

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号  
特許第4690988号  
(P4690988)

(45) 発行日 平成23年6月1日 (2011.6.1)

(24) 登録日 平成23年2月25日 (2011.2.25)

(51) Int.Cl.

F I

G O 6 F 9/50 (2006.01)

G O 6 F 9/46 4 6 5 Z

G O 6 F 9/48 (2006.01)

G O 6 F 9/46 4 5 2 H

請求項の数 45 外国語出願 (全 22 頁)

(21) 出願番号	特願2006-266590 (P2006-266590)	(73) 特許権者	591003943
(22) 出願日	平成18年9月29日 (2006.9.29)		インテル・コーポレーション
(65) 公開番号	特開2007-102781 (P2007-102781A)		アメリカ合衆国 9 5 0 5 2 カリフォル
(43) 公開日	平成19年4月19日 (2007.4.19)		ニア州・サンタクララ・ミッション カレ
審査請求日	平成18年9月29日 (2006.9.29)		ッジ ブールバード・2 2 0 0
(31) 優先権主張番号	11/239, 475	(74) 代理人	100104156
(32) 優先日	平成17年9月30日 (2005.9.30)		弁理士 龍華 明裕
(33) 優先権主張国	米国 (US)	(72) 発明者	ガウタム チンヤ
			アメリカ合衆国、9 7 1 2 3 オレゴン州
			、ヒルズバロ、エス イー 4 9 ティーエ
			イチ シーティー、1 2 5
		(72) 発明者	ホン ワン
			アメリカ合衆国、9 5 0 5 2 カリフォル
			ニア州、サンタ クララ、ミッション カ
			レッジ ブールバード 2 2 0 0
			最終頁に続く

(54) 【発明の名称】 持続的なユーザレベルスレッド用の装置、システムおよび方法

(57) 【特許請求の範囲】

【請求項 1】

オペレーティングシステム (OS) によりスケジューリングされ OS 可視シーケンサで  
実行中の OS スレッドに基づき、命令セットエクステンションを用いて、前記 OS により  
スケジューリングされるスレッドに対するコンテキストスイッチ動作と独立して OS 分離  
シーケンサで実行されるユーザレベルスレッドである持続的なユーザレベルスレッドを生  
成する段階  
を備え、

前記 OS は、前記持続的なユーザレベルスレッドの命令に対してスケジューリングを行  
わず、

前記持続的なユーザレベルスレッドは、前記 OS によりスケジューリングされるスレ  
ッドがコンテキストスイッチされた時でも前記 OS 分離シーケンサで動作を継続し、

前記持続的なユーザレベルスレッドは、前記 OS 可視シーケンサで実行される別のユー  
ザレベルスレッドから生成される

方法。

【請求項 2】

前記持続的なユーザレベルスレッドを前記 OS 分離シーケンサで実行する段階  
をさらに備える請求項 1 に記載の方法。

【請求項 3】

前記 OS によりスケジューリングされるスレッドおよび前記持続的なユーザレベルスレ

ッドは仮想アドレススペースを共有する  
請求項 1 または 2 に記載の方法。

【請求項 4】

前記持続的なユーザレベルスレッドの実行中に OS による処理を必要とするイベントが発生した場合に、前記持続的なユーザレベルスレッドの実行を一時中断する段階をさらに備える請求項 1 から 3 のいずれかに記載の方法。

【請求項 5】

別の OS 可視シーケンサで実行されるサービススレッドに、前記一時中断された持続的なユーザレベルスレッドに対して、OS サービスを提供させる段階をさらに備える請求項 4 に記載の方法。

10

【請求項 6】

前記サービススレッドおよび前記持続的なユーザレベルスレッドは、同じアプリケーションのコンテキストで実行される  
請求項 5 に記載の方法。

【請求項 7】

前記持続的なユーザレベルスレッドに対して OS サービスを提供する段階は、  
前記別の OS 可視シーケンサに対して、制御移行命令を伝える段階と、  
前記制御移行命令を特権レベルで実行する段階と  
を有する請求項 5 または 6 に記載の方法。

20

【請求項 8】

前記特権レベルには Ring 0 レベルが含まれる  
請求項 7 に記載の方法。

【請求項 9】

前記持続的なユーザレベルスレッドに対して OS サービスを提供する段階は、  
前記サービススレッドに前記持続的なユーザレベルスレッドの状態を取得させる段階と、  
前記持続的なユーザレベルスレッドに対して代理実行を行う段階と、  
前記持続的なユーザレベルスレッドに実行後状態を戻す段階と  
を有する請求項 5 から 8 のいずれかに記載の方法。

30

【請求項 10】

前記実行後状態から前記持続的なユーザレベルスレッドの実行を再開する段階をさらに有する請求項 9 に記載の方法。

【請求項 11】

前記サービススレッドは、前記持続的なユーザレベルスレッドと共有している仮想アドレス空間を通じて前記持続的なユーザレベルスレッドの状態を取得する  
請求項 9 または 10 に記載の方法。

【請求項 12】

前記持続的なユーザレベルスレッドは、当該ユーザレベルスレッドを生成したスレッドである前記 OS によりスケジューリングされるスレッドに対するコンテキストスイッチ動作と独立して、動作を継続する  
請求項 1 から 11 のいずれかに記載の方法。

40

【請求項 13】

前記命令セットエクステンションは、前記 OS 可視シーケンサおよび前記 OS 分離シーケンサを識別するオペランドを持つ命令を含む  
請求項 1 から 12 のいずれかに記載の方法。

【請求項 14】

OS によりスケジューリングされ OS 可視シーケンサで実行中のスレッドに基づき、命令セットエクステンションを用いて、前記 OS によりスケジューリングされるスレッドに対するコンテキストスイッチ動作とは独立して OS 分離シーケンサで実行されるユーザレベルスレッドである持続的なユーザレベルスレッドを生成するシーケンサマネージャ

50

を備え、

前記OSは、前記持続的なユーザレベルスレッドの命令に対してスケジューリングを行わず、

前記持続的なユーザレベルスレッドは、前記OSによりスケジューリングされるスレッドがコンテキストスイッチされた時でも前記OS分離シーケンサで動作を継続し、

前記シーケンサマネージャは、前記OS可視シーケンサで実行される別のユーザレベルスレッドから前記持続的なユーザレベルスレッドを生成する装置。

【請求項15】

前記OSによりスケジューリングされるスレッドに対するコンテキストスイッチ動作とは独立して前記持続的なユーザレベルスレッドを実行する前記OS分離シーケンサをさらに備える請求項14に記載の装置。

10

【請求項16】

前記OSによりスケジューリングされるスレッドおよび前記持続的なユーザレベルスレッドは仮想アドレススペースを共有する  
請求項14または15に記載の装置。

【請求項17】

前記シーケンサマネージャは、前記持続的なユーザレベルスレッドの実行中にOSによる処理を必要とするイベントが発生した場合に、前記持続的なユーザレベルスレッドの実行を一時中断する  
請求項14から16のいずれかに記載の装置。

20

【請求項18】

前記シーケンサマネージャは、別のOS可視シーケンサで実行されるサービススレッドに、前記持続的なユーザレベルスレッドに対してOSサービスを提供させる  
請求項14から17のいずれかに記載の装置。

【請求項19】

前記サービススレッドと前記持続的なユーザレベルスレッドは、同じアプリケーションのコンテキストで実行される  
請求項18に記載の装置。

【請求項20】

前記シーケンサマネージャは、前記別のOS可視シーケンサに対して、制御移行命令を伝え、前記制御移行命令を特権レベルで実行させる  
請求項18または19に記載の装置。

30

【請求項21】

前記シーケンサマネージャは、前記サービススレッドに前記持続的なユーザレベルスレッドの状態を取得させ、前記持続的なユーザレベルスレッドに対して代理実行を行い、前記持続的なユーザレベルスレッドに実行後状態を戻す  
請求項18から20のいずれかに記載の装置。

【請求項22】

前記実行後状態から前記持続的なユーザレベルスレッドの実行を再開する  
請求項21に記載の装置。

40

【請求項23】

前記サービススレッドは、前記持続的なユーザレベルスレッドと共有している仮想アドレス空間を通じて前記持続的なユーザレベルスレッドの状態を取得する  
請求項21または22に記載の装置。

【請求項24】

前記OS分離シーケンサは、前記持続的なユーザレベルスレッドを生成したスレッドである前記OSによりスケジューリングされるスレッドに対するコンテキストスイッチ動作と独立して、前記持続的なユーザレベルスレッドの動作を継続させる  
請求項14から23のいずれかに記載の装置。

50

## 【請求項 25】

前記命令セットエクステンションは、前記 OS 可視シーケンサおよび前記 OS 分離シーケンサを識別するオペランドを持つ命令を含む  
請求項 14 から 24 のいずれかに記載の装置。

## 【請求項 26】

システムであって、

OS、少なくとも 1 つのユーザレベルプログラム、前記プログラムを接続するためのアプリケーションプログラムインターフェース、および前記プログラムの実行をスケジューリングするスケジューラを格納しているメモリと、

少なくとも 1 つの OS 可視シーケンサおよび少なくとも 1 つの OS 分離シーケンサを有するマルチシーケンサと、

OS によりスケジューリングされ OS 可視シーケンサで実行中のスレッドに基づき、命令セットエクステンションを用いて、前記 OS によりスケジューリングされるスレッドに対するコンテキストスイッチ動作とは独立して OS 分離シーケンサで実行される ユーザレベルスレッドである持続的なユーザレベルスレッド を生成するシーケンサマネージャとを備え、

前記 OS は、前記持続的なユーザレベルスレッドの命令に対してスケジューリングを行わず、

前記持続的なユーザレベルスレッドは、前記 OS によりスケジューリングされるスレッドがコンテキストスイッチされた時でも前記 OS 分離シーケンサで動作を継続し、

前記シーケンサマネージャは、前記 OS 可視シーケンサで実行される別のユーザレベルスレッドから前記持続的なユーザレベルスレッドを生成する

システム。

## 【請求項 27】

前記 OS によりスケジューリングされるスレッドに対するコンテキストスイッチ動作とは独立して前記持続的なユーザレベルスレッドを実行する前記 OS 分離シーケンサをさらに備える請求項 26 に記載のシステム。

## 【請求項 28】

前記 OS によりスケジューリングされるスレッドと前記持続的なユーザレベルスレッドは仮想アドレススペースを共有する

請求項 26 または 27 に記載のシステム。

## 【請求項 29】

前記シーケンサマネージャは、前記持続的なユーザレベルスレッドの実行中に OS による処理を必要とするイベントが発生した場合、前記持続的なユーザレベルスレッドの実行を一時中断する

請求項 26 から 28 のいずれかに記載のシステム。

## 【請求項 30】

前記シーケンサマネージャは、前記 OS 可視シーケンサで実行されるサービススレッドに、前記持続的なユーザレベルスレッドに対して、OS サービスを提供させる

請求項 26 から 29 のいずれかに記載のシステム。

## 【請求項 31】

前記シーケンサマネージャは、前記 OS 可視シーケンサに対して、制御移行命令を伝え、前記 OS 可視シーケンサに前記制御移行命令の特権レベルで実行させる

請求項 30 に記載のシステム。

## 【請求項 32】

前記シーケンサマネージャは、前記サービススレッドに前記持続的なユーザレベルスレッドの状態を取得させ、前記持続的なユーザレベルスレッドに対して代理実行を行い、前記持続的なユーザレベルスレッドに実行後状態を戻す

請求項 30 または 31 に記載のシステム。

## 【請求項 33】

10

20

30

40

50

前記実行後状態から前記持続的なユーザレベルスレッドの実行を再開する  
請求項 3 2 に記載のシステム。

【請求項 3 4】

前記サービススレッドは、前記持続的なユーザレベルスレッドと共有している仮想アドレス空間を通じて前記持続的なユーザレベルスレッドの状態を取得する

請求項 3 2 または 3 3 に記載のシステム。

【請求項 3 5】

前記 OS 分離シーケンサは、前記持続的なユーザレベルスレッドユーザレベルスレッドを生成したスレッドである前記 OS によりスケジューリングされるスレッドに対するコンテキストスイッチ動作と独立して、前記持続的なユーザレベルスレッドの動作を継続させる

10

請求項 2 6 から 3 4 のいずれかに記載のシステム。

【請求項 3 6】

前記命令セットエクステンションは、前記 OS 可視シーケンサおよび前記 OS 分離シーケンサを識別するオペランドを持つ命令を含む

請求項 2 6 から 3 5 のいずれかに記載のシステム。

【請求項 3 7】

コンピュータに、

OS によりスケジューリングされ OS 可視シーケンサで実行中のスレッドに基づき、命令セットエクステンションを用いて、前記 OS によりスケジューリングされるスレッドに対するコンテキストスイッチ動作とは独立して OS 分離シーケンサで実行されるユーザレベルスレッドである持続的なユーザレベルスレッドを生成する手順

20

を実行させ、

前記 OS は、前記持続的なユーザレベルスレッドの命令に対してスケジューリングを行わず、

前記持続的なユーザレベルスレッドは、前記 OS によりスケジューリングされるスレッドがコンテキストスイッチされた時でも前記 OS 分離シーケンサで動作を継続し、

前記持続的なユーザレベルスレッドを生成する手順は、前記 OS 可視シーケンサで実行される別のユーザレベルスレッドから前記持続的なユーザレベルスレッドを生成する手順を有する

30

プログラム。

【請求項 3 8】

前記コンピュータに、

前記持続的なユーザレベルスレッドを前記 OS 分離シーケンサで実行する手順をさらに実行させる請求項 3 7 に記載のプログラム。

【請求項 3 9】

前記コンピュータに、

前記 OS によりスケジューリングされるスレッドおよび前記持続的なユーザレベルスレッドに仮想アドレススペースを共有させる手順

をさらに実行させる請求項 3 7 または 3 8 に記載のプログラム。

40

【請求項 4 0】

前記コンピュータに、

前記持続的なユーザレベルスレッドの実行中に OS による処理を必要とするイベントが発生した場合に、前記持続的なユーザレベルスレッドの実行を一時中断させる手順をさらに実行させる請求項 3 7 から 3 9 のいずれかに記載のプログラム。

【請求項 4 1】

前記コンピュータに、

別の OS 可視シーケンサで実行されるサービススレッドに、前記持続的なユーザレベルスレッドに対して、OS サービスを提供させる手順

をさらに実行させる請求項 3 7 から 4 0 のいずれかに記載のプログラム。

50

**【請求項 4 2】**

前記 OS サービスを提供させる手順は、

前記別の OS 可視シーケンサに対して、制御移行命令を伝え、前記制御移行命令を特権レベルで実行させる手順  
を有する請求項 4 1 に記載のプログラム。

**【請求項 4 3】**

前記 OS サービスを提供させる手順は、

前記サービススレッドに、前記持続的なユーザレベルスレッドと共有している仮想アドレス空間を通じて前記持続的なユーザレベルスレッドの状態を取得させる手順  
を有する請求項 4 1 または 4 2 に記載のプログラム。

10

**【請求項 4 4】**

前記 OS 分離シーケンサに、前記持続的なユーザレベルスレッドユーザレベルスレッドを生成したスレッドである前記 OS によりスケジューリングされるスレッドに対するコンテキストスイッチ動作と独立して、前記持続的なユーザレベルスレッドの動作を継続させる  
請求項 3 7 から 4 3 のいずれかに記載のプログラム。

**【請求項 4 5】**

前記命令セットエクステンションは、前記 OS 可視シーケンサおよび前記 OS 分離シーケンサを識別するオペランドを持つ命令を含む  
請求項 3 7 から 4 4 のいずれかに記載のプログラム。

20

**【発明の詳細な説明】****【背景技術】****【0001】**

マルチシーケンサコンピュータシステムまたはマルチシーケンサコンピュータプラットフォームにおいて、技術上ならびにコスト上の制約から、特定用途向けのコンピューティング資源、プロセッサまたはプロセッシングユニットの編成が非対称になってしまうことがある。本出願の以下の部分においては、プロセッサまたはプロセッシングユニットも「シーケンサ」または「プロセッシングコア」と呼ぶ。一般的に言って、シーケンサが数多く実装されたコンピュータプラットフォームでは、オペレーティングシステム(OS)がうまく機能しないことがある。特に、複数のシーケンサが非対称に編成された状態で効率よく処理が行えないことがある。

30

**【0002】**

コンピュータプラットフォームは、実装されるシーケンサの種類に関して、その編成が非対称になることがある。例えば、OS に対して「可視」で、OS カーネルが管理するシーケンサがある。本明細書では、このようなシーケンサを「OS 可視シーケンサ」と呼ぶ。ここで、OS カーネルは、OS 可視シーケンサの特権状態を管理することができ、OS 可視シーケンサによってスレッドが実行されている間に、割り込みおよび/または例外を処理する例外ハンドラを提供することができるとしてもよい。このような「OS 可視シーケンサ」に対し、OS に対して「不可視」なシーケンサがある。本明細書では、このようなシーケンサを「OS 分離シーケンサ」と呼ぶ。OS 分離シーケンサは、OS コードを実行せず、シーケンサマネージャに管理されるとしてもよい。

40

**【0003】**

最近になって、マルチシーケンサ・MIMD(複数命令/複数データ)方式ISA(命令セットアーキテクチャ)エクステンションが提案された。これは、1つのOSスレッドコンテキストにおいて複数のユーザレベルスレッドを同時に実行すべく、OS 分離シーケンサをRing 3のユーザレベルアーキテクチャリソースとして扱うことを可能にする命令セットを定義するものである。このため、OS 分離シーケンサをユーザレベルアプリケーションに利用できるようになる。ユーザレベル・マルチシーケンサ・MIMD・ISAエクステンションを用いる場合、OS 可視シーケンサで実行中のアプリケーションスレッドに、OS から分離されたシーケンサを使用することができるようになる。マルチシー

50

ケンサ・M I M D・I S Aエクステンションによれば、O Sに対して可視または不可視なシーケンサに処理を任せることにより、アプリケーションは、O Sとは無関係に、多くのシーケンサに対して対応できるようになる。O Sはこの間、O Sに対して可視なシーケンサ上で動作を継続してもよい。このようなユーザレベル・M I M D・I S Aエクステンションによって、ユーザレベルスレッドをO S分離シーケンサ上で実行できるようになり、該ユーザレベルスレッドは、それらを生成し且つO S可視シーケンサで実行されるO Sスレッドと同時に実行されるとしてもよい。

【図面の簡単な説明】

【0004】

以下では、添付の図面を用いて本発明の実施形態を詳細に説明し、本発明の内容をより明らかにする。

【0005】

【図1】本発明の一実施形態に係るマルチシーケンサシステムを示すブロック図である。

【0006】

【図2】本発明の一実施形態に係るマルチシーケンサシステムの一部であるハードウェアを示す簡略化論理図である。

【0007】

【図3】本発明の一実施形態に係る、持続的なユーザレベルスレッドのサービスを提供することができるマルチシーケンサシステムを示す簡略図である。

【0008】

【図4】本発明の一実施形態に係る、持続的なユーザレベルスレッドのサービスを提供することができるマルチシーケンサシステムを示す概念図である。

【0009】

【図5】本発明の一実施形態に係る、マルチシーケンサシステムの命令セットアーキテクチャを示す図である。

【0010】

【図6】本発明の一実施形態に係る制御移行命令の命令フォーマットの例を示す簡略図である。

【0011】

【図7】本発明の一実施形態に係る、監視命令の命令フォーマットの例を示す簡略図である。

【0012】

【図8】本発明の一実施形態に係る、持続的なユーザレベルスレッドの実行について説明するための簡略化フローチャートである。

【0013】

【図9】本発明の一実施形態に係る、持続的なユーザレベルスレッドの実行について説明するための簡略化フローチャートである。

【0014】

説明を簡潔且つ明瞭なものとできるように、図中に示す構成要素は実寸を反映したものではない。例えば、図示した構成要素の中には分かりやすいようほかの構成要素との相対的なサイズが誇張されているものもある。

【発明を実施するための最良の形態】

【0015】

以下において、具体的且つ詳細な事例を数多く挙げて本発明の実施形態を細部にいたるまで説明する。しかし、以下で説明する内容以外の実施形態を本発明が含むことは、当業者には明らかである。また、公知の方法および処理は、本発明の実施形態の説明をあいまいにしない程度に、詳細に説明する。

【0016】

以下の説明では一部、コンピュータメモリ内でデータビットまたはバイナリデジタル信号に対して行われる処理のアルゴリズムやそのような処理を表す符号を用いている。デー

10

20

30

40

50

タ処理の分野においては、当業者同士間で作業の内容を伝える場合に、このようなアルゴリズムによる説明を用いる。

【 0 0 1 7 】

本明細書内および一般的にアルゴリズムとは、所望の結果を得るために一貫した順序で実施される一連の動作または処理と考えられている。これには、物理量を物理的に操作することも含まれる。必ずしもそうではないが多くの場合、そのような物理量とは電気または磁気信号で、記憶、移動、合成、比較といった操作を行うことができる。こういった信号は、利便性を考慮し一般的な用語を使うべく、ビット、値、成分、符号、文字、項、数字などと呼ばれる。しかし、こういった用語はすべて適切な物理量に対応するものであって、対応する物理量に用いられる便利な名称にすぎない。

10

【 0 0 1 8 】

本明細書において、特に明示されていない限り、「処理」「演算」「計算」「決定」等の用語を用いている部分は、コンピュータや演算処理システム、または同様の演算電子デバイスによる処理および/または動作に関する。こういったコンピュータシステムは、システム内のレジスタおよび/またはメモリ内の物理量（例えば電子量）として表されたデータを操作および/または変換して、同システム内のメモリ、レジスタまたは同様の情報記憶/送信/表示装置内に同様に物理量として表される別のデータを得る。

【 0 0 1 9 】

本発明の実施形態は例えば、命令または命令セットを格納した機械可読媒体または機械可読装置を用いて実施されるときにもよい。このような命令または命令セットが機械によって実行されると当該機械は、本発明の実施形態に係る方法および/または動作を行う。このような機械の例を挙げると、適切なプロセッシングプラットフォーム、コンピューティングプラットフォーム、コンピューティングデバイス、プロセッシングデバイス、コンピューティングシステム、プロセッシングシステム、コンピュータまたはプロセッサなどがある。また、このような機械は、ハードウェアおよび/またはソフトウェアを適切に組み合わせることによって実施するときにもよい。上述の機械可読媒体または機械可読装置は例えば、適切な種類のメモリユニット、メモリデバイス、メモリ装置、メモリ媒体、ストレージデバイス、ストレージ装置、ストレージ媒体および/またはストレージユニットであり、例として、メモリ、取り外し可能/不可能なメディア、消去可能/消去不可能なメディア、書き込み可能/書き換え可能なメディア、デジタル/アナログメディア、ハードディスク、フロッピーディスク、CD-ROM (Compact Disk Read Only Memory)、CD-R (Compact Disk Recordable)、CD-RW (Compact Disk Rewritable)、光ディスク、磁気メディア、さまざまな種類のDVD (Digital Versatile Disk)、テープまたはカセットなどが挙げられる。命令は適切な種類のコードであってもよく、例として、ソースコード、ターゲットコード、コンパイラ型コード、インタプリタ型コード、実行可能コード、スタティックコードまたはダイナミックコードなどが挙げられる。このような命令を実行するには、適切な高レベル、低レベル、オブジェクト指向、ビジュアル、コンパイラ型および/またはインタプリタ型のプログラミング言語を用いるときにもよい。このようなプログラミング言語の例として、C、C++、Java、BASIC、Pascal、Fortran、Cobol、アセンブリ言語、マシンコードなどが挙げられる。

20

30

40

【 0 0 2 0 】

本発明の実施形態は、本明細書に記載する動作を行う装置を含むときにもよい。このような装置は、所望の目的を達成するように構成されてもよいし、コンピュータ内に格納されたコンピュータプログラムが選択的にアクティブ状態にしたり再構成したりする汎用コンピュータであってもよい。このようなコンピュータプログラムは、コンピュータ可読ストレージ媒体に格納されるときにもよい。このようなストレージ媒体としてディスクが挙げられ、例えばフロッピーディスク、光ディスク、CD-ROM、光磁気ディスク、ROM (Read Only Memory)、RAM (Random Access Me

50



mory)、EPROM(Electrically Programmable ROM)、EEPROM(Electrically Erasable and Programmable ROM)、磁気/光カードが含まれる。または、ストレージ媒体の例として、その他の種類の、命令を電子状態で格納することに適しておりコンピュータシステムバスに接続が可能な媒体が挙げられる。

#### 【0021】

本明細書で言及する処理およびディスプレイは必ずしも、特定のコンピュータや装置に関するものではない。本明細書の教示に基づいたプログラムとともに用いるのはさまざまな汎用システムであってよいし、所望の方法を実施するためより特化した装置を構築する方がよい場合もあると考えられる。これらさまざまなシステムの望ましい構造は、以下の説明で明らかにする。また、本発明の実施形態の説明は特定のプログラミング言語に基づくものではない。本明細書で説明する本発明の教示は、さまざまなプログラミング言語を用いて実施することができるとしてもよい。

10

#### 【0022】

以下の部分において、図や表、フローチャート、モデルなどさまざまな手段に基づき、本出願で提案される本発明の実施形態を効率よく説明する。当業者には明らかであるが、以下の説明で挙げる実施形態は単なる例にすぎず、本発明を限定するものではない。

#### 【0023】

図1は、本発明の一実施形態に係るマルチシーケンサシステム100を示すブロック図である。

20

#### 【0024】

図1に示すように、マルチシーケンサシステム100は、メモリ102およびマルチシーケンサハードウェア104を備えるとしてもよい。メモリ102は例えば、ユーザレベルプログラム106、スケジューラ108、API(Application Program Interface)110およびオペレーティングシステム(OS)112を有するとしてもよい。マルチシーケンサハードウェア104は例えば、シーケンサマネージャ114および複数のシーケンサ(例えば、シーケンサ116、118、120および122、図1ではそれぞれSID0、SID1、SID2およびSID3と表す)を有するとしてもよい。図1に図示したシーケンサは4つであるが、マルチシーケンサハードウェア104が有するシーケンサの数は4つに限定されない。

30

#### 【0025】

シーケンサマネージャ114は、ドライバ、OS(例えばOS112)のエクステンション、これ以外のハードウェア、またはハードウェアとソフトウェアの組み合わせとして実施されるとしてもよい。マルチシーケンサハードウェア104は、シーケンサ116、118、120および/または122の種類に関して、対称であってもよいし非対称であってもよい。シーケンサ116、118、120および/または122のうち幾つかは、自分の実行リソースセットを有する物理プロセッサであってもよい。もしくは、シーケンサ116、118、120および/または122のうち幾つかが、ほかの論理プロセッサと物理リソースを共有する論理プロセッサ(例えばハイパースレッド)であってもよい。シーケンサ116、118、120および122を実現するハードウェアコアは、1つであ

40

#### 【0026】

マルチシーケンサシステム100の例を挙げると、デスクトップパソコン、ワークステーション、サーバーコンピュータ、ラップトップコンピュータ、ノート型コンピュータ、携帯型コンピュータ、携帯情報端末(PDA)、携帯電話、ゲームコンソールなどがあるが、これらに限定されない。

#### 【0027】

メモリ102の例を挙げると、半導体デバイス、光学デバイス、磁気デバイスのいずれか、もしくはこれらの組み合わせが考えられるが、これらに限定されない。半導体デバイスとしては、SDRAM(Synchronous Dynamic Random A

50

ccess Memory)デバイス、RDRAM(RAMBUS DRAM)デバイス、DDR(Double Data Rate)メモリデバイス、SRAM(Static RAM)デバイス、フラッシュメモリ(FM)デバイス、EEPROM(Electrically Erasable Programmable Read Only Memory)デバイス、不揮発性(NV)RAMデバイス、取り外し可能なUSB(Universal Serial Bus)メモリデバイスといったものが例として挙げられる。光学デバイスには、CD-ROM(Compact Disk ROM)などがある。磁気デバイスには、ハードディスク、フロッピーディスク、磁気テープなどがある。メモリ102は、システム100内に固定されるとしてもよいし、システム100から取り外し可能な構成としてもよい。

10

#### 【0028】

本発明の一実施形態によれば、ユーザレベルプログラム106がAPI110を介してスレッドライブラリにアクセスするとしてもよい。API110は、ユーザレベルプログラム106に対して、スレッドの作成、制御および同期を行うとしてもよい。スケジューラ108は、マルチシーケンサハードウェア104上での実行に向けて、スレッドの命令をスケジューリングするとしてもよい。

#### 【0029】

マルチシーケンサハードウェア104の例を挙げると、複数のシングルスレッド型/マルチスレッド型中央演算処理装置(CPU)、デジタル信号プロセッサ(DSP)、縮小命令セットコンピュータ(RISC)、復号命令セットコンピュータ(CISC)などがあるが、これらに限定されない。また、マルチシーケンサ104は特定用途向け集積回路(ASIC)の一部であってもよいし、特定用途・一般顧客向けLSI(ASSP)の一部であってもよい。

20

#### 【0030】

本発明の一実施形態によると、マルチシーケンサハードウェア104は、例えば論理プロセッサであるシーケンサ116、118、120および122(SID0-SID3)を実装したシングルコアプロセッサであってもよい。シングルコアプロセッサ104は、例えば並列マルチスレッディング(CMT)、同時マルチスレッディング(SMT)および/またはSwitch-on-Eventマルチスレッディング(SOEMT)に対応可能としてもよい。一実施形態によれば、シーケンサSID0~SID3はそれぞれ独自の次命令ポインタ論理を有し、シングルコアプロセッサ104はスレッドの命令をすべて実行するとしてもよい。論理プロセッサSID1~SID3はそれぞれ、アーキテクチャ状態の独自のバージョンを保持するが、シングルコアプロセッサ104の実行リソースは、複数の並列スレッド、同時スレッドおよび/またはSwitch-On-Eventスレッドの間で共有するとしてもよい。

30

#### 【0031】

上述した一実施形態に係るマルチシーケンサハードウェア104はシーケンサ毎に1つのスレッドという構成であるとしてもよいが、本明細書の開示はシングルスレッドプロセッサに限定されない。本明細書で説明する技術内容は、CMP(チップマルチプロセッシング)システムまたはSMP(同時マルチスレッディングプロセッサ)システムにも同様に適用することができる。また、CMPプロセッサの各コアはSMPプロセッサまたはSwitch-On-Eventマルチプロセッサ(SoeMP)である、CMPプロセッサとSMPプロセッサを備えるハイブリッドシステムにも応用できる。例えば、シングルチップハードウェアパッケージ104内のマルチスレッドプロセッシングコアを備えるシステムに、本明細書で開示する技術内容を利用するとしてもよい。

40

#### 【0032】

上述したが、シーケンサSID0~SID3は、例えば処理速度、処理能力や電力消費量といった演算能力に影響する要因に関して、必ずしも均一である必要はなく非対称であってもよい。例えば、シーケンサSID0は、所定の命令セットアーキテクチャ(例えば、32ビットプロセッサに対応したIA-32命令セットアーキテクチャ)において命令

50

の大半を処理するように構成されている「主」シーケンサとしてもよい。そして、シーケンサ S I D 1 は、これらの命令のうち選択された命令サブセットを処理する「副」シーケンサであるとしてもよい。本発明の別の実施形態によれば、主プロセッサは副プロセッサよりも早い速度で命令を処理するものと定義してもよい。また、OS に対して可視のシーケンサと不可視のシーケンサがあるとしてもよい。例えば、シーケンサ S I D 0 が OS 1 1 2 に対して可視で、シーケンサ S I D 1 ~ S I D 3 は OS 1 1 2 から分離されているとしてもよい。しかし、主シーケンサが OS に対して可視で、副シーケンサが OS から分離されているというわけではない。OS は、分離状態にあるシーケンサ（このようなシーケンサを本明細書では OS 分離シーケンサと呼ぶ）上で実行される命令に対してスケジューリングを行わないということである。

10

#### 【0033】

本発明の実施形態によると、マルチシーケンサシステム 100 は持続的なユーザレベルスレッド (P U L T) のサービスを提供するとしてもよい。P U L T については、以下で図 2 から図 8 を参照しつつ詳述する。本出願において、通常のユーザレベルスレッド（以下では単にユーザレベルスレッドと呼ぶ）は、ユーザレベルまたは R i n g 3 レベルで実行される非特権命令だけから成る一連の命令であるとしてもよい。ユーザレベルスレッドは通常、OS スレッドに「水面下で」サポートされるとしてもよい。このような OS スレッドは、ネイティブ OS スレッドと呼んでもよいし、単にネイティブスレッドとしてもよい。この OS スレッドのコンテキストにおいて、少なくとも 1 つのユーザレベルスレッドが OS 可視シーケンサまたは OS 分離シーケンサの一方で実行されるとしてもよい。OS が OS スレッドをコンテキストスイッチすると、この OS スレッドに属するすべてのシーケンサのコンテキストが保存 / 復元される。従って、これらのシーケンサで動作中のユーザレベルスレッドは中断 / 再開される。本発明の一実施形態によれば、持続的なユーザレベルスレッド (P U L T) は、OS 分離シーケンサ上で動作するユーザレベルスレッドである。P U L T が実行される環境は例えば、該 P U L T を生成する別のユーザレベルスレッドの仮想アドレススペース (V A S) である。P U L T の実行は、当該 P U L T を生成したユーザレベルスレッドの実行とは無関係に、従って同期せずに行われるとしてもよい。ユーザレベルスレッドが属する OS スレッドがコンテキストスイッチされた時でも、P U L T は OS 分離シーケンサ上で動作を継続するとしてよい。本明細書では、このような状態、つまり P U L T の生成元であるスレッドに関する OS スレッドのコンテキストスイッチ動作とは関係なく、ユーザレベルスレッドが動作を継続する特性を「持続的」と呼ぶ。

20

30

#### 【0034】

図 2 は、本発明の一実施形態に係るマルチシーケンサシステムの一部であるハードウェア 200 を示す簡略化論理図である。

#### 【0035】

ハードウェア 200 は、図 1 に示すマルチシーケンサシステム 100 の一部であるマルチシーケンサハードウェア 104 に対応するとしてもよい。マルチシーケンサハードウェア 104 は、図 1 に示すように、シーケンサマネージャ 114 および、例えば複数のシーケンサ 116、118、120 および 122 を有するとしてもよい。本発明の一実施形態によれば、シーケンサマネージャ 114 は、シーケンサ 116、118、120 および 122 がユーザレベルプログラム 106 に対して均一且つ対称であるように見えるように、シーケンサ 116、118、120 および 122 を仮想化できるとしてもよい。つまり、図 1 のシーケンサマネージャ 114 に対応するシーケンサマネージャ 201 は、シーケンサ 211、212、213 および 214 の非対称性を隠して、図 2 に同じサイズの正方形で示すように、アセンブリ言語プログラマが論理的に見た場合、シーケンサ 211、212、213 および 214 が均一且つ対称に見えるようにする。シーケンサ 211、212、213 および 214 は図 1 に示す論理プロセッサ S I D 0、S I D 1、S I D 2 および S I D 3 に対応するとしてもよい。

40

#### 【0036】

50

図 3 は、本発明の一実施形態に係る、持続的なユーザレベルスレッドのサービスを提供できるマルチシーケンサシステム 300 を示した簡略図である。

【0037】

本発明の一実施形態によれば、アプリケーション 301（例えば図 1 に示すユーザレベルプログラム 106）の実行は OS 302 が管理するとしてもよい。OS 302 は、各々がユーザレベルスレッドを含んでいる可能性がある 1 以上のネイティブスレッドを生成およびスケジューリングして、アプリケーション 301 を実行するとしてもよい。このようなネイティブスレッドの例を挙げると、例えば OS 302 に対して可視である一連のシーケンサ（例えばシーケンサ 311 および 312）のうちの 1 つで実行される図 3 のネイティブスレッド 303 がある。OS 302 は、OS カーネルに基づきシーケンサ 311 および 312 を管理するとしてもよい。OS 302 はさらに、シーケンサ 311 および 312 の特権状態を制御し、シーケンサ 311 および 312 で生じる割り込みおよび例外を処理するとしてもよい。

10

【0038】

本発明の一実施形態によれば、シーケンサマネージャ 310 は、OS に対して可視であるシーケンサ 311 および 312 ならびに、別の一連のシーケンサ、例えば OS 302 から分離されており OS 302 のカーネルが管理していないシーケンサ 313、314、315、316、317 および 318 へのアクセスを制御できるとしてもよい。シーケンサ 313、314、315、316、317 および 318 は、OS 分離シーケンサまたは「OS 不可視」シーケンサである。

20

【0039】

本発明の一実施形態によれば、OS がスケジューリングしたスレッド（例えば、ユーザレベルスレッドを含む可能性があるネイティブスレッド）は、ハードウェア、ソフトウェアもしくは両者の組み合わせによって構成されるインターフェースを介して、シーケンサマネージャ 310 と通信を行うとしてもよい。OS 302 に不可視なシーケンサ（例えばシーケンサ 313、314、315、316、317 および 318）を使用したい場合、OS がスケジューリングしたスレッド（例えばネイティブスレッド 303）が、OS 分離シーケンサのうちの 1 つ（例えばシーケンサ 313）で動作するユーザレベルスレッド（例えばスレッド 304）を生成するように、シーケンサマネージャ 310 に要求を送るとしてもよい。本発明の一実施形態によれば、ユーザレベルスレッド 304 は、ネイティブスレッド 303 の OS コンテキストスイッチ動作とは無関係なシーケンサ 313 で実行されるので、持続的なユーザレベルスレッド（PULT）であるとしてもよい。

30

【0040】

以下においては、ネイティブスレッド 303 を例として、持続的なユーザレベルスレッドの作成および実行について説明する。以下の説明がほかのスレッド、例えばネイティブスレッドに属するユーザレベルスレッドにも当てはまることは、当業者には明らかである。

【0041】

本発明の一実施形態によれば、OS 分離シーケンサ 313 は、OS 可視シーケンサ 311 で実行されるユーザレベルスレッド 303 と共有する仮想アドレススペース（VAS）320 で、PULT 304 を実行するとしてもよい。VAS 320 は、OS 302 によってアプリケーション 301 を実行するために生成されるとしてもよい。本発明の一実施形態によれば、VAS 320 が共有されているので、シーケンサマネージャ 310 は、OS 可視シーケンサ 311 で実行されるユーザレベルスレッド 303 の実行環境、例えば、仮想アドレスマップを取得して、この実行環境を OS 分離シーケンサ 313 で実行される PULT 304 に用いることができるとしてもよい。本発明の一実施形態によれば、OS 分離シーケンサ 313 は、仮想アドレススペースの変化を自動的に記録できるスヌープ方式の TLB（変換索引バッファ）を有するとしてもよい。ここで、TLB とはプロセッサ内にある小型のキャッシュで、高速で検索できるように仮想アドレスから物理アドレスへの変換を記録するとしてもよい。プロセッサは、このような TLB を用いることによって、

40

50

仮想アドレスを物理アドレスへと変換する場合の不必要なページウォークを避けることができる。

【0042】

図4は、本発明の一実施形態に係る、持続的なユーザレベルスレッドのサービスを提供できるマルチシーケンサシステム400を示す概念図である。

【0043】

本発明の一実施形態に係るマルチシーケンサシステム400は、複数のOS分離シーケンサ（例えばシーケンサ411、412、413および414）およびOS可視シーケンサ（例えばシーケンサ405および406）を管理するシーケンサマネージャ410を備えるとしてもよい。OS402は、ユーザレベルアプリケーション（例えばアプリケーション401Aおよび401B）の実行を管理するとしてもよい。OS402は、1以上のスレッド（例えば、OS可視シーケンサ405および406で動作するユーザレベルスレッド403および404）ならびにネイティブスレッドを生成することによって、アプリケーション401Aおよび401Bを管理する。

【0044】

シーケンサマネージャ410は、OS402から分離されておりOS402が管理していないシーケンサ411、412、413および414へのアクセスに基づき、アプリケーション401Aおよび401Bに対して実行リソースを提供するとしてもよい。なお、この場合のアクセスは、OS可視シーケンサ405および406で実行されているスレッド403および404またはOSがスケジューリングするネイティブスレッドからの要求に基づいて、行われるとしてもよい。この結果、アプリケーション401Aおよび401Bは、ユーザレベルスレッド403および/または404、もしくはネイティブスレッドに基づき、OS分離シーケンサ411、412、413および414にアクセスできるようになるとしてもよい。

【0045】

本発明の一実施形態によると、シーケンサマネージャ410は、ユーザレベルスレッド403および404によってアプリケーション410Aおよび401B用に生成された持続的なユーザレベルスレッド実行のために、さまざまなタイムスロットをシーケンサ411、412、413および414に割り当てるとしてもよい。例えば、図4に示した参照符号 $t_1$ 、 $t_2$ 、 $t_3$ 、 $t_4$ 、 $t_5$ および $t_6$ はそれぞれ、PUL Tである $T_1$ 、 $T_2$ 、 $T_3$ 、 $T_4$ 、 $T_5$ および $T_6$ の実行用に割り当てられたタイムスロットを示すものであってもよい。

【0046】

本発明の一実施形態によると、シーケンサマネージャ410は、OS分離シーケンサ（例えば、シーケンサ411、412、413および414）用に、割り込み記述子テーブル（例えば、テーブル421、422、423および424）を設定するとしてもよい。テーブル421を例に挙げて説明すると、割り込み記述子テーブルはシーケンサ411で発生するイベントを処理するイベントハンドラ431を含むとしてもよい。OS分離シーケンサ（例えば、シーケンサ411）上でPUL T（例えばPUL T  $T_3$ ）を実行している間（例えば、 $t_3$ ）、OS402による対処が必要となるイベントや状況（例えばページフォルトまたはシステムコール）が発生する可能性がある。この場合、イベントハンドラ431はPUL T  $T_3$ の実行を一時中断し、当該イベントが発生したときのPUL T  $T_3$ の実行状態を保存するとしてもよい。

【0047】

本発明の別の実施形態によると、ページフォルトやシステムコールが発生した場合、OS不可視シーケンサ（例えばシーケンサ411）は、OS可視シーケンサ（例えば、PUL T  $T_3$ ）を起動したシーケンサ405の実行に直接割り込むとしてもよい。この場合、続いてシーケンサ411は、トリガをシーケンサマネージャ410に送って割り込み処理を要求する。本発明のさらに別の実施形態によると、OS分離シーケンサ411はRing 0サービスチャネル、もしくはページフォルトまたはシステムコールを処理するチ

10

20

30

40

50

チャンネルをプログラムするとしてもよい。ここでチャンネルとは、アーキテクチャ状態（例えば、トリガがアーキテクチャが定義された一連のプロセッサ状態であり、レスポンスがサービスルーチンであるトリガ・レスポンス・マッピングを含む）を保持するレジスタであってもよい。またチャンネルは、例えばユーザレベル命令によってプログラムされるとしてもよい。ページフォルトまたはシステムコールが発生した場合、OS分離シーケンサ411は、そのアドレスがチャンネル内で特定されているハンドラを起動するとしてもよい。

#### 【0048】

PULT T<sub>3</sub>の中断は、同じアプリケーション401Aを実行するための新しいユーザレベルスレッド（例えば、スレッド409）がOS可視シーケンサで実行されるようスケジューリングされるまで、継続するとしてもよい。スレッド409（サービススレッドとも呼ぶ）は、一時中断されているPULTがあるかどうか、および例えば代理実行によって提供されるOSサービスを待っているPULTがあるかどうか確認または検証するよう、シーケンサマネージャ410に対して要求するとしてもよい。代理実行は、シーケンサマネージャ410からPULT T<sub>3</sub>の状態を取得することによって、PULT T<sub>3</sub>を模倣するスレッド409に基づいて行われるとしてもよい。PULT T<sub>3</sub>の実行状態は、アドレススペース（例えば、共有仮想アドレススペース（VAS）、図3を参照のこと）を共有しているサービススレッド409によって再開されるとしてもよい。スレッド409は続いて、PULT T<sub>3</sub>を一時停止させた原因である実行条件に対して処理を行うとしてもよい。スレッド409はこの処理が終了すると、実行後状態を保存し、当該状態をシーケンサマネージャ410に戻すとしてもよい。シーケンサマネージャ410は続いて、当該状態をPULT T<sub>3</sub>に渡して、PULT T<sub>3</sub>の実行を再開するとしてもよい。

#### 【0049】

上述した代理実行により、アプリケーションプログラマに対してマルチシーケンサシステムが対称性を有すると見せかけることができ、マルチシーケンサシステム（例えばシステム100）が非対称性を有するものとして構築されたことを隠すとしてもよい。

#### 【0050】

図5は、本発明の一実施形態に係るマルチシーケンサシステムの命令セットアーキテクチャ（ISA）500を示す図である。

#### 【0051】

ISAは、アセンブリ言語プログラマ、バイナリトランスレータ、アセンブラなどから見た場合のシステムの論理図を定義するものである。ISA 500は、論理ストレージ502および命令セット504を備えるとしてもよい。論理ストレージ502は、マルチシーケンサシステム（例えば、図1に示したマルチシーケンサシステム100）用の可視メモリ階層、アドレス割当方法、レジスタセットなどを定義するとしてもよい。一方命令セット504は、命令およびマルチシーケンサシステム100がサポートする命令のフォーマットを定義するとしてもよい。

#### 【0052】

命令セット504は、IA-32命令セットとして公知である命令セットを含むとしてもよい。本発明の実施形態に係る命令セットは、これらに限定されないが、IA-32命令セットのエクステンションやそれ以外の命令セットであってもよい。本発明の一実施形態によると、命令セット504は例えば、SXF R制御・移行命令およびそのエクステンションを含むE-SXF R命令、ならびにSEMONITOR監視命令およびそのエクステンションを含むE-SEMONITOR命令を有するとしてもよい。同じOSネイティブスレッドに属する2つのシーケンサの間で行われ得るSXF RおよびSEMONITORとは違い、E-SXF RおよびE-SEMONITORは、異なる特権レベルで動作する2つの別々のOSネイティブスレッドにそれぞれ属する2つのシーケンサの間で実行することができる。例えば、一方のスレッドはOSコンテキストスイッチされるユーザレベルアプリケーションであり、他方のスレッドは、デバイスドライバのように、特権レベルで持続的に実行されるPULTであり、OSコンテキストスイッチには影響されないとし

てもよい。

【 0 0 5 3 】

本発明の一実施形態によると、E - S X F R命令を用いて、第1シーケンサから第2シーケンサに信号を送り、E - S E M O N I T O R命令を用いて、第1シーケンサから送信される信号を監視するよう、第2シーケンサを構成するとしてもよい。また、E - S X F R制御移行命令およびE - M O N I T O R監視命令はシーケンサを認識しており、シーケンサを認識したほかの複合命令を構築するために用いられるとしてもよい。

【 0 0 5 4 】

本発明の一実施形態によると、E - S X F R命令は図6に例示する命令フォーマットを持つとしてもよい。図6に示すように、E - S X F R命令はOPコード6 0 2、オペラント6 0 4、6 0 6、6 0 8、6 1 0および6 1 2を含むとしてもよい。本発明の一実施形態によると、シーケンサマネージャは、一時中断された持続的なユーザレベルスレッドを処理するための実行環境を作り出すために、シーケンサの特権状態などのオペラントを有するE - S X F R命令を提供するとしてもよい。

【 0 0 5 5 】

一実施形態によると、オペラント6 0 4は、E - S X F R命令信号の送信先であるデスティネーション/ターゲットシーケンサのシーケンサID ( S I D ) に対応するとしてもよい。オペラント6 0 6は、シナリオまたは制御メッセージを含むとしてもよい。このようなシナリオまたは制御メッセージは、条件または予測されるイベントを表す、そのアーキテクチャが定義された識別コードであってもよい。シナリオは非同期制御移行に影響を与えとしてもよい。この詳細は表1を参照して後述する。

【 0 0 5 6 】

一実施形態によると、オペラント6 0 8は、E - S X F R命令を実行するシーケンサで命令を実行するかどうか決定する条件となるパラメータを含むとしてもよい。図6に条件パラメータと示すそのようなパラメータの例を挙げると、「W A I T」パラメータまたは「N O - W A I T」パラメータがある。例えば、E - S X F R命令が代理実行シナリオで用いられる場合、W A I T条件パラメータに基づき、E - S X F R命令を実行するシーケンサでの命令の実行を取りやめ、別のシーケンサでの代理実行の完了を保留にする。一方、N O - W A I T条件パラメータであれば、E - S X F R命令を実行するシーケンサでの命令の実行が別の命令を実行するシーケンサでの代理実行と並行して継続するとしてもよい。

【 0 0 5 7 】

一実施形態によると、オペラント6 1 0は、シナリオ固有ペイロードまたはデータメッセージを含むとしてもよい。例えばF O R Kシナリオの場合、該ペイロードはオペラント6 0 4が示すシーケンサでの実行が始まる箇所である命令ポインタを含むとしてもよい。別の実施形態によると、ペイロードオペラント6 1 0は命令ポインタ、スタックポインタ、一連の制御レジスタなどを含むとしてもよい。ペイロードオペラント6 1 0に含まれるアドレスは、例えばリテラル、レジスタ間接および/またはベース/オフセットアドレッシングなど、さまざまなアドレス指定モードで指定するとしてもよい。

【 0 0 5 8 】

一実施形態によると、オペラント6 1 2は、オペラント6 0 4に含まれるS I Dに関するルーティング方法またはルーティング関数を特定するとしてもよい。ルーティング関数は、E - S X F R命令を実行した結果生成される信号が、ブロードキャスト信号、ユニキャスト信号またはマルチキャスト信号のうちどの信号として送信されるかを制御する。ルーティング関数はさらに、例えば、信号を配信するためのルーティングにおいて基本構成要素であるシーケンサ間のインターコネクトをアシストするために用いられるトポロジ固有情報のような情報を符号化するとしてもよい。

【 0 0 5 9 】

一実施形態によると、E - S E M O N I T O R指示は、図7に例示するような命令フォーマットを有するとしてもよい。図7に示すように、E - S E M O N I T O R命令はOP

10

20

30

40

50

コード702およびオペランド704、706および708を含むとしてもよい。オペランド704、706および708は、持続的なユーザレベルスレッドの実行環境の形成を手助けするべくシーケンサの特権状態に関する情報を含むとしてもよい。例えば、オペランド704はシナリオIDを含み、シナリオを特定するとしてもよい。オペランド706は、シーケンサID(SID)および命令ポインタ(EIP)に関する情報を含むとしてもよい。説明の便宜上、オペランド706に含まれる情報を本明細書では、図7に示すように、「SIDEIP」と呼ぶ。

#### 【0060】

E-SEMONITOR命令は、オペランド704で指定されたシナリオとオペランド706で指定されたSIDEIPをマッピングする。シナリオのSIDEIPへのマッピングを本明細書では、「サービスチャネル」のマッピングと呼ぶとしてもよい。オペランド708によってプログラマは所定のサービスチャネルをどのように処理されるか特定または制御するための制御パラメータを入力できるようになるとしてもよい。例えば、プログラマはE-SEMONITOR命令に基づき、シーケンサが監視するサービスチャネルをプログラムするとしてもよい。

#### 【0061】

上記の説明から、E-SXFR命令およびE-SEMONITOR命令は両方とも、特定のシーケンサを指定しているオペランドを有することから、「シーケンサを認識した」命令であることが分かる。

#### 【0062】

図8および図9は、本発明の実施形態に係る、持続的なユーザレベルスレッドの実行を説明するための簡略化フローチャートである。両図において、同じ参照番号は同じ動作を示している。

#### 【0063】

本発明の一実施形態によると、ブロック801に示すように、OS分離シーケンサ822が持続的なユーザレベルスレッドを実行するとしてもよい。本明細書では説明の便宜上、当該持続的なユーザレベルスレッドを、PULT-1と呼ぶ。PULT-1は例えば、第1OS可視シーケンサによって実行される第1ユーザレベルスレッドによって生成されるとしてもよい。PULT-1の実行中、OSのサービスが必要なイベントが1以上発生するとしてもよい。ここで、以下に詳述するように、第2ユーザレベルスレッドが必要なOSサービスをPULT-1に対して提供するとしてもよい。第2ユーザレベルスレッドはPULT-1を生成した第1ユーザレベルスレッドと同じユーザレベルスレッドであってもよいし、違うものであってもよい。また、第2OS可視シーケンサ(例えば、OS可視シーケンサ821)またはこれ以外のOS可視シーケンサで実行するとしてもよい。

#### 【0064】

本発明の一実施形態によると、図8および図9のブロック802で示すように、PULT-1を実行中の時間 $t_1$ においてOSサービスを必要とするイベントが発生するとしてもよい。このイベントを検出すると、ブロック803に示すように、シーケンサマネージャ820は時間 $t_1$ (例えば、論理ポインタ)でのPULT-1の状態を保存する手順を開始するとしてもよい。ブロック803に示すようにPULT-1の状態が保存されると、ブロック804に示すようにPULT-1の実行は時間 $t_2$ から一時中断されるとしてもよい。本発明の一実施形態によると、OSサービスを必要とするイベントについて通知を受け取った後、外部割込みに基づき、シーケンサマネージャ820はPULT-1の実行を一時中断するとしてもよい。PULT-1の実行の一時中断は、例えばサービススレッドによる代理実行に基づき必要なOSサービスの提供が行われるまで続くとしてもよい。

#### 【0065】

本発明の一実施形態によると、ブロック805に示すように、ある時点において、OSがスケジューリングしたスレッド(例えば新規ユーザレベルスレッド)がOS可視シーケンサ821で動作を開始するとしてもよい。この新規ユーザレベルスレッドは、PULT



- 1と同じアプリケーション用に実行されるとしてもよい。このスレッドは、例えば代理実行によって、P U L T - 1に対してO Sサービスを提供するとしてもよく、このため本明細書ではサービススレッドと呼んでもよい。O Sサービスの提供を開始する前に、ブロック806に示すように、当該サービススレッドは、O Sサービスの提供を待つ一時中断されたP U L Tがあるかどうかシーケンサマネージャ820に確認または検証するとしてもよい。

#### 【0066】

このサービススレッドは、O Sサービスの提供を待つスレッドとして、P U L T - 1を特定するとしてもよい。本発明の一実施形態によると、ブロック807に示すように、続いてサービススレッドは、例えば共有仮想アドレススペース(V A S)に基づきP U L T - 1の実行状態を取得することによって、P U L T - 1を模倣、つまり真似するとしてもよい。サービススレッドによって模倣されたP U L T - 1は、該サービススレッドを実行するO S可視シーケンサで実行され、O Sサービスを受け取るとしてもよい。この結果、ブロック808に示すように、サービススレッドの代理実行によってP U L T - 1に対してO Sサービスが提供され、P U L T - 1が一時中断された条件についてO Sによる処理が行われる。本発明の一実施形態によると、O S可視シーケンサ(例えば、O S可視シーケンサ821)で動作するユーザレベルスレッドは、シナリオを用いてアドレス位置を監視することによって、一時中断中のP U L Tを検出するとしてもよい。ここで、サービスシナリオは、例えばキャッシュミスを監視するための条件である。キャッシュミスが発生してその結果例えばしきい値を超えた場合、サービスチャネルのハンドラが起動されるとしてもよい。

#### 【0067】

本発明の一実施形態によると、P U L T - 1が一時中断された条件に対してO Sが行う処理が完了すれば、ブロック809に示すように、P U L T - 1の実行後状態が保存され、シーケンサマネージャ820を介してO S分離シーケンサ822に戻される。O S分離シーケンサ822は、サービススレッドから実行後状態を受け取った後、ブロック810に示すようにP U L T - 1の実行を再開するとしてもよい。

#### 【0068】

表1に、本発明の実施形態に係るシナリオの例をまとめて示す。

#### 【0069】

#### 【表1】

SPU 特性		出方向サービスシナリオ	入方向サービスシナリオ
OS に対する 可視性	OS 不可視	OS による処理を必要とし、R i n g 間移行が生じる動作。 例： □ ページフォルト処理 □ I/O アクセス	受信制御メッセージ □ Go_shred □ Halt_shred □ Report/Self □ Save/Restore_shred □ Single-step_shred
	OS 可視	送信制御メッセージ □ Fork □ Join □ Yield □ Wait □ Scatter/Set □ Gather/Get □ CSAVE/RSTOR □ Go/Halt/Single-step	OS による処理を必要とし、R i n g 間移行が生じる動作。例： □ ページフォルト処理 □ I/O アクセス
コアの複雑性	単純	SSE13 命令	
	複雑	SSE13 実行	

#### 【0070】

本発明の実施形態によると、表1に示したシナリオには出方向サービスと入方向サービ

スの2つに関するものがある。各サービスについて、動作および制御メッセージは、S P U ( S h r e d P r o c e s s i n g U n i t ) の特性の1つである、O S に対する可視性に基づき特徴付けるとしてもよい。出方向サービスの中で、R N A ( R e s o u r c e n o t a v a i l a b l e : 利用可能なリソースなし) のカテゴリに当てはまる動作のシナリオがある。このR N A というカテゴリは、O S 分離シーケンサで利用不可能なリソースに対するアクセスのため、シーケンサで実行中に生成されたイベント用のものである。本発明の一実施形態によると、R N A のカテゴリに入るシナリオは例えば、直接O S サービスをアクティブにできないO S 分離シーケンサ上でのページフォルト処理および/またはI / O アクセスを含むとしてもよい。

【0071】

10

本発明の実施形態によれば、上述した動作のシナリオには一連の送信制御メッセージが付随する。一実施形態によれば、このようなメッセージは例えば、表1の出方向サービスシナリオの欄に示したメッセージを含むとしてもよい。しかし、本発明はこれらに限定されず、示したものの以外のメッセージを用いるとしてもよい。制御メッセージはO S 可視シーケンサに送信される。

【0072】

上述した出方向サービスシナリオに関連して、O S 分離シーケンサからのサービス要求に対してリソースを提供するO S 可視シーケンサのカテゴリに入る入方向サービスシナリオがある。表1の入方向サービスシナリオの欄に示すように、これらのサービスは出方向サービスシナリオにそのまま対応するが、O S 可視シーケンサを介して利用可能である。一実施形態によると、入方向サービスに付随する受信制御メッセージには、表1の入方向サービスシナリオの欄に示したメッセージが含まれるとしてもよい。しかし、本発明はこれらに限定されず、示したものの以外のメッセージを用いるとしてもよい。受信メッセージはO S 分離シーケンサから受信するとしてもよい。

20

【0073】

本発明の実施形態によると、出方向サービスに対処する命令は例えば、コアの複雑性という点から見れば比較的単純なS S E 1 3 命令である。また、例えばO S 分離シーケンサからの入方向サービスまたは要求に対処する命令は、概してより複雑なものである。

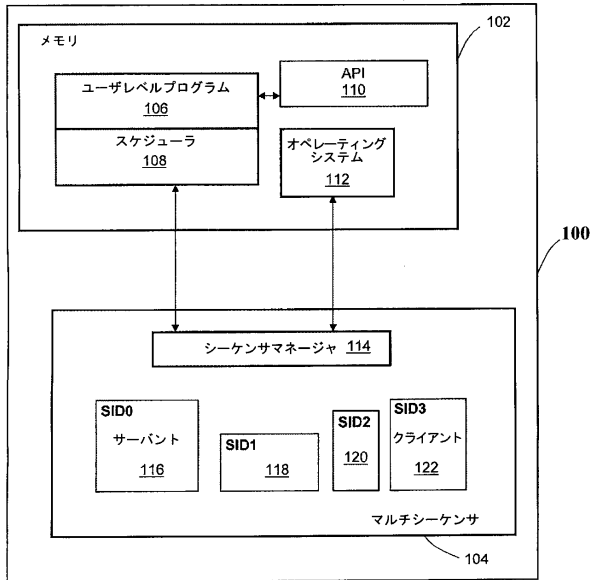
【0074】

本明細書では本発明の特徴を具体的に説明してきたが、当業者であれば数多くの変形例や代替例、変更および同様の技術内容に想到することは明らかである。このため、本願の請求項は、本発明の範囲内にある変形例や変更はすべて含むものと解釈されるべきである。

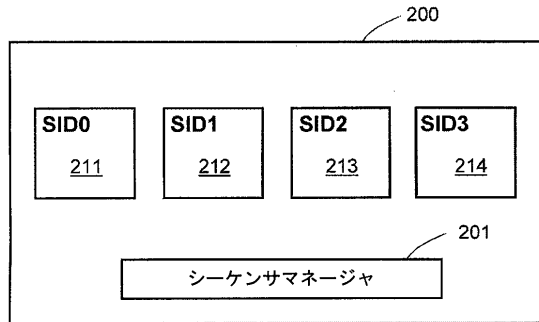
30

。

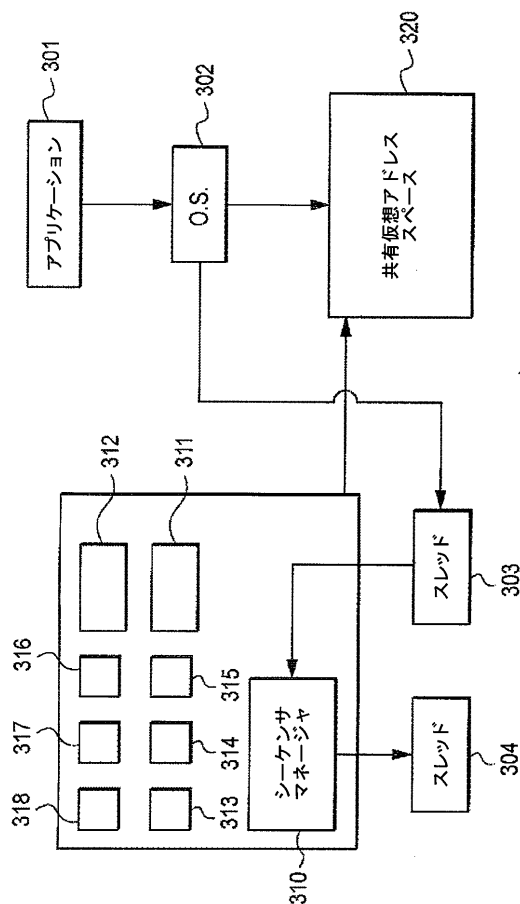
【図 1】



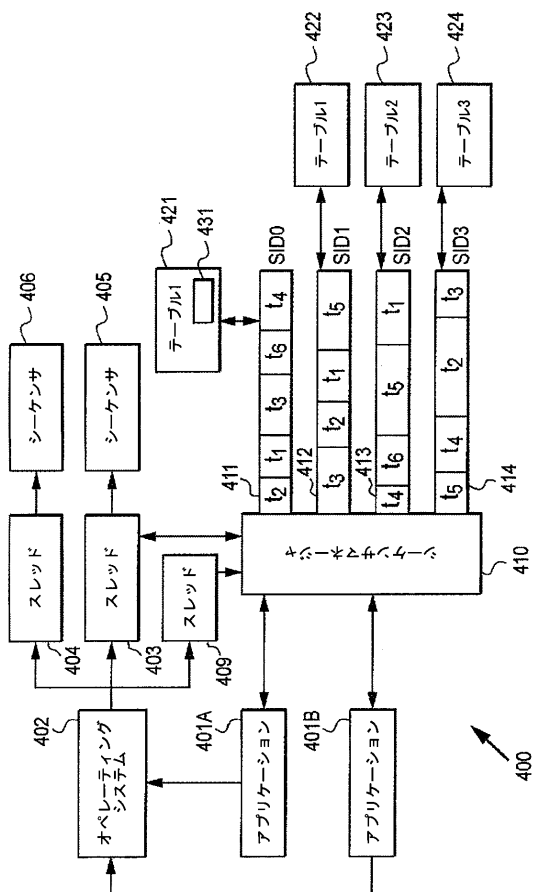
【図 2】



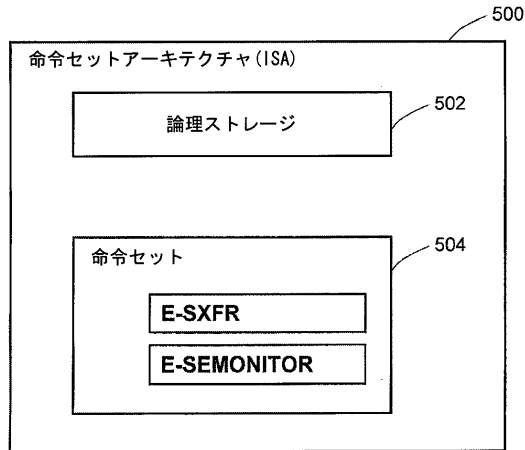
【図 3】



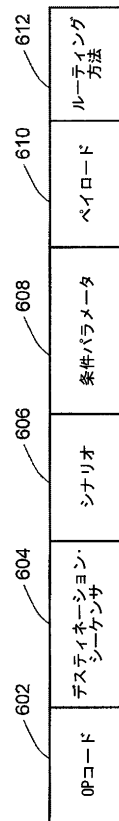
【図 4】



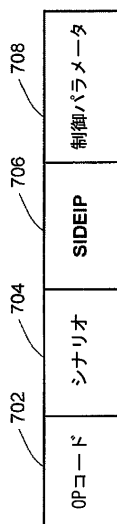
【図 5】



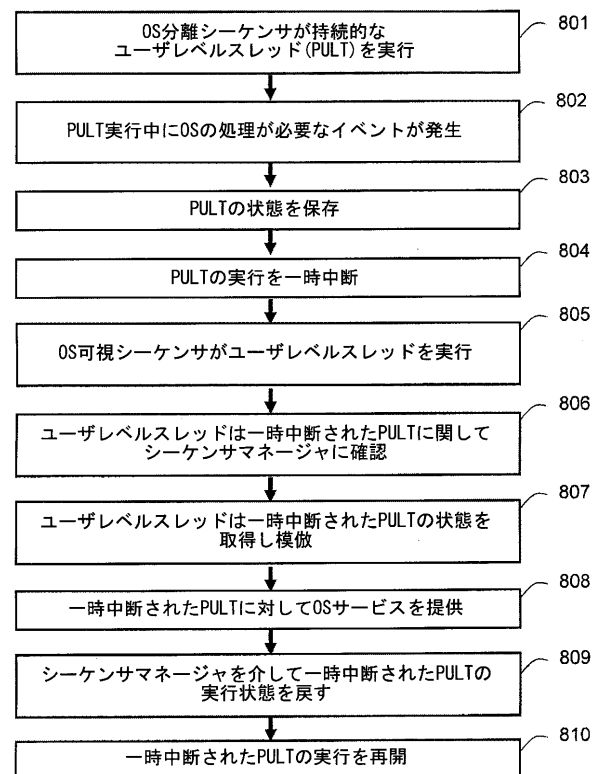
【図 6】



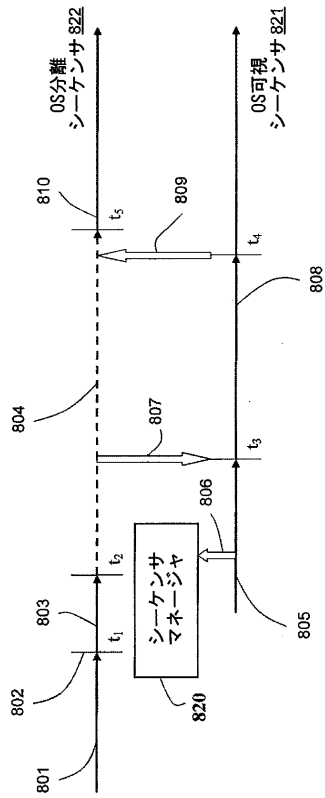
【図 7】



【図 8】



【図 9】



## フロントページの続き

- (72)発明者 ブラシャント セチ  
アメリカ合衆国、 9 5 6 3 0 カリフォルニア州、 フォルサム、 シルバーホーン ドライブ 2 3  
5
- (72)発明者 シブ カウシク  
アメリカ合衆国、 9 7 2 2 9 オレゴン州、 ポートランド、 ノースウエスト ハーバー エルエヌ  
1 4 1 1 6
- (72)発明者 ブライアント ビッグビー  
アメリカ合衆国、 8 5 2 5 9 アリゾナ州、 スコッツデール、 イースト パロミノ ロード 1 0  
8 8 6
- (72)発明者 ジョン シェン  
アメリカ合衆国、 9 5 1 2 1 カリフォルニア州、 サン ノゼ、 クエイル ブラフ プレース 2  
3 0 3
- (72)発明者 リチャード ハンキンス  
アメリカ合衆国、 9 5 0 5 2 カリフォルニア州、 サンタ クララ、 ミッション カレッジ ブー  
ルバード 2 2 0 0
- (72)発明者 シャン ズー  
アメリカ合衆国、 9 7 0 0 6 オレゴン州、 ビーバートン、 サウスウエスト ホートン ウェイ  
9 2
- (72)発明者 バイジュ バテル  
アメリカ合衆国、 9 7 2 2 9 オレゴン州、 ポートランド、 ノースウエスト エルエー カッセル  
クレスト レーン 1 0 5 5 4
- (72)発明者 ジェーソン ブラント  
アメリカ合衆国、 7 8 7 0 3 テキサス州、 オースチン、 サミット ビュー プレース 1 7 1 3  
ビー
- (72)発明者 アニル アガールウォール  
アメリカ合衆国、 9 7 2 2 9 オレゴン州、 ポートランド、 ノースウエスト ケイトリン テラス  
1 5 8 7
- (72)発明者 ジョン レイド  
アメリカ合衆国、 9 7 2 2 9 オレゴン州、 ポートランド、 ノースウエスト ウォーターフォード  
ウェイ 1 6 8 9 3

審査官 漆原 孝治

- (56)参考文献 特開平 0 5 - 2 6 5 8 7 6 ( J P , A )  
特開平 0 4 - 0 9 7 4 4 5 ( J P , A )  
米国特許第 0 5 4 8 5 6 2 6 ( U S , A )  
佐藤未来子、外 6 名、 マルチスレッドアーキテクチャ向け OS 「 F u t u r e 」 におけるプロセ  
ス管理、 コンピュータシステム・シンポジウム論文集、 日本、 社団法人情報処理学会、 2 0 0 3  
年 1 2 月 1 1 日、 Vol. 2003, No. 20, pp. 61 - 70  
HANKINS, R. A. et al., Multiple Instruction Stream Processor, Proceedings of the 33rd  
annual international symposium on Computer Architecture, IEEE, 2 0 0 6 年 6 月 2 1 日

(58)調査した分野(Int.Cl., D B 名)

G 0 6 F 9 / 4 6 - 9 / 5 4