(19)日本国特許庁(JP)　(12)公表特許公報(A)　(11)特許出願公表番号
特表2005-535005
(P2005-535005A)
(43)公表日　平成17年11月17日(2005.11.17)

(51)Int.Cl.[7]	F I		テーマコード（参考）
G06F 12/14	G06F 12/14	510D	5B017
G06F 1/00	G06F 12/14	560C	
	G06F 1/00	370E	

審査請求 有　予備審査請求 未請求　（全 20 頁）

(21)出願番号	特願2003-582616 (P2003-582616)
(86)(22)出願日	平成15年3月20日 (2003.3.20)
(85)翻訳文提出日	平成16年11月26日 (2004.11.26)
(86)国際出願番号	PCT/US2003/008762
(87)国際公開番号	WO2003/085497
(87)国際公開日	平成15年10月16日 (2003.10.16)
(31)優先権主張番号	10/112,169
(32)優先日	平成14年3月29日 (2002.3.29)
(33)優先権主張国	米国 (US)

(71)出願人　593096712
インテル　コーポレイション
アメリカ合衆国　９５０５２　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　100070150
弁理士　伊東　忠彦
(74)代理人　100091214
弁理士　大貫　進介
(74)代理人　100107766
弁理士　伊東　忠重
(72)発明者　サットン，ジェイムズ，セカンド
アメリカ合衆国　９７２２９　オレゴン州　ポートランド　ノースウエスト　ポーリナ　ドライヴ　２０２０５

最終頁に続く

(54)【発明の名称】安全な環境を初期化する命令を実行するシステムおよび方法

(57)【要約】
マイクロプロセッサシステムにおいてセキュアオペレーションを起動する方法と装置が記載されている。一実施形態において、一の起動する論理プロセッサは、他の論理プロセッサの実行を停止し、メモリに初期化およびセキュアバーチャルマシンモニターソフトウェアをロードすることによりプロセスを起動する。起動するプロセッサは、認証と実行のために初期化ソフトウェアをセキュアメモリにロードする。初期化ソフトウェアは、セキュアシステムオペレーションの前に、そのセキュアバーチャルマシンモニターソフトウェアを認証および登録する。

【特許請求の範囲】
【請求項１】
セキュアードエンター命令を実行するセキュアメモリを含む第１の論理プロセッサと、
非プロセッサデバイスによるセキュアードバーチャルマシンモニターへのアクセスを防止するチップセットとを有することを特徴とするシステム。
【請求項２】
請求項１に記載のシステムであって、前記セキュアードエンター命令は、セキュアオペレーションにおいて、第２の論理プロセッサを前記第１の論理プロセッサと同期させるために、前記第１の論理プロセッサに前記第２の論理プロセッサへ特殊なバスメッセージを発行させることを特徴とするシステム。
【請求項３】
請求項１に記載のシステムであって、前記セキュアメモリは前記第１の論理プロセッサのキャッシュ内にあることを特徴とするシステム。
【請求項４】
請求項１に記載のシステムであって、前記セキュアメモリは前記第１の論理プロセッサ以外の回路によるアクセスから保護されていることを特徴とするシステム。
【請求項５】
請求項１に記載のシステムであって、ダイジェストを記憶するためにプラットフォーム構成レジスタを含むセキュリティトークンをさらに有することを特徴とするシステム。
【請求項６】
請求項１に記載のシステムであって、前記セキュアードエンター命令からの第１の特殊なバスメッセージに応答する第２の論理プロセッサをさらに有することを特徴とするシステム。
【請求項７】
請求項６に記載のシステムであって、前記第２の論理プロセッサは、前記第１の特殊なバスメッセージに応じて、カレント命令の実行を終了し、第２の特殊なバスメッセージを発行することを特徴とするシステム。
【請求項８】
請求項７に記載のシステムであって、前記チップセットは前記第２の特殊なバスメッセージの受信に応じてフラグを設定することを特徴とするシステム。
【請求項９】
請求項８に記載のシステムであって、前記第２の論理プロセッサは、第３の特殊なバスメッセージに応じて、前記セキュアバーチャルマシンモニターのエントリポイントにジャンプすることを特徴とするシステム。
【請求項１０】
第１の論理プロセッサと第２の論理プロセッサとを同期させるステップと、
初期化コードモジュールを認証するステップと、
セキュアバーチャルマシンモニターを認証するステップと、
前記セキュアバーチャルマシンモニターを実行するステップとを有することを特徴とする方法。
【請求項１１】
請求項１０に記載の方法であって、特殊なバスメッセージに応じて前記第２の論理プロセッサ上で前記セキュアバーチャルマシンモニターを実行するために前記特殊なバスメッセージを前記第２の論理プロセッサに送信するステップをさらに有することを特徴とする方法。
【請求項１２】
請求項１０に記載の方法であって、前記同期させるステップは、特殊なバスメッセージが前記第２の論理プロセッサに実行を停止させアクノレッジメントを送信させるステップをさらに有することを特徴とする方法。
【請求項１３】

請求項１２に記載の方法であって、前記同期させるステップは、前記アクノレッジメントに応じてチップセットにフラグを設定するステップをさらに有することを特徴とする方法。

【請求項１４】

請求項１０に記載の方法であって、前記初期化コードモジュールを認証するステップは、前記初期化コードモジュールのコピーと公開キーとをセキュアメモリに移動するステップを有することを特徴とする方法。

【請求項１５】

請求項１４に記載の方法であって、前記初期コードモジュールを認証するステップは、前記初期コードモジュール第１のダイジェストを前記初期コードモジュールの第２のダイジェストと比較するステップを含むことを特徴とする方法。

【請求項１６】

請求項１０に記載の方法であって、前記セキュアバーチャルマシンモニターを認証するステップは、前記初期コードモジュールを実行するステップを含むことを特徴とする方法。

【請求項１７】

請求項１６に記載の方法であって、前記セキュアバーチャルマシンモニターを認証するステップは、プラットフォーム構成レジスタに前記バーチャルマシンモニターを登録するステップを含むことを特徴とする方法。

【請求項１８】

第１の論理プロセッサと第２の論理プロセッサとを同期させる手段と、

初期化コードモジュールを認証する手段と、

セキュアバーチャルマシンモニターを認証する手段と、

前記セキュアバーチャルマシンモニターを実行する手段とを有することを特徴とする装置。

【請求項１９】

請求項１８に記載の装置であって、特殊なバスメッセージに応じて前記第２の論理プロセッサ上で前記セキュアバーチャルマシンモニターを実行するために前記特殊なバスメッセージを前記第２の論理プロセッサに送信する手段をさらに有することを特徴とする装置。

【請求項２０】

請求項１８に記載の装置であって、前記初期化コードモジュールのコピーと公開キーとをセキュアメモリに移動する手段を有することを特徴とする装置。

【請求項２１】

請求項２０に記載の装置であって、前記初期コードモジュール第１のダイジェストを前記初期コードモジュールの第２のダイジェストと比較する手段をさらに有することを特徴とする装置。

【請求項２２】

請求項１８に記載の装置であって、前記バーチャルマシンモニターを登録する手段をさらに有することを特徴とする装置。

【請求項２３】

セキュアオペレーション初期化をする第１の命令を実行し、セキュア初期化認証済みコードの実行をする時点を検出するセキュアエンターロジックと、

前記第１の命令に応じて第１の特殊なバスメッセージを送信し、前記検出された時点に第２の特殊なバスメッセージを送信するバスメッセージングロジックとを有することを特徴とするプロセッサ。

【請求項２４】

請求項２３に記載のプロセッサであって、前記時点は第１の論理プロセッサがアクノレッジメントを発行した後であることを特徴とするプロセッサ。

【請求項２５】

請求項２３に記載のプロセッサであって、前記セキュアエンターロジックは、前記時点を決定するためにチップセット中のフラグレジスタをさらに参照することを特徴とするプロセッサ。

【請求項２６】

請求項２３に記載のプロセッサであって、前記セキュアエンターロジックは、さらに、前記時点の後にキーを入力しコードモジュールを認証することを特徴とするプロセッサ。

【請求項２７】

請求項２３に記載のプロセッサであって、前記バスメッセージングロジックは、さらに、コードエントリポイントを含む第３の特殊なバスメッセージを送信することを特徴とするプロセッサ。

【請求項２８】

第１の論理プロセッサからの第１の特殊なバスメッセージに応じてセキュアオペレーションを準備するバスメッセージングロジックと、

前記第１の特殊なバスメッセージに応じて第２の論理プロセッサからのアクノレッジメントを記憶するレジスタとを有することを特徴とするチップセット。

【請求項２９】

請求項２８に記載のチップセットであって、前記チップセットは、前記第１の論理プロセッサにセキュアオペレーション初期化を進める信号をいつ送るかを決定するために、前記レジスタを論理プロセッサアクティビティと比較することを特徴とするチップセット。

【請求項３０】

請求項２９に記載のチップセットであって、前記信号はフラグの設定を含むことを特徴とするチップセット。

【請求項３１】

請求項２８に記載のチップセットであって、セキュアバーチャルマシンモニターをロックするデバイスアクセスロジックをさらに有することを特徴とするチップセット。

【請求項３２】

請求項２８に記載のチップセットであって、前記第１の特殊なバスメッセージの後に、前記第１の論理プロセッサにキーを送信するキーレジスタをさらに有することを特徴とするチップセット。

【請求項３３】

セキュアエンターロジックと、前記セキュアエンターロジックに応じる第１のバスメッセージングロジックとを有する論理プロセッサと、

前記第１のバスメッセージングロジックから第１の特殊なバスメッセージを受信する第２のバスメッセージングロジックと、アクノレッジメントに応じて設定されるフラグとを有するチップセットとを有することを特徴とするシステム。

【請求項３４】

請求項３３に記載のシステムであって、前記セキュアエンターロジックに応じてセキュアオペレーションを起動するセキュア初期化認証済みコードをさらに有することを特徴とするシステム。

【請求項３５】

請求項３４に記載のシステムであって、前記セキュア初期化認証済みコードを認証するするために前記論理プロセッサにより使用されるキーをさらに有することを特徴とするシステム。

【請求項３６】

請求項３４に記載のシステムであって、前記第１のバスメッセージングロジックは第２の特殊なバスメッセージを発行し、前記第２の特殊なバスメッセージの後に前記論理プロセッサは前記セキュア初期化認証済みコードをセキュアメモリに移動することを特徴とするシステム。

【請求項３７】

請求項３４に記載のシステムであって、セキュアバーチャルマシンモニターをさらに有

することを特徴とするシステム。
【請求項３８】
請求項３７に記載のシステムであって、前記セキュア初期化認証済みコードは前記セキュアバーチャルマシンモニターの初期化を実行することを特徴とするシステム。
【請求項３９】
請求項３８に記載のシステムであって、前記初期化は認証を含み、前記チップセットは前記初期化に応じて非プロセッサが前記セキュアバーチャルマシンモニターにアクセスするのを防止するデバイスアクセスロジックを含むことを特徴とするシステム。
【請求項４０】
請求項３８に記載のシステムであって、前記第１のバスメッセージングロジックは、前記初期化に応じて、第３の特殊なバスメッセージを発行することを特徴とするシステム。
【請求項４１】
請求項４０に記載のシステムであって、前記第３の特殊なバスメッセージは、前記セキュアバーチャルマシンモニターのコードエントリポイントを含むことを特徴とするシステム。
【請求項４２】
特殊なバスメッセージを送信するステップと、
第１の論理プロセッサ内の初期化コードを認証するステップと、
セキュアバーチャルマシンモニターを認証するステップと、
前記第１の論理プロセッサ中の前記セキュアバーチャルマシンモニターを実行するステップとを有することを特徴とする方法。
【請求項４３】
請求項４２に記載の方法であって、前記第１のバスメッセージに応じてアクノレッジメントを送信するステップをさらに有することを特徴とする方法。
【請求項４４】
請求項４２に記載の方法であって、第２の論理プロセッサにおける実行を停止してアクノレッジメントを送信するステップをさらに有することを特徴とする方法。
【請求項４５】
請求項４４に記載の方法であって、前記アクノレッジメントに応じてチップセットにフラグを設定するステップをさらに有することを特徴とする方法。
【請求項４６】
請求項４２に記載の方法であって、前記初期化コードを認証するステップは、前記初期化コードのコピーと公開キーをセキュアメモリに移動することを特徴とする方法。
【請求項４７】
請求項４６に記載の方法であって、前記初期化コードを認証するステップは、前記初期化コードの第１のダイジェストを前記初期化コードの第２のダイジェストと比較するステップを含むことを特徴とする方法。
【請求項４８】
請求項４２に記載の方法であって、前記セキュアバーチャルマシンモニターを認証するステップは、前記初期化コードを実行するステップを含むことを特徴とする方法。
【請求項４９】
請求項４８に記載の方法であって、前記セキュアバーチャルマシンモニターを認証するステップは、プラットフォーム構成レジスタに前記バーチャルマシンモニターを登録するステップを含むことを特徴とする方法。

【発明の詳細な説明】
【技術分野】

【０００１】
本発明は、マイクロプロセッサシステムに関し、特に信頼できる（トラステッド）安全な（セキュア）環境において動作するマイクロプロセッサシステムに関する。
【背景技術】
【０００２】
ローカルまたはリモートマイクロコンピュータ上で実行される金融上のおよび個人的取引の数が増加するにつれ、「信頼できる」または「安全な」マイクロプロセッサ環境の確立が促進されつつある。これらの環境が解決しようとする問題は、プライバシーの喪失や、データが汚染されたり乱用されたりする問題である。ユーザは自分のプライベートなデータを公にされたくはない。また、自分のデータが改変されたり不適切な取引で使用されることも望まない。これには、例えば、診療記録の意図しない公開、オンライン銀行等のファンドの電子的窃盗などである。同様に、コンテントプロバイダは、デジタルコンテント（例えば、音楽、オーディオ、ビデオ、その他のタイプの一般的データ）を権限なく複製されることを防止しようとしている。
【０００３】
既存の信頼できるシステムは、完全に閉じた信頼できるソフトウェアのセットを用いているかもしれない。この方法の実装は比較的簡単であるが、一般的商業的に入手可能なオペレーティングシステムやアプリケーションソフトウェアを同時に使用できないという短所がある。この短所により、上記の信頼できるシステムが受け入れられないことがある。
【０００４】
本発明は、限定としてではなく実施例として添付した図面に例示されている。その添付した図面においては、同様の構成要素には類似した参照数字が付されている。
【発明を実施するための最良の形態】
【０００５】
マイクロプロセッサシステムにおいて信頼できる、または安全な環境を初期化するための技術を以下に説明する。以下の説明において、本発明を完全に理解してもらうため、論理実装、ソフトウェアモジュールアロケーション、暗号化方法、バスシグナリング方法、動作の詳細等、多数の具体的詳細を説明する。しかし、本発明はこれらの具体的な詳細なしに実施してもよいことが、当業者には分かるであろう。他の例において、本発明を不明瞭としないように、制御構造、ゲートレベルの回路、ソフトウェア命令シーケンスの全体は詳細には示さなかった。当業者であれば、過剰な説明なしに、ここに含めた説明で、適切な機能を実施することができるであろう。本発明はマイクロプロセッサシステムの形式で開示されている。しかし、本発明は、デジタル信号プロセッサ、ミニコンピュータ、メインフレームコンピュータ等、他の形式のプロセッサで実施してもよい。
【０００６】
図１を参照して、マイクロプロセッサシステムで実行されるソフトウェア環境の実施例の図が示されている。図１に示したソフトウェアは信頼されていない。高い権限レベルで動作しているとき、オペレーティングシステム１５０はサイズが大きく、頻繁に更新しなければならないので、ちょうどよいときに信頼分析を実行することは非常に困難である。多くのオペレーティングシステムは最上位の権限レベルである権限リング０にある。アプリケーション１５２、１５４、および１５６は、より下の権限を有し、一般的には権限リング３にある。異なる権限リングの存在と、オペレーティングシステム１５０とアプリケーション１５２、１５４、１５６が異なる権限リングに分かれていることにより、オペレーティングシステム１５０により提供された便宜を信頼する決定をすることに基づき、図１のソフトウェアが信頼できるモードで動作することが可能であると見えるかもしれない。しかし、実際には、そのように信頼できるとの決定をすることはしばしば現実的ではない。この問題に影響する要因には、オペレーティングシステム１５０のサイズ（コードのライン数）、オペレーティングシステム１５０が（新しいコードモジュールやパッチにより）何度も更新されるという事実、およびオペレーティングシステムがそのデベロッパ以外の第三者により提供されるデバイスドライバのようなコードモジュールも含むという事

実が含まれる。オペレーティングシステム１５０は、マイクロソフト（登録商標）ウィンドウズ（登録商標）、リナックス、ソラリス（登録商標）のような一般的なものであってもよいし、既知のあるいは入手可能な他のいかなるオペレーティングシステムであってもよい。どのタイプまたは名称のアプリケーションまたはオペレーティングシステムであるかは、重要ではない。

【０００７】

図２を参照して、本発明の一実施形態による、トラステッド（信頼できる）、またはセキュア（安全）なソフトウェアモジュールとシステム環境２００の実施例の図が示されている。図２に示した実施形態において、プロセッサ２０２、プロセッサ２１２、プロセッサ２２２、および任意的な他のプロセッサ（図示せず）は、別々のハードウェア実体として示されている。他の実施形態において、様々な構成要素と機能ユニットの境界が異なってもよいように、プロセッサの数が異なってもよい。一部の実施形態において、プロセッサは、１以上のより物理的なプロセッサ上で実行されている別々のハードウェア実行スレッドあるいは「論理プロセッサ」により置き換えられてもよい。

【０００８】

プロセッサ２０２、２１２、２２２は、安全な、または信頼できる動作をサポートする特殊回路または論理要素を含んでもよい。例えば、プロセッサ２０２は、信頼できる動作を起動する特殊なセキュアエンター（SENTER）命令の実行をサポートするSENTERロジック２０４を含んでいてもよい。プロセッサ２０２は、特殊なSENTER動作をサポートするシステムバス２３０上の特殊なバスメッセージをサポートするバスメッセージロジック２０６を含んでもよい。別の実施形態において、チップセット２４０のメモリ制御機能はプロセッサ内の回路に割り当てられてもよく、複数プロセッサの場合、単一のダイに含まれてもよい。これらの実施形態において、特殊なバスメッセージもプロセッサ内のバスに送られてもよい。特殊なバスメッセージの使用により、いくつかの理由により、システムの安全性と信頼性を高めることができる。プロセッサ２０２、２１２、２２２またはチップセット２４０等の回路要素は、本開示の実施形態の適当な論理要素を含んでいるとき、このようなメッセージを発行のみまたは応答のみすることができる。それゆえ、その特殊なバスメッセージを成功裏に交換することにより、適当なシステム構成を確実にすることができる。特殊なバスメッセージによりプラットフォームコンフィギュレーションレジスタ２７８をリセットする等の、通常は禁止されるべき動作も許可されてもよい。特殊なバスメッセージが特殊なセキュリティー命令に応じてのみ発行されることを可能とすることにより、バストランザクションでスパイする潜在的に敵対的な信頼できないコードの能力が削減される。

【０００９】

また、プロセッサ２０２は、セキュアーな初期化動作をサポートするセキュアーメモリ２０８を含んでもよい。一実施形態において、セキュアーメモリ２０８は、特殊モードで動作しているであろう、プロセッサ２０２の内部キャッシュであってもよい。別の実施形態において、セキュアーメモリ２０８は特殊なメモリであってもよい。プロセッサ２１２やプロセッサ２２２等の他のプロセッサも、SENTERロジック２１４、２２４、バスメッセージロジック２１６、２２６、セキュアーメモリ２１８、２２８を含んでもよい。

【００１０】

「チップセット」は、接続されたプロセッサのためにメモリや入出力動作をサポートする一群の回路またはロジックとして定義されてもよい。チップセットの個々の要素は、単一のチップにまとめられてもよいし、一組のチップにまとめられてもよいし、プロセッサを含む複数のチップに分散していてもよい。図２の実施形態において、チップセット２４０は、プロセッサ２０２、２１２、２２２をサポートするためのメモリおよび入出力動作をサポートする回路およびロジックを含んでもよい。一実施形態において、チップセット２４０は、多数のメモリページ２５０－２６２、非プロセッサデバイスがメモリページ２５０－２６２にアクセスしてもよいかどうかを示す制御情報を含むデバイスアクセスページテーブル２４８とインターフェイスしていてもよい。チップセット２４０は、メモリペ

ージ２５０－２６２の選択された部分に入出力デバイスからのダイレクトメモリアクセス（ＤＭＡ）を許可するまたは拒否するデバイスアクセスロジック２４７を含んでいてもよい。一部の実施形態において、デバイスアクセスロジック２４７は、上記のアクセスを許可または拒否するのに要するすべての関連する情報を含んでもよい。他の実施形態において、デバイスアクセスロジック２４７は、デバイスアクセスページテーブル２４８内に保持された上記の情報にアクセスしてもよい。メモリページの実際の数は重要ではなく、システム要求に応じて変化する。他の実施形態において、メモリアクセス機能はチップセット２４０の外部にあってもよい。別の実施形態において、チップセット２４０の機能は１以上の物理的デバイス間にさらに割り当てられていてもよい。

【００１１】

チップセット２４０は、特殊なSENTER動作をサポートするシステムバス２３０上の特殊なバスメッセージをサポートするために、自分のバスメッセージロジック２４２を追加的に含んでいてもよい。これらの特殊なバスメッセージは、キーレジスタ２４４の内容をプロセッサ２０２、２１２、または２２２に転送すること、またはプロセッサ２０２、２１２、または２２２により特殊なALL-JOINEDフラグ２７４が調べられることを許可することを含んでもよい。バスメッセージロジック２４２の付加的機能は、プロセッサによるバスアクティビティをEXISTSレジスタ２７２に登録し、プロセッサによる特殊なバスメッセージアクティビティをJOINSレジスタ２７２に記憶することである。EXISTSレジスタ２７２とJOINSレジスタ２７２の内容が等しいとき、特殊なALL_JOINEDフラグ２７４が設定され、システム中のすべてのプロセッサがセキュアーエンタープロセスに参加していることを示す。

【００１２】

チップセット２４０は、ペリフェラルコンポーネントインターコネクト（PCI）、アクセラレーテッドグラフィクスポート（AGP）、ユニバーサルシリアルバス（USB）、ローピンカウント（LPC）バス、またはその他の種類の入出力バス（図示せず）等の入出力バス上の標準的入出力動作をサポートする。インターフェイス２９０は、チップセット２４０を１以上のプラットフォームコンフィグレーションレジスタ（PCR）２７８、２７９を含むトークン２７６と接続するために用いられる。一実施形態において、インターフェイス２９０は、セキュリティ拡張を追加して変更されたLPCバス（ローピンカウント（LPC）インターフェイス仕様書、インテルコーポレイション、Rev.1.0、1997年12月29日）であってもよい。上記セキュリティ拡張の一実施例は、トークン２７６内のプラットフォームコンフィギュレーションレジスタ（PCR）２７８をターゲットとする以前に予約されたメッセージヘッダとアドレス情報を用いる、ローカルの確認メッセージである。一実施形態において、トークン２７６は、特殊なセキュリティ機能を有してもよく、一実施形態において、トラステッドコンピューティングプラットフォームアライアンス（TCPA）により発行された、2001年12月1日の、TCPA主仕様書バージョン1.1a（本願の出願時点においてwww.trustedpc.comから入手可能）に開示されたトラステッドプラットフォームモジュール（TPM）２８１を含んでもよい。

【００１３】

システム環境２００で識別された２つのソフトウェアコンポーネントが、セキュアバーチャルマシンモニター（SVMM）２８２モジュールとセキュア初期化認証コード（SINIT-AC）２８０モジュールである。SVMM２８２モジュールは、システムディスクまたは他の大容量記憶に記憶されてもよく、必要に応じて移動またはコピーされてもよい。一実施形態において、セキュア起動プロセスの開始の前に、SVMM２８２は１以上のメモリページ２５０－２６２に移動またはコピーされてもよい。セキュアエンタープロセスの後、SVMM282がシステム内の最上位の権限を与えられたコードとして動作するバーチャルマシン環境が生成されてもよく、生成されたバーチャルマシン内でオペレーティングシステムまたはアプリケーションによりシステムリソースへの直接アクセスを許可または拒否するために用いられてもよい。

【００１４】

セキュアエンタープロセスにより要求される動作の一部は単純なハードウェア実装の範囲を超えており、実行が黙示的に信頼されたソフトウェアモジュールを有利に用いてもよい。一実施形態において、これらの動作はセキュア初期化（SINIT）コードにより実行されてもよい。３つの動作がここでは識別されているが、これらの動作は限定的に捉えてはならない。１つの動作は、システム構成の重要な部分を表すさまざまなコントロールが、その構成がセキュアード環境の正しい裏付けをサポートすることを確からしめるためにテストされることを要求する。一実施形態において、１つの要求されるテストは、２以上の異なるシステムバスアドレスがメモリページ２５０－２６２内の同じ位置をアドレスすることを、チップセット２４０により提供されるメモリコントローラの構成が許さないことである。第２の動作は、SVMM２８２のメモリ常駐コピーにより使用されるそれらのメモリページを非プロセッサデバイスによる干渉から保護するために、デバイスアクセスページテーブル２４８とデバイスアクセスロジック２４７を構成することであろう。第３の動作は、SVMM２８２モジュールのアイデンティティを計算および登録し、それにシステムコントロールを転送することである。ここで、「登録」とは、SVMM２８２のトラスト計測をレジスタ、例えばPCR２７８またはPCR２７９に置くことを意味する。この最後の動作が行われると、SVMM２８２の信頼性が潜在的システムユーザにより検査されてもよい。

【００１５】

SINITコードは、プロセッサまたはチップセットの製造者により作られてもよい。この理由により、SINITコードは、チップセット２４０のセキュア起動を手伝うため信頼されてもよい。SINITコードを配布するために、一実施形態において、SINITコード全体から周知の暗号ハッシュを作り、「ダイジェスト」として知られた値を生成してもよい。一実施形態によると、１６０ビットの値をもつダイジェストが作られる。ダイジェストは、ダイジェストシグネチャを形成するため秘密キーにより暗号化されてもよい。一実施形態によると、その秘密キーはプロセッサの製造者が保持していてもよい。SINITコードが対応するデジタル署名とバンドルされているとき、その組み合わせはSINIT認証済みコード（SINIT-AC）２８０と呼ばれることもある。SINIT-AC２８０のコピーは、下で説明するように、後で確認されてもよい。

【００１６】

SINIT-AC２８０は、システムディスク、その他の大規模記憶装置、または固定メディアに記憶され、必要に応じて他の場所へ移動またはコピーされてもよい。一実施形態において、セキュア起動プロセスを開始する前に、SINIT-ACは、メモリ常駐コピーを形成するため、メモリページ２５０－２６２に移動またはコピーされる。

【００１７】

いずれの論理プロセッサがセキュア起動プロセスを起動してもよく、その後は起動する論理プロセッサ（ILP）と呼ばれてもよい。システムバス２３０上のプロセッサのいずれもがILPになることはできるが、本実施例においてはプロセッサ２０２がILPになるものとする。他の理由もあるが、他のプロセッサまたはDMAデバイスがメモリページ２５０－２６２をオーバーライトすることがあるので、この時点では、SINIT-AC２８０のメモリ常駐コピーもSVMM２８２のメモリ常駐コピーもいずれも信頼できるとは考えられないかもしれない。

【００１８】

その後、ILP（プロセッサ２０２）は、特殊な命令を実行する。この特殊な命令はセキュアードエンター（SENTER）命令と呼ばれ、SENTERロジック２０４によりサポートされる。SENTER命令を実行すると、ILP（プロセッサ２０２）がシステムバス２３０に特殊なバスメッセージを発行し、後続のシステム動作を相当な時間待つ。SENTERの実行が開始された後、特殊なバスメッセージの１つであるSENTERバスメッセージがシステムバス２３０にブロードキャストされる。ILP以外の論理プロセッサは、応答する論理プロセッサ（RLP）と呼ばれ、内部のノンマスカブルイベントでSENTERバスメッセージに応答する。本実施例において、RLPはプロセッサ２１２とプロセッサ２２２を含む。RLPはそれぞれ、カレントオペレーションを終了し、システムバス２３０にRLPアクノレッジ（ACK）特殊バスメッセ

ージを送信し、待機状態に入る。ILPもシステムバス２３０にACKメッセージを送信することに注意すべきである。

【００１９】

チップセット２４０は一組のレジスタ、「EXISTS」レジスタ２７０と「JOINS」レジスタ２７２とを含んでもよい。これらのレジスタは、ILPとすべてのRLPがSENTERバスメッセージに適当に応答していることを検証するために用いられる。一実施形態において、チップセット２４０は、論理プロセッサによりなされたシステムバストランザクションにおいて、EXISTSレジスタ２７０の対応するビットに「１」を書き込むことにより、そのシステム中のすべての動作している論理プロセッサを追跡する。本実施形態において、システムバス２３０上の各トランザクションは、論理プロセッサ識別子を含む識別フィールドを含まねばならない。一実施形態において、これは物理プロセッサ識別子と、各物理プロセッサ内のハードウェア実行スレッドの識別子から構成される。例えば、プロセッサ２２２上で実行されているスレッドがシステムバス２３０上の何らかのバストランザクションを生じさせたとき、チップセット２４０は、そのトランザクション内でこの論理プロセッサ識別子を見て、EXISTSレジスタ２７０内の対応する場所２８６に「１」を書き込む。セキュア起動プロセスの間に、プロセッサ２２２上の同じスレッドがシステムバス２３０にACKメッセージを送信したとき、チップセット２４０はこれも見て、JOINSレジスタ２７２内の対応する場所２８８に「１」を書き込むことができる。（図２の実施例においては、明瞭にするため、各物理プロセッサには単一のスレッドが示されている。別の実施形態において、物理プロセッサは複数のスレッド、それにより負狂うの論理プロセッサをサポートしてもよい。）JOINSレジスタ２７２の内容とEXISTSレジスタ２７０の内容とが一致したとき、チップセット２４０は、すべてのプロセッサがSENTERバスメッセージに適当に応答したことを示すALL_JOINEDフラグ２４６を設定することができる。

【００２０】

他の実施形態において、EXISTSレジスタ２７０とJOINSレジスタ２７２は、ALL_JOINEDフラグ２４６の設定の後にセキュリティを支援し続けてもよい。ALL_JOINEDフラグ２４６の設定後トラステッドまたはセキュアオペレーションが終了するまでの間に、チップセット２４０はバスサイクルをモニターし、JOINSレジスタ２７２と比較し続けてもよい。この間に、チップセット２４０がJOINSレジスタ２７２では現在識別されていない論理プロセッサからのバストランザクションを見た場合、チップセット２４０はこの論理プロセッサはともかくも送れて「現れた」ものと推定する。これは暗に、このようなプロセッサはセキュア起動プロセスには参加せず、それゆえアタッカ（セキュリティ脅威）を表しうることを示す。上記の状況において、チップセット２４０はこのアタッカをセキュアード環境の外にとどめるように適当に応答することができる。一実施形態において、チップセット２４０は、上記の状況においてシステムリセットを強制することができる。第２の実施形態において、各論理プロセッサがACKバスメッセージの主張に続いて各トランザクションにおいてシステムバスに特殊な予約された信号を主張することにより、「遅れた」プロセッサが同様に検出される。この実施形態において、ALL_JOINEDフラグ２４６の設定につづいて、チップセット２４０が特殊な信号の主張を伴わないプロセッサにより起動されたバストランザクションが見られたとき、チップセット２４０はこの論理プロセッサがとにかく「遅れて」現われ、攻撃者かもしれないと推定する。

【００２１】

SENTERバスメッセージを発行した後、ILP（プロセッサ２０２）は、いつすべてのプロセッサが適当にACKに応答したか、また応答したかどうかを知るために、ALL_JOINEDフラグ２４６を参照する。フラグ２４６が設定されていなければ、いくつかの実装が可能である。ILP、チップセット、またはその他のウォッチドッグタイマーがシステムリセットをしてもよい。あるいは、システムがハングしてオペレータのリセットを要求してもよい。いずれの場合にも、システムは機能し続けないかもしれないが、（すべてのプロセッサが参加しない限り、セキュア起動プロセスが完了しないという意味で）セキュア環境の主張は保護される。通常のオペレーションにおいて、すぐ後に、ALL_JOINEDフラグ２４６が設

定され、他のすべての論理プロセッサが待機状態に入ったことをILPが補償される。

【００２２】

ALL_JOINEDフラグ２４６が設定されたとき、SINIT-AC２８０に含まれたSINITコードを認証し、その後実行する目的で、ILP（プロセッサ２０２）はSINIT-AC２８０のコピーとキー２８４の両方をセキュアメモリ２０８に移動する。一実施形態において、このセキュアメモリ２０８は、ILP（プロセッサ２０２）の内部キャッシュでもよい。そのキャッシュは特殊なモードで動作していてもよい。キー２８４は、SINIT-AC２８０モジュールに含まれるデジタルシグネチャを符号化するために用いられる秘密キーに対応する公開キーを表し、デジタルシグネチャを検証し、それによりSINITコードを認証するために使用される。一実施形態において、キー２８４は、例えばSENTERロジック２０４の一部として、プロセッサ中にすでに記憶されていてもよい。他の実施形態において、キー２８４は、ILPにより読み出されるチップセット２４０の読み出し専用キーレジスタ２４４に記憶されてもよい。さらに別の実施形態において、プロセッサまたはチップセットのキーレジスタ２４４のいずれかが、キー２８４の暗号ダイジェストを実際に保持していてもよい。ここで、キー２８４自体はSINIT-AC２８０に含まれる。この最後の実施形態において、ILPはキーレジスタ２４４からダイジェストを読み出し、SINIT-AC２８０に組み込まれたキー２８４の等価な暗号ハッシュを計算し、供給されたキー２８４が確かに信頼できることを確認するために２つのダイジェストを比較する。

【００２３】

SINIT-ACのコピーと公開キーのコピーがセキュアメモリ２０８内に存在してもよい。ILPは、公開キーのコピーを用いてSINIT-ACのコピーに含まれるデジタルシグネチャを復号することによりSINIT-ACのコピーを検証する。この復号により、暗号ハッシュのダイジェストのオリジナルコピーが作られる。新たに計算されたダイジェストがこのオリジナルダイジェストと一致したとき、SINIT-ACのコピーとそれに含まれたSINITコードは信頼できると考えられる。

【００２４】

ILPは、セキュアードオペレーションが起動されることを待機しているRLP（プロセッサ２１２、２２２）とチップセット２４０に知らせる、他の特殊なバスメッセージSENTER連続メッセージを、システムバス２３０を介して発行する。ILPは、下で概略を説明するように、セキュリティトークン２７６中のプラットフォーム構成レジスタ２７２にSINIT-ACモジュールの暗号ダイジェスト値を書きこむことにより、SINIT-ACモジュールが唯一のものであることを登録する。ILPのセキュアメモリ２０８内に保持されているSINITコードのトラステッドコピーに実行制御を転送することにより、SENTER命令のILPによる実行は終了する。トラステッドSINITコードは、そのシステムテストと構成動作を実行し、上の「レジスタ」の定義により、SVMMのメモリ常駐コピーを登録してもよい。

【００２５】

SVMMのメモリ常駐コピーの登録は、いくつかの方法で実行される。一実施形態において、ILP上で実行されているSENTER命令が、セキュリティートークン２７６内のPCR２７８に、SINIT-ACの計算されたダイジェストを書き込む。その後、トラステッドSINITコードは、メモリ常駐SVMMの計算されたダイジェストをセキュリティトークン２７６内の同じPCR２７８または他のPCR２７９に書き込む。SVMMダイジェストが同じPCR２７８に書き込まれたとき、セキュリティトークン２７６は新しい値（SVMMダイジェスト）でオリジナルコンテント（SINITダイジェスト）をハッシュし、その結果をPCR２７８に書き戻す。PCR２７８への第１の書き込み（初期化）がSENTER命令に限定されている実施形態においては、結果として得られるダイジェストはシステムへの信頼の根源として使用される。

【００２６】

一旦トラステッドSINITコードが実行を完了し、PCRにSVMMのアイデンティティを登録すると、SINITコードはSVMMにILP実行制御を移す。一般的な実施形態において、ILPにより実行された第１のSVMM命令は、SVMMの自己初期化ルーチンを表す。一実施形態において、ILPは各RLPに個別のRLP JOIN MESSAGE特殊バスメッセージを発行し、RLPの各々をSVMM

の現在実行されているコピーの監視下のオペレーションに入るようにする。この時点から後は、下の図３の説明で概略を説明するように、システム全体はトラステッドモードで動作する。

【００２７】

図３を参照して、本発明の一実施形態による、トラステッドまたはセキュアードソフトウェア環境の実施例の図が示されている。図３の実施形態において、トラステッドソフトウェアとアントラステッドソフトウェアが同時にロードされ、単一のコンピュータシステム上で同時に実行される。SVMM３５０は、１以上のアントラステッドオペレーティングシステム３４０とアントラステッドアプリケーション３１０－３３０からのハードウェアリソース３８０への直接アクセスを選択的に許可または拒絶する。このコンテクストにおいて、「アントラステッド」とは必ずしもオペレーティングシステムまたはアプリケーションが故意に不正を働いているということを意味するのではなく、インターラクションするコードのサイズや多様性により、ソフトウェアが所望のように振舞っており、その実行に干渉するウィルスやその他の外的コードがないと信頼性をもって主張することが実際的ではないということを意味する。典型的な実施形態において、アントラステッドコードは今日、パーソナルコンピュータ上で見られる通常のオペレーティングシステムおよびアプリケーションにより構成されるかもしれない。

【００２８】

SVMM３５０は、また、１以上のトラステッドまたはセキュアなカーネル３６０および1以上のトラステッドアプリケーション３７０からの、ハードウェアリソース３８０への直接アクセスを選択的に許可または拒絶する。上記のトラステッドまたはセキュアなカーネル３６０およびトラステッドアプリケーション３７０は、信頼分析の実行を可能とするために、サイズおよび機能的に限定されたものであってもよい。トラステッドアプリケーション３７０は、セキュア環境において実行可能ないかなるソフトウェアコード、プログラム、ルーチン、または一組のルーチンであってもよい。よって、トラステッドアプリケーション３７０は様々なアプリケーションやコードシーケンスであってもよく、Java（登録商標）アプレットのような比較的小さなアプリケーションであってもよい。

【００２９】

システムリソース保護や権限を改変しうるオペレーティングシステム３４０またはカーネル３６０により通常実行される命令または動作は、SVMM３５０によりトラップされ、選択的に許可され、部分的に許可され、または拒絶される。一実施例として、一般的な実施形態において、オペレーティングシステム３４０またはカーネル３６０により普通に実行されるプロセッサのページテーブルを変更する命令は、SVMM３５０によりトラップされる。これにより、そのバーチャルマシンのドメイン外のページ権限を変更しようとする要求はされないことを保証する。

【００３０】

図４Aを参照して、図３のセキュアードソフトウェア環境をサポートするように適応したマイクロプロセッサシステム４００の一実施形態が示されている。CPU A４１０、CPU B４１４、CPU C４１８、およびCPU D４２２は、特殊な命令の実行をサポートする追加的マイクロコーダまたはロジック回路も有するように構成されている。一実施形態において、この追加的マイクロコーダまたはロジック回路は、図２のSENTERロジック２０４であってもよい。これらの特殊な命令は、セキュア環境を立ち上げる間に、プロセッサの適当な同期を可能とする、システムバス４２０上に特殊なバスメッセージの発行をサポートする。一実施形態において、特殊なバスメッセージの発行は、図２のバスメッセージロジック２０６等の回路によりサポートされていてもよい。同様に、チップセット４３０は、チップセット２４０と同様であり、上で説明したシステムバス４２０上の特殊なサイクルをサポートする。物理プロセッサの数は特定の実施形態の実装により変化してもよい。一実施形態において、プロセッサは、インテル（登録商標）ペンティアム（登録商標）クラスのマイクロプロセッサであってもよい。チップセット４３０は、PCIバス４４６、あるいは、USB４４２、インテグレーテッドコントローラエレクトロニクス（IDE）バス（図示

せず）、スモールコンピュータシステムインターコネクト（SCSI）バス（図示せず）、その他いかなる入出力バスを介して、大容量記憶デバイス、例えば固定メディア４４４やリムーバブルメディア４４８とインターフェイスされる。固定メディア４４４またはリムーバブルメディア４４８は、磁気ディスク、磁気テープ、磁気ディスケット、光磁気ドライブ、CD-ROM、DVD-ROM、フラッシュメモリカード、その他多くの形式の大規模記憶であってもよい。

【００３１】

図４Aの実施形態において、４つのプロセッサCPU　A４１０、CPU　B４１４、CPU　C４１８、CPU　D４２２は、４つの別々のハードウェア実体として示されている。他の実施形態において、マイクロプロセッサの数は異なってもよい。実際に、物理的に個別のプロセッサは、１以上の物理的プロセッサ上で実行された別々のハードウェア実行スレッドにより代替されてもよい。後者の場合、これらのスレッドは、追加的物理的プロセッサの属性の多くを有する。複数の物理的プロセッサとプロセッサ上の複数のスレッドのミックスを用いて一般的に表現するため、「論理プロセッサ」という表現を物理的プロセッサまたは１以上の物理的プロセッサ上で動作するスレッドのいずれかを示すために用いる。よって、単一スレッドのプロセッサは論理プロセッサとみなされ、マルチスレッドまたはマルチコアプロセッサは複数の論理プロセッサとみなされる。

【００３２】

一実施形態において、チップセット４３０はモディファイドLPCバス４５０とインターフェイスしている。モディファイドLPCバス４５０は、チップセット４３０をセキュリティトークン４５４と接続するために用いられる。トークン４５４は、一実施形態において、トラステッドコンピューティングプラットフォームアライアンス（TCPA）により構想されたTPM４７１を含む。

【００３３】

図４Bを参照して、図３のセキュアードソフトウェア環境をサポートするように適応したマイクロプロセッサシステム４９０の別の実施形態が示されている。図４Aの実施形態と異なり、CPU　A４１０とCPU　B４１４はシステムバスA４０２でチップセット４２８に接続されており、一方、CPU　C４１８とCPU　D４２２はシステムバスB４０４でチップセット４２８に接続されている。他の実施形態において、２以上のシステムバスが用いられてもよい。他の別の実施形態において、ポイント・ツー・ポイントバスが用いられてもよい。特殊な命令は、セキュア環境を立ち上げる間に、プロセッサ間の適当な同期を可能とする、システムバスA４０２とシステムバスB４０４上の特殊なバスメッセージの発行をサポートする。一実施形態において、図２のバスメッセージロジック２０６等の回路により、特殊なバスメッセージの発行がサポートされる。

【００３４】

一実施形態において、チップセット４２８は、システムバスA４０２とシステムバスB４０４をわたる一貫性および統一性を維持する役割を果たす。標準であろうと特殊であろうと、バスメッセージがシステムバスAに送信されると、チップセット４２８は（適当であれば）そのメッセージをシステムバスB４０４に反映し、逆もまた同じである。

【００３５】

別の実施形態において、チップセット４２８は、システムバスA４０２とシステムバスB４０４を独立したサブシステムとして扱う。システムバスA４０２に発行された特殊なバスメッセージはいずれもそのバス上のプロセッサにのみ適用され、同様に、システムバスB４０４に発行された特殊なバスメッセージはいずれもそのバス上のプロセッサにのみ適用される。システムバスA４０２に関して確立された保護されたメモリはいずれも、システムバスA４０２に接続されたプロセッサにのみアクセス可能であり、システムバスB４０４上のプロセッサはアントラステッドデバイスとして扱われる。システムバスA４０２上のCPU　A４１０とCPU　B４１４のために確立された保護されたいずれのメモリにアクセスするにも、システムバスB４０４上のプロセッサCPU　C４１８とCPU　D４２２は自分自身のSENTERプロセスを実行して、システムバスA４０２上のプロセッサのために生成された

登録された環境と等しい環境を生成しなければならない。

【００３６】

図５を参照して、本発明の別の実施形態による、図３のセキュアードソフトウェア環境をサポートするように適応したマイクロプロセッサシステム５００の実施例の概略図が示されている。図４Aの実施形態と異なり、各プロセッサ（例えば、CPU A５１０）は、例えば、メモリコントローラ機能とデバイスアクセスロジック機能を実行するチップセット機能の一部（例えば、シップセット機能５９３）を含んでいる。これらのチップセット機能により、メモリ（例えば、メモリA５０２）のプロセッサへの直接接続が可能となる。他のチップセット機能は、別のチップセット５３０に残されてもよい。特殊なバスメッセージがシステムバス５２０をわたって発行されてもよい。各プロセッサは他のプロセッサに接続されたメモリに間接的にアクセスしてもよい。しかし、これらのアクセスはプロセッサ自身のメモリへのアクセスと比較したとき、かなり遅くてもよい。SENTERプロセスの開始の前に、ソフトウェアはSINIT-AC５６６とSVMM５７４のコピーを固定メディア５４４からローカルメモリ５０４に動かし、SINIT－AC５５６のコピーとSVMM５７２のコピーを形成する。一実施形態において、ILPであると意図されたプロセッサ（図５の実施例において、これはCPU B５１４である）により直接アクセスされるので、メモリ５０４が選択されてもよい。あるいは、SINIT-AC５６６とSVMM５７４のコピーは、ILP５１４がアクセス可能である限り、他の（ILPでない）プロセッサに付属した他のメモリに置かれてもよい。図２ですでに説明したように、同様のシークエンスと発行されたバスサイクルで、CPU B ILP５１４は、SENTER命令を発行することによりセキュアエンタープロセスを開始する。すべてのプロセッサがSENTER BUS MESSAGEに適当に応答したかどうかを決定し、この情報をILPに知らせるため、図２に関して上で説明したように、チップセット５３０はEXISTSレジスタ５７６、JOINSレジスタ５８０、ALL_JOINEDフラグ５８４を利用する。ILP（CPU B５１４）は、公開キー５６４のコピーと共に、SINIT－AC５５６のメモリ常駐コピーをセキュアメモリ５６０に動かしてもよい。SINIT-AC５５６の検証と登録をするとき、ILPはSVMM５７２のメモリ常駐コピーの検証と登録を続けてもよい。

【００３７】

次に図６を参照して、本発明の一実施形態による、様々なオペレーションの時間ライン図を示す。図６の時間ラインは、上で図２に関して説明したシステムの実施例に関連して説明したオペレーションの全体的スケジュールを示す。セキュアまたはトラステッドオペレーションが望ましいとソフトウェアが決定したとき、時間６１０において、SINIT-AC２８０とSVMM２８２のコピーを後続のSENTER命令に入手可能とする。本実施例において、ソフトウェアはSINIT－AC２８０のコピーとSVMM２８２のコピーを１以上のメモリページ２５０－２６２にロードする。１つのプロセッサ、本実施例においてはプロセッサ２０２がILPに選択され、時間６１２にSENTER命令を発行する。時間６１４に、ILPのSENTER命令はSENTER BUS MESSAGE６１６を発行する。ILPは、時間６２８でチップセットフラグ待機状態に入る前に、時間６１８にそれ自身のSENTER ACK６０８を発行する。

【００３８】

各RLP、例えばプロセッサ２２２は、時間６２０の間にカレント命令を完了することによりSENTER BUS MESSAGEに応答する。RLPはSENTER ACK６２２を発行し、SENTER CONTINUE MESSAGEを待ち受ける状態６３４に入る。

【００３９】

チップセット２４０は、時間６２４で、システムバス２３０上で観測したSENTER ACKメッセージに応じて、JOINSレジスタ２７２を設定する。JOINSレジスタ２７２の内容がEXISTSレジスタ２７０の内容と一致したとき、チップセット２４０は時間６２６でALL_JOINEDフラグ２４６を設定する。

【００４０】

この時間中、ILPは、ALL_JOINEDフラグ２４６を調べている間、ループにいる。ALL_JOINEDフラグ２４６が設定されており、時間６３０でALL_JOINEDフラグ２４６が設定されているとILPが決定したとき、ILPは時間６３２の間にSENTER CONTINUE MESSAGEを発行す

る。SENTER CONTINUE MESSAGEがシステムバス２３０に時間６３６でブロードキャストされると、RLPは参加待ち状態に入る。例えば、プロセッサ２２２のRLPは時間６３８において参加待ち状態に入る。

【００４１】

SENTER CONTINUE MESSAGEを発行すると、公開キーのコピーとSINIT-ACのコピーを形成するため、ILPは（時間６４０において）チップセット２４０のキーレジスタ２４４の公開キーとSINIT-ACのコピーをセキュアメモリ２０８に持ってくる。他の実施形態において、キーレジスタ２４４は公開キーのダイジェストを含んでおり、実際の公開キーはSINIT-ACに含まれるか、それに付属している。図２に関して上で説明したように、SINIT-ACのコピーを認証するとき、ILPはセキュアメモリ２０８内でSINIT-ACのコピーを実際に実行してもよい。

【００４２】

セキュアメモリ２０８内でSINIT-ACのコピーの実行が開始された後、SVMMのメモリ常駐コピーを検証および登録する。SVMMのコピーがセキュリティトークン２７６のPCR２７８に登録された後、SVMMのメモリ常駐コピー自体が実行を開始する。この時、進行中の期間６５０に、ILPでSVMMのオペレーションが確立される。

【００４３】

ILP SVMMオペレーションで最初になされることの１つは、個々のRLP JOIN MESSAGEをシステムバス２３０に発行することである。例えば、プロセッサ２２２JOIN MESSAGE６４４である。このメッセージには、SVMMの登録されたメモリ常駐コピーの実行において、RLPプロセッサ２２２が参加することができるメモリの場所を含む。あるいは、ILP SVMMオペレーションは、チップセットまたはメモリの所定の場所にメモリロケーションを登録し、JOIN MESSAGEを受信するとRLPはこの場所から開始アドレスを読み出してもよい。プロセッサ２２２JOIN MESSAGEを受信し、開始アドレスを決定した後、時間６４６の間に、RLPプロセッサ２２２はこの場所にジャンプし、SVMMの登録されたメモリ常駐コピーの実行に参加する。

【００４４】

すべてのRLPがSVMMの登録されたメモリ常駐コピーに参加した後、セキュアードオペレーションがマイクロコンピュータシステム２００全体で確立される。

【００４５】

図７を参照して、本発明の一実施形態による、ソフトウェアと他のプロセスブロックのフローチャートを示す。説明の明瞭化のため、図７には単一の代表的なRLPのプロセスブロックのみが示されている。他の実施形態においては、数個の応答論理プロセッサがあってもよい。

【００４６】

プロセス７００は、ブロック７１０において、論理プロセッサによりSINIT-ACとSVMMモジュールのコピーが後続のSENTER命令によりアクセス可能とされたときに開始する。本実施例において、ブロック７１２において、ILPは大規模記憶から物理メモリにSINIT-ACとSVMMコードをロードする。別の実施形態において、ILPだけでなくいずれの論理プロセッサがそうしてもよい。プロセッサは、ブロック７１４で識別されたように、SENTER命令を実行することによりILPとなる。ブロック７１６において、ILP SENTER命令はSENTER BUS MESSAGEを発行する。ブロック７１８において、ILPはチップセットに自分のSENTER ACKメッセージを発行する。ILPはその後、判断ブロック７２０に示したように待機状態に入り、チップセットがそのALL_JOINEDフラグを設定するのを待つ。

【００４７】

各RLPがブロック７７０においてSENTER BUS MESSAGEを受信した後、カレント命令が終わると実行を停止し、ブロック７７２において自分自身のSENTER ACKを発行する。判断ブロック７７４に示したように、各RLPは待機状態に入り、ILPからSENTER CONTINUE MESSAGEが届くのを待つ。

【００４８】

SENTER ACKメッセージが受信されたとき、チップセットはJOINSレジスタ内の対応するビットを設定する。JOINSレジスタの内容がEXISTSレジスタの内容と一致するとき、チップセットはそのALL_JOINEDフラグを設定し、ILPに判断ブロック７２０から進むように知らせる。

【００４９】

既存の判断ブロック７２０でYESの場合、ILPはブロック７２２においてSENTER CONTINURE MESSAGEを発行する。これにより各RLPが判断ブロック７７４から進むように信号で知らされる。各RLPは、判断ブロック７７６として示した第２の待機状態に入り、SENTER JOIN MESSAGEを待つ。

【００５０】

さしあたり、ILPはブロック７２４において、セキュア実行のために、チップセットの公開キーとSINIT-ACのメモリ常駐コピーをそれ自身のセキュアメモリに移動する。ILPはブロック７２６において、SINIT-ACのセキュアメモリ常駐コピーを検証するために使用し、それを実行する。SINIT-ACの実行により、ブロック７２８において、システム構成とSVMMコピーのテストが実行され、SVMMのアイデンティティが登録され、最後にSVMMの実行が開始される。ブロック７５４で示したように、ブロック７２８で実行される動作の一部として、SVMM２８２のメモリ常駐コピーにより使用されるメモリページを非プロセッサデバイスによる干渉から保護するように、メモリとチップセットのデバイスアクセスページテーブル２４８とデバイスアクセスロジック２４７がILP SINITコードにより構成される。

【００５１】

ILPがSVMMの制御下で実行を開始した後、ブロック７３０において、そのILPは各RLPに個別のSENTER JOIN MESSAGEを送信する。SENTER JOIN MESSAGEを発行した後、ILPはブロック７３２においてSVMMオペレーションを開始する。

【００５２】

SENTER JOIN MESSAGEを受信すると、各RLPは判断ブロック７７６で表された待機状態をYESの方に抜け、ブロック７８０でSVMMオペレーションを開始する。SENTER JOIN MESSAGEには、SVMMオペレーションに参加するときにRLPが分岐するSVMMエントリポイントが含まれる。あるいは、ILP SVMMコードがシステムロケーション（例えば、チップセット）中に適当なRLPエントリポイントを登録して、SENTER JOIN MESSAGEを受信した時にRLPにより読み出されるようにしてもよい。

【００５３】

開示した様々な実施形態には２以上のプロセッサ（論理または物理プロセッサのいずれか）が含まれるが、上記のマルチプロセッサおよび／またはマルチスレッドシステムは、複数の論理または物理プロセッサを伴うシステムの安全に関連する複雑性を説明する詳細に説明されたものであることを理解すべきである。より複雑でないシステムにおいて有利であろう実施形態は、プロセッサが１つだけのものである。一部の場合において、１つの物理的プロセッサが複数のスレッドを有し、それゆえ複数の論理プロセッサを含んで（したがって、説明したようにILPとRLPを有して）いてもよい。しかし、他の場合において、単一プロセッサ、単一スレッドのシステムを使用して、開始したセキュア処理技術を利用してもよい。このような場合、RLPは存在しない。しかし、セキュア処理技術は依然としてデータが盗まれたり、権限を与えられていない方法で改ざんされたりする見込みを減少させるのに役立つ。

【００５４】

以上の説明により、本発明をその特定の実施形態を参照して説明した。しかし、添付した請求項に記載した本発明のより広い精神と範囲から逸脱することなく、様々な修正や変更をできることは明らかである。したがって、本明細書と図面は限定的ではなく例示としてみなすべきである。

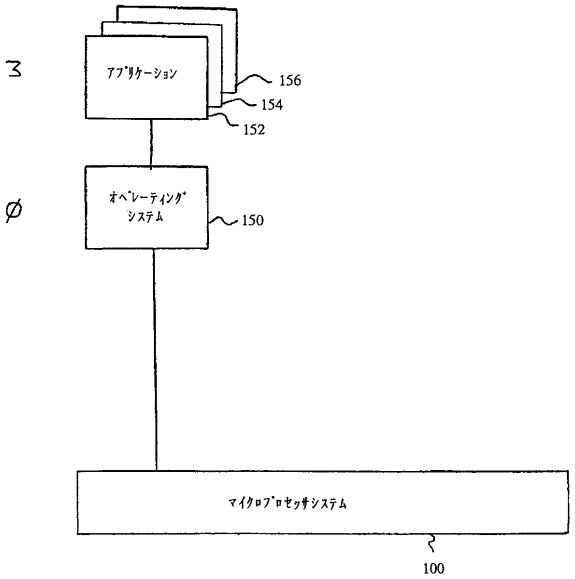
【図面の簡単な説明】

【００５５】

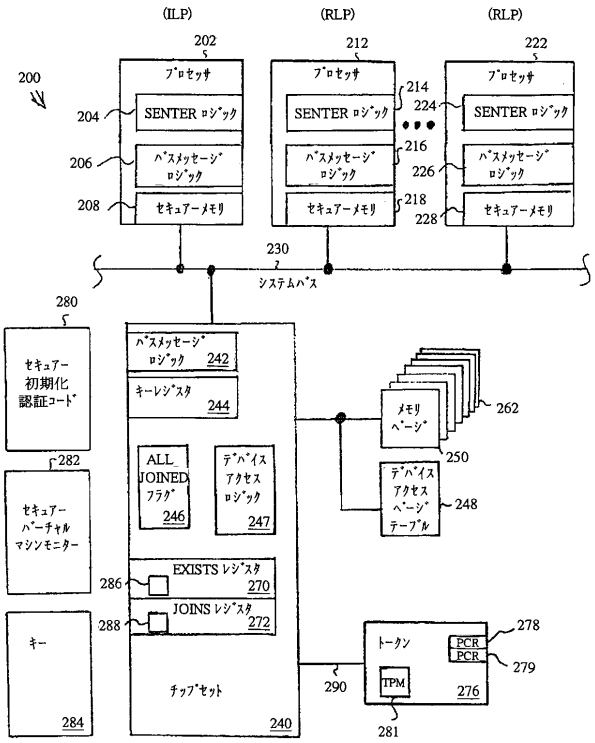
【図１】マイクロプロセッサシステムで実行されるソフトウェア環境を示す図である。

【図２】本発明の一実施形態による、信頼された、または安全なソフトウェアモジュールとシステム環境の実施例を示す図である。
【図３】本発明の一実施形態による、信頼された、または安全なソフトウェア環境の実施例を示す図である。
【図４Ａ】本発明の一実施形態による、図３の安全なソフトウェア環境をサポートするように適応したマイクロプロセッサシステムの実施例を示す概略図である。
【図４Ｂ】本発明の別の実施形態による、図３の安全なソフトウェア環境をサポートするように適応されたマイクロプロセッサシステムの実施例を示す概略図である。
【図５】本発明の別の実施形態による、図３の安全なソフトウェア環境をサポートするように適応されたマイクロプロセッサシステムの実施例を示す概略図である。
【図６】本発明の一実施形態による、ソフトウェアコンポーネントの実行を示すタイムチャートである。
【図７】本発明の一実施形態による、ソフトウェアおよびその他のプロセスブロックのフローチャートである。

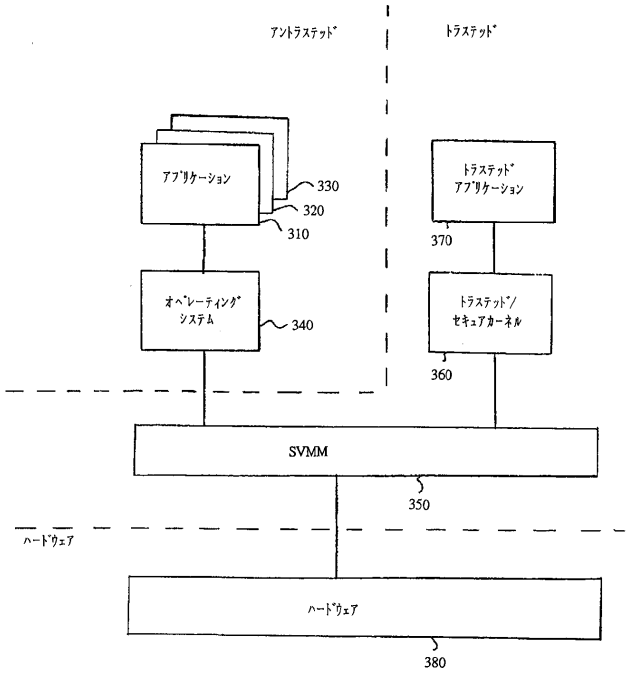
【図１】



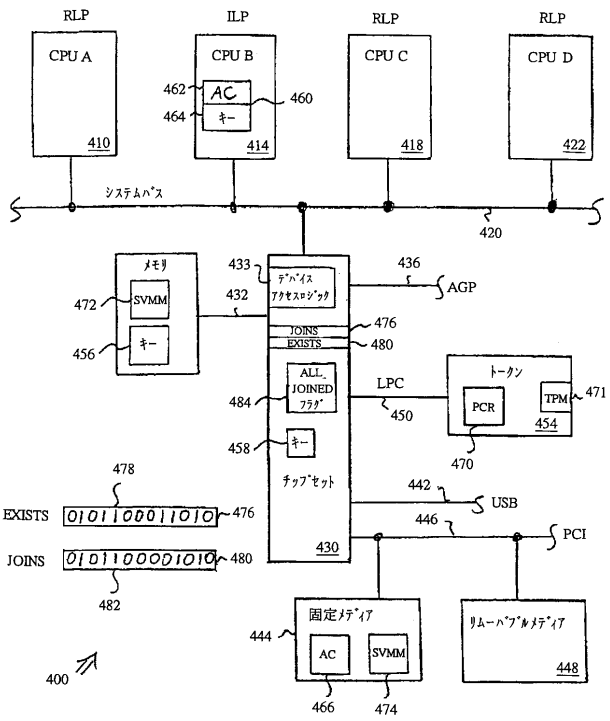
【図２】



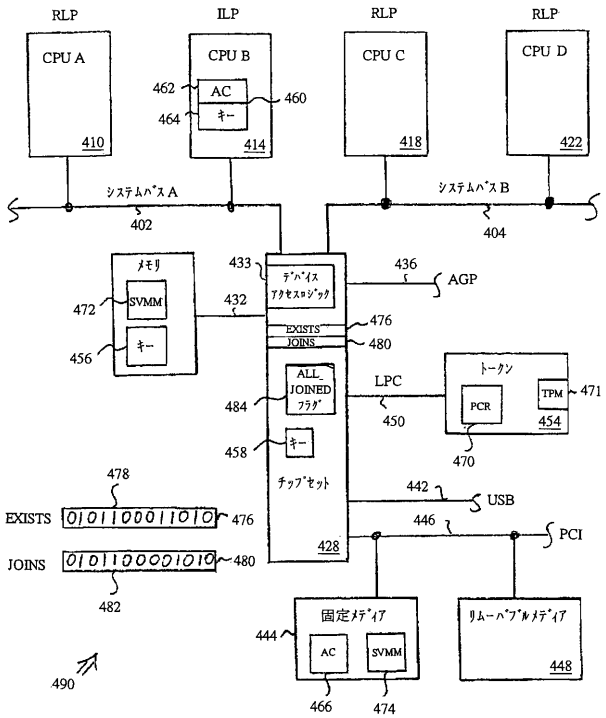
【図３】


アントラステッド
トラステッド
アプリケーション
330
320
310
トラステッド アプリケーション
370
オペレーティング システム
340
トラステッド/ セキュアカーネル
360
SVMM
350
ハードウェア
ハードウェア
380

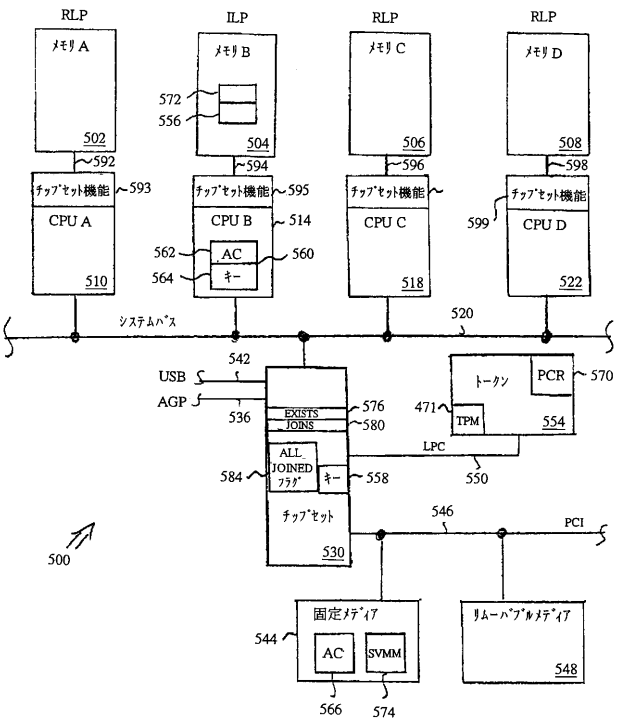
【図４Ａ】


RLP
CPU A
410
ILP
CPU B
462
AC
460
464
キー
414
RLP
CPU C
418
RLP
CPU D
422
システムバス
420
メモリ
472
SVMM
456
キー
433
デバイス アクセスロジック
432
436
AGP
JOINS EXISTS
476
480
ALL_ JOINED フラグ
484
458
キー
チップセット
430
LPC
450
トークン
PCR
TPM
471
454
470
478
EXISTS
476
JOINS
480
482
442
USB
446
PCI
固定メディア
444
AC
SVMM
466
474
リムーバブルメディア
448
400

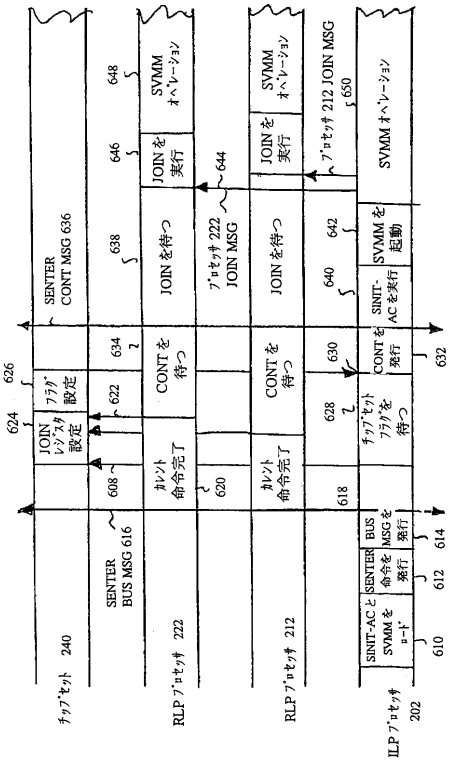
【図４Ｂ】


RLP
CPU A
410
ILP
CPU B
462
AC
460
464
キー
414
RLP
CPU C
418
RLP
CPU D
422
システムバス A
402
システムバス B
404
メモリ
472
SVMM
456
キー
433
デバイス アクセスロジック
432
436
AGP
EXISTS JOINS
476
480
ALL_ JOINED フラグ
484
458
キー
チップセット
428
LPC
450
トークン
PCR
TPM
471
454
470
478
EXISTS
476
JOINS
480
482
442
USB
446
PCI
固定メディア
444
AC
SVMM
466
474
リムーバブルメディア
448
490

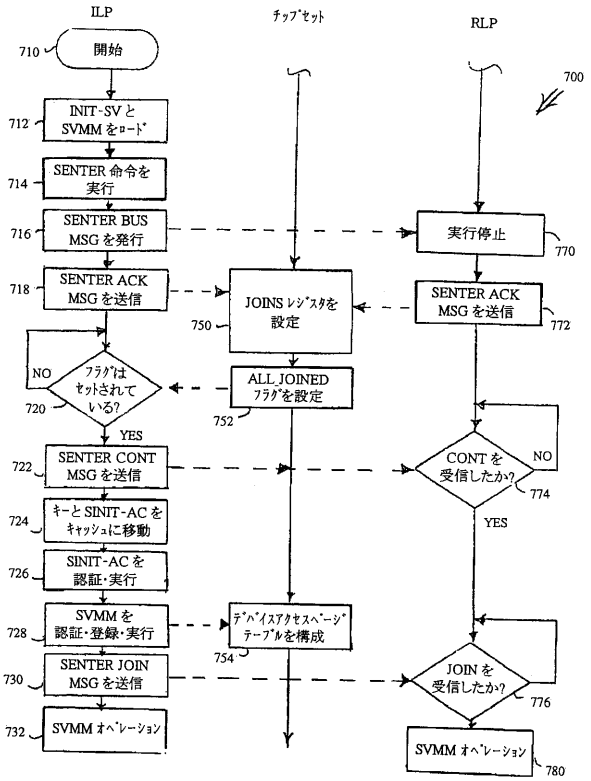
【図５】


RLP
メモリ A
502
592
チップセット機能
593
CPU A
510
ILP
メモリ B
572
556
504
594
チップセット機能
595
CPU B
514
562
AC
560
564
キー
RLP
メモリ C
506
596
チップセット機能
CPU C
518
RLP
メモリ D
508
598
599
チップセット機能
CPU D
522
システムバス
520
542
USB
AGP
536
EXISTS JOINS
576
580
ALL_ JOINED フラグ
584
キー
558
チップセット
530
LPC
550
トークン
PCR
570
471
TPM
554
546
PCI
固定メディア
544
AC
SVMM
566
574
リムーバブルメディア
548
500

【図６】


チップセット 240
RLP プロセッサ 222
RLP プロセッサ 212
ILP プロセッサ 202
SENTER CONT MSG 636
SENTER BUS MSG 616
JOIN レジスタ 設定
フラグ 設定
JOIN を待つ
CONT を待つ
カレント 命令完了
プロセッサ 222 JOIN MSG
SVMM オペレーション
JOIN を実行
プロセッサ 212 JOIN MSG
SINIT-AC を実行
SVMM を起動
CONT を発行
チップセット フラグを待つ
SINIT-AC と SVMM をロード
SENTER 命令を発行
BUS MSG を発行
624
626
608
620
618
622
634
638
646
648
644
640
642
630
628
632
614
612
610

【図７】


ILP
チップセット
RLP
700
710 開始
712 INIT-SV と SVMM をロード
714 SENTER 命令を実行
716 SENTER BUS MSG を発行
718 SENTER ACK MSG を送信
720 フラグはセットされている?
NO
YES
722 SENTER CONT MSG を送信
724 キーと SINIT-AC をキャッシュに移動
726 SINIT-AC を認証・実行
728 SVMM を認証・登録・実行
730 SENTER JOIN MSG を送信
732 SVMM オペレーション
750 JOINS レジスタを設定
752 ALL JOINED フラグを設定
754 デバイスアクセスページテーブルを構成
770 実行停止
772 SENTER ACK MSG を送信
774 CONT を受信したか?
776 JOIN を受信したか?
780 SVMM オペレーション

フロントページの続き

(81)指定国 AP(GH,GM,KE,LS,MW,MZ,SD,SL,SZ,TZ,UG,ZM,ZW),EA(AM,AZ,BY,KG,KZ,MD,RU,TJ,TM),EP(AT,BE,BG,CH,CY,CZ,DE,DK,EE,ES,FI,FR,GB,GR,HU,IE,IT,LU,MC,NL,PT,RO,SE,SI,SK,TR),OA(BF,BJ,CF,CG,CI,CM,GA,GN,GQ,GW,ML,MR,NE,SN,TD,TG),AE,AG,AL,AM,AT,AU,AZ,BA,BB,BG,BR,BY,BZ,CA,CH,CN,CO,CR,CU,CZ,DE,DK,DM,DZ,EC,EE,ES,FI,GB,GD,GE,GH,GM,HR,HU,ID,IL,IN,IS,JP,KE,KG,KP,KR,KZ,LC,LK,LR,LS,LT,LU,LV,MA,MD,MG,MK,MN,MW,MX,MZ,NO,NZ,OM,PH,PL,PT,RO,RU,SC,SD,SE,SG,SK,SL,TJ,TM,TN,TR,TT,TZ,UA,UG,UZ,VC,VN,YU,ZA,ZM,ZW

（特許庁注：以下のものは登録商標）
リナックス

(72)発明者 グロウロック，デイヴィッド
アメリカ合衆国 ９７００７ オレゴン州 アローハ サウスウエスト １８４ス アヴェニュー ８２８５

Ｆターム(参考) 5B017 AA01 AA08 BA01 BA09 CA11 CA15