることができるデータ（例えば、ディスプレイインターフェイスに表示されるシミュレートされた車線境界線またはシミュレートされた安全領域）は、各センシングデータの融合（fusion）または空間認識によって取得される。

【 0 0 7 5 】

本出願のいくつかの実装では、タスクセット内の各タスクを形成するデータは、搭載されたセンサデバイス（例えば、カメラ）を使用することにより車両によって取得され得ることが留意されるべきである。

【 0 0 7 6 】

本出願のいくつかの実装では、１つのスケジューリングされたタスク内の分割されたタスク間の依存関係は図 7 の 2 次元チャートを用いて表され得ることが留意されるべきである。2 次元チャートはメモリに格納され得る。

【 0 0 7 7 】

以下は、本出願の本実施形態で説明した方法が適用される実際のシナリオを説明する。図 1 1 は、サラウンドビューステッチング（surround-view stitching）およびイメージセンシング部の典型的なデータフローグラフを示す。一方向矢印の線は、データフローのフロー方向を示す。双方向矢印の線は、タスクスケジューラと各ハードウェアアクセラレータとの間にインターフェイスがあることを示す。タスクスケジューラは、これらのインターフェイスを介して各ハードウェアアクセラレータをスケジューリングし得る。図 1 1 に示すように、タスクスケジューラは、画像処理アクセラレータ 1 ~ 4、ビジョン前処理アクセラレータ 1 ~ 3、および AI モジュール 1 ~ 2 のいずれか 1 つをスケジューリングし得る。車両の前後左右にそれぞれ設置された 4 つの魚眼カメラは、まず画像データを収集するように構成されている。タスクスケジューラは、画像処理アクセラレータ 1 ~ 4 をスケジューリングして、対応する分岐（branches）の画像データに対して画像処理を行い、その後、4 チャンネルの画像データに対して画像歪み補正およびサラウンドビューステッチングを行い、1 チャンネルのパノラマを出力する。パノラマは 3 か所に出力される（すなわち、3 つのタスクセットが形成される）。第 1 のチャンネル（すなわち、第 1 のタスクセット）がビジョン前処理アクセラレータ 1 に送信されて処理されたデータを取得し、その後処理されたデータは AI モジュール 1 に計算のために出力される。計算されたフィードバックデータは、車線境界線および停止線を検知するために使用される。同様に、第 2 のチャンネル（すなわち、第 2 のタスクセット）が別のビジョン前処理アクセラレータ 2 に送信されて処理されたデータを取得し、その後処理されたデータは別の AI モジュール 2 に計算のために出力される。計算されたフィードバックデータは、車両が走行する安全領域を検知するために使用される。第 3 のチャンネル（すなわち、第 3 のタスクセット）は、画像のエンコーディング、圧縮、およびストレージのためにビデオエンコーディングアクセラレータに直接送信される。各タスクセットに含まれる各タスクは、前述のスケジューリング方法に従って処理され得る。詳細は、図 3 に対応する本実施形態の方法を参照されたい。詳細はここでは説明しない。

【 0 0 7 8 】

図 3 に対応する実施形態によれば、本出願のこの実施形態を実装するための上述の解決策をよりよく説明するために、以下は上述の解決策を実装するように構成されたタスクスケジューラを提供する。詳しくは、図 1 2 は、本出願の一実施形態によるタスクスケジューラの構造の概略図である。タスクスケジューラ 1 2 0 0 は、取得モジュール 1 2 0 1、決定モジュール 1 2 0 2、およびスケジューリングモジュール 1 2 0 3 を含む。取得モジュール 1 2 0 1 は、ターゲットタスクを取得するように構成される。決定モジュール 1 2 0 2 は、依存関係に基づいて、ターゲットタスクに関連付けられた第 1 の関連付けられた

10

20

30

40

50

タスクを決定するように構成される。依存関係は、タスクセット内のタスクの実行順序を示す。スケジューリングモジュール 1203 は、決定モジュール 1202 が第 1 の関連付けられたタスクが実行されたと決定するときに、ターゲットタスクを実行するよう少なくとも 1 つのハードウェアアクセラレータをスケジューリングするように構成される。

【0079】

可能な設計では、スケジューリングモジュール 1203 がターゲットタスクを実行するために少なくとも 1 つのハードウェアアクセラレータをスケジューリングするとき、少なくとも 1 つのハードウェアアクセラレータがターゲットタスクを順に実行する。

【0080】

可能な設計では、少なくとも 1 つのハードウェアアクセラレータのそれぞれが実行キューに対応する。ターゲットタスクの識別子が、各ハードウェアアクセラレータに対応する実行キューに格納される。

10

【0081】

可能な設計では、スケジューリングモジュール 1203 はさらに：スケジューリングモジュール 1203 がターゲットタスクを実行するように少なくとも 1 つのハードウェアアクセラレータをスケジューリングした後、少なくとも 1 つのハードウェアアクセラレータによって返された指示メッセージを受信し、その指示メッセージは少なくとも 1 つのハードウェアアクセラレータがターゲットタスクを実行したことを示し、少なくとも 1 つのハードウェアアクセラレータに対応する実行キューからターゲットタスクの識別子を削除する、ように構成される。

20

【0082】

可能な設計では、スケジューリングモジュール 1203 は、さらに：タスクセット内の各タスクが実行された後に取得されたデータを格納するように構成される。タスクセット内のすべてのタスクは、スケジューリングされたタスクを形成する。

【0083】

可能な設計では、タスクセット内のタスクを形成するデータは、端末装置に搭載されたカメラ装置を使用することにより端末装置によって取得される。

【0084】

可能な設計では、タスクセット内の各タスクが実行された後に取得されるデータは、AI モジュールによって計算されたフィードバックデータを含む。フィードバックデータは、端末装置の操作を案内するために使用される。AI モジュールは、少なくとも 1 つのハードウェアアクセラレータに属する。

30

【0085】

可能な設計では、端末装置は車両である。フィードバックデータは、車線境界線および/または停止線を感知するために使用されるデータ、または安全エリアを感知するために使用されるデータ、障害物を感知するために使用されるデータを含む。フィードバックデータの種別および機能はここでは限定されない。対応するフィードバックデータは、実際のアプリケーションシナリオに基づいて取得され得る。したがって、車両所有者はフィードバックデータに基づいて車両を操作するように案内され、それによって運転の安全性および運転性能を向上させる。

40

【0086】

タスクスケジューラ 1200 におけるモジュール/ユニット間の情報交換および実行プロセスなどの内容は、本出願における図 3 に対応する方法の実施形態と同じ概念に基づいていることが留意されるべきである。具体的な内容については、本出願の上述の方法の実施形態の説明を参照されたい。詳細はここでは説明しない。

【0087】

本出願の一実施形態は、さらにタスクスケジューラを提供する。図 13 は、本出願の一実施形態によるタスクスケジューラの構造の概略図である。説明を容易にするため、本出願の本実施形態に関連する部分のみを示す。開示されていない具体的な技術的詳細については、本出願の実施形態における方法の部分を参照されたい。図 12 に対応する実施形態

50

で説明したタスクスケジューラのもジュールは、タスクスケジューラ 1 3 0 0 に展開され得、図 1 2 に対応する実施形態のタスクスケジューラの機能を実装するように構成される。具体的には、タスクスケジューラ 1 3 0 0 は、1 つまたは複数のサーバによって実装される。タスクスケジューラ 1 3 0 0 は、構成または性能が異なるために大きく異なる場合があり、1 つまたは複数の中央処理装置 (central processing units、CPU) 1 3 2 2 (例えば、1 つまたは複数のプロセッサ)、メモリ 1 3 3 2、およびアプリケーションプログラム 1 3 4 2 またはデータ 1 3 4 4 を格納する 1 つまたは複数の記憶媒体 1 3 3 0 (例えば、1 つまたは複数の大容量記憶装置) を含み得る。メモリ 1 3 3 2 および記憶媒体 1 3 3 0 は、一時記憶または永続記憶であり得る。記憶媒体 1 3 3 0 に格納されたプログラムは、1 つまたは複数のもジュール (図には示されていない) を含み得、各もジュールは、ハードウェアアクセラレータのための一連の命令操作を含み得る。さらに、中央処理装置 1 3 2 2 は、記憶媒体 1 3 3 0 と通信し、記憶媒体 1 3 3 0 内の一連の命令操作をタスクスケジューラ 1 3 0 0 上で実行するように構成され得る。例えば、中央処理装置 1 3 2 2 は、ターゲットタスクを取得し、依存関係に基づいて、ターゲットタスクに関連付けられた第 1 の関連付けられたタスクを決定するように構成され得る。依存関係は、タスクセット内のタスクの実行順序を示す。第 1 の関連付けられたタスクが実行されたことを決定する場合、中央処理装置 1 3 2 2 はターゲットタスクを実行するように少なくとも 1 つのハードウェアアクセラレータをスケジューリングする。

10

【0088】

タスクスケジューラ 1 3 0 0 はさらに、1 つまたは複数の電源 1 3 2 6、1 つまたは複数の有線または無線ネットワークインターフェイス 1 3 5 0、1 つまたは複数の入出力インターフェイス 1 3 5 8、および / または Windows Server (商標)、Mac OS X (商標)、Unix (商標)、Linux (商標)、FreeBSD (商標) などの 1 つまたは複数のオペレーティングシステム 1 3 4 1 を含み得る。

20

【0089】

本出願の本実施形態では、図 3 に対応する実施形態のタスクスケジューラによって実行されるステップが、図 1 3 に示す構造に基づいて実装され得る。詳細はここでは説明しない。

【0090】

加えて、記載されている装置の実施形態はあくまで一例であることが留意されるべきである。別々の部分として記述されるユニットは、物理的に分かれていてもそうでなくてもよく、ユニットとして表示される部分は、物理的なユニットであってもそうでなくてもよく、1 つの位置に配置されていてもよく、または複数のネットワークユニットに分散されていてもよい。いくつかまたはすべてのもジュールは、実施形態の解決策の目的を達成するために、実際のニーズにしたがって選択され得る。加えて、本出願によって提供される装置の実施形態の添付図面では、もジュール間の接続関係は、もジュールが相互に通信接続を持つことを示しており、これは 1 つ以上の通信バスまたは信号ケーブルとして具体的に実装され得る。

30

【0091】

上述の実装の説明に基づいて、当業者は、本出願が必要なユニバーサルハードウェアに加えてソフトウェアによって、または専用の集積回路、専用の CPU、専用のメモリ、専用のコンポーネントなどを含む専用のハードウェアによって実装され得ることを明確に理解することができる。概して、コンピュータプログラムによって実行することができる機能は、対応するハードウェアを使用することによって簡単に実装することができる。さらに、同じ機能を実現するために使用される特定のハードウェア構造は、例えば、アナログ回路、デジタル回路、または専用回路の形態の、さまざまな形態のものであり得る。しかし、本出願については、ソフトウェアプログラムの実装はほとんどの場合、より優れた実装である。このような理解に基づき、本出願の技術的な解決策は本質的に、または、従来の技術に貢献する部分は、ソフトウェア製品の形で実装され得る。コンピュータソフトウェア製品は、コンピュータのフロッピー (登録商標) ディスク、USBフラッシュドライブ

40

50

、リムーバブルハードディスク、ROM、RAM、磁気ディスク、または光ディスクなどの可読記憶媒体に格納され、本出願の実施形態に記載されている方法を実行するようにコンピュータデバイス（これは、パーソナルコンピュータ、トレーニングデバイス、またはネットワークデバイスであり得る）に指示するためのいくつかの命令を含む。

【0092】

上述の実施形態の全てまたはいくつかは、ソフトウェア、ハードウェア、ファームウェア、またはそれらの任意の組み合わせによって実装され得る。ソフトウェアを使用して実施形態を実装する場合、実施形態の全てまたは一部はコンピュータプログラム製品の形で実装され得る。

【0093】

コンピュータプログラム製品は、1つ以上のコンピュータ命令を含む。コンピュータプログラム命令がコンピュータにロードされて実行されるとき、本出願の実施形態に従った手順または機能が全てまたは部分的に生成される。コンピュータには、汎用コンピュータ、専用コンピュータ、コンピュータネットワーク、またはその他のプログラム可能な装置であり得る。コンピュータ命令は、コンピュータ可読記憶媒体に格納され得るまたはコンピュータ可読記憶媒体から別のコンピュータ可読記憶媒体に送信され得る。例えば、コンピュータの命令は、ウェブサイト、コンピュータ、トレーニングデバイス、またはデータセンターから別のウェブサイト、コンピュータ、トレーニングデバイス、またはデータセンターに有線（例えば、同軸ケーブル、光ファイバ、またはデジタル加入者線（DSL））または無線（赤外線、ラジオ、またはマイクロ波）で送信され得る。コンピュータ可読記憶媒体は、コンピュータによってアクセス可能な任意の使用可能な媒体、または1つ以上の使用可能な媒体を統合したトレーニング装置もしくはデータセンターなどのデータストレージデバイスであり得る。使用可能な媒体は、磁気媒体（例えば、フロッピー（登録商標）ディスク、ハードディスク、または磁気テープなど）、光媒体（例えば、DVD）、半導体媒体（例えば、ソリッドステートディスク（Solid State Disk、SSD））等であり得る。

10

20

30

40

50

【図面】

【図 1】

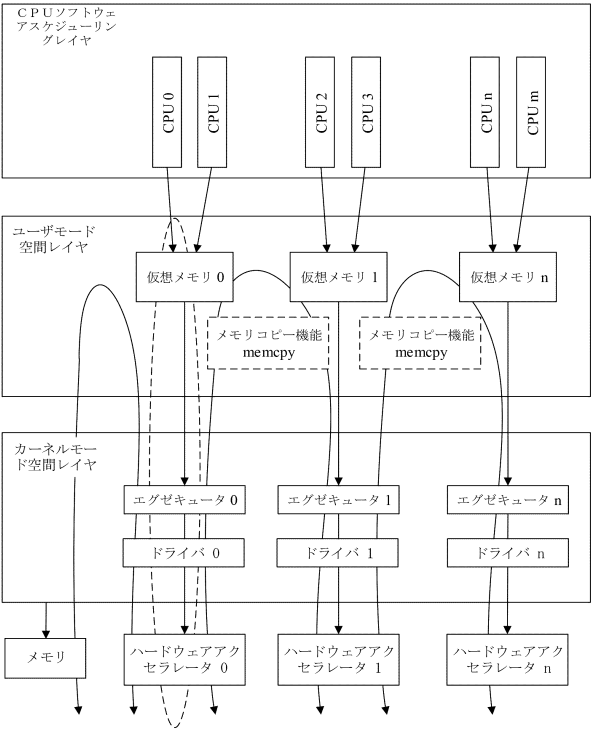


FIG. 1

【図 2】

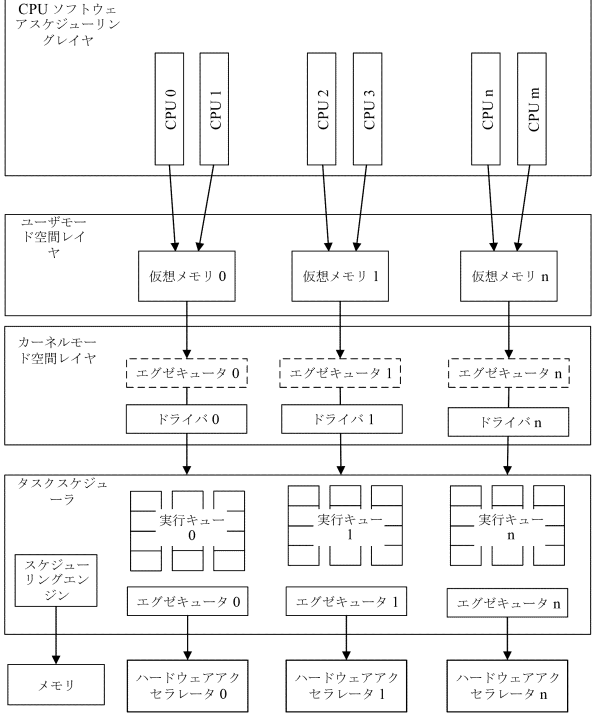


FIG. 2

【図 3】

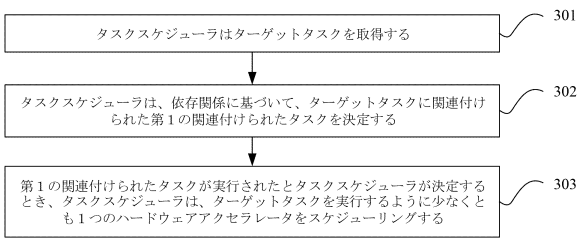


FIG. 3

【図 4】

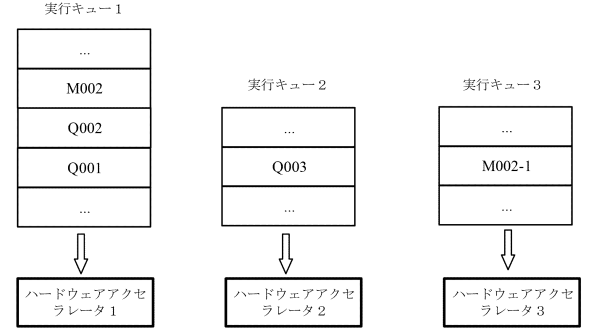


FIG. 4

10

20

30

40

50

【図 5】

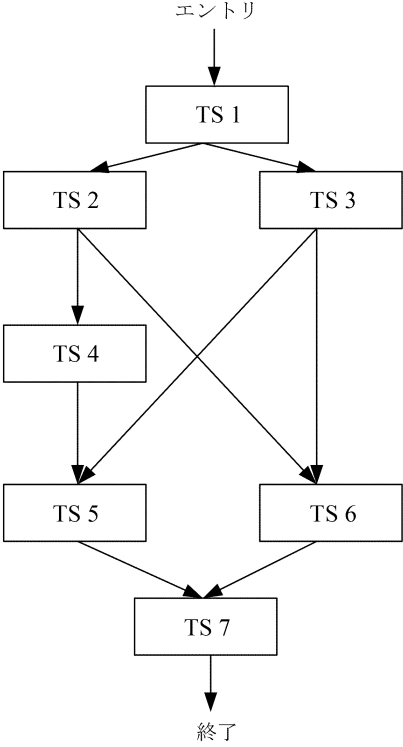


FIG. 5

【図 6】

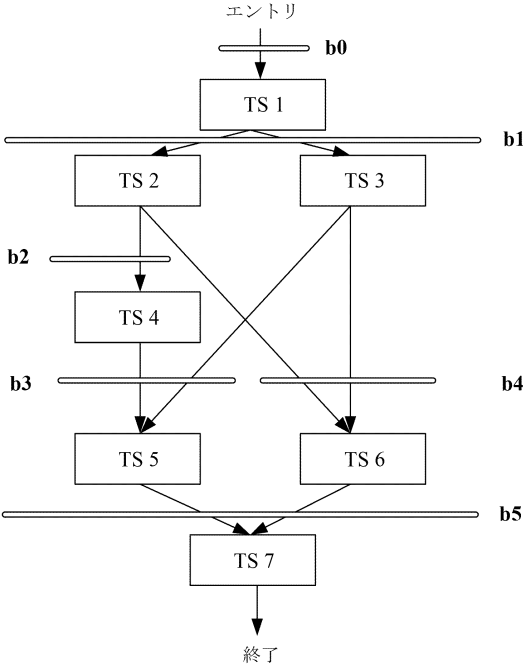


FIG. 6

【図 7】

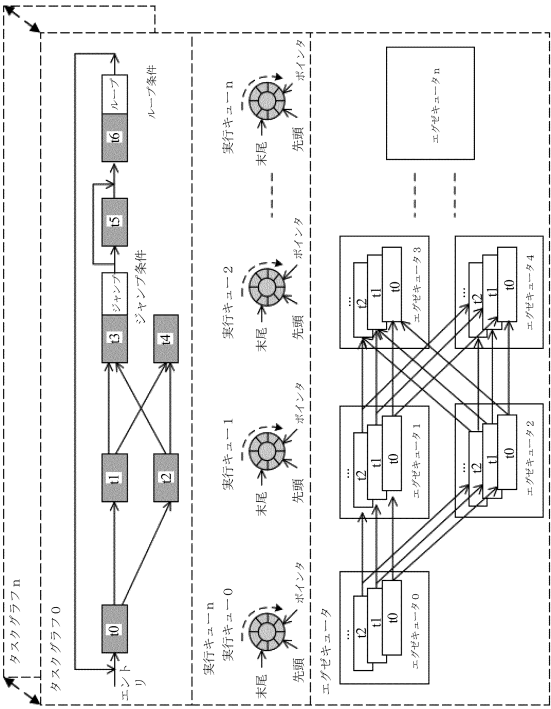


FIG. 7

【図 8】

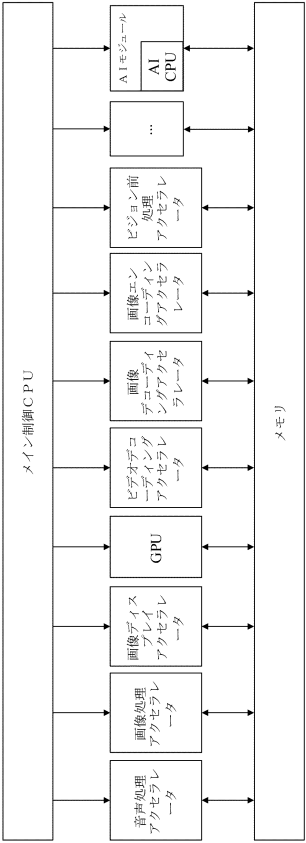


FIG. 8

【図 9】

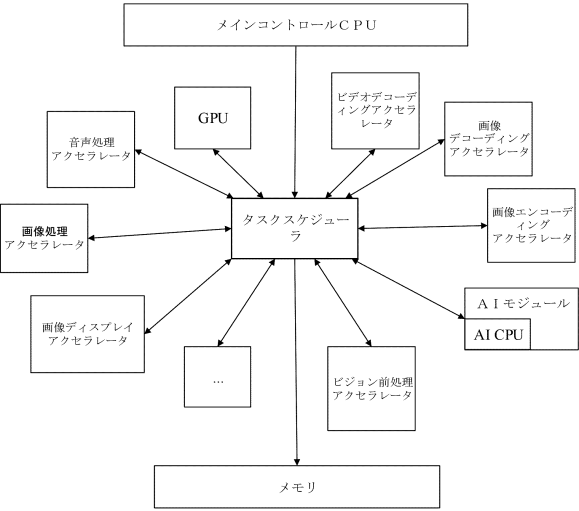


FIG. 9

【図 10】

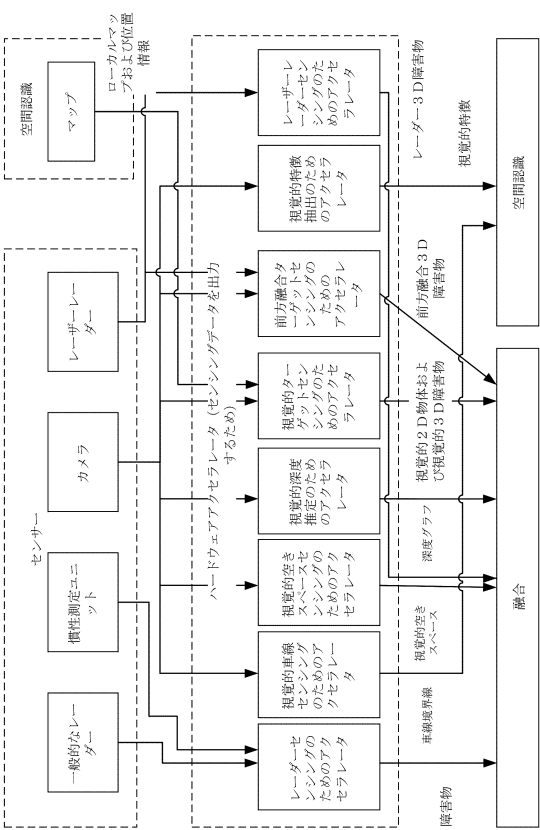


FIG. 10

【図 11】

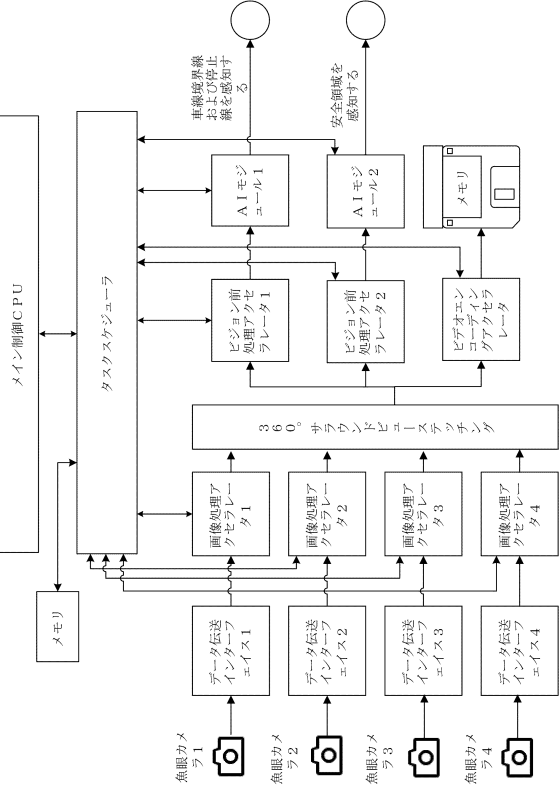


FIG. 11

【図 12】

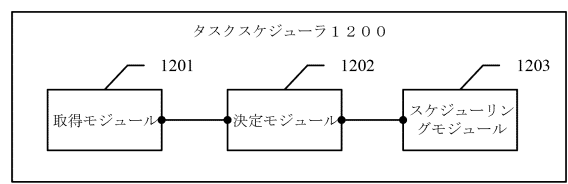


FIG. 12

10

20

30

40

50

【図 13】

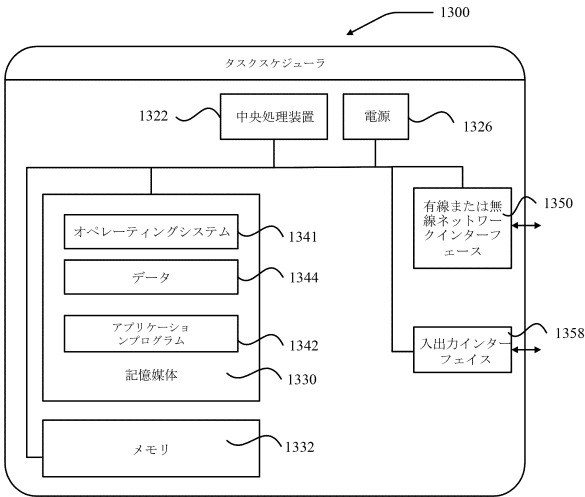


FIG. 13

10

20

30

40

50

フロントページの続き

- (74)代理人 100070150
弁理士 伊東 忠彦
- (74)代理人 100135079
弁理士 宮崎 修
- (72)発明者 ホアン, ウエイチャオ
中国 5 1 8 1 2 9 グァンドン シェンチェン ロンガン・ディストリクト バンティエン ホァウ
エイ・アドミニストレーション・ビルディング
- (72)発明者 シアオ, ロォン
中国 5 1 8 1 2 9 グァンドン シェンチェン ロンガン・ディストリクト バンティエン ホァウ
エイ・アドミニストレーション・ビルディング
- 審査官 漆原 孝治
- (56)参考文献 国際公開第 2 0 1 7 / 1 3 1 1 8 7 (W O , A 1)
特開 2 0 0 9 - 0 6 9 9 2 1 (J P , A)
特開 2 0 0 9 - 0 9 3 2 1 8 (J P , A)
- (58)調査した分野 (Int.Cl., D B 名)
G 0 6 F 9 / 4 8