(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第5801372号
(P5801372)
(45)発行日　平成27年10月28日(2015.10.28)　(24)登録日　平成27年9月4日(2015.9.4)

(51)Int.Cl.　　F I
G06F 9/48 (2006.01)　　G06F 9/46 455Z

請求項の数 18 (全 17 頁)

(21)出願番号　特願2013-250360(P2013-250360)
(22)出願日　平成25年12月3日(2013.12.3)
(62)分割の表示　特願2012-517932(P2012-517932)の分割
原出願日　平成22年8月2日(2010.8.2)
(65)公開番号　特開2014-75147(P2014-75147A)
(43)公開日　平成26年4月24日(2014.4.24)
審査請求日　平成25年12月3日(2013.12.3)
(31)優先権主張番号　12/550,737
(32)優先日　平成21年8月31日(2009.8.31)
(33)優先権主張国　米国(US)

(73)特許権者　593096712
インテル　コーポレイション
アメリカ合衆国　９５０５４　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　２２００
(74)代理人　100107766
弁理士　伊東　忠重
(74)代理人　100070150
弁理士　伊東　忠彦
(74)代理人　100091214
弁理士　大貫　進介
(72)発明者　ナトゥ，マヘシュ，エス．
アメリカ合衆国　９７２２９　オレゴン州　ポートランド　ノースウェスト　バニスター　ドライヴ　５５５４
最終頁に続く

(54)【発明の名称】システム管理モードのためのプロセッサにおける状態記憶の提供

(57)【特許請求の範囲】
【請求項１】
コアのアクティブな状態を記憶する記憶ユニットと；
命令を実行し、システム管理モード（SMM）にはいる前記コアとを有するプロセッサであって、前記コアは前記記憶ユニットに結合されており、
前記コアが、前記コアの状態記憶に存在しているアクティブな状態を、システム管理ランダム・アクセス・メモリ（SMRAM）にではなく前記記憶ユニットに記憶できるようにされていることを示すインジケータを記憶する第一の状態レジスタを含む複数の状態および構成設定レジスタを前記コアは含み、
SMMにはいる際、前記コアは、前記アクティブな状態を前記記憶ユニットに記憶し、SMMに関連する値を前記状態記憶中に挿入することによってSMM実行環境をセットアップする、プロセッサ。
【請求項２】
前記第一の状態レジスタが、SMM外では変更可能でない、請求項１記載のプロセッサ。
【請求項３】
前記コアが、前記SMRAMに記憶されているSMMコードを実行する、請求項１記載のプロセッサ。
【請求項４】
SMMがメモリ・エラーを解決するとき、前記コアが不揮発性メモリから復元SMMコードを取得するのであって、前記SMRAMから前記SMMコードを取得するのではない、請求項３記載

のプロセッサ。
【請求項５】
前記複数の状態および構成設定レジスタが、前記コアの論理プロセッサが長いフローの動作中であることを示すインジケータを記憶する第二の状態レジスタをさらに有しており、長いフローの動作とは少なくとも10^3クロックかかる動作である、請求項１記載のプロセッサ。
【請求項６】
前記複数の状態および構成設定レジスタが、前記コアの前記論理プロセッサがシステム管理割り込み（SMI）禁止状態にあることを示すインジケータを記憶する第三の状態レジスタをさらに有する、請求項５記載のプロセッサ。
【請求項７】
SMMにはいった前記コアの各論理プロセッサの指標を記憶するSMMインジケータ・マップをさらに有する、請求項６記載のプロセッサ。
【請求項８】
前記第一、第二および第三の状態レジスタがSMM外では書き込み不能である、請求項６記載のプロセッサ。
【請求項９】
プロセッサであって：
フロントエンド・ユニットと；
前記フロントエンド・ユニットに結合された複数のレジスタ・ファイルと；
前記フロントエンド・ユニットに結合された複数のレジスタ・セットであって、各レジスタ・セットは、当該プロセッサが、システム管理モード（SMM）にはいる際、当該プロセッサの記憶ユニット中の前記複数のレジスタ・ファイルの一つに記憶されているアクティブな状態を記憶できるようにされていることを示すインジケータを記憶する第一のレジスタを含む、複数のレジスタ・セットと；
命令を実行する複数の実行ユニットと；
当該プロセッサがSMMにあるときに前記アクティブな状態を記憶する前記記憶ユニットとを有する、
プロセッサ。
【請求項１０】
前記第一のレジスタが、SMM外では変更可能でない、請求項９記載のプロセッサ。
【請求項１１】
当該プロセッサが、当該プロセッサに結合されているシステム管理ランダム・アクセス・メモリ（SMRAM）に記憶されているSMMコードを実行する、請求項９記載のプロセッサ。
【請求項１２】
SMMがメモリ・エラーを解決するとき、当該プロセッサが不揮発性メモリから復元SMMコードを取得するのであって、前記SMRAMから前記SMMコードを取得するのではない、請求項１１記載のプロセッサ。
【請求項１３】
前記記憶ユニットが静的ランダム・アクセス・メモリである、請求項９記載のプロセッサ。
【請求項１４】
当該プロセッサがさらに、前記複数の実行ユニットにおいて実行される命令をリタイアさせるリタイア・ユニットを有する、請求項９記載のプロセッサ。
【請求項１５】
前記プロセッサが、複数のコアと、統合メモリ・コントローラと、共有されるキャッシュ・メモリとを含むマルチコア・プロセッサである、請求項９記載のプロセッサ。
【請求項１６】
システム管理割り込み（SMI）イベントに応答して、プロセッサ上で実行中のすべてのスレッドがシステム管理モード（SMM）集結状態にはいったかどうかを判定する段階と；

もしまだである場合、残っているスレッドが、選択された命令の集合の一つを実行中であるまたは選択されたプロセッサ状態にあるかどうかを判定し、もしそうであれば、前記残っているスレッドが前記選択された命令の前記一つを実行するまたは前記選択されたプロセッサ状態にある間に、モナーク・スレッドを使って、前記SMIイベントを処理し、そうでなければ、前記残っているスレッドがSMM集結状態にはいるのを待ってから前記SMIイベントを処理する段階とを含む方法であって、

当該方法がさらに、

SMM集結状態にはいる各スレッドについて、SMMインジケータ・マップ内のインジケータをセットする段階と；

スレッドが前記選択された命令の前記一つを実行中であることを示すよう、前記プロセッサの第一の状態レジスタのインジケータを設定する段階と；

スレッドが前記選択されたプロセッサ状態にあることを示すよう、前記プロセッサの第二の状態レジスタのインジケータを設定する段階とを含む、

方法。

【請求項１７】

前記判定する処理が、前記SMMインジケータ・マップと、前記第一の状態レジスタと、前記第二の状態レジスタとの間のビットごとのOR演算を実行することを含む、請求項１６記載の方法。

【請求項１８】

スレッドがSMMにはいる際、前記モナーク・スレッドが、前記プロセッサの状態記憶に存在するアクティブな状態を、前記プロセッサの記憶ユニット中に記憶し、SMM状態を前記状態記憶に記憶する、請求項１６記載の方法。

【発明の詳細な説明】

【技術分野】

【０００１】

本願はプロセッサのシステム管理モードに関する。

【背景技術】

【０００２】

たいていのコンピュータ・システム・プロセッサは、システム管理モード（SMM: system management mode）と呼ばれる特別な動作モードをサポートしている。SMMはオペレーティング・システム（OS: operating system）ソフトウェアにとって透明な、独特な動作環境を提供する。このモードはしばしば、相手先ブランド装置製造業者（OEM: original equipment manufacturer）によって、システム管理、デバイス、電力および熱管理といった特別なタスクを実行するために使用される。サーバーに関係した信頼性、可用性および保守性（RAS: reliability, availability and serviceability）機能は通例SMMを使って実装される。典型的にはシステム管理割り込み（SMI: system management interrupt）メッセージをプロセッサに送ることによって、SMMにはいる。SMIを受け取り確認すると、プロセッサは、プロセッサ保存状態（Processor Save State）とも呼ばれる現在のプロセッサ・コンテキストを、SMMに特に割り当てられている、システム管理ランダム・アクセス・メモリ（SMRAM: system management random access memory）と称されるシステム・メモリの部分に保存し、SMRAMに含まれるSMIハンドラ・コードを実行する。SMIハンドラがその動作を完了すると、特別な（SMMにおいてのみ有効な）再開命令を実行し、それによりプロセッサは保存されたプロセッサ・コンテキストをSMRAMから改めてロードし、中断されたタスクの実行を再開する。

【０００３】

マルチプロセッサ・システムでは、一般にSMIメッセージはすべてのプロセッサにブロードキャストされる。SMIハンドラは、そのイベントを処理するために、SMMモナーク（monarch）と称される一つのプロセッサを選択する。このプロセッサは、SMIイベントを処理する前に、他のすべてのプロセッサがSMM内部に集結〔ランデブー〕するまで待つ。非モナーク・プロセッサは、モナークがイベント処理を完了するまでSMMに留まる。SMMイベン

トが処理されたら、モナークは他のプロセッサにSMMを出るよう合図する。この同期された入場および退場振る舞いは、二つの並列環境（OSおよびSMM）の間のいかなる資源の衝突も防止するよう実装される。すなわち、いくつかのプロセッサがOS環境においてアクティブであり、残りは同時にSMM環境においてアクティブである場合、それらが共有された資源を修正し、それにより互いの動作に干渉してシステム・クラッシュを引き起こすことがありうる。さらに、ある種のSMMイベントは、特定の論理プロセッサまたは一組の論理プロセッサによってのみ処理されることができる。ブロードキャストは、すべての論理的プロセッサがSMIにはいるので、この条件が常に満たされることを保証する。

【発明の概要】

【発明が解決しようとする課題】

【０００４】

このように、マルチプロセッサにおけるSMI処理（SMI handling）は複雑で、すべてのシステム資源を消費して、他の有用な作業を処理することを妨げることがある。プロセッサがSMM内にある間は、オペレーティング・システムにとって利用可能でないからである。

【課題を解決するための手段】

【０００５】

本願の課題は請求項記載の手段によって解決される。

【図面の簡単な説明】

【０００６】

【図１】本発明のある実施形態に基づくプロセッサのブロック図である。

【図２】本発明のある実施形態に基づくマイクロプロセッサ・システムのブロック図である。

【図３】本発明のある実施形態に基づく方法の流れ図である。

【図４】本発明のもう一つの実施形態に基づく方法の流れ図である。

【発明を実施するための形態】

【０００７】

さまざまな実施形態において、SMM入場／退場に際して個々のスレッドの保存状態を記憶するために外部の物理的メモリを使うことに対する代替として、ダイ上記憶を使用することができる。対照的に、現行のシステムはSMMに入場および退場するために、外部の物理的メモリに依存する。このSMMのシステムRAMへの依存の結果、ミッションクリティカルな応用におけるスケーリング、パフォーマンスおよび信頼性に関係した制限が生じるが、そのような依存は、本発明のある実施形態を使って回避できる。本稿での用法では、「スレッド」という用語は、プロセスに関連付けられたアーキテクチャ状態についてのプロセッサ中の記憶（たとえば、レジスタ・ファイルおよび関連する構成設定および状態レジスタ）を含むハードウェア・スレッドを指しうることを注意しておく。本稿での用法では、「ハードウェア・スレッド」という用語は、「論理プロセッサ」という用語と同義に使われる。各プロセッサ・コアは複数の論理プロセッサを含んでいてもよく、各論理プロセッサは、専用のアーキテクチャ状態記憶を有するが、フロントエンド・ユニット、実行ユニットなどといった他のコア資源を共有する。

【０００８】

種々の実装において、SMMの間、アクティブ・スレッドがあればその保存状態を記憶しておくために設けられるダイ上記憶は、保存状態記憶のための小さな専用のメモリとしてはたらく、ダイ上静的RAM（SRAM: static RAM）またはプロセッサ自身の中のレジスタ・ファイルであることができる。いくつかのプロセッサは、電力管理のような特定のタスクのためのダイ上SRAMを含むことがある。該電力管理とは、たとえば、先進構成設定および電力インターフェース（ACPI: Advanced Configuration and Power Interface）状態に基づくようなOSに管理される低電力状態（たとえば、C6状態または他の電力管理動作）である。そのようなプロセッサでは、スレッド毎に分割した、このSRAMの一部を、各スレッドのSRAM保存状態のためにリザーブすることができる。一例として、各論理プロセ

ッサは、SMM保存状態のために１キロバイト（KB）のSRAM記憶を使ってもよい。所与のプロセッサがこの量のSRAMをSMM保存状態のために割くことができない場合、ある実施形態は、C6フローのためにリザーブされているSRAMを利用できるよう実装されることができる。この場合、SMM内部のC6/C7遷移はより低い低電力状態（たとえばC3）に降格されることができる。SMM状態保存のために共有されるSRAM空間の互いに排他的な使用を保証するためである。いくつかのプロセッサはC6状態保存のための専用のSRAMを実装せず、代わりに最終レベル・キャッシュ（LLC: last level cache）の一部を、C6状態保存の間、プロセッサ状態を保存するために利用する。これらのプロセッサにおいて、SMM保存状態はLLC内に記憶されることができる。

【０００９】

いったん保存されると、この内部SMM保存状態は種々の仕方でアクセスされうる。例として、内部状態は、スレッド毎のモデル固有レジスタ（MSR: model-specific register）アドレッシングまたは上位互換な機構を使ってアクセスされることができる。従来では、プロセッサは、ある種のシステム・メモリ・アドレスにおいてSMM保存状態にアクセスできる。上位互換な機構は、これらのレガシー・メモリ・アドレスへの論理プロセッサのアクセスを捕捉して、それらを適切なSRAM位置にリダイレクトするプロセッサ中の論理を含む。そのようなリダイレクトは、既存の基本入出力システム（BIOS: basic input/output system）ソフトウェアとの絶対的な上位互換性が要求される場合に実装されることができる。これらのMSRは、SMMモードでのみ読まれたり書かれたりすることができ、SMM保存状態に関連する制約に従う。ある論理プロセッサが別のプロセッサの保存状態へのアクセスを必要とする場合、これはソフトウェア・プロトコルを介して達成できる。

【００１０】

いくつかの実施形態では、専用のプロセッサ識別子リーフ（leaf）（たとえばCPUIDリーフ）またはそのフィールドまたは機能イネーブルMSR（Model-specific Register）ビットが、内部SRAMの使用を有効にするために使うことができる。ここで図１を参照するに、本発明のある実施形態に基づくプロセッサのブロック図が示されている。図１に示されるように、プロセッサ１００は多段（multi-stage）パイプライン式（pipelined）順序外（out-of-order）プロセッサであってもよい。プロセッサ１００は、本稿で記載されるSMM技法に関連して使われるさまざまな機能を例解するために比較的単純化した図として示している。見て取れるように、プロセッサ１００は、複数のプロセッサ・コア１０５を含み、単一の半導体ダイ上に形成されうるマルチコア・プロセッサであってもよい。図１の実施形態では四つのそのようなコアをもって示されているが、本発明の範囲がこの点に関して限定されないことを理解されたい。図１においてさらに見て取れるように、追加的なコンポーネントがプロセッサ１００内に存在していてもよい。たとえば、統合メモリ・コントローラ（IMC: integrated memory controller）１０８が、静的ランダム・アクセス・メモリ（SRAM）１０６とともに存在していてもよい。上で論じたように、いくつかの実装では、このメモリは、普通ならSMRAMに記憶されるところのコンテキスト状態を本発明の実施形態に基づいて記憶するために使用されてもよい。さらに、プロセッサ１００は、すべてのプロセッサ・コアの間で共有されている共有キャッシュであってもよい最終レベル・キャッシュ（LLC）１０９を含んでいてもよい。

【００１１】

図１に示されるように、プロセッサ１００は、実行されるべきマクロ命令をフェッチしてそれらをコアにおけるのちの使用のために用意するために使用されうるフロントエンド・ユニット１１０を含む。たとえば、フロントエンド・ユニット１１０は命令プリフェッチャー、命令デコーダおよびトレース・キャッシュを、マイクロコード記憶およびマイクロ命令（μop）記憶とともに含んでいてもよい。命令プリフェッチャーは、メモリからマクロ命令をフェッチし、それらを命令デコーダに供給してもよい。命令デコーダはそれらのマクロ命令をプリミティブ、すなわちプロセッサによる実行のためのマイクロ命令にデコードする。トレース・キャッシュはデコードされたマイクロ命令を受け、それらをプログラム順序付けシーケンス（program ordered sequence）にまとめてもよい。もちろん

、追加的なコンポーネントおよび機能がフロントエンド・ユニット１１０において実装されていてもよい。

【００１２】

フロントエンド・ユニット１１０および実行ユニット１２０の間には、マイクロ命令を受け取ってそれらを実行のために用意するために使用されうる順序外（OOO: out-of-order）エンジン１１５が結合される。より具体的には、OOOエンジン１１５は、マイクロ命令フローを並べ替え、実行のために必要とされるさまざまな資源を割り当てるとともに、レジスタ・ファイル１３０ａのようなさまざまなレジスタ・ファイル内の記憶位置への論理レジスタのリネーミング（renaming）を提供するためのさまざまなバッファを含んでいてもよい。レジスタ・ファイル１３０は、整数演算および浮動小数点演算のための別個のレジスタ・ファイルを含んでいてもよい。それぞれ異なる論理プロセッサ用の複数のレジスタ・ファイル１３０ａ～ｎが存在していてもよいことを注意しておく。さらなるレジスタ、すなわち状態および構成設定レジスタ１３５も存在していてもよい。見て取れるように、レジスタ１３５ａ～ｎの各セットは異なる論理プロセッサ用であってもよい。これらさまざまなレジスタは、異なる動作モードのためにコアを構成設定するために、また実行されるスレッドおよび異なる命令に関して状態情報を提供するために、使用されてもよい。

【００１３】

図１に示した例では、そのようなレジスタはSMM保存状態レジスタ１３６を含んでいてもよい。さまざまな実装では、それぞれコア上で動作する所与のスレッドに関連付けられている複数のそのようなレジスタが存在していてもよい。上で論じたように、そのようなレジスタは、SMMにはいるときにコア自身の内部などにスレッドの状態が記憶されることができるようにするインジケータ、たとえばイネーブル・ビットを記憶してもよい。このインジケータが有効にされていない場合は、その代わりに、SMMにはいった時点でスレッドのコンテキストがSMRAMに記憶される。いくつかの実施形態では、このMSRは、他のプロセッサ機能を制御できる他のビットを含んでいてもよい。いくつかの実施形態では、インジケータを含むこのレジスタ・ファイル１３５は、SMMにおいてのみ変更可能であるようにされることができる。こうして、SMMの外部のマルウェア・コンポーネントによって悪意をもって変更されることから保護され、システムのセキュリティおよび堅牢性の両方が向上する。

【００１４】

さらに見て取れるように、レジスタ・ファイル１３５は一つまたは複数のSMM状態インジケータ・レジスタ１３８をも含んでいてもよい。そのようなインジケータ・レジスタは、ビットマップまたはビット・ベクトルの形であってもよく、ここで、各論理プロセッサが、該論理プロセッサがいつSMMにはいることを禁止されるか、あるいは該論理プロセッサが長いフロー実行の中にあるかどうかを指示するための位置をもつ。ある実施形態では、別個のレジスタがそのような各指示について存在していてもよい。あるいはまた、単一のレジスタが存在していてもよく、論理的に組み合わされたインジケータが、各論理プロセッサについてこれらの状態の一つの存在を指示するために使用されてもよい。これらのレジスタの使用に関するさらなる詳細は以下で述べる。

【００１５】

引き続き図１を参照するに、さまざまな資源が実行ユニット１２０中に存在していてもよい。それにはたとえば、数ある特化されたハードウェアの中でも、整数、浮動小数点および単一命令複数データ（SIMD: single instruction multiple data）論理ユニットが含まれる。結果は、リアイアメント（retirement）ユニット１４０に与えられてもよい。リアイアメント・ユニット１４０は、実行された命令が有効にリタイアされ、結果データがプロセッサのアーキテクチャ状態にコミットされることができるかどうか、あるいは命令の適正なリアイアメントを妨げる一つまたは複数の例外が発生したかどうかを決定するよう動作しうる。

【００１６】

図１に示されるように、リアイアメント・ユニット１４０はキャッシュ・メモリ１５０に結合されている。キャッシュ・メモリ１５０は一実施例では低レベル・キャッシュ（たとえばL1キャッシュ）であってもよいが、本発明の範囲はこの点に関して限定されるものではない。また、実行ユニット１２０はキャッシュ１５０に直接結合されることができる（図１には示さず）。キャッシュ・メモリ１５０から、より高いレベルのキャッシュ、システム・メモリなどとデータ通信が行われてもよい。図１の実施形態ではこの高レベルで示されているが、本発明の範囲がこの点に関して限定されるものではないことを理解されたい。たとえば、他の実施形態は、順序内（in-order）プロセッサにおいて実装されてもよい。

【００１７】

SMM保存状態を内部的にプロセッサに保存することによって、システムの信頼性および堅牢性が改善されうる。すなわち、典型的にはSMRAMが存在する外部の動的ランダム・アクセス・メモリ（DRAM）デバイスの組である物理的メモリはメモリ・エラーを起こしやすい。本発明の実施形態なしでは、SMM動作はその外部メモリを使い果たして、よってエラー状態において頼られることができない。その代わり、本発明の実施形態を使うと、エラーを処理する際に不揮発性空間からSMIハンドラを実行することによって、SMRAMメモリ信頼性が改善できる。たとえば、SMMハンドラは、メモリ・エラーを処理する間、BIOSフラッシュまたは外部SRAMのようなより堅牢な記憶から走らせることができる。また、SMM保存状態がプロセッサ内部であるとき、この記憶のアーキテクチャ状態は、ソフトウェア外部には、MSRを通じてのみ暴露されることができる。SMMコードが「再開（RSM: resume）」命令を実行した後に機械実行状態を復元するために必要とされるプロセッサのマイクロアーキテクチャ状態は、外部のソフトウェアに暴露される必要がない。外部のソフトウェアはこの内部機械状態にとって何の正当な用途もないからである。これは、悪意のあるソフトウェア・コードが機微なマイクロアーキテクチャ状態へのアクセスをもたないことをも意味する（保存されたデータ記憶がSMRAM内にあればそのようなアクセスをもつはずである）。それにより当該機械がより安全で、堅牢になる。

【００１８】

諸実施形態は、パフォーマンスおよびレイテンシも改善しうる。多くのサーバー・アプリケーション／オペレーティング・システムは非一様メモリ・アーキテクチャ（NUMA: non-uniform memory architecture）最適化をされており、BIOSは典型的には、連続するメモリ範囲であるSMRAM全体が単一のソケットにマッピングされるようメモリを構成設定する。したがって、すべてのSMM保存状態／復元状態動作は、SMRAMにとってローカルな一つのソケットに含まれるものを除いて、すべての論理CPUについてリモート書き込み／リモート読み出しのように見えるであろう。それぞれ12個のコアをもつ四つのソケットをもつサーバー構成についてのパフォーマンス解析によると、SMM保存状態書き込み動作は相互接続およびメモリ帯域幅によって制限されることができ、高々５マイクロ秒かかることができることが示される。アプリケーションがよりNUMA最適化されるにつれ、プロセッサはリモート・トラフィックのためにさらに少数のバッファを割り当てることがありうる。そうなると、SMRAM保存状態書き込みおよび読み出し動作は、さらに長い時間がかかる。オペレーティング・システムは典型的には、受け容れ可能なリアルタイム・パフォーマンスを維持し、高速ネットワーク・リンク上でのタイムアウトを回避するために、CPUがどれくらい長くSMM内にあることができるかについて制限をもつ。この制限を超えることは、OSの反応性、アプリケーション・レイテンシに影響し、さらにはオペレーティング・システムの誤動作につながることもありうる。したがって、本発明のある実施形態に基づくダイ上SMM保存状態を使うことは、レイテンシを短縮し、よってSMMイベントにサービスする（SMMの有用な作業）ためのSMMハンドラのために割り当てられるさらなる時間を可能にする。

【００１９】

さらに、諸実施形態はスケーラビリティを改善しうる。マルチプロセッサ・システムにおいて、SMIが発生するとき、システム中の全スレッドがその保存状態を、システム・ブ

ートの際にシステムBIOSによって画定され、リザーブされる、外部システム・メモリ中のそれ自身の専用の保存状態領域に記憶しなければならない。システム中のすべてのスレッドの保存状態すべてを取り込むために必要とされるSMRAM空間としてリザーブされるべき物理的メモリの総量は、システム中のスレッド数に対して線形に増大する。対称的なマルチスレッディング・サポートをもつマルチコア、マルチソケット・システムについて、空間の量はかなり大きいことがある（ある実施形態では約256KBのオーダーになりうる）。SMM保存状態のためのダイ上記憶を設けることによって、すべてのコアおよびそのスレッドを収容するための拡大し続けるSMRAM領域の必要性が回避でき、それによりスケーリングが容易にされる。また、BIOSがスレッド毎にSMRAM中の一意的な重複しない領域を見出し、割り当てる必要性もなくなる。さらにまた、これはメモリ保護領域がシリコンにおいて実装されることを免除する。ホット・プラグ・シナリオでは、SMRAMにおけるアーキテクチャ的に画定されるSMM保存状態領域は1MB未満である。本発明の実施形態なしでは、BIOSはメモリ保護範囲を設定し、新しいプロセッサを追加するときは、OS攻撃および／または干渉を避けるために、データを退避する。諸実施形態では、保存される状態はもはやOS可視メモリ内に記憶されないので、これを行う必要がなくなる。

【００２０】

ここで図２を参照すると、本発明のある実施形態に基づくマルチプロセッサ・システムのブロック図が示される。図２に示されるように、マルチプロセッサ・システム２００は複数のプロセッサ２１０_{1}～２１０_{n}（概括してプロセッサ２１０）を含む。図２の実施形態ではそのようなプロセッサ四つが示されているが、本発明の範囲がこの点に関して限定されるものではないことを理解されたい。図２に示される実施形態では、非一様メモリ・アーキテクチャ（NUMA）システムが存在しており、システム・メモリ２２０_{1}および２２０_{3}が相互接続２１７_{1}および２１７_{3}を介してプロセッサ２１０_{1}および２１０_{3}にローカルに取り付けられる。こうして、プロセッサ２１０_{2}および２１０_{n}によるメモリへのアクセスは、プロセッサ２１０_{1}および２１０_{3}の一方との複数のポイントツーポイント（PTP: point-to-point）相互接続２１５のうちの一つを通じた通信を必要とする。図２の実装において見て取れるように、DRAMであってもよいメモリ２２０_{1}はSMRAM　２２５を含む。このNUMA最適化アーキテクチャでは、SMRAM　２２５はシステム全体のためのシステム管理ストアである。よって、本発明の実施形態なしでは、各プロセッサは、SMM入場または退場の際、このSMRAM　２２５にコンテキストを保存／復元する必要がある。これはPTP相互接続２１５および相互接続２１７_{1}での帯域幅の大幅な使用を引き起こすとともに、SMMへの入場およびSMMからの退場のためのレイテンシを増加させる。したがって、さまざまな実施形態において、各プロセッサ２１０は、一つまたは複数のコア２１２および統合メモリ・コントローラ２１４に加えて、SRAM　２１６を含んでいてもよい。さまざまな実施形態では、SRAM　２１６はSMM保存状態の記憶のために専用とされてもよい。すなわち、システム管理割り込みが起こるとき、各プロセッサ２１０のさまざまな論理プロセッサについてのコンテキスト状態がそのSRAM　２１６にローカルに記憶されてもよい。それによりSMRAM　２２５との状態情報の通信の必要性が回避される。他の実施形態では、専用のダイ上記憶の代わりに、このコンテキスト状態は、たとえばレジスタ・ファイルまたはキャッシュ・メモリのような他の位置の、チップ上レジスタに記憶されることができる。図２の実施形態ではこの特定の実装で示しているが、本発明の範囲はこの点について限定されるものではない。たとえば、諸実施形態はさらに、一様メモリ・アーキテクチャ・システムとともに使用されてもよい。

【００２１】

ここで図３を参照するに、本発明のある実施形態に基づく方法の流れ図が示されている。図３に示されるように、方法３００は、状態情報を保存するためにSMRAMにアクセスする必要なしにSMMへの入場を扱うよう実行されうる。議論の簡単のため、単一のハードウェア・スレッドしか存在していないと想定されるが、多くの実装では複数のスレッドが一緒にSMMにはいってもよいことを注意しておく。図３で見て取れるように、方法３００は、システム管理割り込みを受け取ることによって開始されうる（ブロック３１０）。この

割り込みを受領すると、（たとえば所与のハードウェア・スレッドの）現在のアクティブ状態がダイ上記憶に保存されてもよい（ブロック３２０）。上で論じたように、このダイ上記憶は、専用のSRAM、別の目的（たとえば電力管理状態）のために使われるSRAM、レジスタ記憶、ダイ上キャッシュ記憶などであってもよい。

【００２２】

引き続き図３を参照するに、プロセッサ状態は、たとえばプロセッサ仕様によって定義されるところのSMM入場状態にマッチするよう修正される（ブロック３３０）。この状態は、さまざまな制御および構成設定レジスタについての値およびレジスタ・ファイルについての初期値を含む。よって、このセットアップは、SMM入場状態に関連付けられた所定の値を状態記憶にロードすることによって、SMMハンドラのために適切なSMM実行環境を用意する。SMM状態がセットアップされたら、制御はブロック３４０に進む。ブロック３４０では、SMMがSMRAMからのコードおよびデータを使って実行されてもよい。したがって、所望されるSMM動作が実行されてもよい。本発明の範囲はこの点に関して限定されるものではないが、SMM動作の例は電力管理動作、エラー処理動作などを含む。

【００２３】

次いで、SMM動作が完了したかどうかが判定されてもよい（菱形３５０）。まだであれば、SMMにおける実行は継続してもよい。完了していれば、プロセッサは再開命令を実行する（ブロック３６０）。この命令の結果として、前の状態が、ダイ上記憶からプロセッサのレジスタにロードし戻されてもよい（ブロック３７０）。次いで、プロセッサは、アクティブ状態に復元し戻されたこの、前の状態に対応するスレッドの実行を再開してもよい（ブロック３８０）。図３の実施形態ではこの特定の実装をもって示しているが、本発明の範囲がこの点に関して限定されるものではないことを理解されたい。たとえば、いくつかの実装では、特にSMMがDRAMエラーのようなエラーを処理するためであるとき、SMM動作をSMRAMから実行するのではなく、諸実施形態はSMM状態情報、SMMコードおよびデータをフラッシュメモリのような不揮発性記憶から取得してもよい。

【００２４】

上記のように、アクティブ状態のシリコン記憶はSMMレイテンシを減らすことができる。諸実施形態はさらに、ある種の状況においてより高速にSMMにはいることを可能にすることによって、さらにレイテンシを減らしてもよい。それについてこれから論じる。

【００２５】

SMMレイテンシは、単一SMI当たりプロセッサがSMM環境内にある期間の長さとして定義される。全SMMレイテンシに対する主として二つの寄与因子がある。プロセッサ・オーバーヘッドおよびOEM BIOSコードである。このレイテンシは、タイムアウトおよびクロック・ドリフトのようなOS環境に対する副作用を回避するために制御下に保たれる必要がある。将来の需要はこのレイテンシが減らされることを要求するであろうが、それは実現するのが難しくなる。現在のところ、SMIレイテンシは、約190マイクロ秒未満であるよう指定されている。インターネット・ポータル・データ・センターおよびユーティリティー・コンピューティングのような新しい使用モデルは、アプリケーションから、より予測可能なレイテンシを期待する。結果として、OSベンダーはSMMレイテンシのさらなる削減を求めている。他方、他の技術は時間とともにSMIレイテンシを増大させる可能性がある。たとえば、マルチコア・プロセッサに向けた業界の圧力は、SMIハンドラが増大し続ける数のプロセッサ・コアを集結させなければならないことを意味する。新しいSMMベースの機能も、SMMレイテンシに対して追加的な圧力をかける。たとえば、ハイエンドRAS機能はSMMに依拠する。さらに、一部のOEMは、自分たちの製品を差別化するために独特な電力管理機能を与えるため、SMMを利用する。多くのOEMは、１秒当たり８回もSMIを生成することが知られている。

【００２６】

ある種の命令セット・アーキテクチャ（ISA: instruction set architecture）は、すべてのキャッシュ・ラインを無効にしてメモリに書き戻すライトバック（write back）および無効化（invalidate）命令（たとえばwbinvd）のような命令を含む。これらの動

作は完了までに、特に大きなキャッシュ・サイズをサポートするプロセッサでは、たとえば10^3ないし10^7プロセッサ・サイクルのオーダーの長い時間がかかることがある。さらに、SMI応答が遅延されることのできるある種のプロセッサ状態がある（たとえばC3およびC6低プロセッサ状態）。まとめて、これらの命令およびプロセッサ状態は「長いフロー（long flow）」状態と称される。これは、完了するのに異例なほど長いサイクル数（たとえば10^3クロックのオーダー）がかかることがあり、SMMにはいるのを遅らせることができる命令またはプロセスを意味するものと定義される。ある実施形態では、SMM入場を５ミリ秒より長く遅らせるいかなるフローも長いフローと称されることができる。SMMに関しては、一つまたは複数の論理プロセッサが長いフロー中にある場合、それはSMMにはいるのを遅らせる。

【００２７】

上に説明したように、SMMモナークは、すべての期待される論理プロセッサがSMMにはいってしまうまで待つ。SMMにはいると、各プロセッサはSMRAM中の自らのビットをセットして、SMMにはいったことを示す。モナークは、すべての期待されるプロセッサがそのビットをセットしてしまうまで待つ。一つまたは複数の論理プロセッサが長いフロー中にあってSMMに遅れてはいるときは、SMMモナークは引き留められ、よってSMMレイテンシが増す。さらに、スタートアップ・プロセッサ間割り込み待ち（WFS: wait for startup interprocessor interrupt）およびTXTスリープ状態のような、SMIイベントが禁止されるある種のアーキテクチャ状態がある。OS/BIOSが一つまたは複数の論理プロセッサをSMI禁止状態に入れる場合、その論理プロセッサは、OS/BIOSが明示的にこの状態から出すまで、SMMにはいらない。SMIイベントは他のすべてのプロセッサをSMMに入れるので、OSはSMIをマスク解除することができない。このシナリオのもとでは、SMMモナークは、SMI禁止されたプロセッサの存在を判別するためには、長いタイムアウトに頼らなければならない。これらのタイムアウトはSMM集結を遅らせ、全体的なSMMレイテンシを増大させるか、SMMイベント処理のために利用可能な時間を減らすかする。

【００２８】

さまざまな実施形態において、たとえいくつかの論理プロセッサが長いフロー中にある場合でも、SMM内部のタイムアウトの必要性が回避できる。そのようなタイムアウトをなくすことは、平均SMMレイテンシを10～20%改善でき、最悪ケースのSMMレイテンシを少なくとも数ミリ秒改善できる。

【００２９】

諸実施形態は、長いフロー中にあるまたはSMI禁止状態にあるプロセッサは共有資源にアクセスする可能性は低いという事実に依拠している。さらに、そのようなプロセッサはSMIを引き起こした可能性は低く、よってその参加はSMI処理のために必要ではない。したがって、SMMモナークは、そのようなプロセッサがSMMにはいる前にSMM処理を進めることができる。

【００３０】

しかしながら、先に進む前に、SMMモナークは、どのプロセッサが長いフロー中にあるおよび／またはSMI禁止状態にあるかを信頼できる仕方で検出できなければならない。長いフロー中またはSMI禁止状態にあってビジーであるプロセッサを検出するために、諸実施形態は、それらの状態についてのインジケータを、たとえばビットマップによって、設けてもよい。ある実施形態では、そのような指標はグローバルに可視である、LONG_FLOW_INDICATION〔長いフロー指標〕およびSMI_INHIBITED_INDICATION〔SMI禁止指標〕と呼ばれる構成設定レジスタを介して提供されることができる。この実施形態では、ソケット内の各論理プロセッサに１ビットが割り当てられることができる。一例として、レジスタは図１のレジスタ１３８によって表現されてもよい。プロセッサ・マイクロコードが長いフローおよびSMI禁止状態への出入りに関わる実装では、マイクロコード／ハードウェアがそれらのレジスタ・ビットの中身を入れることができる。長いフローのいくつかは、５マイクロ秒より長い時間をかけることがあり、したがって、これらの状態にあるプロセッサを待たない能力は、SMMレイテンシの有意な節減を提供できる。将来のプロセッサはSMMマ

イクロコード入場フローについて５ミリ秒を超える時間がかかることがありえ、それ自身が長いフローと考えられることができる。SMMモナークはすべてのプロセッサの説明が付く、すなわちSMMに加わるか長いフローまたはSMI禁止状態にあると報告されるまで待つことができる。そのような判定において支援するために、下記で述べるように、SMRAMに記憶されるビットマップのような一つまたは複数のテーブルが、使用されることができる。

【００３１】

ある実装では、モナーク・プロセッサはその状態を保存し、インジケータ・レジスタのチェックを実行する前にSMMプリアンブル・コードを走らせる。これらのステップは、容易に0.5マイクロ秒より長くかかることがある。この継続時間は、いかなるインフライト割り込み（in-flight interrupt）のための伝搬時間よりずっと長く、コアへのSMI送達とそのインジケータ・レジスタの読み出しの間に競合条件がないことが保証される。遅延がある種の構成のもとでより小さい場合、モナーク・プロセッサは、埋め合わせるために小さな遅延ループを挿入することができる。

【００３２】

ここで図４を参照するに、本発明のもう一つの実施形態に基づく方法の流れ図が示されている。特に、図４は、すべての論理プロセッサがSMM状態において集結する必要がないときの、SMMへの出入りを扱うための流れ図を示している。このようにして、すべての論理プロセッサを待ってからSMM動作を実行することに関わるレイテンシが回避できる。図４で見て取れるように、方法４００は、SMIイベントの生成によって開始されうる（ブロック４１０）。このSMIイベントはすべてのスレッドに伝搬されてもよい。議論の簡単のため、図４のスレッドは単一プロセッサ・ソケットに関してであると想定されていることを注意しておく。ただし、実装は、複数のソケットにまたがってSMMを集結させるために使われることができる。

【００３３】

次に、SMM集結状態にはいる各スレッドについて、SMMインジケータ・マップにおいてインジケータが設定されてもよい（ブロック４２０）。たとえば図３に関して上述した状態保存のような、SMMにはいるためのさまざまな準備動作が先にスレッドによって実行されることができることは理解しておくものとする。SMM集結状態にはいる各スレッドは、SMRAM内に記憶されていてもよいSMMインジケータ・マップにおいて、インジケータをセットしてもよい。ある実施形態では、このマップは、各論理プロセッサがマップのあるビットと関連付けられており、各ソケットの論理プロセッサがマップの異なるセグメントに分離されることのできるビットマップであってもよい。このように、所与のスレッドがSMMにはいるとき、ビットマップにおけるその対応するビットがセットされてもよい。次いで、SMM内部のスレッドの一つが、モナークまたは実行スレッドとして選択されてもよい（ブロック４３０）。さまざまな実施形態において、どのスレッドが実行スレッドとなるかの決定は多様でありうる。たとえば、モナークはあらかじめ選択されていてもよいし（たとえば、ソケット０上の論理プロセッサ０）、あるいは選出機構を介して動的に選択されることもできる。

【００３４】

引き続き図４を参照するに、各スレッドは次いで、該スレッドがモナークとして選択されたかどうかを判定する（菱形４３５）。そうでなければ、そのスレッドはスリープ状態にはいって、モナーク・スレッドが完了を合図するのを待ってもよい（ブロック４７０）。

【００３５】

こうして、制御はモナーク・スレッドのためのブロック４４０に移る。このブロックでは、すべてのスレッドについてACCOUNTED〔説明が付けられた〕状態が決定される。ある実施形態では、この状態は、SMRAM内にあってもよいスレッド存在マップに加えて、さまざまな構成設定レジスタ、SMMインジケータ・マップに基づいていてもよい。この存在マップは、SMMインジケータ・マップと同様のビットマップであってもよく、システム中に存在するスレッドを示すためにSMM初期化の際に設定されてもよい。ある実施形態では、

ブロック４４０における決定は次のようなビットごとのOR演算であってもよい：
OR(LONG_FLOW_INDICATION,SMI_INHIBITED_INDICATION,IN_SMM_INDICATION)
ここで、LONG_FLOW_INDICATIONは、各ビットが対応するスレッドが長いフロー動作中にあるかどうかを示すビット・ベクトルを記憶する状態レジスタから得られる。SMI_INHIBITED_INDICATIONは、各ビットが対応するスレッドがSMI禁止状態にあるかどうかを示すビット・ベクトルを記憶する状態レジスタから得られる。IN_SMM_INDICATIONはSMMインジケータ・マップである。このビットごとのORの結果であるACCOUNTEDは、たとえばSMRAM中のビットマップに記憶されてもよい。この解析後、制御は菱形４５０に移り、ACCOUNTED状態〔説明が付けられたかどうかの状態〕がすべての存在するスレッドについてアクティブであるかどうかが判定されてもよい（菱形４５０）。これは、ACCOUNTED演算の結果と存在マップとの間の比較に基づいて決定できる。もしそうでない場合には、制御はもとのブロック４４０に移る。それ以外の場合には、制御はブロック４５５に移り、SMIイベントが処理されうる。こうして、モナーク・スレッドは所望されるSMMコードを実行しうる。モナーク・スレッドによって実行されるSMMの終結時に、制御はブロック４６０に移る。ブロック４６０では、ACCOUNTED状態およびSMMインジケータ・マップがリセットされてもよい（ブロック４６０）。すなわち、モナーク・スレッドはこれら両方のビットマップにおける値をリセットしてもよい。次いで、モナーク・スレッドは他の論理プロセッサに、SMIから復帰してもよいことを合図してもよい（ブロック４６５）。このようにして、他のスレッドは待ちループから解放される。こうして、ブロック４７５において、すべてのスレッドがSMMから復帰してもよい。図４の実施形態ではこの特定の実装をもって示されているが、本発明の範囲はこの点に関して限定されるものではない。

【００３６】

このように、諸実施形態は、メモリ依存性なしにSMMハンドラ実行を可能にし、信頼性を改善する。この機構は、SMMに付随するパフォーマンスおよびスケーラビリティの問題にも対処する。そのため、SMI処理は、マルチコア／マルチソケット・システムにおけるボトルネックになることを回避できる。このように、諸実施形態は、DRAM依存性をもつSMMコードの実行を回避し、高い可用性使用モデルを可能にする。ここで、SMMコードはメモリ・エラーを診断および訂正する。

【００３７】

諸実施形態はさらに、長いフローまたはSMI禁止状態にある論理プロセッサがあるときに低減したレイテンシをもってSMMにはいることを可能にする。対照的に、現在のところ、SMMコードが一つまたは複数のプロセッサがSMMに遅れて加わるまたはSMM禁止状態にあるかどうかを判定できる信頼できる機構はなく、よって、最大の長いフロー状態よりも大きいタイムアウトが設定される。この解決策は、信頼できず、実装が難しいことに加えて、SMMレイテンシを増大させ、OSリアルタイム応答を低下させるが、本発明の実施形態を使って克服できる。

【００３８】

諸実施形態は、コードにおいて実装されてもよく、システムが命令を実行するようプログラムするために使用できる命令が記憶されている記憶媒体上に記憶されてもよい。記憶媒体は、これに限られないが、フロッピー（登録商標）ディスク、光ディスク、光学式ディスク、固体ドライブ（SSD: solid state drive）、コンパクトディスク読み出し専用メモリ（CD-ROM）、書き換え可能型コンパクトディスク（CD-RW）および光磁気ディスクを含む任意の型のディスク、読み出し専用メモリ（ROM）、動的ランダム・アクセス・メモリ（DRAM）、静的ランダム・アクセス・メモリ（SRAM）のようなランダム・アクセス・メモリ（RAM）、消去可能型プログラム可能読み出し専用メモリ（EPROM）、フラッシュメモリ、電気的に消去可能なプログラム可能読み出し専用メモリ（EEPROM）のような半導体デバイス、磁気もしくは光学式カードまたは電子的な命令を記憶するのに好適な他の任意の型の媒体を含んでいてもよい。

【００３９】

本発明は限られた数の実施形態に関して記述されてきたが、当業者は、それから数多く

の修正および変形を理解するであろう。付属の請求項は、本発明の真の精神および範囲内にはいるそのようなすべての修正および変形をカバーすることが意図されている。

【００４０】

いくつかの態様を記載しておく。

〔態様１〕

命令を実行し、システム管理モード（SMM）にはいるプロセッサ・コア内の論理プロセッサを有する装置であって、SMMにはいる際、前記論理プロセッサは、前記論理プロセッサについての前記プロセッサ・コアの状態記憶に存在しているアクティブな状態を前記論理プロセッサについての前記プロセッサ・コアの記憶ユニットに記憶し、SMMに関連する値を前記状態記憶中に挿入することによってSMM実行環境をセットアップし、

前記論理プロセッサが、システム管理ランダム・アクセス・メモリ（SMRAM）にではなく、前記記憶ユニットに、前記アクティブな状態を記憶できるようにされていることを示すインジケータを記憶する第一の状態レジスタをさらに有する、

装置。

〔態様２〕

前記第一の状態レジスタが、SMM外では変更可能でない、態様１記載の装置。

〔態様３〕

前記プロセッサ・コアが、前記SMRAMに記憶されているSMMコードを実行する、態様１記載の装置。

〔態様４〕

SMMがメモリ・エラーを解決するものであり、前記プロセッサ・コアが不揮発性メモリから復元SMMコードを取得するのであって、前記SMRAMから前記SMMコードを取得するのではない、態様３記載の装置。

〔態様５〕

前記プロセッサ・コアの前記論理プロセッサが長いフローの動作中であることを示すインジケータを記憶する第二の状態レジスタをさらに有しており、長いフローの動作とは少なくとも10^3クロックかかる動作である、態様１記載の装置。

〔態様６〕

前記プロセッサ・コアの前記論理プロセッサがシステム管理割り込み（SMI）禁止状態にあることを示すインジケータを記憶する第三の状態レジスタをさらに有する、態様５記載の装置。

〔態様７〕

SMMにはいった前記プロセッサ・コアの各論理プロセッサの指標を記憶するSMMインジケータ・マップをさらに有する、態様６記載の装置。

〔態様８〕

前記第一、第二および第三の状態レジスタがSMM外では書き込み不能である、態様６記載の装置。

〔態様９〕

命令を実行し、システム管理モード（SMM）にはいるプロセッサ・コア内の論理プロセッサを有する装置であって、SMMにはいる際、前記論理プロセッサは、前記論理プロセッサについての前記プロセッサ・コアの状態記憶に存在しているアクティブな状態を前記論理プロセッサについての前記プロセッサ・コアの記憶ユニットに記憶し、SMMに関連する値を前記状態記憶中に挿入することによってSMM実行環境をセットアップし、

前記プロセッサ・コアが複数の論理プロセッサを含み、

当該方法が、前記プロセッサ・コアのすべての論理プロセッサがSMMはいることなく、SMM動作を実行するモナーク・プロセッサをさらに有する、

装置。

〔態様１０〕

前記モナーク・プロセッサは、前記プロセッサ・コアの論理プロセッサが長いフローの動作中であることを示す第一の状態レジスタ、前記プロセッサ・コアの論理プロセッサが

SMI禁止状態にあることを示す第二の状態レジスタおよびSMMにはいった前記プロセッサ・コアの各論理プロセッサを示すSMMインジケータ・マップにアクセスし、それに基づいて、前記論理プロセッサのすべてが集結を実行することなく、要求されたSMM動作を実行するかどうかを決定し、長いフローの動作とは少なくとも10^3クロックかかる動作である、態様９記載の装置。

〔態様１１〕

前記モナーク・プロセッサは、前記プロセッサ・コアの各論理プロセッサがSMMにはいった、長いフローの動作中であるまたはSMI禁止状態にある場合、前記論理プロセッサのすべてが前記集結を実行したかどうかにかかわりなく、前記要求されたSMM動作を実行する、態様１０記載の装置。

〔態様１２〕

システム管理割り込み（SMI）イベントに応答して、プロセッサ上で実行中のすべてのスレッドがシステム管理モード（SMM）集結状態にはいったかどうかを判定する段階と；

もしまだである場合、残っているスレッドが、長いフローの動作を実行中であるまたはSMI禁止状態にあるかどうかを判定し、もしそうであれば、前記残っているスレッドが前記長いフローの動作を実行するまたは前記SMI禁止状態にある間に、モナーク・スレッドを使って、前記SMIイベントを処理し、そうでなければ、前記残っているスレッドがSMM集結状態にはいるのを待ってから前記SMIイベントを処理する段階とを含み、長いフローの動作とは少なくとも10^3クロックかかる動作である、

方法。

〔態様１３〕

SMM集結状態にはいる各スレッドについて、SMMインジケータ・マップ内のインジケータをセットする段階をさらに含む、態様１２記載の方法。

〔態様１４〕

スレッドが長いフローの動作中であることを示すよう、前記プロセッサの第一の状態レジスタのインジケータを設定する段階と；

スレッドがSMI禁止状態にあることを示すよう、前記プロセッサの第二の状態レジスタのインジケータを設定する段階とをさらに含む、

態様１３記載の方法。

〔態様１５〕

前記判定する処理が、前記SMMインジケータ・マップと、前記第一の状態レジスタと、前記第二の状態レジスタとの間のビットごとのOR演算を実行することを含む、態様１４記載の方法。

〔態様１６〕

スレッドがSMMにはいる際、前記モナーク・スレッドが、前記プロセッサの状態記憶に存在するアクティブな状態を、前記プロセッサの記憶ユニット中に記憶し、SMM状態を前記状態記憶に記憶する、態様１２記載の方法。

〔態様１７〕

第一のプロセッサと、第二のプロセッサと、動的ランダム・アクセス・メモリ（DRAM）とを有するシステムであって：

前記第一のプロセッサは、命令を実行しシステム管理モード（SMM）にはいる第一のコアと、前記第一のコアで実行されているスレッドが長いフローの動作中であるかどうかを示す第一のインジケータと、前記スレッドがシステム管理割り込み（SMI）禁止状態にあるかどうかを示す第二のインジケータと、記憶ユニットとを有し、SMMにはいる際、前記第一のコアは前記第一のコアの状態記憶に存在するアクティブな状態を前記記憶ユニット中に記憶し、SMM実行状態を前記状態記憶中に記憶し、前記記憶ユニットはSMMの間前記アクティブな状態を記憶するのに専用であり、

前記第二のプロセッサは、命令を実行しSMMにはいる第二のコアと、前記第二のコアで実行されている第二のスレッドが長いフローの動作中であるかどうかを示す第一のインジケータと、前記第二のスレッドがSMI禁止状態にあるかどうかを示す第二のインジケータ

と、第二の記憶ユニットとを有し、SMMにはいる際、前記第二のコアは前記第二のコアの状態記憶に存在するアクティブな状態を前記第二の記憶ユニット中に記憶し、SMM実行状態を前記状態記憶中に記憶し、前記第二の記憶ユニットはSMMの間前記アクティブな状態を記憶するのに専用であり、

前記DRAMは前記第一および第二のプロセッサに結合されており、前記DRAMの一部は当該システムのためのシステム管理ランダム・アクセス・メモリ（SMRAM）であり、

長いフローの動作とは少なくとも10^3クロックかかる動作である、

システム。

〔態様１８〕

前記DRAMが非一様メモリ・アーキテクチャ（NUMA）において結合されており、前記第二のプロセッサは前記DRAMと前記第一のプロセッサを通じて通信する、態様１７記載のシステム。

〔態様１９〕

前記第二のプロセッサが、SMI信号に応答して前記SMRAMに前記アクティブな状態を記憶せず、その代わり、前記アクティブな状態を前記第二の記憶ユニットに記憶する、態様１７記載のシステム。

〔態様２０〕

前記第一のプロセッサおよび前記第二のプロセッサがそれぞれ少なくとも一つの論理プロセッサを含み、前記第一および第二のプロセッサの複数の論理プロセッサがSMMにはいったが前記第一および第二のプロセッサの少なくとも一つの論理プロセッサがSMMにはいっていない状態後にSMM動作を実行するモナーク・プロセッサを含む、態様１７記載のシステム。

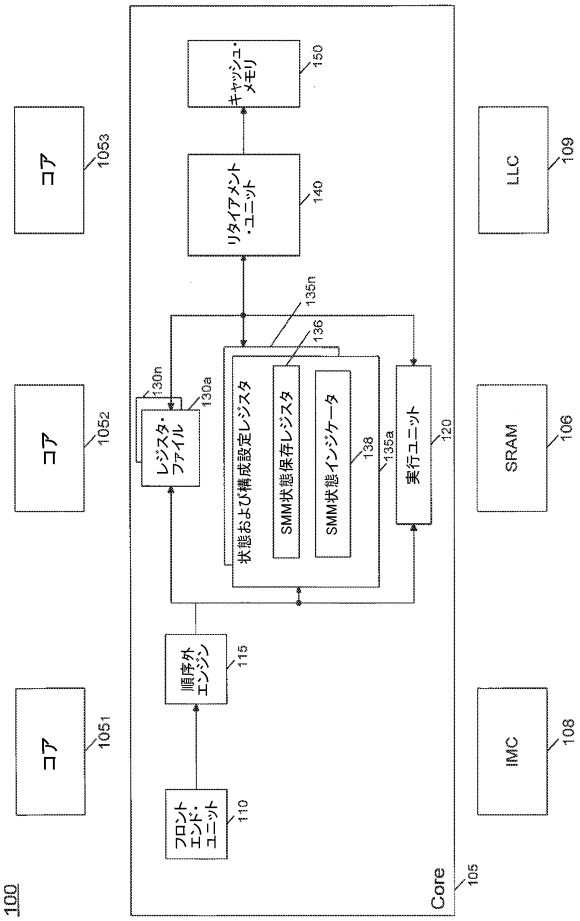
〔態様２１〕

前記少なくとも一つの論理プロセッサが長いフローの動作中であるまたはSMI禁止状態にある、態様２０記載のシステム。

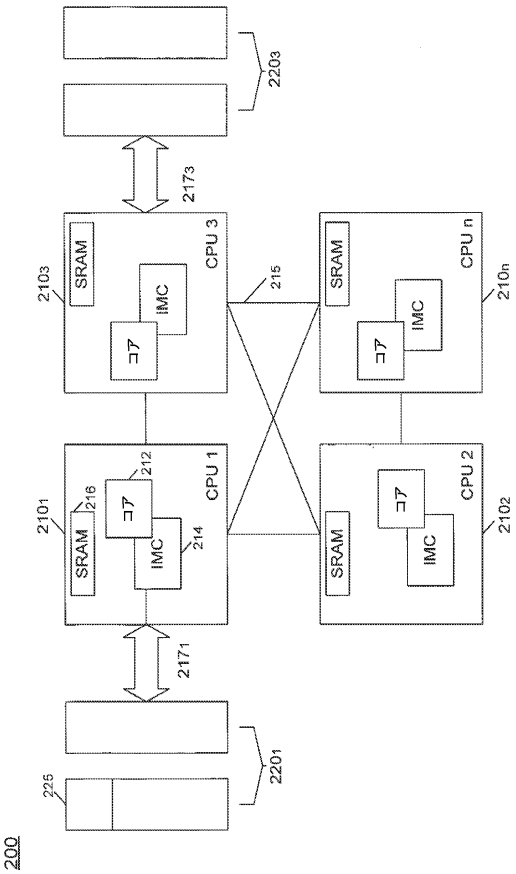
〔態様２２〕

態様２０記載のシステムであって、前記モナーク・プロセッサが、前記第一のプロセッサのいずれかの論理プロセッサが長いフローの動作中であるかどうかを示す第一のビットマップ、前記第一のプロセッサのいずれかの論理プロセッサがSMI禁止状態にあるかどうかを示す第二のビットマップおよび前記第一のプロセッサの各論理プロセッサがSMM集結状態にはいったかどうかを示す第三のビットマップにアクセスし、それに基づいて、前記少なくとも一つの論理プロセッサがSMM集結状態にはいっていないときにSMM集結状態動作を実行するかどうかを決定する、システム。

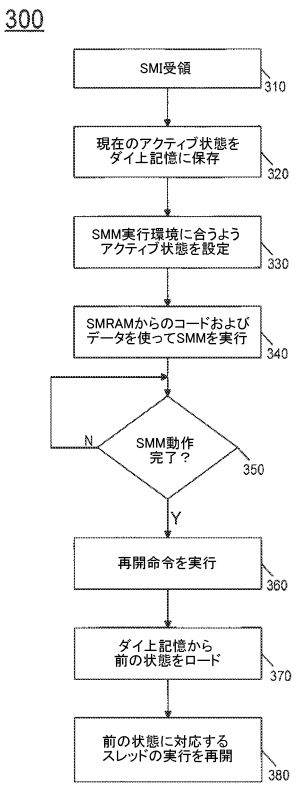
【図１】


100
Core
105
コア
1053
コア
1052
コア
1051
キャッシュ・メモリ
150
リタイアメント・ユニット
140
レジスタ・ファイル
130a
130n
状態および構成設定レジスタ
135n
SMM状態保存レジスタ
136
SMM状態インジケータ
138
135a
実行ユニット
120
順序外エンジン
115
フロント・エンド・ユニット
110
LLC
109
SRAM
106
IMC
108

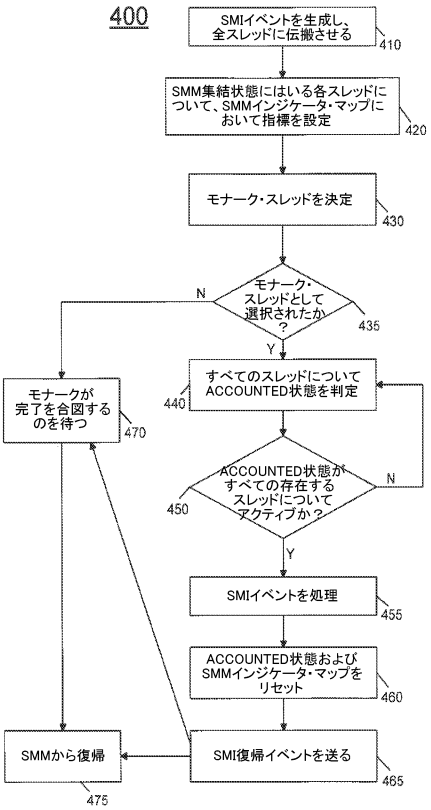
【図２】


200
2203
2173
CPU 3
2103
SRAM
IMC
コア
215
CPU n
210n
CPU 1
2101
216
212
214
CPU 2
2102
2171
2201
225

【図３】


300
SMI受領
310
現在のアクティブ状態をダイ上記憶に保存
320
SMM実行環境に合うようアクティブ状態を設定
330
SMRAMからのコードおよびデータを使ってSMMを実行
340
SMM動作完了？
350
N
Y
再開命令を実行
360
ダイ上記憶から前の状態をロード
370
前の状態に対応するスレッドの実行を再開
380

【図４】


400
SMIイベントを生成し、全スレッドに伝搬させる
410
SMM集結状態にはいる各スレッドについて、SMMインジケータ・マップにおいて指標を設定
420
モナーク・スレッドを決定
430
モナーク・スレッドとして選択されたか？
435
N
Y
すべてのスレッドについてACCOUNTED状態を判定
440
ACCOUNTED状態がすべての存在するスレッドについてアクティブか？
450
N
Y
SMIイベントを処理
455
ACCOUNTED状態およびSMMインジケータ・マップをリセット
460
SMI復帰イベントを送る
465
SMMから復帰
475
モナークが完了を合図するのを待つ
470

フロントページの続き

(72)発明者 ガネサン, バスカラン
インド国 ５６００３７ カルナータカ州 バンガロール エアポート ロード １３６
(72)発明者 ランガラジャン, タヌナタン
インド国 ５６００４３ バンガロール カンマナハリ メイン ロード ジャル ヴァユ ヴィハール エー－８９
(72)発明者 クマール, モーハン, ジェイ.
アメリカ合衆国 ９７００７ オレゴン州 アロハ サウスウェスト マルコ レーン １８６８０
(72)発明者 ドシ, ガウタム, ビー.
インド国 ５６０００８ カルナータカ州 バンガロール エアポート ロード ダイアモンド ディストリクト オー－７３
(72)発明者 パルタサラティ, ラジェシュ, エス.
アメリカ合衆国 ９７１２４ オレゴン州 ヒルズボロ ノースイースト クリークスエッジ ドライヴ １２０９
(72)発明者 ダッタ, シャンマンナ, エム.
アメリカ合衆国 ９７１２４ オレゴン州 ヒルズボロ ノースイースト レノックス ストリート ５３２
(72)発明者 ビンズ, フランク
アメリカ合衆国 ９７２２９ オレゴン州 ポートランド ノースウェスト ピナクル ドライヴ ２４２０
(72)発明者 ムルティ, ラジェシュ ナガラジャ
インド国 ５６００３４ カルナータカ州 バンガロール エイチエスアール レイアウト セクター － アイアイ ２２ンド “ビー” メイン
(72)発明者 スワンソン, ロバート, シー.
アメリカ合衆国 ９８５１６ ワシントン州 オリンピア グレイホーク レーン ７１４２

審査官 篠塚 隆

(56)参考文献 米国特許出願公開第２００５／００８６４０５（ＵＳ，Ａ１）

(58)調査した分野(Int.Cl., ＤＢ名)
Ｇ０６Ｆ ９／４６ － ９／５４
Ｇ０６Ｆ １３／２４
Ｇ０６Ｆ １５／１６ －１５／１７７