(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号

特許第5316592号
(P5316592)

(45)発行日　平成25年10月16日(2013.10.16)　(24)登録日　平成25年7月19日(2013.7.19)

(51)Int.Cl.		F I		
G06F 12/14	(2006.01)	G06F	12/14	510E
G06F 21/62	(2013.01)	G06F	21/24	166A
H04L 9/32	(2006.01)	H04L	9/00	675A

請求項の数 2　(全 57 頁)

(21)出願番号　特願2011-129097(P2011-129097)
(22)出願日　平成23年6月9日(2011.6.9)
(62)分割の表示　特願2009-185006(P2009-185006)の分割
原出願日　平成16年6月30日(2004.6.30)
(65)公開番号　特開2011-181107(P2011-181107A)
(43)公開日　平成23年9月15日(2011.9.15)
審査請求日　平成23年6月9日(2011.6.9)

(73)特許権者　308014341
富士通セミコンダクター株式会社
神奈川県横浜市港北区新横浜二丁目１０番２３
(74)代理人　100074099
弁理士　大菅　義之
(74)代理人　100133570
弁理士　▲徳▼永　民雄
(72)発明者　後藤　誠司
神奈川県川崎市中原区上小田中４丁目１番１号　富士通株式会社内
(72)発明者　蒲田　順
神奈川県川崎市中原区上小田中４丁目１番１号　富士通株式会社内

最終頁に続く

(54)【発明の名称】セキュアプロセッサ用プログラム

(57)【特許請求の範囲】
【請求項１】
実行コードを含むページをメモリにページインする計算機によって使用されるプログラムであって、
該計算機内の直接メモリアクセス機構に前記ページのメモリへの転送を依頼する手順と、
該転送の成功後に、該計算機のトランスレーション・ルックアサイド・バッファ内のページ・テーブル・エントリに、該ページ内の実行コードに対応するプロセスの実行に先立って該実行コードを格納するページが正しく認証されたことを示すセキュアページフラグが設定されたページに対応するセキュアプロセス識別子と比較するための識別子であって、該プロセスの生成命令が発行された時点で生成されたセキュアプロセス識別子を含み、該ページについてのデータを設定する手順と、
前記ページの認証と、該認証の成功を示すセキュアページフラグの該ページ・テーブル・エントリへのセットとをハードウェアに要求する手順とを計算機に実行させることを特徴とするセキュアプロセッサ用プログラム。
【請求項２】
実行コードを含むページの認証を行う計算機によって使用されるプログラムであって、
メモリに読み込まれた該ページに対するハッシュ演算を行う手順と、
該ページに付与されている認証情報を復号する手順と、
該ハッシュ演算結果と該復号結果とを比較する手順と、

該比較の結果として一致が検出された時、該計算機のトランスレーション・ルックアサイド・バッファ内のページ・テーブル・エントリに該ページの認証が成功したことを示すセキュアページフラグをセットする手順とを計算機に実行させることを特徴とするセキュアプロセッサ用プログラム。

【発明の詳細な説明】

【技術分野】

【０００１】

本発明は、計算機などの情報処理システムの安全性確保方式に係り、さらに詳しくは計算機や各種プロセッサ組込機器などにおいて悪意を持った実行コードの動作を防止することが可能なセキュアプロセッサ、およびセキュアプロセッサ用プログラムに関する。

【背景技術】

【０００２】

プロセッサを使用するシステムでは、動作をプログラムによって記述することができ、ハードウェアによってすべてを構成するシステムに比べ、その動作の柔軟性が大きく、多種類の機能を実装することが容易である。これらの特徴のために、プロセッサはパーソナルコンピュータ、ＰＤＡ、携帯電話、情報家電など多くのシステムに使用されており、またその普及とともに電子商取引のように高度にセキュリティを要求される処理も広範に行われるようになっている。セキュリティを強固にするために回線データに対する暗号化、ユーザ認証などの各種のシステム的な措置が施されているが、近年ではシステムレベルでの安全性のみならず、コンピュータウィルスや不正アクセス等の蔓延に対応するために、ソフトウェアレベル、プロセッサレベルの安全性が問題となっている。

【０００３】

例えば携帯電話や情報家電など、各種のプロセッサ組込機器などがネットワークに接続されることにより、パーソナルコンピュータ等に対すると同等の脅威をこれらの機器も外部から受ける可能性が高くなっている。不正侵入などを細かく見ると、悪意を持った実行コードが端末内で動作することにその原因がある。悪意のあるコード、所望しないコードをプロセッサ上で動作させないようにすることが重要であるが、従来においては悪意のあるコードを動作させないようにするためのプロセッサ側の対策があまり十分ではなく、結果的に安全なソフトウェア実行環境が提供されていないという問題点があった。

【０００４】

次に従来においては、例えばデータや命令の実行コードを主記憶装置や二次記憶装置に格納するに当り、安全性を確保するために暗号化を行って、実際の命令実行に当って暗号化されたデータなどを復号し、プロセッサ内のキャッシュメモリに格納して処理を実行することも行われているが、このような場合暗号処理を実行するハードウェアなどはプロセッサチップと別のチップに搭載され、外付けで使用されるために処理速度など、暗号処理性能が低くなってしまうという問題点があった。

【０００５】

またこのような暗号化処理において、データなどの暗号化に用いられる暗号鍵は、外付けされたチップ上の暗号処理側で決定されており、プロセッサ側で実行される命令の種類やスーパバイザ／ユーザモードなどの区別、あるいはデータや命令フェッチのアクセスアドレスなどに無関係であり、プロセッサ側の実行ユニットが暗号化、および復号において用いられるべき鍵を指定することができないために、実行中の命令に対応して適切な暗号鍵を選択することができないという問題点もあった。

【０００６】

このようなソフトウェア実行環境の安全性に関する従来技術として次の文献がある。

特許文献１：特開２００２－３５３９６０号公報　「コード実行装置およびコード配布方法」

この文献には暗号化された実行コードの認証を行って暗号化コードの有効性を確認し、セキュアプロセッサがその暗号化コードに対応する命令をフェッチし、セキュアタスクとして実行するコード実行装置が開示されている。

【０００７】
しかしながらこのコード実行装置では実行コードに対応するプロセスと認証に用いられる鍵との間に関連が無く、例えばオペレーティングシステム（ＯＳ）に対して悪意のある操作が行われ、プログラムに別の認証鍵が割り付けられると、結果的に悪意のあるコードが動作してしまうという問題点を解決できなかった。
【先行技術文献】
【特許文献】
【０００８】
【特許文献１】特開２００２－３５３９６０号公報
【発明の概要】
【発明が解決しようとする課題】
【０００９】
本発明の第１の課題は、暗号化された命令コードを書き換え不可能な形式で記憶するメモリの記憶内容を基本として、例えば二次記憶装置上に格納されているプログラムの実行コードを次々と認証し、確実に信頼できるアプリケーションの範囲を段階的に広げ、信頼できる動作だけを実行することが可能なセキュアプロセッサを提供することである。
【００１０】
本発明の第２の課題は、暗号処理ブロックを、例えばプロセッサと同一チップ上に備えて暗号処理性能を向上させるとともに、実行中の命令によってデータや実行コードの暗号化／復号に用いるべき鍵を選択可能とすることである。
【００１１】
本発明の第３の課題は、プロセスの実行コードの主記憶への格納のタイミングでプロセスに対応する認証鍵を用いて実行コードの認証を行い、認証に成功した実行コードのみを実行可能とすることによって、プロセッサによる情報処理の安全性を向上させることである。
【課題を解決するための手段】
【００１２】
図１は本発明のセキュアプロセッサの原理構成ブロック図である。同図において本発明のセキュアプロセッサ１は、固有鍵記憶手段２、命令コード記憶手段３、認証処理手段４、および暗号処理手段５を備える。
【００１３】
固有鍵記憶手段２は、セキュアプロセッサにおいて命令コードを実行するコアに固有の鍵、例えばＣＰＵ固有鍵を記憶するものであり、命令コード記憶手段３、例えば暗号化ＲＯＭコード領域は暗号化された命令コードを書き換え不可能な形式で記憶するものであり、認証処理手段４は固有鍵を用いて命令コード記憶手段３に記憶された命令コードを含む命令コードの認証を行うものであり、暗号処理手段５はコアと外部のメモリとの間で入出力されるデータを暗号化するものである。
【００１４】
発明の実施の形態においては、暗号処理手段５が、認証された命令コードを暗号化してページ単位でセキュアプロセッサ１に接続された記憶装置、例えば主記憶に格納することもでき、また認証処理手段４が認証対象とする命令コードに認証情報が付加されていることもできる。
【００１５】
次に図１のセキュアプロセッサ１において、命令コードを実行するコアとして認証処理手段４によって認証された命令コードのみを実行するセキュアコアと、認証されていない通常の命令コードを実行するノーマルコアとを備えることもできる。
【００１６】
この場合、命令コード記憶手段３に記憶された暗号化された命令コードを用いてセキュアコアがブート（起動）されるとともに、セキュアコアがそのブート完了後にノーマルコアのブートを行わせるノーマルコアブート手段を備えることもでき、さらにセキュアコア

がノーマルコアのブート後にノーマルコアの動作を監視し、異常状態を検出した時、ノーマルコアの動作停止、または特定処理への分岐を行わせるノーマルコア監視手段を備えることもできる。

【００１７】

次に本発明のセキュアプロセッサ用プログラムは、暗号化された命令コードが書き換え不可能な形式で記憶されたメモリ内のプログラムを用いて起動処理を行う手順と、そのメモリ内に記憶された命令コードを含む命令コードの認証処理を行う認証処理ブロックと、プロセッサ固有の鍵を管理する鍵管理処理と、認証処理ブロックによって認証された命令コードの暗号化／復号処理に用いられる鍵が格納された鍵テーブルに対する操作処理とをセットアップする手順と、認証処理ブロックを用いて二次記憶上のプログラムの認証処理を行う手順と、起動されたオペレーティングシステムを含み、認証処理済のプログラムの実行時に必要となる鍵処理を実行する鍵処理モニタとしての動作を行う手順とを計算機に実行させるものである。

【００１８】

本発明のセキュアプロセッサは、命令を実行する命令実行手段、例えば実行ユニットと、命令実行手段からのコマンドに対応して外部のメモリに対するデータのロード／ストアを制御するロード／ストア制御手段、例えばロードストアユニットと、ロード／ストア制御手段と外部のメモリとの間でデータの暗号化／復号化を行う暗号処理手段、例えば暗号化回路と復号化回路とを備え、命令実行手段が、実行中の命令に対応して暗号処理手段に対してデータ暗号化／復号化に使用すべき鍵を指定するものである。

【００１９】

発明の実施の形態においては、このセキュアプロセッサにおいて複数個の鍵を記憶する鍵記憶手段、例えば鍵テーブルメモリをさらに備え、命令実行手段が、鍵記憶手段に対して前述の鍵を指定する鍵番号を出力し、該鍵記憶手段がその鍵番号に対応して暗号処理手段に対して、データ暗号化／復号化に使用すべき鍵を与えることもできる。

【００２０】

またこのセキュアプロセッサにおいて、外部からロードされた命令フェッチデータの復号に使用されるべき鍵を記憶する鍵記憶手段をさらに備え、命令実行手段が命令フェッチ状態にある時、鍵記憶手段が暗号処理手段に対してフェッチされた命令の復号に使用されるべき鍵を与えることもできる。

【００２１】

また本発明のセキュアプロセッサは命令を実行する命令実行手段と、命令実行手段からのコマンドに対応して外部のメモリに対するデータのロード／ストアを制御するロード／ストア制御手段と、ロード／ストア制御手段と外部のメモリとの間でデータの暗号化／復号化を行う暗号処理手段とを備え、命令実行手段が、実行中の命令によるデータ／命令フェッチのアクセスアドレスに対応させて、暗号処理手段に対してデータ、および命令の暗号化／復号化に使用すべき鍵を指定する信号を与えるものである。

【００２２】

発明の実施の形態においては、このセキュアプロセッサにおいて複数個の鍵を記憶する鍵記憶手段をさらに備え、命令実行手段が、前述のアクセスアドレスとしての論理アドレスを出力し、鍵記憶手段がその論理アドレスに対応して暗号処理手段に対して暗号化／復号化に使用すべき鍵を与えることもできる。

【００２３】

あるいはこのセキュアプロセッサにおいて、複数個の鍵を記憶する鍵記憶手段をさらに備え、ロード／ストア制御手段が、命令実行手段から与えられるコマンドに対応してアクセスアドレスとしての物理アドレスを出力し、鍵記憶手段がその物理アドレスに対応して暗号処理手段に対して暗号化／復号化に使用すべき鍵を与えることもできる。

【００２４】

本発明のセキュアプロセッサは、実行コードに対応するプロセスの実行に先立ってその実行コードが正しく認証されたことを示すセキュアページフラグが設定されるページに対

応するセキュアプロセス識別子と比較するためのセキュアプロセス識別子を、そのプロセスの生成命令が発行された時点で生成するセキュアプロセス（コンテキスト）識別子生成手段と、生成されたセキュアプロセス識別子をそのプロセスに関連する情報として保持するプロセス情報保持手段、例えばコンテキスト情報格納部とを備える。

【００２５】

発明の実施の形態においては、前述のプロセスに対応する実行コードに認証情報が付与されるとともに、プロセス情報保持手段が生成されたプロセスの生存期間中に行われる実行コード認証のための認証鍵をさらに保持することもできる。

【００２６】

またこのセキュアプロセッサは、前述のプロセスに対応する実行コードがメモリの空きページに格納され、そのページのアドレスに対応させてセキュアプロセス識別子がプロセッサ内のバッファに格納された後に、ページ単位の認証鍵を用いてその実行コードの認証を行い、認証が成功した時、そのバッファにセキュアページフラグをセットする認証手段をさらに備えることもできる。

【００２７】

あるいはこのセキュアプロセッサは、実行コードの実際の実行に先立って前述のバッファ内に格納されたセキュアプロセス識別子であって、対応するセキュアページフラグがセットされているセキュアプロセス識別子と、プロセス情報保持手段に保持され、実行すべき命令コードに対応するセキュアプロセス識別子とを比較し、両者が一致した時に実行コードが格納されたメモリ上のページへのアクセスを、命令を実行する命令実行部に許可するメモリアクセス制御手段をさらに備えることもできる。

【００２８】

またこのセキュアプロセッサは、それぞれ命令実行ユニットとキャッシュとを備えるコアとして、認証された実行コードのみを実行するセキュアコアと、認証されていない通常の実行コードを実行するノーマルコアとを備えることもできる。

【００２９】

さらにこのセキュアプロセッサは、実行コードのメモリへの格納と並行して実行コードの認証に必要となる演算を行い、その演算の結果を保持して認証手段に与える直接メモリアクセス手段をさらに備えることもできる。

【００３０】

次に本発明のセキュアプロセッサ用プログラムは、実行コードを含むページをメモリにページインする計算機によって使用されるプログラムであり、計算機内の直接メモリアクセス機構に前述のページのメモリへの転送を依頼する手順と、その転送の成功後に計算機のトランスレーション・ルックアサイド・バッファ内のページ・テーブル・エントリに、そのページ内の実行コードに対応するプロセスの実行に先立ってその実行コードを格納するページが正しく認証されたことを示すセキュアページフラグが設定されたページに対応するセキュアプロセス識別子と比較するための識別子であって、そのプロセスの生成命令が発行された時点で生成されたセキュアプロセス識別子を含み、そのページについてのデータを設定する手順と、前述のページの認証と、認証の成功を示すセキュアページフラグのページ・テーブル・エントリへのセットとをハードウェアに要求する手順とを計算機に実行させるものである。

【００３１】

さらに本発明のセキュアプロセッサ用プログラムは、実行コードを含むページの認証を行う計算機によって使用されるプログラムであって、メモリに読み込まれたページに対するハッシュ演算を行う手順と、そのページに付加されている認証情報を復号する手順と、ハッシュ演算結果と復号結果とを比較する手順と、比較の結果として一致が検出された時、その計算機のトランスレーション・ルックアサイド・バッファ内のページ・テーブル・エントリにページの認証が成功したことを示すセキュアページフラグをセットする手順とを計算機に実行させるものである。

【発明の効果】

【００３２】
本発明によれば、プロセッサ内に保持されている書き換え不可能な形式の暗号化された命令コードを基本的な信頼点として、例えばオペレーティングシステムを含むプログラムの認証を行い、信頼できるプログラムの範囲を拡大していくことによって、システムのセキュリティレベルを本質的に向上させることが可能となる。
【００３３】
また本発明によれば、暗号処理ブロックを、例えばプロセッサと同一チップ内に備え、暗号処理性能を向上させるとともに、実行中の命令に応じたデータや実行コードの暗号化を行うことが可能となる。実行中の命令に対応して、例えば暗号化のレベルを変化させることもでき、システムとしてのセキュリティレベルを向上させることができる。
【００３４】
さらに本発明によれば、命令コードの実行の前にその命令コードの認証を行い、セキュアページフラグがセットされたプロセスに対応するプロセス識別子と実行中のプロセスのプロセス識別子との一致を検出してプロセスの実行を行うことにより、悪意を持って改ざんされた実行コードのプロセッサ上での動作を防止することができ、安全なソフトウェア実行環境が提供される。
【図面の簡単な説明】
【００３５】
【図１】本発明のキュアプロセッサの原理構成ブロック図である。
【図２】第１の実施例におけるプロセッサの基本構成を示すブロック図である。
【図３】第１の実施例におけるプロセッサの基本処理フローチャートである。
【図４】コード認証処理ブロックと暗号処理ブロックによる処理のフローチャートである。
【図５】命令領域とデータ領域とによって異なる鍵が指定されている場合の暗号処理ブロックの処理フローチャートである。
【図６】公開鍵で暗号化された暗号鍵の格納方式の説明図である。
【図７】公開鍵で暗号化された暗号鍵の格納処理フローチャートである。
【図８】認証局の署名が付与された暗号鍵の格納方式の説明図である。
【図９】認証局の署名が付与された暗号鍵の格納処理フローチャートである。
【図１０】不正命令検出時の処理フローチャートである。
【図１１】データ領域に格納された命令に対する鍵付け替え処理のフローチャートである。
【図１２】第２の実施例におけるプロセッサの基本構成を示すブロック図である。
【図１３】第２の実施例におけるプロセッサの基本処理フローチャートである。
【図１４】セキュアコアとノーマルコアを備えるプロセッサの基本構成を示すブロック図である。
【図１５】図１４のプロセッサにおける処理の基本フローチャートである。
【図１６】図１４のプロセッサにおけるセキュアコアによるノーマルコアの動作の停止制御方式の説明図である。
【図１７】図１４のプロセッサにおけるセキュアコアによるノーマルコアの動作の停止制御処理のフローチャートである。
【図１８】セキュアコアに対応する鍵生成機構を備えるプロセッサの構成ブロック図である。
【図１９】図１８のプロセッサにおける鍵処理方式の具体例の説明図である。
【図２０】第３の実施例におけるプロセッサの基本構成を示すブロック図である。
【図２１】第３の実施例において鍵テーブルメモリを備えるプロセッサの構成ブロック図である。
【図２２】第３の実施例において命令アクセス状態にあるプロセッサの構成を示すブロック図である。
【図２３】鍵テーブルメモリに対する鍵選択レジスタを備えるプロセッサの構成ブロック

図である。
【図２４】命令アクセス状態にあり、鍵テーブルメモリに対する鍵選択レジスタを備えるプロセッサの構成ブロック図である。
【図２５】鍵テーブルメモリの構成例を示す図である。
【図２６】暗号化回路、復号化回路の構成例を示すブロック図である。
【図２７】データ追い越し機能付き暗号化回路、復号化回路の構成例を示す図である。
【図２８】キャッシュスルー方式のロードストアユニットに対応するリードモディファイライト方式の説明図である。
【図２９】第４の実施例におけるプロセッサの基本構成を示すブロック図である。
【図３０】論理アドレスが与えられる鍵テーブルメモリを備えるプロセッサの構成ブロック図である。
【図３１】物理アドレスが与えられる鍵テーブルメモリを備えるプロセッサの構成ブロック図である。
【図３２】第４の実施例における鍵テーブルメモリの構成例（その１）を示す図である。
【図３３】第４の実施例における鍵テーブルメモリの構成例（その２）を示す図である。
【図３４】第４の実施例における鍵テーブルメモリの構成例（その３）を示す図である。
【図３５】論理アドレスと物理アドレスとが与えられる鍵テーブルメモリを備えるプロセッサの構成を示すブロック図である。
【図３６】図３５の鍵テーブルメモリに対してアドレス選択指示を与える鍵選択レジスタを備えるプロセッサの構成ブロック図である。
【図３７】図３５、図３６における鍵テーブルメモリの構成例を示す図である。
【図３８】メモリ管理ユニット内に鍵テーブルが備えられるプロセッサの構成を示すブロック図である。
【図３９】図３８におけるデータアクセス方式の説明図である。
【図４０】アドレスマップレジスタに鍵テーブルが併設されている場合のデータアクセス方式の説明図である。
【図４１】メモリ管理ユニットのＯＮ／ＯＦＦ状態に応じて鍵を切り替えるプロセッサの構成を示すブロック図である。
【図４２】メモリ管理ユニットのＯＮ／ＯＦＦ状態に応じて鍵を切り替える暗号化／復号方式の説明図である。
【図４３】第３、および第４の実施例における実行ユニットの入出力信号の説明図である。
【図４４】第５の実施例におけるプロセッサシステムの詳細構成ブロック図である。
【図４５】セキュアコンテキスト識別子生成方式の説明図である。
【図４６】セキュアコンテキスト識別子生成方法の説明図である。
【図４７】セキュアコンテキスト識別子消滅方式の説明図である。
【図４８】実行コードに付加された認証情報の説明図である。
【図４９】公開鍵の認証鍵レジスタへの格納方式の説明図である。
【図５０】公開鍵の認証鍵レジスタへの格納処理のフローチャートである。
【図５１】暗号化された共通鍵の認証鍵レジスタへの格納方式の説明図である。
【図５２】暗号化された共通鍵の認証鍵レジスタへの格納処理フローチャートである。
【図５３】物理メモリへのページイン時の処理方式の説明図である。
【図５４】物理メモリへのページイン時の処理フローチャートである。
【図５５】認証部の構成を示すブロック図である。
【図５６】認証部の動作フローチャートである。
【図５７】第５の実施例におけるページ利用時のメモリアクセス制御部によるアクセスチェック方式の説明図である。
【図５８】メモリアクセス制御部の動作例を説明する図である。
【図５９】命令フェッチ時のメモリアクセス制御部の処理フローチャートである。
【図６０】セキュアコアとノーマルコアからのページ利用時のアクセス制御方式を説明す

る図である。
【図６１】セキュアモードとノーマルモードを切り替えるためのモードレジスタを備えるプロセッサの構成図である。
【図６２】セキュアＤＭＡの構成を示すブロック図である。
【図６３】セキュアＤＭＡによるデータ転送処理のフローチャートである。
【図６４】ＯＳによるページイン時の処理のフローチャートである。
【図６５】第７の実施例におけるコンテキスト情報暗号化方式の説明図である。
【図６６】コンテキスト情報の復号方式の説明図である。
【図６７】コンテキスト情報に対する改ざん検出情報付加方式の説明図である。
【図６８】コンテキスト情報に対する改ざん検出方式の説明図である。
【図６９】セキュア動作用コンテキスト情報の暗号化方式の説明図である。
【図７０】セキュア動作用コンテキスト情報に対する改ざん検出情報付加方式の説明図である。
【図７１】ページ・テーブル・エントリの暗号化方式の説明図である。
【図７２】ページ・テーブル・エントリの復号方式の説明図である。
【図７３】ページ・テーブル・エントリへの改ざん検出情報付加方式の説明図である。
【図７４】ページ・テーブル・エントリに対する改ざん検出方式の説明図である。
【図７５】本発明を実現するためのプログラムのコンピュータへのローディングを説明する図である。
【発明を実施するための形態】
【００３６】
以下、本発明の実施形態について詳細に説明するが、まず本発明のセキュアプロセッサの全体的な構成と、その処理の概要を第１の実施例として説明する。
図２は、第１の実施例としてのセキュアプロセッサの基本構成を示すブロック図である。同図においてプロセッサ１０は、実行ユニットとキャッシュを含むコア１１、外部インタフェースとのコマンド処理およびバスデータ（プログラムのコードまたはデータ）の暗号化やその復号などを行う暗号処理ブロック１２、命令コードの認証を行うコード認証処理ブロック１３、プロセッサの起動時に用いられる最も基本的なプログラムなどが暗号化されて格納されている暗号化ＲＯＭコード領域１４、およびこのコード領域１４に格納されているプログラムなどの復号などを行うためのＣＰＵ固有鍵１５を備えている。なお、暗号処理ブロック１２の動作については後述の第３の実施例などで、コード認証処理ブロック１３の動作については第５の実施例などでより詳細に説明する。
【００３７】
そしてコア１１と暗号処理ブロック１２との間では、コマンド、およびデータのやり取りが行われるとともに暗号化のための鍵の制御が行われ、またコア１１とコード認証処理ブロック１３との間には認証インタフェースが備えられる。さらに暗号処理ブロック１２、およびコード認証処理ブロック１３は主記憶１７に対するアクセスを実行し、またコード認証処理ブロック１３は二次記憶１８に対するアクセスを実行するものとする。
【００３８】
図３は、第１の実施例におけるセキュアプロセッサの全体処理フローチャートである。同図において電源が投入されると、図２のコア１１はステップＳ１で暗号化ＲＯＭコード領域１４に格納されているプログラムをＣＰＵ固有鍵１５を用いて復号し、ブート（起動）処理を実行する。内蔵ＲＯＭであるため、プログラムの改ざんは本来物理的に困難であるが、もし何らかの方法で改ざんされた場合でも、プログラムは暗号化されており、意味のある改ざんは困難である。従って正確にブートできた場合は、プログラムの改ざんがなかったと判断でき、暗号化ＲＯＭコード領域１４に格納されているプログラムは絶対的に信頼できるものといえることになり、この状態をプログラムの基本的な信頼点として定義することが可能となる。
【００３９】
なお、暗号化ＲＯＭコード領域１４については、６４ビット単位の暗号化を行うＤＥＳ

（データ・エンクリプション・スタンダード）方式よりも秘匿強度の大きいＡＥＳ（アドバンスド・エンクリプション・スタンダード）方式を用いる場合などはプロセッサ内部ではなく外付けとすることもできる。その場合には命令コードのＮＯＰ、データパターンのＡＬＬ０／ＡＬＬ１など、頻発するパターンに対応して暗号鍵の推定を可能とさせないように、同一の平文に対して常に同一の暗号文が出力されるＥＣＢ（エレクトリック・コードブック）以外のモードを使用することも可能である。

【００４０】

続いてステップＳ２で暗号処理ブロック１２内に備えられ、後述する鍵テーブル（メモリ）に対する操作処理、ＣＰＵ固有鍵１５を用いた公開鍵や秘密鍵の生成などを行う鍵管理処理、およびコード認証処理ブロック１３のセットアップなどが実行され、これらの処理の内容は同様な信頼点として定義される。

【００４１】

続いてステップＳ３で二次記憶１８に格納されているプログラムに対する認証処理が行われる。この第１の実施例ではオペレーティングシステム（ＯＳ）を含む一般のプログラムは、ハードディスクやネットワーク経由など二次記憶１８上に格納されており、これらのプログラムに対する認証処理が実行される。この認証処理については、さらに後述する。

【００４２】

前述の鍵テーブル操作処理などを実行するためのプログラム群はライブラリィ化され、鍵処理モニタと呼ばれる。暗号処理ブロック１２、コード認証処理ブロック１３、およびＣＰＵ固有鍵１５等のセキュアハードウェア２０に対するアクセスは、ステップＳ４でこの鍵処理モニタの動作している区間だけに制限される。この鍵処理モニタが動作し、セキュアハードウェア２０にアクセス可能となる状態をアクセスレベル１と呼ぶことにする。アクセスレベル１は、プログラムカウンタが、固定領域にあるステップＳ４の鍵処理モニタのアドレスを指しているかどうかの監視を行うハードウェアにより実現する。

【００４３】

このアクセスレベル１に比較して前述のＯＳを含む一般のプログラムによる動作はアクセスレベル２、またはアクセスレベル３に分類される。第１の実施例ではＯＳはアクセスレベル２に分類され、ステップＳ５でＯＳなどの起動が行われると、ステップＳ６で認証済みプログラムの実行が行われる。アクセスレベル２における認証済みプログラムは、アクセスレベル１におけるステップＳ４、すなわち鍵処理モニタに対し、鍵処理を依頼する事が出来、自空間の暗号化、データの暗号化や復号など鍵処理モニタ経由で間接的に行うことができる。このようにＣＰＵ外部からのプログラムであっても、認証されたものについてはアクセスレベル２として位置づけられ、鍵処理を行うことが出来るが、公開鍵以外のすべての鍵またはセキュアハードウェア２０には直接アクセスすることは出来ないため、レベル２のプログラムに何らかの障害が生じたとしても公開鍵を除く鍵情報が外部に流出することはない。

【００４４】

アクセスレベル３における未認証のプログラムの実行は、ステップＳ７においてステップＳ５のＯＳなどの起動の後に行われるが、このアクセスレベル３のプログラムは公開鍵以外のすべての鍵へのアクセス、および鍵処理モニタへの鍵処理の依頼などを行うことは一切できないものとする。なおステップＳ４からステップＳ７の処理は各アクセスレベルのプログラム間のプロセス間通信を利用して実行される。

【００４５】

以上説明したように第１の実施例においては、まずプロセッサの起動時に暗号化ＲＯＭコード領域１４に格納されたプログラムを用いて行われるブート処理の成功の時点でプログラムの基本的な信頼点が確立され、その後その信頼点を用いてＯＳを含む各種のプログラムの認証を行いながら信頼できるプログラム範囲を拡大していくことによって、システムのセキュリティレベルをプロセッサ自らが段階的に向上させるという目的が達成される。また運用開始後は認証単位毎にコードやデータの暗号化を行うことも可能となり、プロ

グラムの間の秘密性保持に関しても十分な信頼性を維持することができる。なおこの第１の実施例では、アクセスレベル１の処理をプロセッサのコアが実行するソフトウェアとして実現する方式を説明したが、このレベル１の処理の一部、または全てをマイクロコード、あるいはワイヤードロジックとして実現することも可能である。

【００４６】

図４は、図２のコード認証処理ブロック１３と暗号処理ブロック１２による処理の概要を示すフローチャートである。同図においてまずステップＳ１０のコード認証処理ブロックによる処理に続いて、ステップＳ１１で暗号処理ブロックによる処理が行われるものとする。

【００４７】

図４においてまずステップＳ１２で、例えば主記憶１７、または二次記憶１８に格納されているプログラムに対するコード認証処理が実行される。この処理の詳細については後述する。そしてステップＳ１３で認証が成功したか失敗したかが判定され、失敗した場合にはステップＳ１４でコード実行に対する停止処理が実行される。

【００４８】

認証が成功した場合には、暗号処理ブロックによる処理が開始され、ステップＳ１６で暗号化のための鍵が、例えばページ単位に指定されているか否かが判定され、指定されていない場合にはステップＳ１７で乱数生成器などを用いてランダム鍵が生成され、指定されている場合にはステップＳ１８でその指定鍵が取り出される。ここで、鍵が指定されていない場合とは、新規にそのページが生成される場合などが含まれ、鍵が指定されている場合とは、生成されたページが一度ページアウト後、再度ページインする場合や、外部からの暗号化ページの格納などが含まれる。鍵が確定後、ステップＳ１９で暗号化ページエントリ、すなわち後述するトランスレーション・ルックアサイド・バッファ（ＴＬＢ）内のページ・テーブル・エントリ（ＰＴＥ）が生成され、暗号化ページが割り当てられてコードまたはデータの暗号化が行われる。

【００４９】

図５は、同一プロセスの命令領域とデータ領域とにそれぞれ異なる暗号鍵が割り当てられてコードの暗号化が行われる場合の、コード認証とその暗号化処理の全体フローチャートである。同図においてステップＳ１０、すなわちコード認証処理ブロックによる処理は図４における場合と同様である。

【００５０】

図５においてコード認証が成功すると、ステップＳ２１で命令領域に対する鍵としての命令鍵の指定があるか否かが判定され、指定がない場合にはステップＳ２２でランダム鍵が生成され、指定がある場合にはステップＳ２３でその指定鍵が取り出され、ステップＳ２４でランダム鍵、または指定鍵が使用されて、暗号化命令ページ・テーブル・エントリ、すなわちＰＴＥが生成され、暗号化ページが命令領域に割り当てられて命令領域の暗号化が行われる。

【００５１】

その後ステップＳ２６でデータ領域に対する暗号鍵としてのデータ鍵の指定があるか否かが判定され、指定がない場合にはステップＳ２７でランダム鍵が生成され、指定がある場合にはステップＳ２８で指定された鍵が取り出され、ステップＳ２９でデータに対するページ・テーブル・エントリが生成され、暗号化ページが割り当てられてデータ領域に対する暗号化が実行される。

【００５２】

続いて第１の実施例における暗号鍵の取得動作について図６から図９を用いて説明する。図６、および図７は、暗号鍵取得動作例（その１）におけるプロセッサ内部の構成例とその処理のフローチャートである。この例ではプロセッサ固有のＲＳＡ秘密鍵が予め安全な方式によってプロセッサ内部に保持されており、対応するＲＳＡ公開鍵は何らかの方法によってプロセッサの外部に出力され、外部から与えられる暗号鍵はこの公開鍵によって暗号化されているものとする。すなわち、例えばページ単位の暗号化、および復号用の暗

号鍵は共通鍵であり、その秘密性保持のために公開鍵による再暗号化が必須のものとなっている。

【００５３】

図６は、プロセッサ内部への暗号鍵設定処理実行のためのプロセッサ１０の構成を示し、プロセッサの内部には必要なブロックとして暗号鍵設定部２１、復号部２２、プロセッサ固有ＲＳＡ秘密鍵２３、およびトランスレーション・ルックアサイド・バッファ（ＴＬＢ）２４が備えられ、ＴＬＢの内部には前述のページ・テーブル・エントリ（ＰＴＥ）に相当する論理アドレステーブル２５、物理アドレステーブル２６、および鍵テーブル２７が備えられている。そして暗号鍵設定部２１に対して、プロセッサ固有のＲＳＡ公開鍵で暗号化された暗号鍵を含む暗号鍵設定要求が外部から与えられる。

【００５４】

図７は、暗号鍵取得処理のフローチャートである。同図において処理が開始されると、まずステップＳ３１で暗号鍵設定部２１によって暗号鍵設定要求が受付けられ、ステップＳ３２で復号部２２によって受け取った暗号化暗号鍵がプロセッサ固有ＲＳＡ秘密鍵２３を用いて復号され、ステップＳ３３で暗号鍵設定部２１によって復号された暗号鍵がＴＬＢ２４の内部の鍵テーブル２７に格納されて処理を終了する。

【００５５】

図８は、暗号鍵取得動作例（その２）におけるプロセッサの構成例である。同図においては、その１の例における図６と比較すると、プロセッサ１０の内部に復号部２２に代わって署名検証部２８が備えられ、またプロセッサ固有ＲＳＡ秘密鍵２３の代わりに認証局公開鍵としての認証局証明書２９が格納されている点が異なっている。この認証局証明書２９はその不正な置き換えが不可能なようにプロセッサ内部に記録されているものとし、暗号鍵設定部２１に対しては、認証局の署名が付けられた暗号鍵を含む暗号鍵設定要求が与えられるものとする。

【００５６】

図９は、暗号鍵取得動作例（その２）における処理のフローチャートである。同図において処理が開始されると、まずステップＳ３６で暗号鍵設定部２１によって暗号鍵が署名とともに受付けられ、ステップＳ３７で署名検証部２８によって受け取った暗号鍵が署名と認証局公開鍵を用いて検証され、ステップＳ３８で検証が成功したか否かが判定され、成功した場合にはステップＳ３９で暗号鍵設定部２１によって受け取った暗号鍵がＴＬＢ２４の内部の鍵テーブル２７に格納された後に、また検証が失敗した場合には直ちに処理を終了する。なおさらに暗号鍵の信頼性を向上させるためにその１の動作例、すなわち暗号鍵の秘密性保持と、動作例その２、すなわち暗号鍵の身元証明とを組み合わせて実施することも当然可能である。

【００５７】

図１０は、第１の実施例において暗号化された命令領域の命令を実行中に不正命令を検出した場合の不正命令対応処理のフローチャートである。同図においてステップＳ４１で不正命令が検出されると、ステップＳ４２でその不正命令が暗号化ページ内の命令であるか否かが判定され、非暗号化ページ内の命令である場合にはステップＳ４３で通常の不正命令対応処理が行われるが、暗号化ページ内の命令であると判定されるとステップＳ４４で命令改ざんが行われたと判定され、その改ざんに対する命令改ざん対応処理としてプロセスのロックダウンや、保留中処理のキャンセルなどの処理が実行され、その命令コードの実行は停止される。

【００５８】

図１１は、図５で説明したように同一プロセスの命令領域とデータ領域とに異なる暗号鍵が割り当てられている場合に、例えばデータ領域に格納されている命令コードの実行に先立ってその命令が不正命令として検出されることを防ぐための鍵付け替え処理のフローチャートである。このような命令コードのデータ領域への格納はプログラムドＩＯ（ＰＩＯ）、すなわちプログラムによる命令のコピーの実行時に起こるものである。

【００５９】

図１１においてまずステップＳ４６で命令コードがＰＩＯによってデータ領域にコピーされたものとし、ステップＳ４７で鍵付け替え処理が起動される。この処理ではステップＳ４８で命令が格納されていたデータページに対応するデータＰＴＥが読み出され、ステップＳ４９でそのエントリに格納されていた暗号鍵が取り出された後にそのＰＴＥが消去され、ステップＳ５０でそのデータＰＴＥの内容、すなわち暗号鍵を用いて、その暗号鍵が、例えば図６の鍵テーブル２７に格納された命令ＰＴＥが生成され、ステップＳ５１でその命令ＰＴＥがＴＬＢに書き込まれた後にステップＳ５２で命令がコピーされた領域への分岐が行われ、その後の処理、すなわちコピー領域に格納された命令の実行処理が行われる。

【００６０】

このように第１の実施例では、図２で説明したようにプロセッサ１０の内部に実行ユニットとキャッシュを含むコア１１が１つだけ備えられ、そのコア１１がセキュアコアとしてセキュアプロセッサとしての動作の中心的役割を果たすものとしたが、マルチプロセッサシステム、あるいはマルチコアシステムと呼ばれるシステムでは、例えば複数のコアを、セキュア動作を実行するセキュアコアと、ノーマル動作を実行するノーマルコアとに分類し、処理を分担させることも可能である。そのようなプロセッサシステムを次に第２の実施例として説明する。

【００６１】

図１２は、第２の実施例としてのプロセッサの基本構成ブロック図である。同図においては第１の実施例を示す図２と比較すると、コア１１に代わってセキュアコア３１とノーマルコア３２とが備えられ、またこれらの２つのコア３１、３２と暗号処理ブロック１２、およびコード認証処理ブロック１３との間にバスインタフェース３３が備えられ、またセキュアコア３１と暗号処理ブロック１２との間では鍵制御が行われ、セキュアコア３１とコード認証処理ブロック１３との間で認証制御が行われ、さらにＣＰＵ固有鍵１５はセキュアコア３１だけに接続されている点が異なっている。すなわち第２の実施例では、図３で説明したセキュアハードウェア２０としての暗号処理ブロック１２、コード認証処理ブロック１３、およびＣＰＵ固有鍵１５に対する制御がセキュアコア３１だけによって行われるという点に基本的な特徴がある。

【００６２】

この第２の実施例では、セキュアハードウェア２０に対するアクセスはセキュアコア３１のみに限定される。第１の実施例ではセキュア動作としての図２のステップＳ４における鍵処理モニタの動作においてユーザのソフトウェアが介在する余地があり、そのため前述のようにプログラムカウンタのハードウェア監視によるアクセス制限などが行われるが、第２の実施例ではこのようなソフトウェアの介在がなく、ソフトウェアバグによる問題も発生しない。

【００６３】

また、第１の実施例では例えば同一のコアを使用して時分割方式でアクセスレベル間の共有を行うことも必要となるが、第２の実施例ではコアが別になるため、アクセスレベル切替時点におけるレジスタクリアなどのソフトウェアに対する要求処理量も少なくなる。

【００６４】

図１３は、第２の実施例におけるプロセッサの基本処理フローチャートである。第１の実施例における図３と比較して異なる処理を中心にその処理を説明する。図１２におけるセキュアコア３１とノーマルコア３２とが基本的に対等な関係にあるものとすると、電源が投入された時点でそれぞれのコアは、暗号化ＲＯＭコード領域１４に格納されたプログラムを用いてブート処理を実行する。すなわち前述のようにステップＳ１でセキュアコアによるブート処理においてＣＰＵ固有鍵１５を用いて暗号化されたプログラムが復号され、ブート処理が実行される。このブート処理が成功した場合には、前述のようにその状態がプログラムの基本的な信頼点として定義され、セキュアコアはその後、例えばもっぱら鍵処理モニタとしての動作を継続することになる。

【００６５】

これに対してノーマルコア３２は、主としてＯＳなどのアクセスレベル２に相当する処理を担当することになる。図１３のステップＳ３ではセキュアコア側で二次記憶上のプログラムの認証処理が実行されるのに対応してノーマルコア３２側で電源がオンとなり、ステップＳ５５で暗号化ＲＯＭコード領域１４内のプログラムによるブート処理が実行される。この時点では、セキュアコア３１によって暗号化ＲＯＭコード領域１４内のプログラムは絶対的に信頼可能であることが判明しているものとすると、ノーマルコア側のブート処理は基本的に問題なく終了し、ステップＳ５におけるＯＳなどの起動処理が続いて実行されることになる。

【００６６】

図１４は、第２の実施例においてセキュアコアとノーマルコアとが対等な関係でなく、セキュリティの厳密な適用を行うために、セキュアコアによってノーマルコアの制御が行われる場合のプロセッサの構成ブロック図である。セキュアコア３１とノーマルコア３２とが基本的に対等な関係にある図１２と比較してプロセッサの構成要素は同一であるが、セキュアコア３１からノーマルコア３２に対してコア制御信号が与えられる点が異なっている。このコア制御信号の具体的な例としてはリセット信号や割込み信号などが挙げられる。

【００６７】

図１５は、図１４のプロセッサによる全体処理のフローチャートである。同図においてセキュアコア３１側では、ステップＳ１のブート処理に続いてステップＳ５７でステップＳ２の代わりに、鍵テーブル操作処理、鍵管理処理、および認証処理ブロックのセットアップに加えてシステム監査が行われる。このシステム監査では、システム構成の変更の有無や、二次記憶上のプログラムの変更の有無などの検証が行われ、システムのセキュリティ機能、およびシステム構成に問題がないことが確認される。

【００６８】

その後ステップＳ５８でセキュアコア３１側からノーマルコアの起動が行われ、これに対応してノーマルコア３２側では、ステップＳ５９で暗号化ＲＯＭコード領域１４に格納されたプログラムを用いた起動処理が実行される。その後の処理は、例えば図３における場合と同様である。

【００６９】

図１６は、図１４のプロセッサにおけるセキュアコア３１によるノーマルコア３２の制御処理の１つとしてのノーマルコアの停止制御処理の説明図である。同図において、例えばノーマルコア側でステップＳ６の認証済みプログラムの実行において、セキュアコア３１側にデータなどの認証のための鍵処理が依頼され、ステップＳ４における鍵処理モニタの動作において認証の失敗や、セキュリティ基準への違反が検出された場合には、セキュアコア３１側からの指示によってノーマルコア３２による処理、すなわちステップＳ６の認証済みプログラムの実行、およびステップＳ７の未認証プログラムの実行が停止される。

【００７０】

図１７は、図１４におけるセキュアコア３１によるノーマルコア３２の制御処理のフローチャートである。セキュアコア３１側ではステップＳ６１でブート処理が実行され、ステップＳ６２でその処理が完了すると、ノーマルコア３２側への起動制御が行われ、ステップＳ６３でノーマルコアの起動が行われ、ステップＳ６４で公開鍵以外の鍵や認証処理を必要としない通常処理がノーマルコア側で実行される。セキュアコア３１側では、常にステップＳ６５でノーマルコア３２側から送られる監視情報を用いた認証・監視処理を行っており、ステップＳ６６でエラー発生があったか否かを判定し、発生がない場合にはステップＳ６５以降の処理を続行し、エラー発生があった場合にはノーマルコア３２側に対する停止要求、または割込みを行ってノーマルコア３２側の処理を停止させる。セキュアコアによるノーマルコアの制御については、前述のようにリセット信号をその例として用いるものとしたが、その他の方法としてはＣＰＵに対するＮＭＩ（マスク不可割込み）を利用することもできる。

【００７１】
図１８は、第２の実施例において鍵生成機構を有するプロセッサの構成ブロック図である。同図において図１２の構成に加えて、鍵生成機構３４を備えている点が異なっている。

【００７２】
図１９は、第２の実施例におけるセキュアコアによる鍵の生成と、生成された鍵を用いた暗号化処理の説明図である。同図においてプロセッサ内のセキュアコア３１はＣＰＵ固有鍵１５、および鍵生成機構３４を用いて公開鍵Ｋｅ、Ｎ、および秘密鍵Ｋｄ３５を生成し、例えばノーマルコア３２を経由して公開鍵Ｋｅ、Ｎを外部に通知するものとする。このときノーマルコア３２側には秘密鍵Ｋｄが渡されることはなく、ノーマルコア３２は前述のように公開鍵以外の鍵処理を実行することはできない。

【００７３】
そして例えば外部において公開鍵と原文Ｐを用いて暗号化された暗号文Ｃがノーマルコア３２に入力されると、ノーマルコア３２は秘密鍵Ｋｄを保持していないためセキュアコア３１に復号処理を依頼する。セキュアコア３１は秘密鍵Ｋｄを用いて原文Ｐを復号する。

【００７４】
次に本発明の第３の実施例について説明する。図２０は、第３の実施例におけるプロセッサの基本構成ブロック図である。同図においてプロセッサ４０は実行ユニット４１、ロードストアユニット４２、暗号化回路４３、および復号化回路４４を備え、またロードストアユニット４２は、キャッシュメモリ４５とメモリ管理ユニット４６を備えている。

【００７５】
この第３の実施例は前述の第１、および第２の実施例と同様に基本的にはセキュアな動作を実行するプロセッサであるが、プロセッサ４０の内部で第１の実施例における暗号処理ブロックと同様にストアデータの暗号化を行う暗号化回路４３と、フェッチされる命令を含むロードデータを復号する復号化回路４４とに対して、実行ユニット４１からストア用の暗号化鍵とロード用の復号化鍵の指定が行われるところに基本的な特徴がある。

【００７６】
図２０の第３の実施例において、実行ユニット４１からロードストアユニット４２に対しては、コマンドとストアデータとしての平文が与えられ、ロードストアユニット４２から実行ユニット４１に対しては平文としてのロードデータが与えられる。このうちコマンドはロードストアユニット４２を介して、例えば図２で説明した主記憶や二次記憶に与えられるが、平文としてのストアデータは暗号化回路４３に与えられ、暗号化されたストアデータとして、例えば主記憶に出力され、また例えば主記憶から入力される暗号化されたロードデータは復号化回路４４によって復号され、平文としてのロードデータとしてロードストアユニット４２に与えられる。

【００７７】
図２１は、第３の実施例において暗号化鍵、復号化鍵を格納する鍵テーブルメモリを備えるプロセッサの構成ブロック図である。同図において鍵テーブルメモリ４７はストアデータ暗号化用の暗号化鍵を格納するものであり、また鍵テーブルメモリ４８はロードデータを復号するための復号化鍵を格納するものである。実行ユニット４１からは鍵テーブルメモリ４７に対してストア用の鍵番号指示とストア用の暗号化鍵の更新の指示が与えられ、また鍵テーブルメモリ４８に対してはロード用の鍵番号の指示とロード用復号化鍵の更新の指示が与えられる。鍵テーブルメモリの構成については後述する。

【００７８】
図２２は、第３の実施例においてフェッチされる命令の復号を行うための命令フェッチ用復号化鍵を格納する鍵テーブルメモリを備えるプロセッサの構成ブロック図である。同図において実行ユニット４１は、例えば主記憶に格納されている命令をフェッチする命令アクセス状態の処理を行っており、例えば主記憶からのロードデータとして命令フェッチデータが復号化回路４４に与えられ、このとき実行ユニット４１は鍵テーブルメモリ４８

に対して命令アクセス状態フラグを与え、復号化回路４４は鍵テーブルメモリ４８から出力される命令フェッチ用復号化鍵を用いて命令フェッチデータの復号を行い、平文としての命令フェッチデータはロードストアユニット４２を介して実行ユニット４１に与えられる。実行ユニット４１からは鍵テーブルメモリ４８に対して、必要に応じて命令フェッチ用復号化鍵の更新の指示が与えられる。

【００７９】

図２３は、第３の実施例において鍵テーブルメモリに対して使用すべき鍵番号の指示を与える鍵選択レジスタを備えるプロセッサの構成ブロック図である。同図においてストア用暗号化鍵を格納する鍵テーブルメモリ４７と実行ユニット４１の間にストア用鍵番号指示を鍵テーブルメモリ４７に与える鍵選択レジスタ５１が、またロード用復号化鍵を格納する鍵テーブルメモリ４８と実行ユニット４１の間に鍵テーブルメモリ４８にロード用鍵番号指示を与える鍵選択レジスタ５２が備えられる。実行ユニット４１から鍵選択レジスタ５１に対してはストア用鍵選択レジスタの更新指示が与えられ、また鍵選択レジスタ５２に対してはロード用鍵選択レジスタの更新指示が与えられる。

【００８０】

すなわち図２１では実行ユニット４１からは実行命令のそれぞれに対応して鍵番号の指示が出力されるのに対して、図２３では命令のある区間毎にレジスタの更新指示が与えられ、次の更新指示が与えられるまでは同一の鍵を使用して暗号化／復号化が行われる。なお実行ユニットから鍵テーブルメモリに対して直接に鍵番号指示を与える経路と、鍵選択レジスタを経由した経路との両方を設け、例えば実行ユニット４１が実行命令に対応してどちらの経路の指示を用いるべきかの信号を鍵テーブルメモリに与えるような構成も当然可能である。

【００８１】

図２４は、第３の実施例において実行ユニットの命令アクセス状態に対応する鍵選択レジスタを備えるプロセッサの構成ブロック図である。同図においては図２２におけると同様に、実行ユニット４１は、例えば主記憶から命令フェッチを行うべき命令アクセス状態であり、実行ユニット４１から鍵選択レジスタ５２に対して命令アクセス状態フラグが与えられ、鍵選択レジスタ５２は、それに対応して命令フェッチ用の鍵番号指示を、命令フェッチ用復号化鍵を格納する鍵テーブルメモリ４８に与える。

【００８２】

図２５は、第３の実施例における鍵テーブルメモリの構成例の説明図である。同図において鍵テーブルメモリには、暗号鍵とその属性とが対応して格納されており、実行ユニット４１から直接に、あるいは鍵選択レジスタを介して鍵番号の指示が与えられ、その鍵番号がリードアドレスとして用いられ、暗号化鍵、または復号化鍵が暗号化回路４３、または復号化回路４４に対する暗号化方式の指定情報や、暗号化の可否を指示する属性データとともに与えられる。また実行ユニット４１から与えられる鍵更新番号指示がライトアドレスとして用いられ、鍵更新データの書込みが行われる。

【００８３】

各エントリの属性データはエントリの有効／無効、暗号化のオン／オフ、暗号化方式および暗号化モードなどを示し、暗号鍵は指定された暗号化方式に依存するものとなる。なお暗号化のオン／オフを指示するデータは後述するように暗号化／復号化を行うことなく平文データをロード、ストアする場合の指示に対応する。

【００８４】

図２６は、第３の実施例における暗号化回路、および復号化回路の構成例の説明図である。例えば図２０の復号化回路４４は、復号パイプ５５とバスアービタ５７とによって基本的に構成され、復号パイプ５５は実行ユニット５１からのコマンドバッファ５９を介したコマンド情報の入力に対応して動作する。復号パイプ５５は、例えば主記憶からバスを介して入力される暗号データを平文データに復号するためのＮ段のパイプであり、このＮ段のパイプは共通鍵暗号処理一段の概念的な例である処理５６がＮ段接続されたものである。復号パイプ５５から出力される平文データは、バスアービタ５７を介して、例えば図

２０のキャッシュメモリ４５に格納される。

【００８５】

暗号化回路４３は暗号パイプ６０とバスアービタ６１とによって基本的に構成される。暗号パイプ６０に対しては、キャッシュメモリ４５から例えば３２ｂｉｔの平文データが与えられ、実行ユニット４１から指定される暗号化の鍵を用いてＮ段のパイプによって暗号化された暗号データが出力され、その暗号データはバスアービタ６１を介して、例えば主記憶に接続されたバスに与えられる。暗号パイプ６０の動作は復号パイプ５５と同様にコマンドバッファ５９を介して実行ユニット４１から与えられるコマンド情報によって制御される。また暗号パイプ６０の各段における処理の基本構造は復号パイプ５５におけると同様である。さらに暗号方式としてはＡＥＳ１２８、ＤＥＳ、およびＳＣ２０００などの各種の暗号化方式を用いることができる。なおＡＥＳ方式としてはＡＥＳ１９２、ＡＥＳ２５６の仕様も規定されている。

【００８６】

なお例えばバスアービタ６１は主記憶装置や二次記憶装置に接続されたバスに対する調停を行うものであり、本発明におけるセキュアプロセッサの動作とは基本的に無関係である。

【００８７】

図２７は、第３の実施例において全てのデータを暗号化するのではなく、一部のデータを平文データのままで、例えば主記憶との間で入出力するためのデータ追い越し機能付き暗号化回路と復号化回路の構成を示すブロック図である。同図において暗号化回路と復号化回路の基本的な構成は図２６におけると同様であるが、例えば暗号化回路側では、キャッシュメモリ４５から与えられる平文データのうちで暗号化の必要がないデータについては暗号パイプ６０を介することなく、直接にバイパスセレクタ６３にそのデータが与えられ、暗号パイプ６０から出力された暗号化データとともに複数のバイパスバッファ６４の何れかに格納され、バスアービタ６１を介して、例えば主記憶に接続されたバスに与えられる。

【００８８】

バイパスセレクタ６３による平文データ、または暗号化データの選択もコマンドバッファ５９を介した実行ユニット４１からのコマンド情報によって制御される。暗号パイプ６０による処理は時間を要するため、暗号化の必要のない平文データは、バイパスセレクタ６３の制御によって暗号化データを追い越して主記憶側に与えられることが可能となる。なお図２７で暗号化に必要な鍵は鍵レジスタ６９を介して暗号パイプ６０に与えられている。

【００８９】

例えば主記憶に接続されたバスからのデータのうち、暗号化されていない平文データは復号パイプ５５を経由することなく直接にバイパスセレクタ６６に与えられ、復号パイプ５５によって復号された平文データとともにバイパスセレクタ６６によって複数のバイパスバッファ６７のいずれかに格納され、バスアービタ５７を介してキャッシュメモリ４５に出力される。

【００９０】

図２８は、第３の実施例においてライトスルーキャッシュ方式に対応するためのリード・モディファイ・ライト方式の説明図である。キャッシュメモリ４５がライトスルー方式を採用している場合には、ストア時にキャッシュミスが発生するとキャッシュメモリ４５にそのデータが格納されず主記憶にそのままデータの格納が行われる。ストアすべきデータが、例えば１バイトしか無いような場合には、主記憶に１バイトデータの格納が行われる。しかしながら第３の実施例においては、基本的にストアデータは暗号化回路４３によって暗号化された後に主記憶に格納されるものであり、一般に暗号化の処理においてはストアデータとしてある程度の量のデータを必要とし、１バイトだけのデータを暗号化して主記憶に格納したとしてもその正しい復号は困難である。

【００９１】

図２８のロードストアユニット４２は、例えば１バイトのデータを主記憶に格納する必要があるときに、暗号化の処理に必要な長さのデータを主記憶からロードし、ロードされたデータとストアすべき１バイトのデータを結合し、結合されたデータを暗号化して主記憶に格納するリード・モディファイ・ライト動作を行うものである。

【００９２】

すなわち、例えば１バイトのデータをキャッシュにストアすべきキャッシュストア命令（１）が（２）でキャッシュミスと判定されると、（３）でキャッシュメモリ４５から主記憶に対してコマンドとしてのロードが発行され、（４）で復号化回路４４を介して平文のロードデータがリード・モディファイ・ライト（ＲＭＷ）バッファ７１に格納され、（５）でストアすべきデータがＲＭＷバッファ７１に与えられ、（６）でストアすべきデータとロードデータとを結合したデータが暗号化回路４３に与えられ、（７）でコマンドとしてのストアが主記憶に対して発行される。

【００９３】

次に本発明の第４の実施例について説明する。この第４の実施例では、第３の実施例において暗号化回路によって使用される暗号化鍵、および復号化回路によって使用される復号化鍵の例えば鍵番号が実行ユニット４１によって指定されるのに対して、実行ユニット４１による命令実行時にストア、またはロード対象となるデータのアクセスアドレスが実行ユニット４１によって指定され、そのアドレスに応じて暗号化鍵、または復号化鍵が選択される点が異なっている。

【００９４】

図２９は、第４の実施例におけるプロセッサの基本構成ブロック図である。同図においてプロセッサ４０は実行ユニット４１、暗号化回路４３、復号化回路４４に加えて、実行ユニット４１から与えられるアドレスに対応してストア用暗号化鍵を暗号化回路４３に与え、ロード用復号化鍵を復号化回路４４に与える鍵テーブルメモリ７３を備えている。

【００９５】

図３０は、実行ユニットから指定されるストアデータ、またはロードデータの論理アドレスに対応して鍵が選択されるプロセッサの構成ブロック図である。同図においてプロセッサ４０は、図２９と異なってストア用暗号化鍵を格納する鍵テーブルメモリ７４と、ロード用復号化鍵を格納する鍵テーブルメモリ７５とを備えるとともに、キャッシュメモリ４５、メモリ管理ユニット４６を備えるロードストアユニット４２を、例えば図２０とおけると同様に備えている。実行ユニット４１からロードストアユニット４２に与えられるアドレス、すなわちストアデータ、またはロードデータのアドレスは論理アドレスであり、この論理アドレスが鍵テーブルメモリ７４、または７５に与えられてストア用暗号化鍵、またはロード用復号化鍵が選択され、それぞれ暗号化回路４３、または復号化回路４４に与えられる。また実行ユニット４１から鍵テーブルメモリ７４に対してはストア用暗号化鍵の更新の指示が、鍵テーブルメモリ７５に対してはロード用復号化鍵の更新の指示が与えられる。

【００９６】

図３１は、第４の実施例においてデータの物理アドレスに対応して鍵が選択されるプロセッサの構成ブロック図である。同図を図３０と比較すると、鍵テーブルメモリ７４と７５のそれぞれに対してロードストアユニット４２からストアデータの物理アドレス、またはロードデータの物理アドレスが与えられ、そのアドレスに対応してストア用暗号化鍵が暗号化回路４３に与えられ、ロード用復号化鍵が復号化回路４４にそれぞれ与えられることになる。

【００９７】

図３２は、第４の実施例における鍵テーブルメモリの構成図である。同図を第３の実施例における図２５と比較すると、実行ユニット側からデータのアクセスアドレスとして０ｂｉｔ目から３１ｂｉｔ目までの３２ｂｉｔが与えられると、そのアドレスをリードアドレスとして格納されている暗号鍵が選択され、暗号化回路４３、または復号化回路４４に暗号属性とともに与えられる。メモリのリードアドレスとして４ｋバイト毎に異なる鍵が

使用される場合には、アドレスの１２ｂｉｔ目から３１ｂｉｔ目が使用されて暗号鍵の選択が行われる。なおこの４ｋバイトは後述するように例えば主記憶における１ページの大きさに相当する。またこの４ｋバイトを暗号化のアドレス単位と呼ぶことにすると、鍵テーブルメモリのエントリデータには全エントリ数×アドレス単位分を除いたアドレスタグが含まれる。例えば全エントリ数が３２（５ビット）であれば、アドレスの１７ビット目から３１ビット目までがタグとなる。

【００９８】

図３３は、第３の実施例において複数のウエイの構成を持つ鍵テーブルメモリの説明図である。同図において鍵テーブルメモリは、鍵テーブル１から鍵テーブル４までの複数のテーブルによって構成されており、実行ユニット側から与えられるアクセスアドレスに対応して４つのテーブルのうちの何れかに格納されている鍵と暗号属性とが選択され、暗号化回路４３、または復号化回路４４に与えられる。

【００９９】

図３４は、連想記憶方式を用いる鍵テーブルメモリの構成例の説明図である。同図においてはアクセスアドレス３２ｂｉｔが比較選択器７７によって格納されている暗号鍵のそれぞれに対応する対象アドレス範囲の何れかに分類され、その範囲に対応する暗号鍵が選択されて、暗号属性とともに暗号化回路４３、または復号化回路４４に与えられる。なお図３４では全エントリ数に無関係にアドレス単位分を除いたアドレスタグがエントリに含まれる。アドレス単位が４ｋバイトのときには、アドレスの１２ビット目から３１ビット目がタグとなる。

【０１００】

図３５は、第４の実施例においてデータの論理アドレス、または物理アドレスの何れかに対応して鍵を選択するプロセッサの構成ブロック図である。同図において実行ユニット４１からはデータの論理アドレスが、またロードストアユニット４２からは物理アドレスがそれぞれ鍵テーブルメモリ７４、７５に与えられ、また実行ユニット４１から鍵テーブルメモリ７４に対してはストアデータの論理アドレスと物理アドレスの選択指示が与えられ、鍵テーブルメモリ７５に対してはロードデータの論理アドレスと物理アドレスの選択指示が与えられる。そしてこれらの選択指示に対応して論理アドレス、または物理アドレスの何れかに対応する鍵が選択されて暗号化回路４３、復号化回路４４に与えられる。

【０１０１】

図３６は、鍵テーブルメモリに対して論理アドレスと物理アドレスの選択指示を与える鍵選択レジスタを備えるプロセッサの構成例である。同図を図３５と比較すると実行ユニット４１と鍵テーブルメモリ７４、７５との間にそれぞれ鍵選択レジスタ７８、７９が備えられ、それぞれストアデータに対する論理アドレスと物理アドレスの選択指示、ロードデータに対する論理アドレスと物理アドレスの選択指示を鍵テーブルメモリ７４、７５に出力する。実行ユニット４１から鍵選択レジスタ７８、７９に対してはそれぞれ鍵選択レジスタの更新指示が与えられる。

【０１０２】

図３７は、図３５と図３６における鍵テーブルメモリの構成例を示す。同図において鍵テーブルメモリは、物理アドレス鍵テーブルと論理アドレス鍵テーブルとを備え、それぞれ物理アドレスと論理アドレスに対応して物理鍵、論理鍵を出力し、実行ユニット４１側からの鍵選択指示、または鍵選択レジスタからの選択指示に対応して物理・論理鍵選択部８１によって物理鍵、または論理鍵の何れかが暗号属性とともに暗号化回路４３、または復号化回路４４に出力される。

【０１０３】

図３８は、第４の実施例において鍵テーブルメモリの内容を鍵テーブルとしてロードストアユニット４２の内部のメモリ管理ユニット（ＭＭＵ）４６に備えたプロセッサの構成例である。

【０１０４】

図３９、および図４０は、このメモリ管理ユニット内の鍵情報の格納形式とキャッシュ

メモリアクセス方式の説明図である。一般的にＭＭＵ４６の内部のトランスレーション・ルックアサイド・バッファ（ＴＬＢ）には論理アドレスと物理アドレスとの対応が、例えば物理メモリ内の各ページに対応して各エントリに格納されているが、図３９ではＴＬＢの各エントリにそのページに対応する鍵情報を格納し、例えばデータアクセスアドレスが論理アドレスである場合には論理アドレスが一致するエントリが選択され、そのエントリに対応するデータの属性とアクセス属性とが属性チェック８３によってチェックされ、キャッシュコマンド生成８４によって生成されたコマンドがキャッシュメモリ４５に送られる。

【０１０５】

キャッシュメモリ４５側では、受け取ったコマンドの内容に対応してタグを検索し、キャッシュヒットの場合には直ちにデータ応答を実行ユニット４１側に返すことになるが、キャッシュミスの場合には暗号化回路４３、復号化回路４４を含む暗号化・復号化バスインタフェース８５に対してタグに対応するコマンドが発行される。このときエントリから読み出された鍵情報や物理アドレスが使用され、例えば主記憶からの応答データが復号化された後にキャッシュメモリに格納され、実行ユニット４１にデータ応答が返されることになる。

【０１０６】

図４０は、メモリ管理ユニット内にＴＬＢに代わるアドレス・マップ・レジスタ（ＡＭＲ）を設けた場合の鍵情報格納形式の説明図である。同図においてはＴＬＢの格納内容に対応する情報がメモリでなくレジスタに格納されており、例えばページサイズを可変とすることができ、大きなデータ領域を１つのエントリでカバーすることも可能となる。

【０１０７】

図４１は、ロードストアユニットの内部のメモリ管理ユニット（ＭＭＵ）が動作停止の状態、すなわちＯＦＦの状態において実行ユニット４１から暗号化鍵を暗号化回路４３に与え、復号化鍵を復号化回路４４に与えるプロセッサの構成例である。同図においてＭＭＵのＯＮ／ＯＦＦ信号は暗号化回路４３と復号化回路４４に与えられ、その信号がＯＦＦのときには暗号化回路４３、復号化回路４４は実行ユニット４１から与えられる鍵、ＯＮのときにはメモリ管理ユニット４６の内部のＴＬＢ８７、またはＡＭＲ８８から与えられる鍵を用いて暗号化、または復号化の処理を実行することになる。

【０１０８】

図４２は、図４１の暗号化回路、復号化回路における鍵の切り替え方式の説明図である。同図において暗号化回路、および復号化回路の構成は第３の実施例における図２６と基本的に同じであるが、鍵セレクタ９０が追加され、実行ユニットから与えられるＭＭＵＯＮ／ＯＦＦ信号の値に応じてＯＦＦの時には実行ユニットから与えられる鍵が、ＯＮの時にはＴＬＢ、またはＡＭＲから与えられる鍵の何れかが鍵セレクタ９０によって選択され、暗号パイプ６０、復号パイプ５５に与えられる。

【０１０９】

図４３は、第３の実施例、および第４の実施例における実行ユニットの入出力信号の説明図である。まず第３の実施例における図２０では、出力信号としてロード復号化鍵、ストア暗号化鍵、ストアデータ、およびコマンド、入力信号としてロードデータが必須の信号（○印）であり、アクセスアドレス、ロード・ストア状態信号などは構成的には存在する信号（△印）である。

【０１１０】

図２１に対しては、ロード復号化鍵の代わりにロード鍵番号指示、ストア暗号化鍵の代わりにストア鍵番号指示の出力信号が必須のものとなる。また図２１では、鍵テーブルメモリに対する更新が、実行ユニットから見てレジスタアクセスと等価であるため、レジスタ関連の入出力信号も必須となる。

【０１１１】

図２２に対しては、命令アクセス状態に対応する入出力信号が必要であり、出力信号として実行状態信号、入力信号として命令フェッチデータが必須となる。

図２３、図２４では、図２０、図２１に加え鍵選択レジスタを使用するため、レジスタ関連の入出力信号も必須となる。

【０１１２】

以下説明を簡略化し、特徴的な部分について説明すると図２１＋図２２＋図２３では３つの図を組み合わせた場合の入出力信号に加えて、スーパバイザ、あるいはユーザのいずれに対応するプロセスの実行であるかを示すスーパバイザ・ユーザ状態信号とコンテキスト、すなわちプロセスのＩＤ（識別子）のデータが追加された場合が示されている。これらスーパバイザ／ユーザ状態信号とコンテキストＩＤデータは、第３の実施例において実行ユニットから出力される鍵番号指示の信号に加えて暗号化鍵、復号化鍵の選択のために用いられる。

【０１１３】

図２９以降は第４の実施例に対応するものであり、データへのアクセスアドレスが必須の出力信号となるとともに、図３５、図３６では論理アドレスと物理アドレスの何れかを選択するための鍵選択指示信号が出力される。

【０１１４】

図３８に対しては、メモリ管理ユニットの内部のＴＬＢに鍵テーブルが併設されるために、レジスタ関係の信号が構成的には存在する信号となり、またスーパバイザ・ユーザ状態信号とコンテキストＩＤデータとが追加された場合が示されている。これら追加された信号も第３の実施例と同様にアクセスアドレスとともに暗号化鍵、復号化鍵の選択に用いられる。

【０１１５】

図４１に対しては、メモリ管理ユニット（ＭＭＵ）のＯＮ／ＯＦＦを示す状態信号の値に対応して実行ユニットから出力される鍵が用いられる場合と、例えばＴＬＢから出力される鍵が用いられる場合とがあり、スーパバイザ・ユーザ状態信号、コンテキストＩＤデータも追加されて、すべての入出力信号が必須のものとなっている。

【０１１６】

以上のように第３、第４の実施例ではデータや命令コードの暗号化／復号化用の鍵が実行ユニットから指定されるために、実行される命令に対応したレベルで暗号化処理を行うことも可能となる。また、鍵選択レジスタ、あるいはアクセスアドレスによる暗号化／復号化鍵の指定により、プログラム単位またはアクセス単位での暗号化処理も可能であり、様々な状況に応じて使い分けることが可能である。

【０１１７】

続いて本発明の第５の実施例について説明する。この第５の実施例としては、例えば第１の実施例としてのセキュアプロセッサのセキュア動作を実現するためのより詳細な構成を示し、その構成に対応させてプロセス（プログラム）の信頼点を拡大していくための認証鍵の設定や、プロセスの認証などの動作について詳細に説明する。

【０１１８】

図４４は、第５の実施例を説明するためのプロセッサ内の必要な機能構成図である。同図においてプロセッサ１００は、物理メモリ１０１、例えば図２では主記憶１７と、Ｉ／Ｏ装置１０２、例えば二次記憶１８と接続されている。

【０１１９】

プロセッサ１００は、物理メモリ１０１、およびＩ／Ｏ装置１０２へのアクセスを制御するメモリアクセス制御部１０５、実行すべき命令を解釈する命令解釈部１０６、実行コードの格納されているページの認証などを行う認証部１０７、例えば認証後のページの暗号化／復号などを行う暗号化／復号、署名生成／検証部１０８、プロセスの生成時にそのプロセス、すなわちコンテキストに対応するセキュアコンテキスト識別子を生成するセキュアコンテキスト識別子生成部１０９、プロセスの消滅時に対応する識別子を消滅させるセキュアコンテキスト識別子消滅部１１０、暗号化などに用いられるプロセッサ固有鍵１１１、例えば物理メモリ１０１に格納されている物理ページに対応する認証情報を格納する認証情報一次格納部１１２、メモリアクセス時に直接メモリアクセスを行うためのセキ

ュアＤＭＡ１１３を備えている。

【０１２０】

プロセッサ１００の内部には、さらに例えば図３９で説明したトランスレーション・ルックアサイド・バッファ（ＴＬＢ）１１４とコンテキスト情報格納部１１５が備えられている。ＴＬＢ１１４には、例えば物理ページに対応させて論理アドレスと物理アドレスとの対応などを示すページ・テーブル・エントリ（ＰＴＥ）１２２が格納され、コンテキスト情報格納部１１５にはプログラムカウンタ（の値を保持するカウンタ）１１７、セキュアコンテキスト識別子を格納するセキュアコンテキスト識別子レジスタ１１８、認証に必要な鍵を格納する認証鍵レジスタ１１９、およびレジスタ群１２０を備えている。

【０１２１】

また物理メモリ１０１には、例えば実行コードが物理ページ１２４の単位で格納されており、Ｉ／Ｏ装置１０２には実行コードやデータがページ１２５の単位で認証情報１２６が付加された形式で格納されている。なお第７の実施例では、セキュアコンテキスト識別子レジスタ１１８にセキュアコンテキスト識別子が格納されたコンテキスト（プロセス）の実行コードの認証は認証鍵レジスタ１１９に格納される認証鍵を用いて行われる。

【０１２２】

図４５は、プロセッサ上で動作するプログラム、例えばユーザによって使用されるプログラムが起動されて、コンテキスト生成命令を発行した時点でのそのコンテキストに対応するセキュアコンテキスト識別子の生成方法の説明図である。コンテキスト生成命令は、プロセッサ内部の命令解釈部１０６に与えられ、その解釈結果に対応してセキュアコンテキスト識別子生成部１０９によってセキュアコンテキスト識別子が生成され、セキュアコンテキスト識別子レジスタ１１８にその値がセットされる。第５の実施例ではセキュアコンテキスト識別子レジスタ１１８への値のセットはこの方法によってのみ可能となるように構成される。これによってセキュアコンテキスト識別子を改ざんし、他のコンテキストになりすますことは不可能となる。なおコンテキストは基本的にオブジェクト指向プログラミングにおける概念であり、より一般的にはプロセス、すなわちプログラムの実行状態に対応するが、この第５の実施例ではプロセスの代わりにコンテキストという用語を用いる。

【０１２３】

図４６は、セキュアコンテキスト識別子の具体的な生成方法の説明図である。その生成には図に示すように乱数発生器１２７を用いることも、また単調増加カウンタ１２８を用いることもできる。乱数として同じ値が生成される確率は０でなく、カウンタの値も一巡すると同じ値になるため、厳密には識別子としてのユニーク性が保証できないが、十分長いビット長のセキュアコンテキスト識別子を用いることによって実質的に問題が起きないようにすることができる。

【０１２４】

あるいは図に示すようにプロセッサが持つ既存のコンテキストＩＤと乱数発生器１２７の出力とを結合部１２９で結合するか、単調増加カウンタ１２８の出力と結合することによってセキュアコンテキスト識別子を生成しても良い。なお既存のコンテキストＩＤは、例えばＯＳによって設定される任意の値であり、一般にユニーク性が保証されているものではない。

【０１２５】

図４７は、セキュアコンテキスト識別子消滅方法の説明図である。同図においてプロセッサ上で動作するプログラムがコンテキスト消滅命令を発行した場合に、プロセッサはセキュアコンテキスト識別子レジスタ１１８の内容を無効とする。無効とする方法は０クリアでも良く、レジスタ中に有効／無効を表すフラグの格納領域を設け、そのフラグを無効にセットしても良い。

【０１２６】

図４８は、図４４において例えばページ単位の実行コード１２５に付加される認証情報１２６の説明図である。プロセッサ内のコンテキスト情報１１５の内部の認証鍵レジスタ

１１９には認証情報１２６を用いて実行コード１２５を認証処理するための鍵が格納される。認証情報として、例えばＲＳＡによる電子署名を用いる場合には、認証鍵はＲＳＡ公開鍵となり、共通鍵系のＳＨＡ（セキュアハッシュアルゴリズム）－１ＨＭＡＣ（ハッシュベースド・メッセージ・オーセンティケーション・コード）を用いる場合には認証鍵は２０バイトの値となる。

【０１２７】

認証鍵レジスタ１１９への鍵の格納は、ＯＳによるコンテキストの生成時、すなわちコンテキスト情報が初期化される段階で行われ、同時に認証鍵の正当性のチェックがおこなわれる。認証鍵自体が悪意を持った者によって生成され、その鍵によって悪意を持った実行コードに対応する認証情報が生成されてしまうと、認証処理自体は何の問題もなく成功し、プロセッサによる認証機能が働かなくなることになる。従って認証鍵の正当性をいかに保証するかは重要な課題である。

【０１２８】

図４９は、認証鍵が公開鍵の場合の認証鍵レジスタへの鍵設定方式の説明図である。同図において認証鍵はＲＳＡ公開鍵であるものとし、認証局証明書、すなわち認証局公開鍵１３４は、例えばプロセッサ内に工場出荷時に埋め込まれ、それ以降その置き換えや変更は不可能となっているものとする。認証鍵レジスタ１１９に設定すべき認証鍵は認証局秘密鍵による署名が付与された形で、例えばコンテキスト生成の時点でプロセッサに対して認証鍵設定命令として与えられ、その命令は命令解釈部１０６によって解釈され、署名検証部１０８によって認証鍵の検証が行われた後に認証鍵レジスタ１１９に格納される。これにより認証局のお墨付きのある公開鍵のみが認証鍵レジスタに設定される。

【０１２９】

図５０は、図４９における認証鍵設定処理のフローチャートである。同図においてまずステップＳ７１で命令解釈部１０６によって認証鍵設定命令がフェッチされ、ステップＳ７２で署名検証部１０８によってフェッチされた公開鍵が署名と認証局公開鍵を用いて検証され、ステップＳ７３で検証が成功した否かが判定され、成功した場合には命令解釈部１０６によってフェッチした設定命令に含まれる公開鍵が認証鍵レジスタ１１９に格納された後に、検証が失敗した場合には直ちに処理を終了する。

【０１３０】

図５１は認証鍵が共通鍵の場合の鍵設定方式を示し、図５２は鍵設定処理のフローチャートである。証明鍵として共通鍵を用いる場合には、実行コードに対して認証情報を付加する側から安全な方法で認証鍵を受け取る必要がある。ここではＲＳＡ公開鍵を用いて暗号化されたＨＭＡＣ鍵を認証鍵設定命令とともに受け取るものとし、プロセッサ側でプロセッサ固有ＲＳＡ秘密鍵１３７を用いて復号部１０８によって復号した後に認証鍵レジスタ１１９に格納するものとする。

【０１３１】

図５２のフローチャートにおいて、まずステップＳ７６で命令解釈部１０６によって認証鍵設定命令がフェッチされ、ステップＳ７７でその命令に含まれる暗号化されたＨＭＡＣ鍵がプロセッサ固有ＲＳＡ秘密鍵１３７を用いて復号部１０８によって復号され、ステップＳ７８で復号されたＨＭＡＣ鍵が命令解釈部１０６によって認証鍵レジスタ１１９に格納されて処理を終了する。

【０１３２】

図５３は既にセキュアコンテキスト識別子が生成されているコンテキストの実行コードを主記憶、すなわち物理メモリ１０１の物理ページに格納し、その物理ページの認証を行って処理の実行開始を可能とするためのページイン方式の説明図であり、図５４はこのページインにおける処理のフローチャートである。

【０１３３】

このページインの処理では、まずＯＳによって実行コードの物理ページへの格納やページ・テーブル・エントリ（ＰＴＥ）内の各種データの設定が行われた後に、ＯＳから認証部１０７に対してコンテキスト、すなわち物理ページの認証要求としてのセキュアページ

フラグフィールドのセット要求がなされ、その要求に対応して認証部１０７によって物理ページの認証が行われた後にセキュアページフラグフィールドのフラグがセットされ、以後ＰＴＥ使用が可能となる。

【０１３４】

図５４の処理フローチャートにおいて処理が開始されると、ステップＳ８０でＯＳによって空き物理ページに実行コードが格納され、ステップＳ８１でＯＳによってその物理ページの先頭アドレスと対応する論理ページの先頭アドレスとが物理アドレス、論理アドレスとしてＴＬＢの内部のＰＴＥにセットされ、ステップＳ８２でセキュアコンテキスト識別子の値がそのＰＴＥにセットされる。例えば図４５、４６で説明したようにコンテキスト生成命令の発行時点で生成され、セキュアコンテキスト識別子レジスタ１１８に格納されたセキュアコンテキスト識別子がＯＳによって読み出し可能となっているものとして、ＯＳは読み出したセキュアコンテキスト識別子をＰＴＥに設定する。

【０１３５】

その後ステップＳ８３でＯＳによって、必要に応じてそのページに対するリード／ライト属性などがＰＴＥにセットされ、ステップＳ８４でＯＳから、例えばハードウェアとしての認証部１０７に対してセキュアページフラグフィールドのセットが要求される。なお、ＯＳ自体は既に認証済みであることは大前提であり、セキュアページフラグフィールドのセットは基本的にＯＳの仕事であるが、ここでは認証済みのＯＳからハードウェアに対してフラグのセット要求が行われる。

【０１３６】

ステップＳ８５で認証部１０７による認証処理が実行される。この処理の詳細については後述する。この処理ではセキュアコンテキスト識別子に対応するコンテキストの認証鍵と、認証情報一次格納部１１２に格納された認証情報とを用いて物理ページの認証が行われ、ステップＳ８６で認証が成功したか否かが判定され、成功した場合にはセキュアページフラグフィールドのフラグがセットされ、以後そのＰＴＥが使用可能となる。これに対して失敗した場合にはそのフィールドのフラグはリセットされ、そのＰＴＥは使用不可とされ、ステップＳ８９でＯＳによるリカバリまたはエラー処理が行われる。

【０１３７】

なお例えば図５４の処理はＴＬＢ内のＰＴＥに対して直接に値の設定可能なプロセッサを対象としているが、例えば主記憶上のＰＴＥに値が設定され、ＴＬＢはそのキャッシュとして働くようなプロセッサにおいては、ステップＳ８０からステップＳ８３までの処理が主記憶上のＰＴＥに対して行われ、その内容をＴＬＢにキャッシュするタイミングでステップＳ８４以降の動作が行われることになる。

【０１３８】

図５５は図５３の認証部１０７の構成例であり、図５６は図５４におけるステップＳ８５の認証処理のフローチャートである。ここではページ全体からＳＨＡ－１ハッシュ値が計算され、電子署名の復号結果と比較されるものとする。なお図５６に示すように認証部１０７の動作をハードウェアによってではなく、ソフトウェアによる処理として実現することも当然可能である。

【０１３９】

図５５において物理ページ１２５は６４バイトずつに分割され、ＳＨＡ－１ハッシュ演算器１４０に与えられ、ページ全体のハッシュ値が計算されて比較器１４２に与えられる。一方、認証情報一次格納部１１２に格納されたＲＳＡ電子署名は、認証鍵レジスタ１１９に格納されたＲＳＡ公開鍵とともにＲＳＡ復号器１４１に与えられ、その出力としての復号済みハッシュ値が比較器１４２によってＳＨＡ－１ハッシュ演算器１４０の出力と比較されて一致する場合には認証成功、不一致の場合には失敗と判定される。

【０１４０】

図５６の認証処理において、まずステップＳ９０で物理ページが６４バイト単位で読み込まれ、ステップＳ９１でハッシュ演算が行われ、ステップＳ９２でページ終端に達したか否かが判定され、達していない場合にはステップＳ９０以降の処理が繰り返される。

【０１４１】
終端に達した場合にはステップＳ９３でＲＳＡ公開鍵を用いて電子署名が復号処理され、ステップＳ９４で復号結果とハッシュ演算の結果が比較され、一致している場合にはステップＳ９５でセキュアページフラグフィールドがセットされ、不一致の場合にはステップＳ９６でセキュアページフラグフィールドがリセットされて処理を終了する。
【０１４２】
以上のように第５の実施例では、実行コードの物理メモリ（主記憶）へのページインにあたって実行コードの認証が行われ、認証が成功したことを示すセキュアページフラグのセットが行われる。
【０１４３】
次に本発明における物理ページ上の命令実行時のメモリアクセス制御について第６の実施例として説明する。図５７は、物理ページ上の命令実行時のメモリアクセス制御方式の説明図である。同図においてセキュアコンテキスト識別子レジスタ１１８に意味のある値が入っており、またＰＴＥ１２２のセキュアページフラグフィールドがセットされており、さらにセキュアコンテキスト識別子レジスタ１１８内に格納されている識別子の値と、ＰＴＥ１２２上のコンテキスト識別子の値が一致する場合に物理ページ１２４上の命令の実行が許可される。この制御はメモリアクセス制御部１０５によって行われる。なお物理ページに対するデータのリード／ライト属性やスーパバイザ属性などのチェックは本発明の内容と直接の関係はなく、別途行われているものとする。
【０１４４】
図５８は、メモリアクセス制御部１０５の動作例の説明図である。同図において太い一点鎖線の中味がメモリアクセス制御部１０５に相当し、またＴＬＢ１１４の内部のＰＴＥの属性データとしてセキュアページフラグフィールドやセキュアコンテキスト識別子が含まれるものとする。
【０１４５】
図３９と同様に、例えば論理アドレスをアクセスアドレスとしたアクセスが行われると、そのアドレスによって選択されたＰＴＥの属性データが読み出され、アクセス属性と属性チェック１４６によって比較され、チェック結果がＯＫであれば論理アドレスに対応して読み出された物理アドレスと属性チェック結果を用いてキャッシュコマンド生成１４７が行われ、例えば図２０のキャッシュメモリ４５の内部のタグ１４８が検索されて、キャッシュヒット時の場合にはそのままデータ応答が返され、キャッシュミス時の場合にはキューおよびバスインタフェース１４９を介して、例えば主記憶からロードされたデータがキャッシュメモリに格納されるとともにデータ応答が実行ユニットに返される。
【０１４６】
図５９は、命令フェッチ時のメモリアクセス制御部１０５の処理フローチャートである。図５７の命令実行部１４４によって命令フェッチのための論理アドレスが出力されると、ステップＳ９８で指定された論理アドレスに対応するＰＴＥの属性データが選択され、ステップＳ９９でカレントコンテキスト、すなわち現在実行すべきコンテキストがセキュアコンテキストであるか否か、すなわち有効なセキュアコンテキスト識別子を持っているかどうかがチェックされる。セキュアコンテキストである場合には、ステップＳ１００でそのコンテキストに対応するＰＴＥのセキュアページフラグフィールド（ＳＰＦ）がセットされているか否かがチェックされ、セットされている場合にはステップＳ１０１でカレントコンテキストのセキュアコンテキスト識別子、すなわちセキュアコンテキスト識別子レジスタ１１８に格納されている識別子とＰＴＥに格納されているセキュアコンテキスト識別子が一致しているか否かが判定される。
【０１４７】
一致している場合にはステップＳ１０２で、例えばコンテキストに対応するページの属性としてのリード／ライト属性やスーパバイザ属性などのチェックが行われ、ＯＫの場合にはステップＳ１０３で命令フェッチのための物理アドレスをキャッシュに出力するためのキャッシュコマンドが生成されて処理を終了する。

【０１４８】
ステップＳ９９でカレントコンテキストが有効なセキュアコンテキスト識別子を持っていない場合には、対応するＰＴＥの中のセキュアページフラグフィールド（ＳＰＦ）がセットされているかが否かがステップＳ１０４で判定され、セットされていない場合にはセキュアコンテキスト識別子が設定されておらず、また認証の行われていない従来と同様の実行コードを処理すべきことになり、ステップＳ１０２の処理に移行する。ステップＳ１００、Ｓ１０１における判定結果がＮｏである場合、またステップＳ１０４の判定結果がＹｅｓである場合、ステップＳ１０２の判定結果がＮｏである場合にはいずれもステップＳ１０５でエラー処理が行われて処理を終了する。なおここでは論理アドレスを論理ページの先頭アドレスとページ内のオフセットの値とに分解する処理と、物理ページの先頭アドレスとそのオフセットの値を加算する処理が必要であるが、これらの処理については本発明と直接の関係はなくその説明を省略する。
【０１４９】
図６０は、セキュアコアとノーマルコアとが備えられたプロセッサにおけるメモリアクセス制御方式の説明図である。同図においてノーマルコア１５２は、第１の実施例における図１２と同様に暗号処理ブロック１２とコード認証処理ブロック１３による処理とは無関係の従来と同様の処理だけを行うものであり、セキュアコア１５１は図４４においては説明しない暗号処理ブロックによる動作の制御を含めて、コード認証処理ブロックの認証制御を含むセキュア動作を実行可能なものである。
【０１５０】
図６０においてメモリアクセス制御部１０５の制御によって、セキュアコア１５１からはセキュアページフラグフィールドのセットされたＰＴＥに対応する物理ページの使用は許可されるが、ノーマルコアからはそのページが利用できないようにする制御が行われる。
【０１５１】
なお図４４において、セキュアコアから制御されるコード認証処理ブロックはメモリアクセス制御部１０５を含み、認証部１０７、暗号化／復号、署名生成／検証部１０８、セキュアコンテキスト識別子生成部１０９、セキュアコンテキスト識別子消滅部１１０、プロセッサ固有鍵１１１、認証情報一次格納部１１２、セキュアＤＭＡ１１３、セキュアコンテキスト識別子レジスタ１１８、認証鍵レジスタ１１９、およびＰＴＥ１２２の内部のセキュアページフラグフィールドとセキュアコンテキスト識別子に相当する。
【０１５２】
なお図６０においてセキュアコア１５１は、認証処理が終了し、セキュアページフラグフィールドがＰＴＥにセットされた物理ページ内の実行コードのみを実行し、ノーマルコア１５２は認証されていない通常コードのみを実行することを基本とするが、ノーマルコアが通常コードに加えて認証されたコードを実行可能とするよう構成することも可能である。
【０１５３】
図６１は、セキュアモードとノーマルモードとを切り替えるコアを備えるプロセッサの構成ブロック図である。同図においてはコア１５４の内部にモードレジスタ１５５が備えられ、セキュアモードの場合だけセキュアページフラグフィールドが設定されたページを利用可能とするものである。なおセキュアモードとノーマルモードの切り替えは、例えば通常のユーザモードとカーネルモードとの切り替えのように割込みをトリガとする方法でもよく、他の方法を用いても良い。
【０１５４】
図６２はメモリアクセス制御方式としての、図４４におけるセキュアＤＭＡ１１３による物理メモリ１０１へのページデータ転送方式の説明図であり、図６３はセキュアＤＭＡ１１３によるデータ転送処理のフローチャートである。例えば図５３や図５４のページイン処理では、物理ページに実行コードとしてのページデータが格納された後にその実行コードの認証を行うものとしたが、認証処理においてはハッシュ値の計算などの処理を必要

とするため、ここではページデータの転送単位毎にハッシュ値を計算し、その結果をハッシュ演算の中間結果として保持する動作を繰返し、転送終了時点ではハッシュ演算を終了して、その結果をその後の認証処理に利用するものである。

【０１５５】

図６２においてセキュアＤＭＡ１１３は、コア１５４からのデータの転送元アドレス、転送先アドレス、転送サイズを受け取る転送管理部１５７、Ｉ／Ｏ装置１０２からデータを読み出すデータ読み出し器１５８、ハッシュ演算を行うハッシュ演算器１５９、物理メモリ１０１にページデータを書き込むデータ書込み器１６０、物理ページの先頭アドレスとそのページに対するハッシュ値を保持する物理ページ先頭アドレス保持部１６１を備えている。

【０１５６】

図６３において処理が開始されると、コア１５４上で動作するプログラム、一般にはＯＳから転送元アドレスなどの指示を受けた転送管理部１５７によって、データ読み出し器１５８に対してＩ／Ｏ装置１０２から次の６４バイトのデータの読み出しが指示され、ステップＳ１１１でデータ読み出し器１５８によって６４バイトのデータが読み出され、ステップＳ１１２で転送管理部１５７によってハッシュ演算器１５９に対してハッシュ演算が指示され、ステップＳ１１３でハッシュ演算器１５９によってハッシュ演算が行われ、その中間結果が内部に保持され、ステップＳ１１４で転送管理部１５７からデータ書き込み器１６０に対して物理メモリ１０１への６４バイトのデータの書込みが指示され、ステップＳ１１５でデータ書き込み器１６０によって６４バイトのデータが物理メモリ１０１に書き込まれ、ステップＳ１１６で１ページのデータ転送が終了したか否かが判定され、終了していない場合にはステップＳ１１０からの処理が繰り返され、終了している場合には転送管理部１５７によって転送先アドレスとしての物理ページ先頭アドレスとハッシュ値のペアが保持部１６１に与えられて処理を終了する。

【０１５７】

図６４はメモリアクセスの制御を含むコード実行時の処理フローチャートである。同図は代表的にはＯＳによるページイン時の処理のフローチャートであり、本発明の特徴は太線で囲まれた処理にある。処理が開始されるとまずステップＳ１２０でセキュアＤＭＡ１１３に対して転送元／先アドレス、転送サイズなどが指示され、ステップＳ１２１で転送が成功したか否かが判定され、成功した場合にはステップＳ１２２で図５４のステップＳ８１からＳ８３におけると同様にＴＬＢ内のＰＴＥに各種情報が設定され、ステップＳ１２３でステップＳ８４と同様にセキュアページフラグフィールドのセットが要求され、認証部による認証処理が実行された後にステップＳ１２４でフラグのセットが成功したか否かが判定され、成功した場合には処理を終了する。またステップＳ１２１で転送が失敗した場合、ステップＳ１２４でセットが失敗した場合には直ちに処理を終了する。

【０１５８】

以上のように第６の実施例によれば、既に認証が成功した実行コードへのアクセスに対してもセキュアコンテキスト識別子やセキュアページフラグフィールドのチェックが行われた後にアクセスが許可される。

【０１５９】

最後に本発明の第７の実施例について図６５から図７４を用いて説明する。この第７の実施例では、例えばコンテキストスイッチに対応したコンテキスト情報やＰＴＥの、例えば主記憶への退避に当って、データを保護するための暗号化、または改ざん検出情報の付加が行われる。例えば第１の実施例では認証された実行コードが暗号化されて物理メモリに格納されるものとしたが、第７の実施例では、例えばコンテキスト情報が暗号化されて物理メモリに格納される。

【０１６０】

図６５は、そのコンテキスト情報暗号化方式の説明図である。同図においては図４４で説明したコンテキスト情報格納部１１５に格納されているコンテキスト情報のすべてが暗号器１６５によってプロセッサ固有鍵１１１を用いて暗号化され、暗号化コンテキスト情

報１６６として物理メモリ１０１に格納される。

【０１６１】

図６６は、図６５に対応するコンテキスト情報の復号方式の説明図である。物理メモリ１０１に格納されている暗号化コンテキスト情報１６６は、コンテキストスイッチによって必要となった時点でプロセッサ固有鍵１１１を用いて復号器１６８によって復号され、コンテキスト情報格納部１１５に格納される。

【０１６２】

図６７は、コンテキスト情報への改ざん検出情報の付加方式の説明図である。同図においてコンテキスト情報格納部１１５に格納されているコンテキスト情報に対して、プロセッサ固有鍵１１１を用いて改ざん検出情報１７０が改ざん検出情報生成器１６９によって生成され、物理メモリ１０１にコンテキスト情報とともに格納される。

【０１６３】

図６８は、図６７に対応する改ざん検出情報を用いたコンテキスト情報に対する改ざん検出方式の説明図である。同図においてコンテキスト情報に付加された改ざん検出情報１７０を用いて、改ざん検出器１７２によってプロセッサ固有鍵１１１を用いた改ざん検出が行われる。

【０１６４】

図６９は、コンテキスト情報格納部１１５に格納されたコンテキスト情報のうちでセキュアな動作に必要となるコンテキスト情報と、通常のコンテキスト情報とを区分し、セキュア動作用コンテキスト情報１７５だけを暗号化するコンテキスト情報暗号化方式の説明図である。この方式では、プロセッサの核となる部分の変更は極力行わないように通常のコンテキスト情報１７６、すなわち既存コンテキストＩＤなどのコンテキスト情報は従来と同様に暗号化せずに扱うこととし、認証鍵レジスタ１１９、セキュアコンテキスト識別子レジスタ１１８の格納内容をセキュア動作用コンテキスト情報１７５として暗号化するものである。

【０１６５】

既存コンテキストＩＤとしては、例えばＯＳの動作としてセキュアコンテキスト識別子と同じ値を格納することも可能であるものとする。例えばＯＳが悪意をもったコードに書き換えられたような場合には２つの識別子の値が同じとなる保証がなくなるが、同じ値であるときのみプロセッサを動作可能とさせるように構成することで、同じ値でない場合には動作しないという安全サイドに倒れ、問題は発生しない。

【０１６６】

図６９においてセキュア動作用コンテキスト情報１７５だけがプロセッサ固有鍵１１１を用いて暗号器／復号器１７４によって暗号化され、暗号化コンテキスト情報１７７として物理メモリ１０１に格納され、通常のコンテキスト情報１７６は平文コンテキスト情報１７６としてそのまま物理メモリ１０１に格納される。

【０１６７】

図７０は、セキュア動作用コンテキスト情報１７５に対して改ざん検出情報を付加して物理メモリ１０１に格納するコンテキスト情報格納方式の説明図である。同図においては改ざん検出情報生成器／改ざん検出器１７９によって、プロセッサ固有鍵１１１を用いてセキュア動作用コンテキスト情報１７５に対する改ざん検出情報１８０が生成され、物理メモリ１０１にセキュア動作用コンテキスト情報１７５と通常のコンテキスト情報、すなわち平文コンテキスト情報１７６とともに格納される。なおここでは通常のコンテキスト情報としてのプログラムカウンタの値やレジスタ群の値に対しては暗号化などを行わないものとしたが、さらに信頼性を向上させるためにはこのような通常コンテキスト情報についても暗号化、あるいは改ざん検出情報の付加を行うことも当然可能である。

【０１６８】

図７１から図７４は、ページ・テーブル・エントリ（ＰＴＥ）１２２の格納内容の保護方式の説明図である。図７１はＰＴＥの暗号化方式を示し、ＰＴＥ１２２の格納内容、すなわちセキュアページフラグフィールド、セキュアコンテキスト識別子、論理アドレス、

および物理アドレスの値がプロセッサ固有鍵１１１を用いて暗号器１６５によって暗号化され、暗号化ＰＴＥ１８３として物理メモリ内のページテーブル１８２に格納される。

【０１６９】

図７２は、図７１に対応する暗号化ＰＴＥの復号方式の説明図である。同図において物理メモリ１０１に格納されている暗号化ＰＴＥ１８３は、プロセッサ固有鍵１１１を用いて復号器１６８によって復号され、ＴＬＢ１１４の内部にＰＴＥとして格納される。

【０１７０】

図７３はＰＴＥへの改ざん検出情報付加方式、図７４はＰＴＥに対する改ざん検出方式の説明図である。図７３においては改ざん検出情報生成器１６９によってプロセッサ固有鍵１１１を用いてＰＴＥ１２２に対する改ざん検出情報１８５が生成され、ＰＴＥ１２２とともにページテーブル１８２に格納される。

【０１７１】

図７４においては改ざん検出情報１８５とプロセッサ固有鍵１１１とを用いて、改ざん検出器１７２によってページテーブル１８２に格納されているＰＴＥ１２２に対する改ざん検出が行われる。

【０１７２】

以上のように第７の実施例では、セキュアプロセッサによって使用されるコンテキスト情報とＰＴＥに対しても暗号化や改ざん検出の処理が行われ、情報処理の安全性がさらに向上する。

【０１７３】

以上において本発明のセキュアプロセッサ、およびセキュアプロセッサ用プログラムについてその詳細を説明したが、このセキュアプロセッサを一般的なコンピュータシステムの基本要素とすることが可能である。図７５はそのようなコンピュータシステム、すなわちハードウェア環境の構成ブロック図である。

【０１７４】

図７５においてコンピュータシステムは中央処理装置（ＣＰＵ）２００、リードオンリメモリ（ＲＯＭ）２０１、ランダムアクセスメモリ（ＲＡＭ）２０２、通信インタフェース２０３、記憶装置２０４、入出力装置２０５、可搬型記憶媒体の読取り装置２０６、およびこれらの全てが接続されたバス２０７によって構成されている。

【０１７５】

記憶装置２０４としてはハードディスク、磁気ディスクなど様々な形式の記憶装置を使用することができ、このような記憶装置２０４、またはＲＯＭ２０１に図３～図５、図７、図９～図１１、その他のフローチャートに示されたプログラムや、本発明の特許請求の範囲の請求項７、１９、および２０のプログラムなどが格納され、そのようなプログラムがＣＰＵ２００によって実行されることにより、本実施形態におけるセキュアプロセッサの動作、暗号鍵の設定、コード認識処理、および暗号処理などが可能となる。

【０１７６】

このようなプログラムは、プログラム提供者２０８からネットワーク２０９、および通信インタフェース２０３を介して、例えば記憶装置２０４に格納されることも、また市販され、流通している可搬型記憶媒体２１０に格納され、読取り装置２０６にセットされて、ＣＰＵ２００によって実行されることも可能である。可搬型記憶媒体２１０としてはＣＤ－ＲＯＭ、フレキシブルディスク、光ディスク、光磁気ディスク、ＤＶＤなど様々な形式の記憶媒体を使用することができ、このような記憶媒体に格納されたプログラムが読取り装置２０６によって読取られることにより、本実施形態におけるセキュアプロセッサの動作が可能となる。

【０１７７】

（付記１）

命令コードを実行するコアを備えるプロセッサであって、

該コアに固有の鍵を記憶する鍵記憶手段と、

暗号化された命令コードを書き換え不可能な形式で記憶する命令コード記憶手段と、

該命令コード記憶手段に記憶された命令コードを含む命令コードの認証を前記固有鍵あるいは固有鍵による認証済鍵を用いて行う認証処理手段と、
該コアと外部との間で入出力されるデータを暗号化する暗号処理手段とを備えることを特徴とするセキュアプロセッサ。
（付記２）
前記暗号処理手段が、前記認証処理手段によって認証された命令コードを暗号化し、ページ単位で前記セキュアプロセッサに接続された記憶装置に格納することを特徴とする付記１記載のセキュアプロセッサ。
（付記３）
前記記憶装置に格納されたページ単位の暗号化命令コードの実行時に不正命令が検出された時、該ページ単位の暗号化命令コードの実行を停止する不正命令実行停止手段を備えることを特徴とする付記２記載のセキュアプロセッサ。
（付記４）
前記認証処理手段が認証対象とする命令コードに認証情報が付加されていることを特徴とする付記１記載のセキュアプロセッサ。
（付記５）
前記認証情報内に暗号化の鍵が指定されている時、前記暗号処理手段が該指定されている鍵を使用してさらに前記命令コードの暗号化を行うことを特徴とする付記４記載のセキュアプロセッサ。
（付記６）
前記認証情報内に暗号化の鍵が指定されていない時、前記暗号処理手段が任意のページ鍵を使用してさらに前記命令コードの暗号化を行うことを特徴とする付記４記載のセキュアプロセッサ。
（付記７）
前記暗号処理手段が、前記認証された命令コードに対応する同一プロセスのデータに対して、該命令コードに対する暗号鍵と異なる暗号鍵を用いて該データの暗号化を行うことを特徴とする付記１記載のセキュアプロセッサ。
（付記８）
前記セキュアプロセッサに接続された記憶装置内でデータの格納領域に格納された命令コードの実行時に前記異なる暗号鍵に代わって命令コードに対する暗号鍵を使用することを特徴とする付記７記載のセキュアプロセッサ。
（付記９）
前記セキュアプロセッサにおいて、
前記認証処理手段による認証が失敗した命令コードの実行を停止させるコード実行停止処理手段をさらに備えることを特徴とする付記１記載のセキュアプロセッサ。
（付記１０）
前記コアとして、
前記認証処理手段によって認証された命令コードのみを実行するセキュアコアと、
前記認証処理手段によって認証されていない通常の命令コードも実行可能なノーマルコアとを備えることを特徴とする付記１記載のセキュアプロセッサ。
（付記１１）
前記命令コード記憶手段に記憶された暗号化命令コードを用いて前記セキュアコアがブートされるとともに、
該セキュアコアが該ブート完了後に前記ノーマルコアのブートを行わせるノーマルコアブート手段を備えることを特徴とする付記１０記載のセキュアプロセッサ。
（付記１２）
前記セキュアコアが、前記ノーマルコアのブート後に該ノーマルコアの動作を監視し、異常状態を検出した時、該ノーマルコアの動作停止、または特定処理への分岐を行わせるノーマルコア監視手段を備えることを特徴とする付記１１記載のセキュアプロセッサ。
（付記１３）

前記セキュアコアが、前記ノーマルコアに対してコア制御信号を与え、ノーマルコアの動作を制御することを特徴とする付記１０記載のセキュアプロセッサ。
（付記１４）
前記コア固有鍵に対するアクセスが、前記セキュアコアに対して許可され、前記ノーマルコアに対して禁止されることを特徴とする付記１０記載のセキュアプロセッサ。
（付記１５）
前記セキュアコアの制御のもとで、前記コア固有鍵を用いて公開鍵と秘密鍵のペア、および共通鍵を生成する鍵生成手段をさらに備えることを特徴とする付記１４記載のセキュアプロセッサ。
（付記１６）
前記セキュアコアが、前記ノーマルコアを経由して前記鍵生成手段によって生成された公開鍵を外部に通知し、
外部から該公開鍵によって暗号化された原文をノーマルコアを経由して受け取り、前記秘密鍵を用いて原文を復号することを特徴とする付記１５記載のセキュアプロセッサ。
（付記１７）
前記原文が情報の暗号化に使用された鍵であることを特徴とする付記１６記載のセキュアプロセッサ。
（付記１８）
プロセッサにおいて命令コードを実行するコアによって使用されるプログラムであって、
暗号化された命令コードが書き換え不可能な形式で記憶されたメモリ内のプログラムを用いて自コアの起動処理を行う手順と、
該メモリ内に記憶された命令コードを含む命令コードの認証処理を行う認証処理ブロックと、前記コア固有の鍵を管理する鍵管理処理と、該認証処理ブロックによって認証された命令コードの暗号化／復号処理に用いられる鍵が格納された鍵テーブルに対する操作処理とをセットアップする手順と、
前記認証処理ブロックを用いて二次記憶上のプログラムの認証処理を行う手順と、
起動されたオペレーティングシステムを含む該認証処理済みのプログラムの実行時に前記命令コードの暗号化／復号のための鍵処理を含む処理を実行する鍵処理モニタとしての動作を行う手順とを計算機に実行させるためのセキュアコア用プログラム。
（付記１９）
プロセッサにおいて命令コードを実行するコアによって使用されるプログラムであって、
暗号化された命令コードが書き換え不可能な形式で記憶されたメモリ内のプログラムを用いて、自コアの起動処理を行う手順と、
オペレーティングシステムを起動する手順と、
該プロセッサ内で前記メモリ内に記憶された命令コードを含む命令コードの認証処理を行う認証処理ブロックによって認証されたプログラム、または認証されていないプログラムを実行し、該認証されたプログラムの実行処理としては、該認証された実行コードに対応して、暗号化／復号用の鍵を用いた処理を含む鍵処理を実行する鍵処理モニタに対する鍵処理の依頼を含みうる処理を実行する手順とを計算機に実行させることを特徴とするノーマルコア用プログラム。
（付記２０）
命令を実行する命令実行手段と、
該命令実行手段からのコマンドに対応して外部のメモリに対するデータのロード／ストアを制御するロード／ストア制御手段と、
該ロード／ストア制御手段と外部のメモリとの間でデータの暗号化／復号化を行う暗号処理手段とを備え、
前記命令実行手段が、実行中の命令に対応して該暗号処理手段に対してデータ暗号化／復号化に使用すべき鍵を指定することを特徴とするセキュアプロセッサ。

（付記２１）
前記プロセッサにおいて複数個の鍵を記憶する鍵記憶手段をさらに備え、
前記命令実行手段が、該鍵記憶手段に対して前記鍵を指定する鍵番号を出力し、該鍵記憶手段が該鍵番号に対応して前記暗号処理手段に対して、データ暗号化／復号化に使用すべき鍵を与えることを特徴とする付記２０記載のセキュアプロセッサ。
（付記２２）
前記セキュアプロセッサにおいて、外部からロードされた命令フェッチデータの復号化に使用されるべき鍵を記憶する鍵記憶手段をさらに備え、
前記命令実行手段が命令フェッチ状態にある時、該鍵記憶手段が前記暗号処理手段に対して該復号化用の鍵を与えることを特徴とする付記２０記載のセキュアプロセッサ。
（付記２３）
前記セキュアプロセッサにおいて、
複数個の鍵を記憶する鍵記憶手段と、
前記命令実行手段によって出力され、前記鍵を指定するための鍵番号を記憶する鍵番号記憶手段とを備え、
該鍵記憶手段が、該鍵番号記憶手段から与えられる鍵番号に対応して前記データ暗号化／復号化に使用すべき鍵を前記暗号処理手段に対して与えることを特徴とする付記２０記載のセキュアプロセッサ。
（付記２４）
前記セキュアプロセッサにおいて、
外部からロードされる命令フェッチデータの復号化に使用されるべき鍵を含む複数の鍵を記憶する鍵記憶手段と、
外部からロードされた命令フェッチデータの復号化に使用されるべき鍵の鍵番号を記憶する鍵番号記憶手段とをさらに備え、
前記命令実行手段が命令フェッチ状態にある時、該鍵番号記憶手段から出力される鍵番号に対応して該鍵記憶手段が、命令フェッチデータの復号化に使用されるべき鍵を前記暗号処理手段に与えることを特徴とする付記２０記載のセキュアプロセッサ。
（付記２５）
前記命令実行手段が、前記鍵を指定するための信号として鍵の番号に加えて命令に対応するスーパバイザ／ユーザ切り替え信号を出力することを特徴とする付記２０記載のセキュアプロセッサ。
（付記２６）
前記命令実行手段が、前記鍵を指定するための信号として鍵の番号に加えて実行中の命令が含まれるプロセスの識別子を出力することを特徴とする付記２０記載のセキュアプロセッサ。
（付記２７）
前記ロード／ストア制御手段が、
ライトスルー方式のキャッシュメモリと、
外部のメモリにストアすべきデータと該外部メモリから前記暗号処理手段を介してロードされたデータとを結合して暗号処理手段に与えるリードモディファイライト手段とをさらに備えることを特徴とする付記２０記載のセキュアプロセッサ。
（付記２８）
前記セキュアプロセッサにおいて、
前記ロード／ストア制御手段と外部メモリとの間で、前記暗号処理手段をバイパスして、暗号化／復号化を行うことなく、平文データの転送を行うデータバイパス手段をさらに備えることを特徴とする付記２０記載のセキュアプロセッサ。
（付記２９）
命令を実行する命令実行手段と、
該命令実行手段からのコマンドに対応して外部のメモリに対するデータのロード／ストアを制御するロード／ストア制御手段と、

該ロード／ストア制御手段と外部のメモリとの間でデータの暗号化／復号化を行う暗号処理手段とを備え、
前記命令実行手段が、実行中の命令によるデータ／命令フェッチのアクセスアドレスに対応させて、該暗号処理手段に対してデータ暗号化／復号化に使用すべき鍵を指定する信号を与えることを特徴とするセキュアプロセッサ。
（付記３０）
前記プロセッサにおいて複数個の鍵を記憶する鍵記憶手段をさらに備え、
前記命令実行手段が、前記アクセスアドレスとしての論理アドレスを該鍵記憶手段に対して出力し、該鍵記憶手段が該論理アドレスに対応して前記データ暗号化／復号化用の鍵を前記暗号処理手段に与えることを特徴とする付記２９記載のセキュアプロセッサ。
（付記３１）
前記セキュアプロセッサにおいて複数個の鍵を記憶する鍵記憶手段をさらに備え、
前記ロード／ストア制御手段が、前記命令実行手段から与えられるコマンドに対応して前記アクセスアドレスとしての物理アドレスを該鍵記憶手段に対して出力し、該鍵記憶手段が該物理アドレスに対応して前記データ暗号化／復号化用の鍵を前記暗号処理手段に与えることを特徴とする付記２９記載のセキュアプロセッサ。
（付記３２）
前記セキュアプロセッサにおいて、前記アクセスアドレスとしての論理アドレスと物理アドレスとのそれぞれに対応させてそれぞれ複数の鍵を記憶する鍵記憶手段をさらに備え、
該鍵記憶手段に対して前記ロード／ストア制御手段から与えられる前記アクセスアドレスとしての物理アドレスと、前記命令実行手段から与えられる論理アドレスとのいずれを選択すべきかを示す該命令実行手段からの指示に対応して、該鍵記憶手段が選択したアドレスに対応する前記データ暗号化／復号化用の鍵を前記暗号処理手段に与えることを特徴とする付記２９記載のセキュアプロセッサ。
（付記３３）
前記セキュアプロセッサにおいて、
前記アクセスアドレスとしての論理アドレスと物理アドレスとのそれぞれに対応させてそれぞれ複数の鍵を記憶する鍵記憶手段と、
前記命令実行手段によって出力され、前記暗号処理手段に対して論理アドレスと物理アドレスのいずれに対応する鍵を与えるべきかを示すアドレス選択指示のデータを記憶するアドレス選択指示記憶手段とをさらに備え、
該鍵記憶手段が、該アドレス選択指示記憶手段の記憶内容に従って論理アドレスと物理アドレスとのいずれかに対応する鍵を前記データ暗号化／復号化用の鍵として前記暗号処理手段に与えることを特徴とする付記２９記載のセキュアプロセッサ。
（付記３４）
前記ロード／ストア制御手段が、前記アクセスアドレスに対応して複数の鍵を記憶する鍵記憶手段をさらに備え、
該ロード／ストア制御手段が、前記命令実行手段から命令実行中に与えられたアクセスアドレスに対応して該鍵記憶手段に記憶された鍵を選択し、前記データ暗号化／復号化用の鍵として前記暗号処理手段に与えることを特徴とする付記２９記載のセキュアプロセッサ。
（付記３５）
前記命令実行手段が前記暗号処理手段に対して、前記鍵記憶手段のＯＮ／ＯＦＦを示す信号と、該鍵記憶手段がＯＦＦの時に前記データ暗号化／復号化に使用されるべき鍵を与える信号とを出力し、
該暗号処理手段が前記ＯＮ／ＯＦＦ信号に対応して、前記鍵記憶手段がＯＮの時には鍵記憶手段から与えられる鍵を、ＯＦＦの時には該命令実行手段から与えられる鍵を前記データ暗号化／復号化用の鍵として使用することを特徴とする付記３４記載のセキュアプロセッサ。

（付記３６）
前記命令実行手段が前記鍵を指定するための信号として、前記アクセスアドレスに加えて、実行中の命令に対応するスーパバイザ／ユーザ切り替え信号を出力することを特徴とする付記２９記載のセキュアプロセッサ。
（付記３７）
前記命令実行手段が、前記鍵を指定するための信号として前記アクセスアドレスに加えて、実行中の命令が含まれるプロセスの識別子を出力することを特徴とする付記２９記載のセキュアプロセッサ。
（付記３８）
前記ロード／ストア制御手段が、
ライトスルー方式のキャッシュメモリと、
外部のメモリにストアすべきデータと該外部メモリから前記暗号処理手段を介してロードされたデータを結合して該暗号処理手段に与えるリードモディファイライト手段とをさらに備えることを特徴とする付記２９記載のセキュアプロセッサ。
（付記３９）
前記セキュアプロセッサにおいて、前記ロード／ストア制御手段と外部メモリとの間で前記暗号処理手段をバイパスして、暗号化／復号化を行うことなく、平文データの転送を行うデータバイパス手段をさらに備えることを特徴とする付記２９記載のセキュアプロセッサ。
（付記４０）
実行コードに対応するプロセスの実行に先立って、該実行コードを格納するページが正しく認証されたことを示すセキュアページフラグが設定されたページに対応するセキュアプロセス識別子と比較するためのセキュアプロセス識別子を、該プロセスの生成命令が発行された時点で生成するセキュアプロセス識別子生成手段と、
該生成されたセキュアプロセス識別子を該プロセスに関連する情報として保持するプロセス情報保持手段とを備えることを特徴とするセキュアプロセッサ。
（付記４１）
前記生成され、前記プロセス情報保持手段に保持されているセキュアプロセス識別子を、前記プロセスの消滅時に消去するセキュアプロセス識別子消去手段をさらに備えることを特徴とする付記４０記載のセキュアプロセッサ。
（付記４２）
前記プロセスに対応する実行コードに認証情報が付与されるとともに、
前記プロセス情報保持手段が、前記生成されたプロセスの生存期間中に行われる実行コード認証のための認証鍵をさらに保持することを特徴とする付記４０記載のセキュアプロセッサ。
（付記４３）
前記実行コードに付与された認証情報がメモリにおけるページ単位の情報であることを特徴とする付記４２記載のセキュアプロセッサ。
（付記４４）
前記プロセスに対応する実行コードがメモリの空きページに格納され、該ページのアドレスに対応させて前記セキュアプロセス識別子が前記プロセッサ内のバッファに格納された後に前記ページ単位の認証鍵を用いた該実行コードの認証が成功した時、該バッファに前記セキュアページフラグをセットする認証手段をさらに備えることを特徴とする付記４３記載のセキュアプロセッサ。
（付記４５）
前記セキュアプロセッサにおいて、
前記実行コードの実際の実行に先立って前記バッファ内に格納されたセキュアプロセス識別子であって、対応する前記セキュアページフラグがセットされているセキュアプロセス識別子と、前記プロセス情報保持手段に保持され、実行すべき命令コードに対応するセキュアプロセス識別子とを比較し、両者が一致した時に前記実行コードが格納されたメモ

リ上のページへのアクセスを、命令を実行する命令実行部に許可するメモリアクセス制御手段をさらに備えることを特徴とする付記４４記載のセキュアプロセッサ。

（付記４６）

前記セキュアプロセッサにおいて、

前記実行コードのメモリへの格納に並行して前記実行コードの認証に必要となる演算を行い、該演算の結果を保持して前記認証手段に与える直接メモリアクセス手段をさらに備えることを特徴とする付記４４記載のセキュアプロセッサ。

（付記４７）

前記セキュアプロセッサにおいて、該プロセッサに固有の暗号化／復号化用の鍵と、

前記バッファ内に格納されたセキュアページフラグ、セキュアプロセス識別子、および実行コードが格納されたメモリページのアドレスの情報を外部に退避、または外部から復帰するに当り、該プロセッサ固有鍵を用いて該情報の暗号化／復号化を行う暗号処理手段とをさらに備えることを特徴とする付記４４記載のセキュアプロセッサ。

（付記４８）

前記セキュアプロセッサにおいて、

該プロセッサに固有の鍵と、

前記バッファ内に格納されたセキュアページフラグ、セキュアプロセス識別子、および実行コードが格納されたメモリページのアドレスの情報を外部に退避するに当り、該プロセッサ固有鍵を用いて該情報に対する改ざん検出情報を生成して付与し、外部から復帰するに当り該固有鍵を用いて該情報に対する改ざん検出を行う改ざん検出手段とをさらに備えることを特徴とする付記４４記載のセキュアプロセッサ。

（付記４９）

前記セキュアプロセッサにおいて、それぞれ命令実行ユニットとキャッシュとを備えるコアであって、

前記認証された実行コードのみを実行するセキュアコアと、

該認証されていない通常の実行コードを実行するノーマルコアとを備えることを特徴とする付記４０記載のセキュアプロセッサ。

（付記５０）

前記ノーマルコアが前記通常コードに加えて前記認証されたコードをも実行することを特徴とする付記４９記載のセキュアプロセッサ。

（付記５１）

前記セキュアプロセッサにおいて、

実行ユニットとキャッシュメモリとを備えるコアが、前記認証された実行コードのみを実行すべきセキュアモードと、認証されていない通常の実行コードのみを実行すべきノーマルモードとのいずれかの指示が設定されるモード指定手段をさらに備え、

該コアが該指示に対応してセキュアモード、またはノーマルモードのいずれかを実行することを特徴とする付記４０記載のセキュアプロセッサ。

（付記５２）

前記セキュアプロセッサにおいて、

該プロセッサに固有の暗号化／復号化用の鍵と、

前記プロセス情報保持手段に保持され、前記セキュアプロセス識別子を含む情報を外部に退避、または外部から復帰するに当り、該プロセッサ固有鍵を用いて該情報の暗号化／復号を行う暗号処理手段とをさらに備えることを特徴とする付記４０記載のセキュアプロセッサ。

（付記５３）

前記セキュアプロセッサにおいて、

該プロセッサに固有の鍵と、

前記プロセス情報保持手段に保持され、前記セキュアプロセス識別子を含む情報を外部に退避するに当り、該プロセッサ固有鍵を用いて該情報に対する改ざん検出情報を生成して付与し、外部から復帰するに当り該固有鍵を用いて該情報に対する改ざん検出を行う改

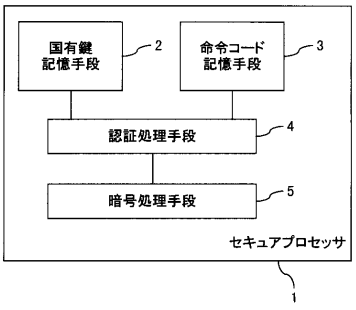
ざん検出手段とをさらに備えることを特徴とする付記４０記載のセキュアプロセッサ。
（付記５４）
実行コードを含むページをメモリにページインする計算機によって使用されるプログラムであって、
該計算機内の直接メモリアクセス機構に前記ページのメモリへの転送を依頼する手順と、
該転送の成功後に、該計算機のトランスレーション・ルックアサイド・バッファ内のページ・テーブル・エントリに、該ページ内の実行コードに対応するプロセスの実行に先立って該実行コードを格納するページが正しく認証されたことを示すセキュアページフラグが設定されたページに対応するセキュアプロセス識別子と比較するための識別子であって、該プロセスの生成命令が発行された時点で生成されたセキュアプロセス識別子を含み、該ページについてのデータを設定する手順と、
前記ページの認証と、該認証の成功を示すセキュアページフラグの該ページ・テーブル・エントリへのセットとをハードウェアに要求する手順とを計算機に実行させることを特徴とするセキュアプロセッサ用プログラム。
（付記５５）
実行コードを含むページをメモリにページインする計算機によって使用される記憶媒体であって、
該計算機内の直接メモリアクセス機構に前記ページのメモリへの転送を依頼するステップと、
該転送の成功後に、該計算機のトランスレーション・ルックアサイド・バッファ内のページ・テーブル・エントリに、該ページ内の実行コードに対応するプロセスの実行に先立って該実行コードを格納するページが正しく認証されたことを示すセキュアページフラグが設定されたページに対応するセキュアプロセス識別子と比較するための識別子であって、該プロセスの生成命令が発行された時点で生成されたセキュアプロセス識別子を含み、該ページについてのデータを設定するステップと、
前記ページの認証と、該認証の成功を示すセキュアページフラグの該ページ・テーブル・エントリへのセットとをハードウェアに要求するステップとを計算機に実行させるセキュアプロセッサ用プログラムを格納した計算機読出し可能可搬型記憶媒体。
（付記５６）
実行コードを含むページの認証を行う計算機によって使用されるプログラムであって、
メモリに読み込まれた該ページに対するハッシュ演算を行う手順と、
該ページに付与されている認証情報を復号する手順と、
該ハッシュ演算結果と該復号結果とを比較する手順と、
該比較の結果として一致が検出された時、該計算機のトランスレーション・ルックアサイド・バッファ内のページ・テーブル・エントリに該ページの認証が成功したことを示すセキュアページフラグをセットする手順とを計算機に実行させることを特徴とするセキュアプロセッサ用プログラム。
（付記５７）
実行コードを含むページの認証を行う計算機によって使用される記憶媒体であって、
メモリに読み込まれた該ページに対するハッシュ演算を行うステップと、
該ページに付与されている認証情報を復号するステップと、
該ハッシュ演算結果と該復号結果とを比較するステップと、
該比較の結果として一致が検出された時、該計算機のトランスレーション・ルックアサイド・バッファ内のページ・テーブル・エントリに該ページの認証が成功したことを示すセキュアページフラグをセットするステップとを計算機に実行させるセキュアプロセッサ用プログラムを格納した計算機読出し可能可搬型記憶媒体。
【符号の説明】
【０１７８】
１０、４０、１００ プロセッサ

１１、１５４ コア
１２ 暗号処理ブロック
１３ コード認証処理ブロック
１４ 暗号化ＲＯＭコード領域
１５ ＣＰＵ固有鍵
１７ 主記憶
１８ 二次記憶
２０ セキュアハードウェア
２１ 暗号鍵設定部
２２ 復号部
２３、１３７ プロセッサ固有ＲＳＡ秘密鍵
２４、１１４ トランスレーション・ルックアサイド・バッファ（ＴＬＢ）
２５ 論理アドレステーブル
２６ 物理アドレステーブル
２７ 鍵テーブル
２８ 署名検証部
２９、１３４ 認証局証明書（認証局公開鍵）
３１、１５１ セキュアコア
３２、１５２ ノーマルコア
３４ 鍵生成機構
４１ 実行ユニット
４２ ロードストアユニット
４３ 暗号化回路
４４ 復号化回路
４５ キャッシュメモリ
４６ メモリ管理ユニット
４７、４８、７３、７４、７５ 鍵テーブルメモリ
５１、５２、７８、７９ 鍵選択レジスタ
７１ リードモデファイライトバッファ
８８ アドレスマップレジスタ（ＡＭＲ）
１０１ 物理ページ
１０２ Ｉ／Ｏ装置
１０５ メモリアクセス制御部
１０６ 命令解釈部
１０７ 認証部
１０８ 暗号化／復号、署名生成／検証部
１０９ セキュアコンテキスト識別子生成部
１１０ セキュアコンテキスト識別子消滅部
１１１ プロセッサ固有鍵
１１２ 認証情報一次格納部
１１３ セキュアＤＭＡ
１１５ コンテキスト情報格納部
１１７ プログラムカウンタ
１１８ セキュアコンテキスト識別子レジスタ
１１９ 認証鍵レジスタ
１２０ レジスタ群
１２４ 物理ページ
１２５ ページ
１２６ 認証情報
１４０ ＳＨＡ－１ハッシュ演算器

１４１　ＲＳＡ復号器
１４２　比較器
１４４　命令実行部
１５５　モードレジスタ
１６５　暗号器
１６８　復号器
１６９　改ざん検出情報生成器
１７２　改ざん検出器
１７４　暗号器／復号器
１７９　改ざん検出情報生成器／改ざん検出器

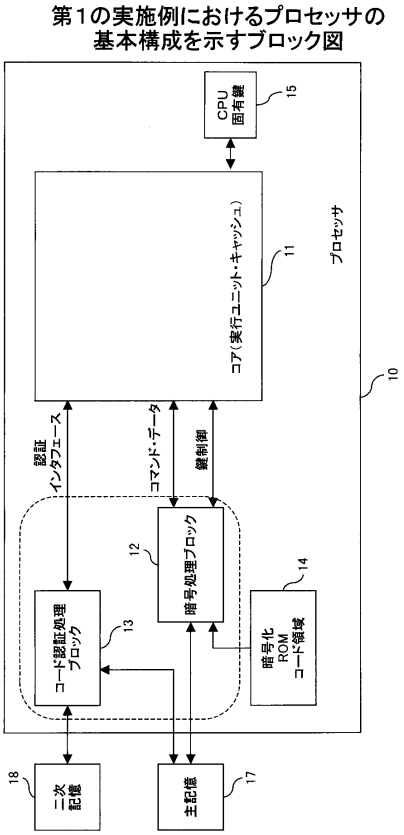
【図１】

本発明のセキュアプロセッサの
原理構成ブロック図



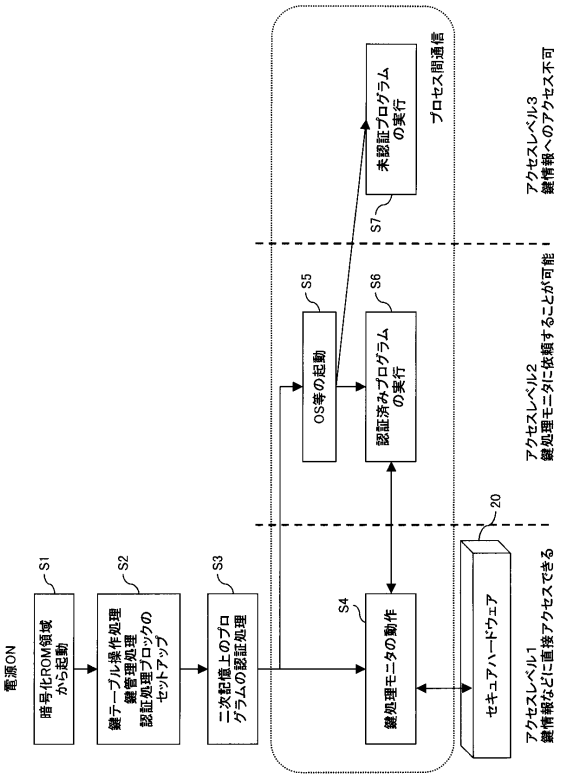
【図２】

第1の実施例におけるプロセッサの
基本構成を示すブロック図



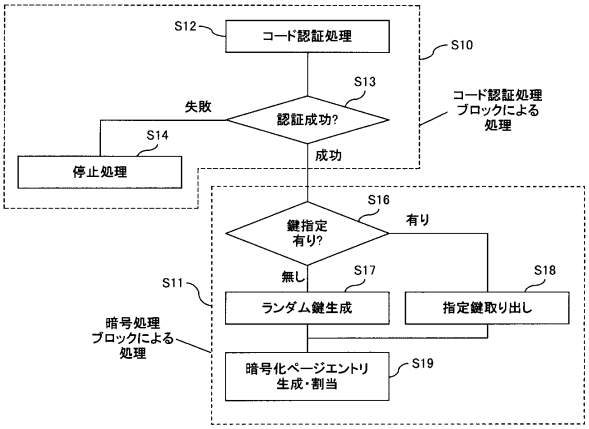
【図３】

第1の実施例におけるプロセッサの
基本処理フローチャート



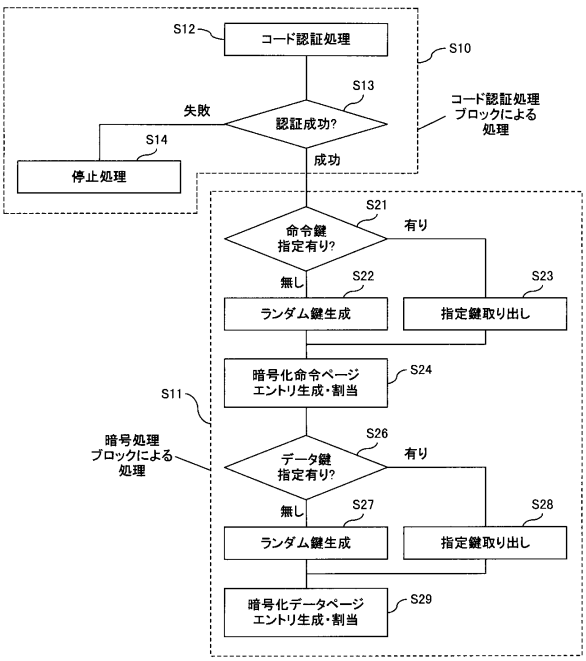
【図４】

コード認証処理ブロックと暗号処理
ブロックによる処理のフローチャート



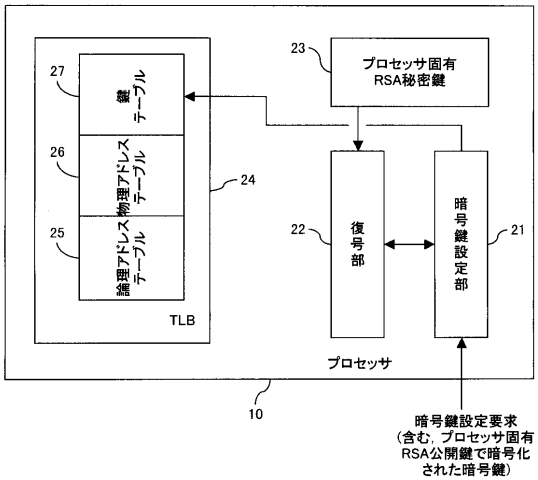
【図５】

命令領域とデータ領域とによって異なる鍵が
指定されている場合の暗号処理ブロックの
処理フローチャート



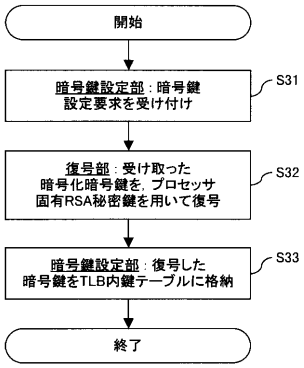
【図６】

公開鍵で暗号化された暗号鍵の
格納方式の説明図



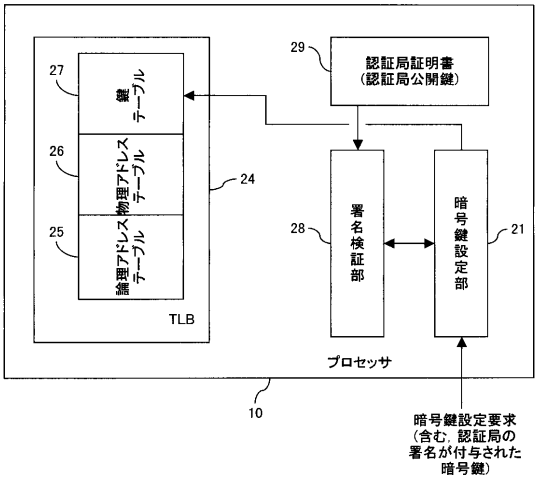
【図７】

公開鍵で暗号化された暗号鍵の
格納処理フローチャート



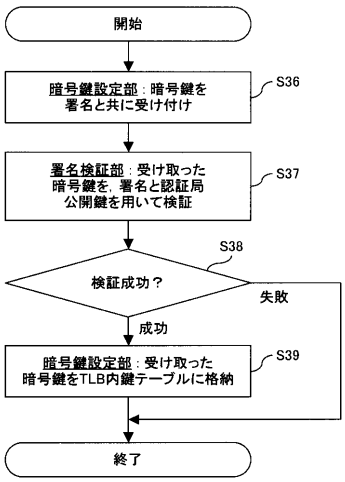
【図８】

認証局の署名が付与された暗号鍵の
格納方式の説明図



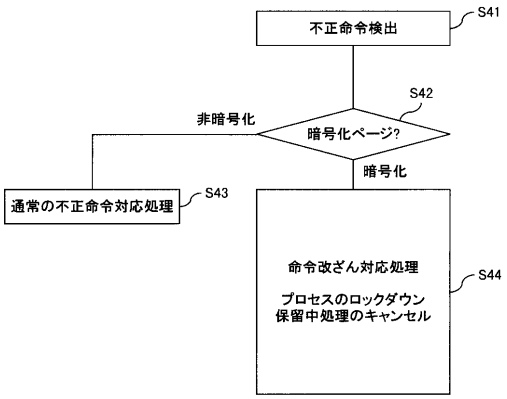
【図９】

認証局の署名が付与された暗号鍵の
格納処理フローチャート



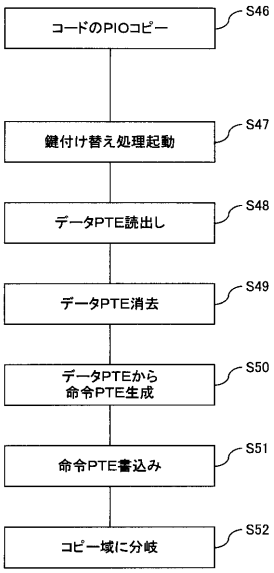
【図１０】

不正命令検出時の処理フローチャート



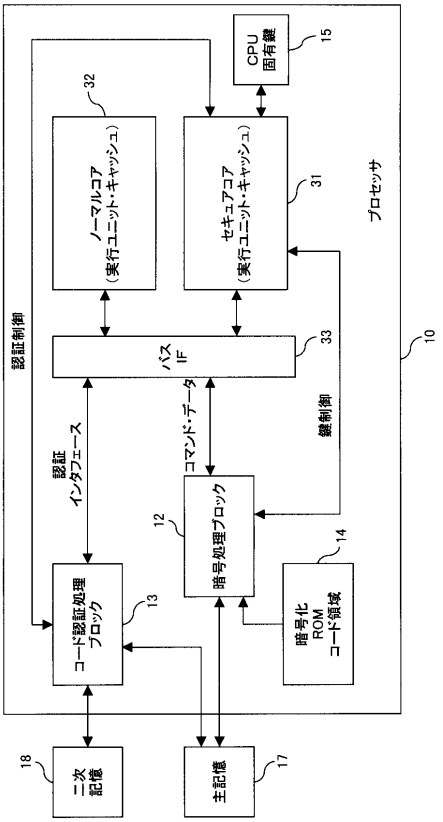
【図１１】

データ領域に格納された命令に対する
鍵付け替え処理のフローチャート



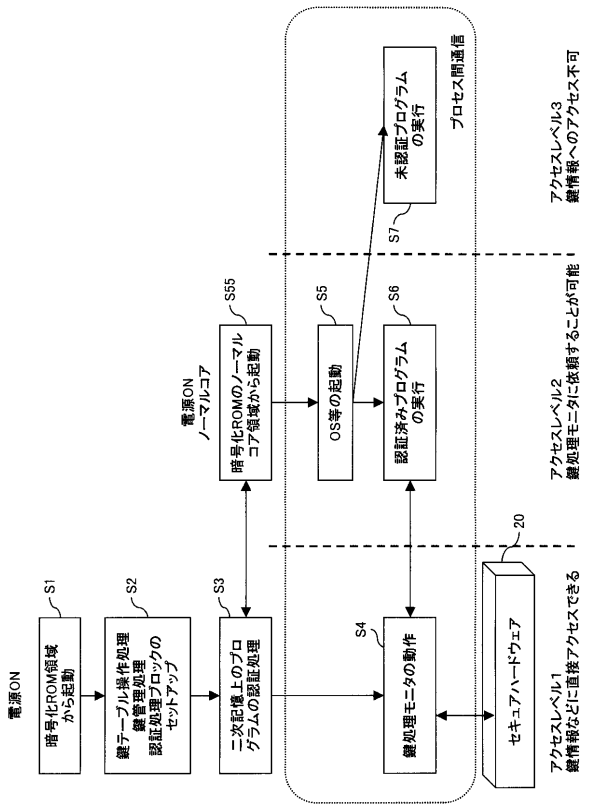
【図１２】

第2の実施例におけるプロセッサの
基本構成を示すブロック図



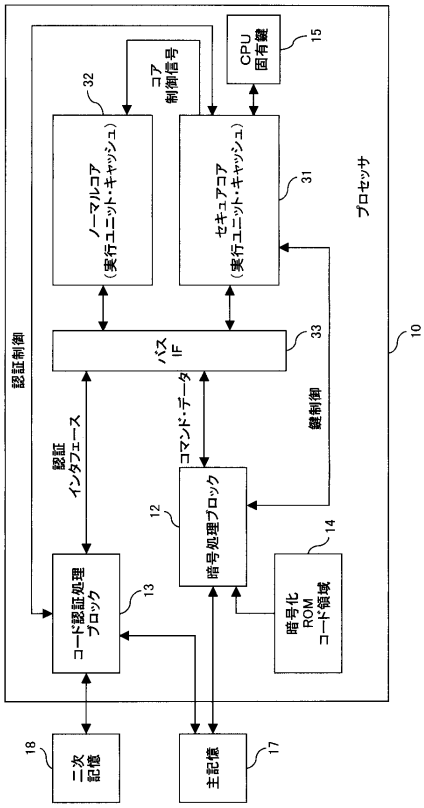
【図１３】

第2の実施例におけるプロセッサの
基本処理フローチャート



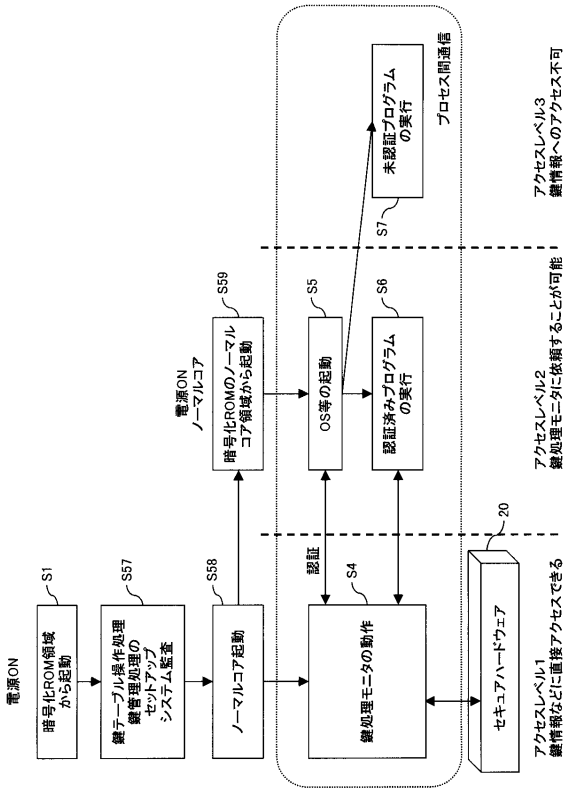
【図１４】

セキュアコアとノーマルコアを備えるプロセッサの
基本構成を示すブロック図



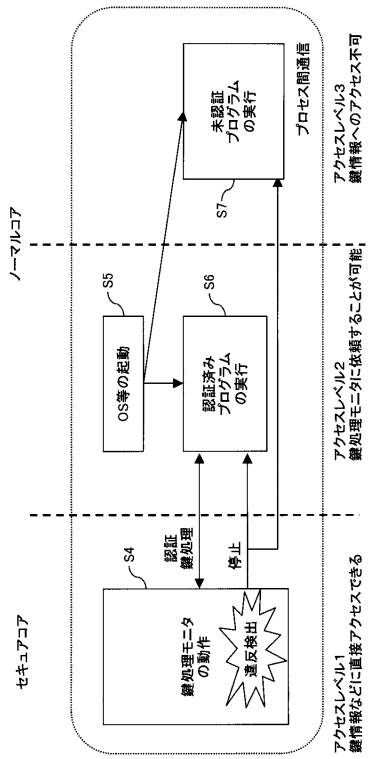
【図１５】

図14のプロセッサにおける処理の
基本フローチャート



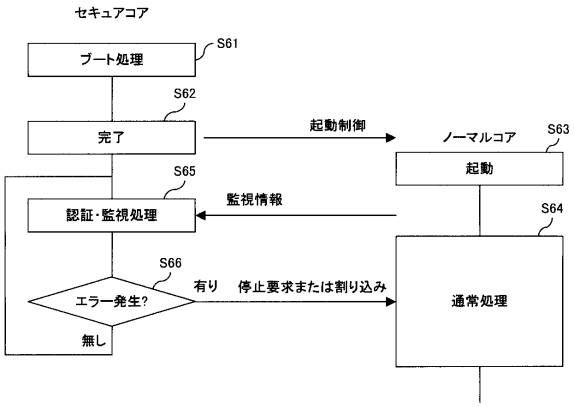
【図１６】

図14のプロセッサにおけるセキュアコアによる
ノーマルコアの動作の停止制御方式の説明図



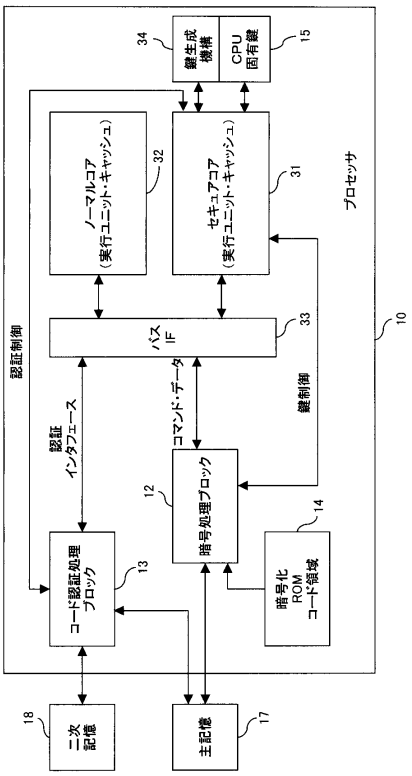
【図１７】

図14のプロセッサにおけるセキュアコアによる
ノーマルコアの動作の停止制御処理のフローチャート



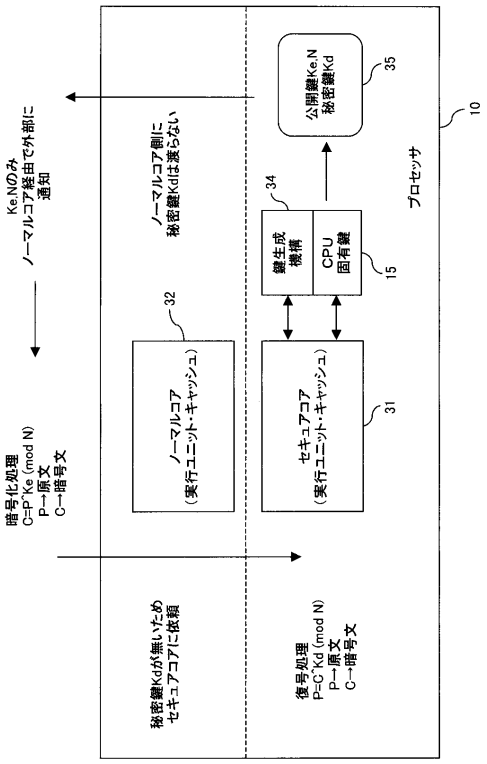
【図１８】

セキュアコアに対応する鍵生成機構を備える
プロセッサの構成ブロック図



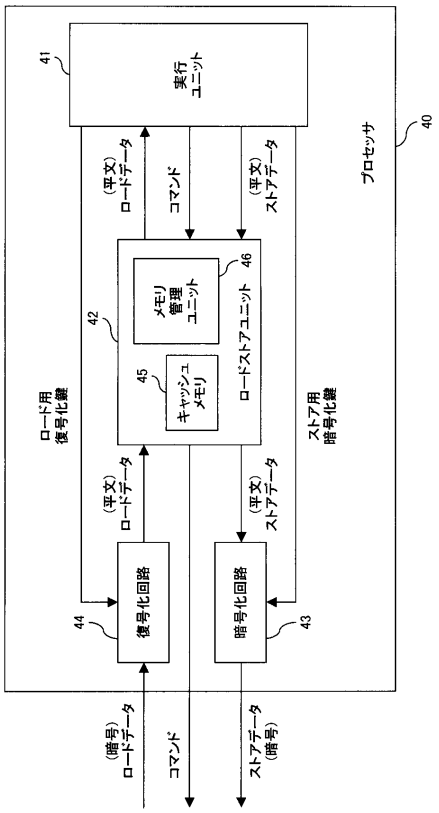
【図１９】

図18のプロセッサにおける鍵処理方式の具体例の説明図



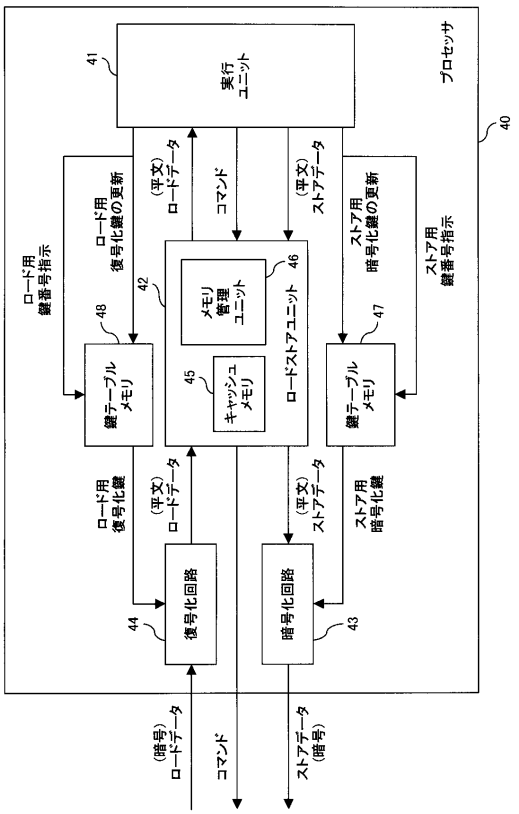
【図２０】

第3の実施例におけるプロセッサの基本構成を示すブロック図



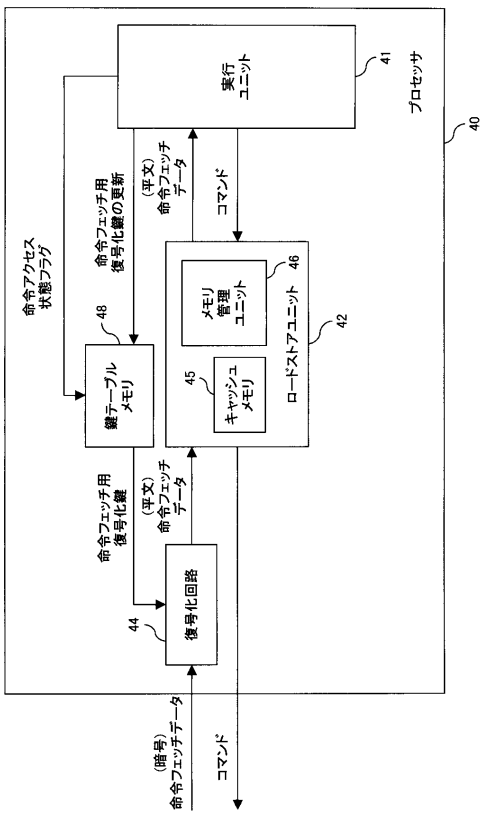
【図２１】

第3の実施例において鍵テーブルメモリを備えるプロセッサの構成ブロック図



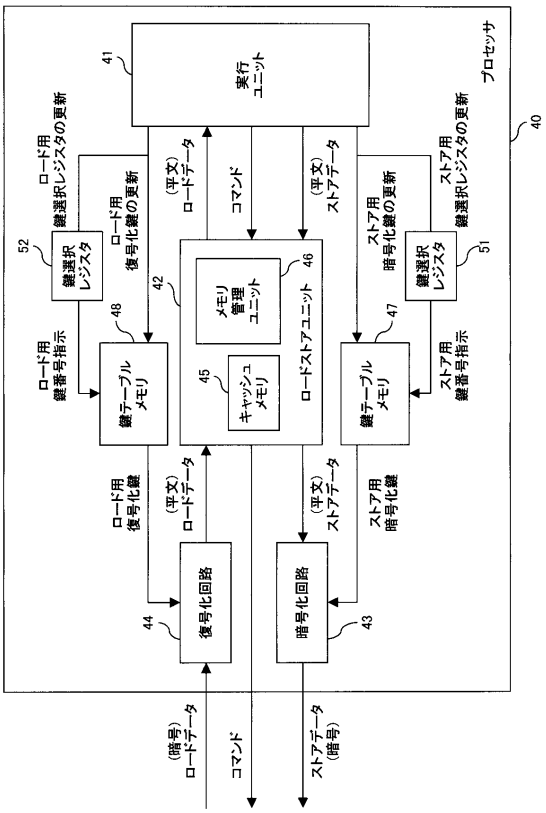
【図２２】

第3の実施例において命令アクセス状態にあるプロセッサの構成を示すブロック図



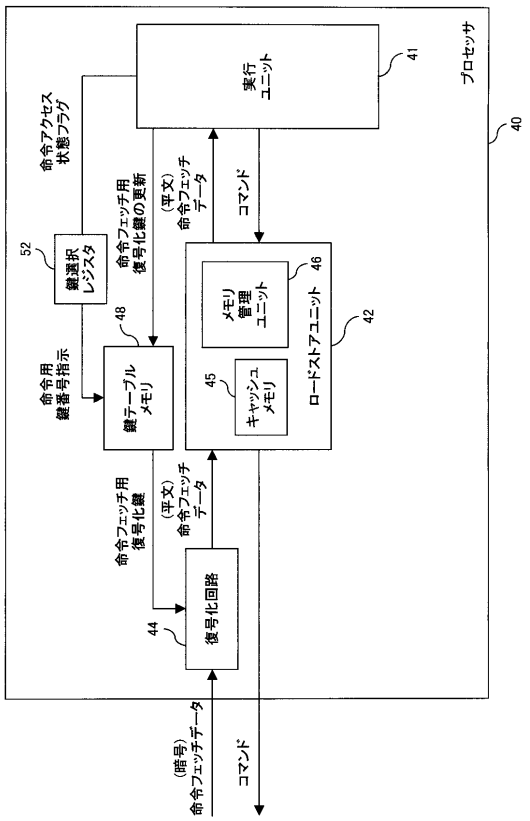
【図２３】

鍵テーブルメモリに対する鍵選択レジスタを備えるプロセッサの構成ブロック図



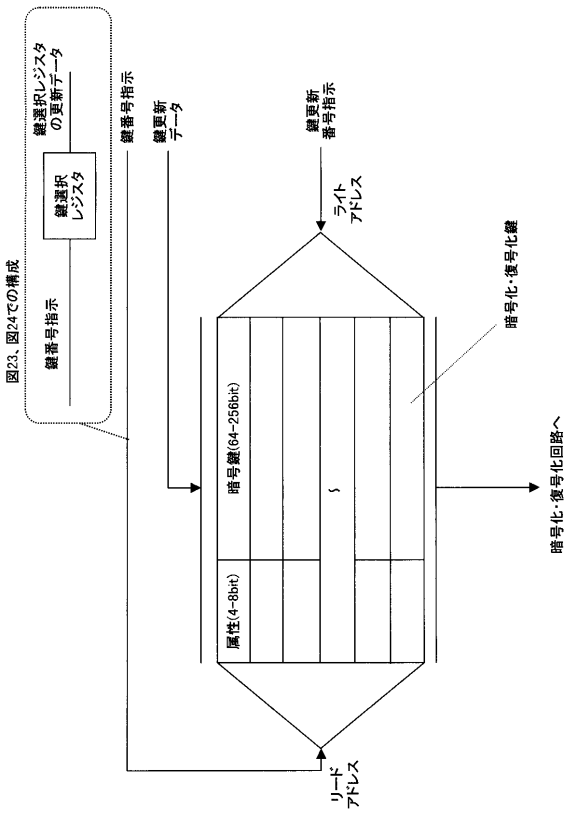
【図２４】

命令アクセス状態にあり、鍵テーブルメモリに対する鍵選択レジスタを備えるプロセッサの構成ブロック図



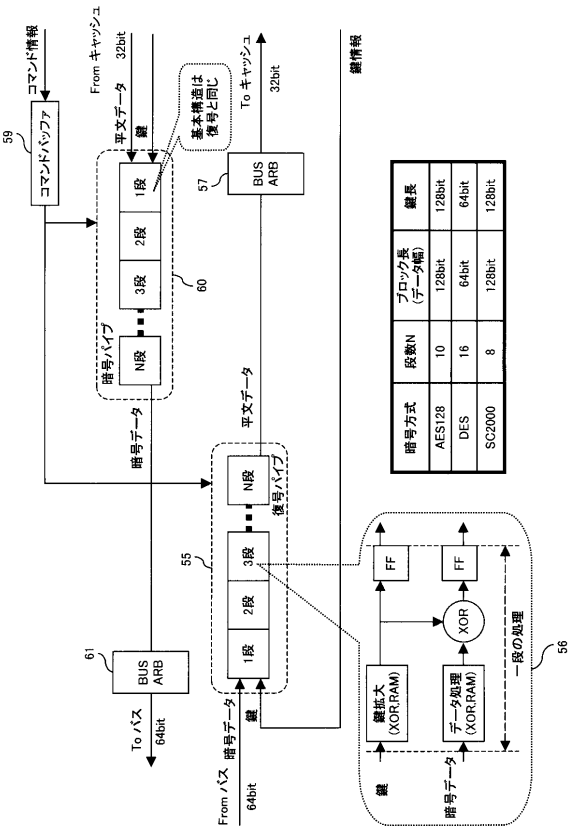
【図２５】

鍵テーブルメモリの構成例を示す図



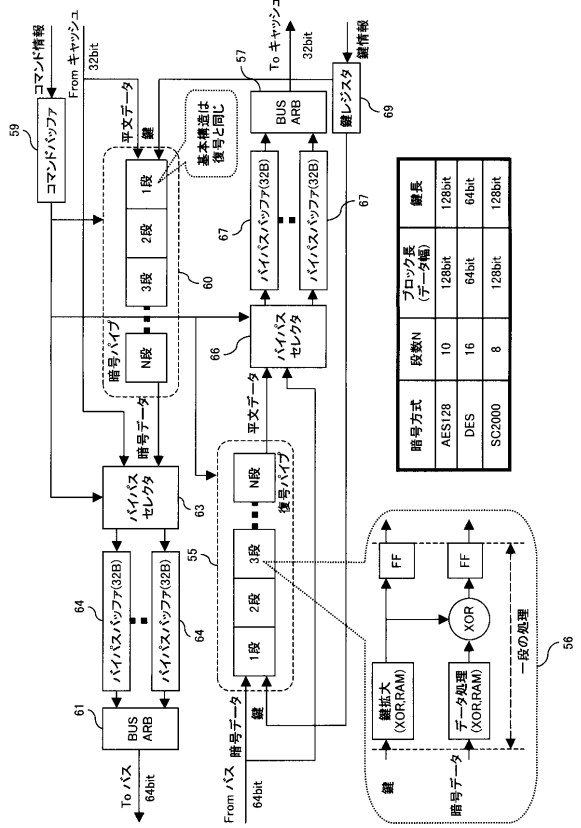
【図２６】

暗号化回路、復号化回路の構成例を示すブロック図



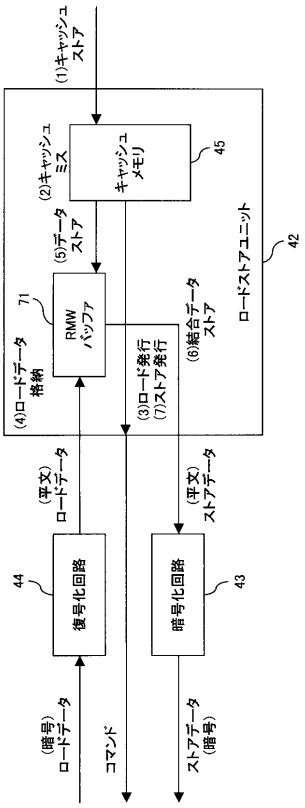
【図２７】

データ追い越し機能付き暗号化回路、
復号化回路の構成例を示す図



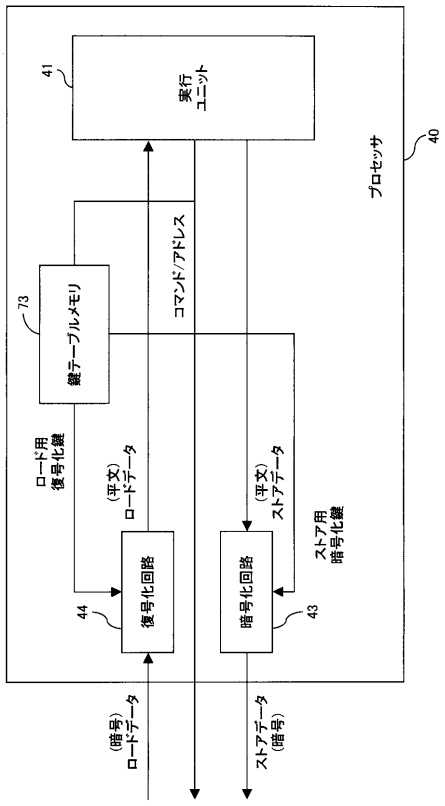
【図２８】

キャッシュスルー方式のロードストアユニットに対応する
リードモディファイライト方式の説明図



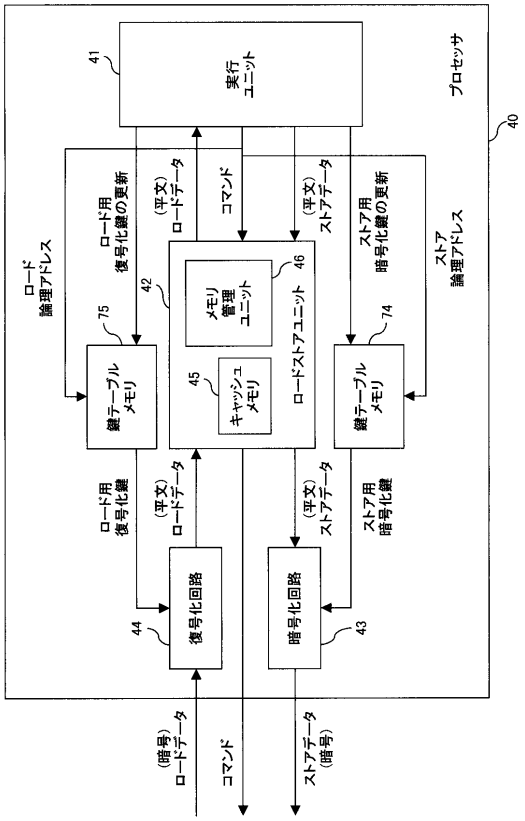
【図２９】

第4の実施例におけるプロセッサの
基本構成を示すブロック図



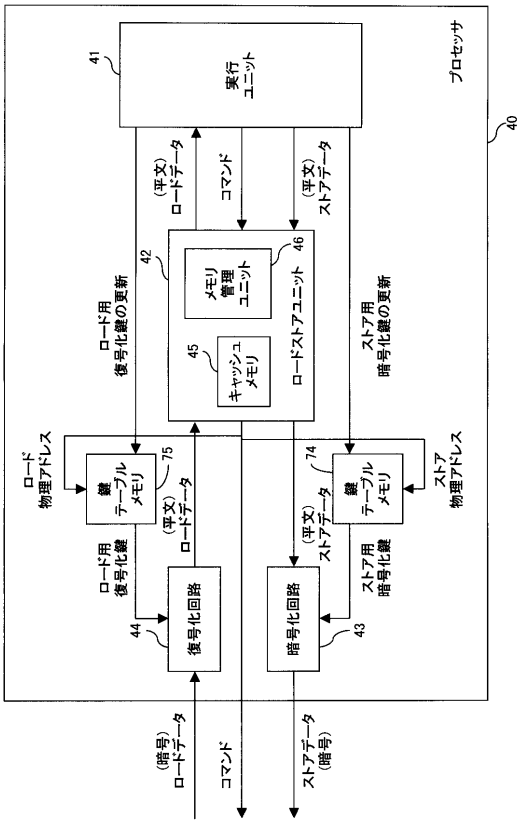
【図３０】

論理アドレスが与えられる鍵テーブルメモリを
備えるプロセッサの構成ブロック図



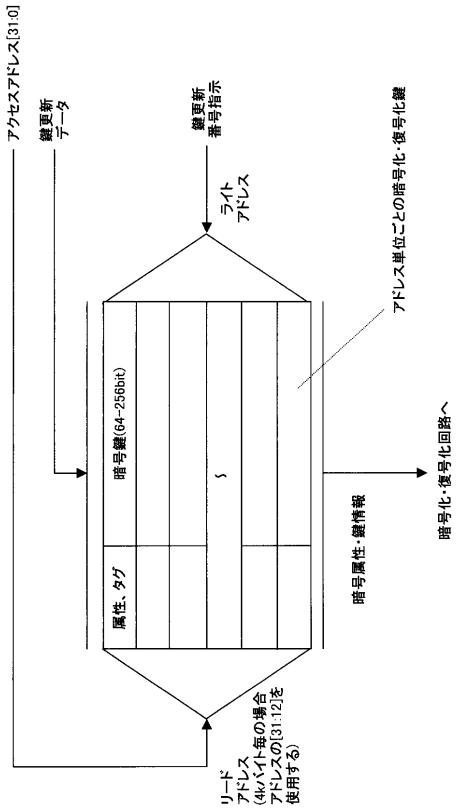
【図３１】

物理アドレスが与えられる鍵テーブルメモリを備えるプロセッサの構成ブロック図



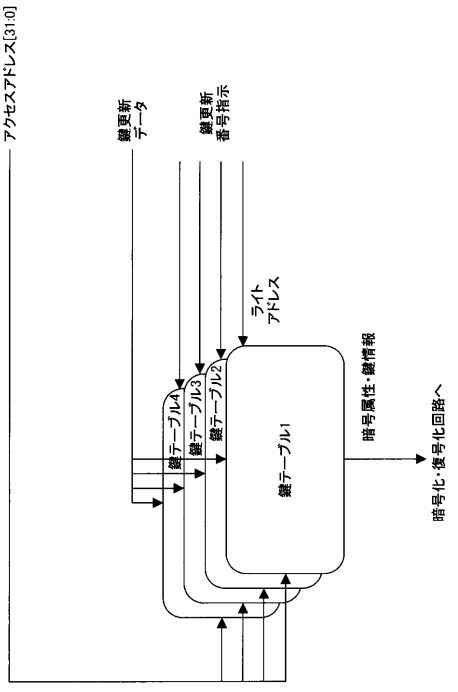
【図３２】

第４の実施例における鍵テーブルメモリの構成例（その１）を示す図



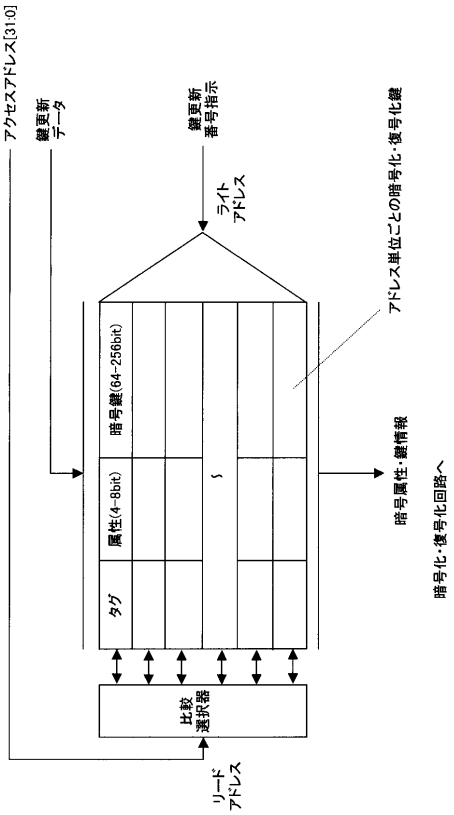
【図３３】

第４の実施例における鍵テーブルメモリの構成例（その２）を示す図



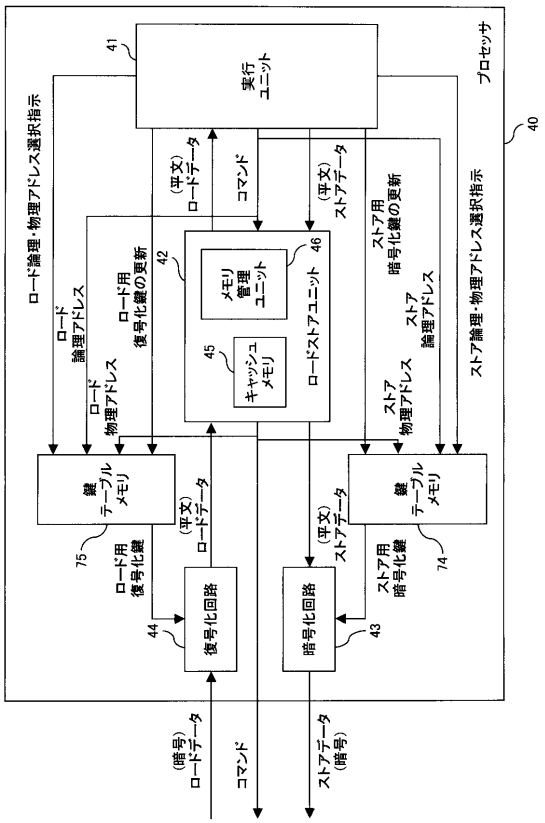
【図３４】

第４の実施例における鍵テーブルメモリの構成例（その３）を示す図



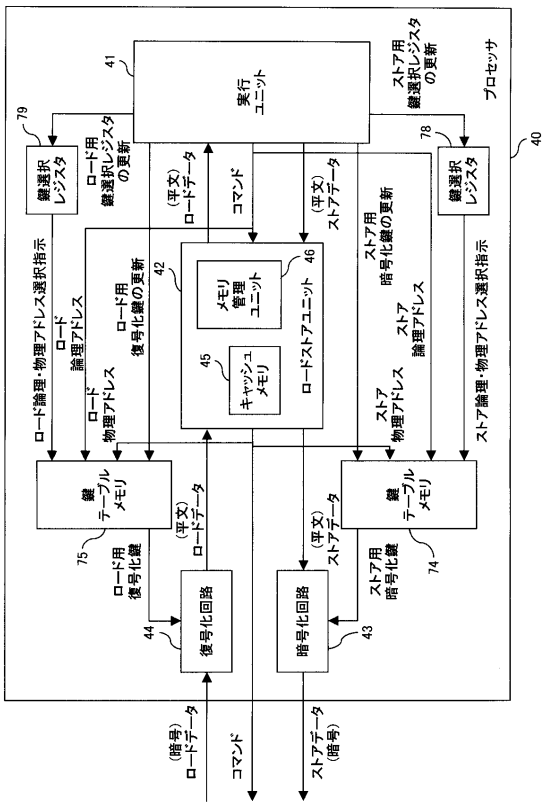
【図３５】

論理アドレスと物理アドレスとが与えられる鍵テーブルメモリを備えるプロセッサの構成を示すブロック図



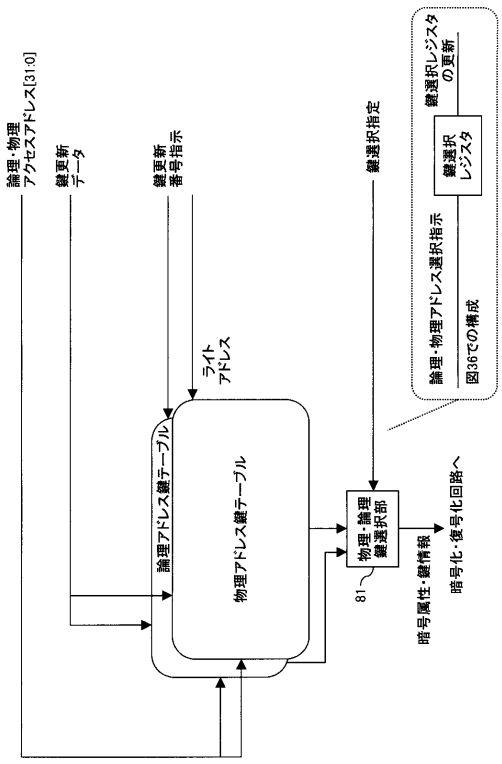
【図３６】

図35の鍵テーブルメモリに対してアドレス選択指示を与える鍵選択レジスタを備えるプロセッサの構成ブロック図



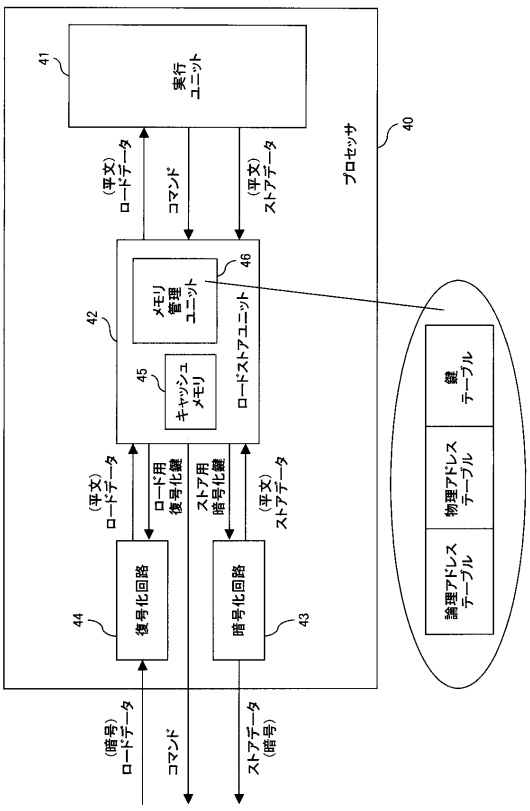
【図３７】

図35、図36における鍵テーブルメモリの構成例を示す図



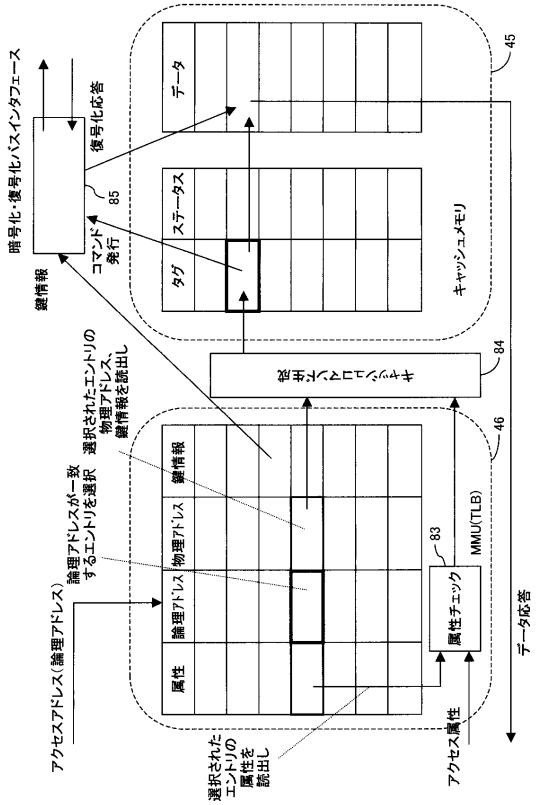
【図３８】

メモリ管理ユニット内に鍵テーブルが備えられるプロセッサの構成を示すブロック図



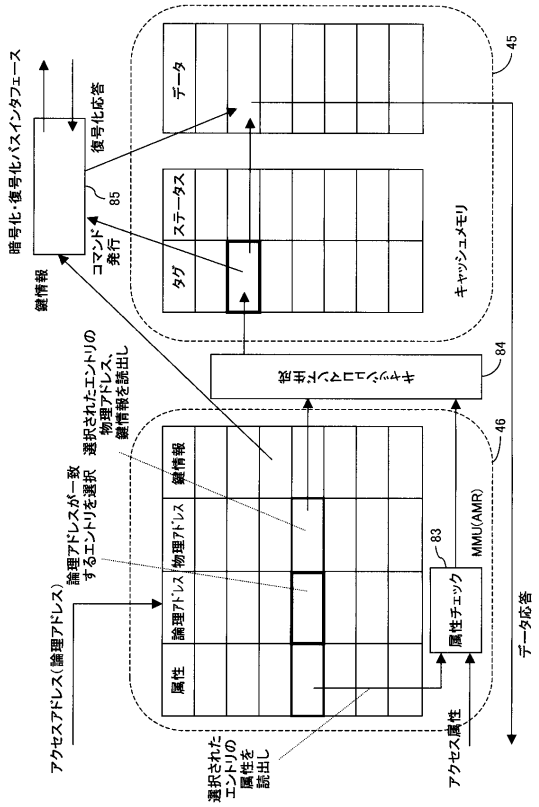
【図３９】

図38におけるデータアクセス方式の説明図



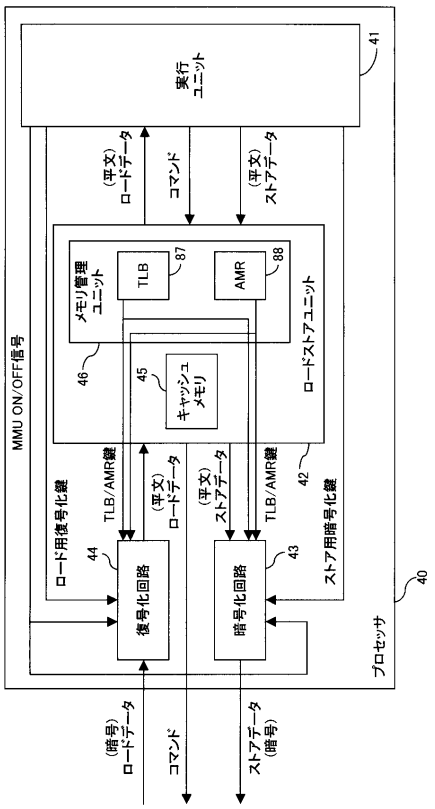
【図４０】

アドレスマップレジスタに鍵テーブルが併設されている場合のデータアクセス方式の説明図



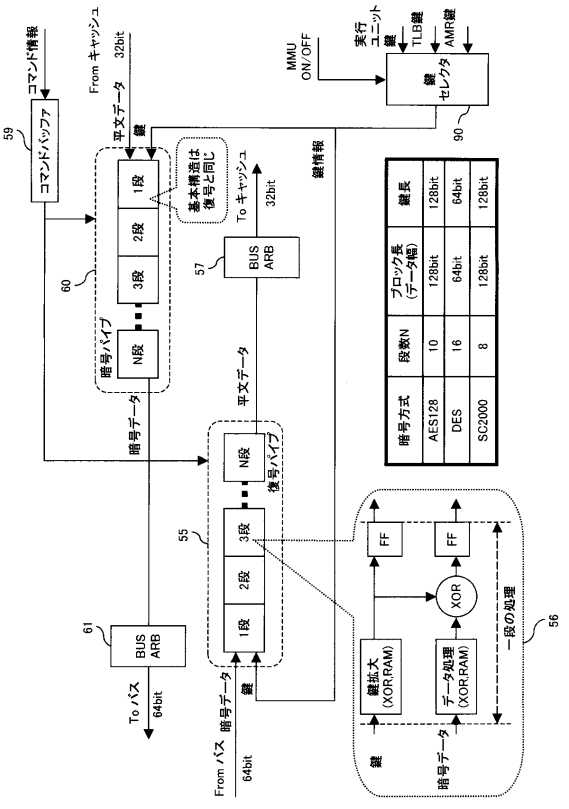
【図４１】

メモリ管理ユニットのON／OFF状態に応じて鍵を切り替えるプロセッサの構成を示すブロック図



【図４２】

メモリ管理ユニットのON／OFF状態に応じて鍵を切り替える暗号化／復号方式の説明図



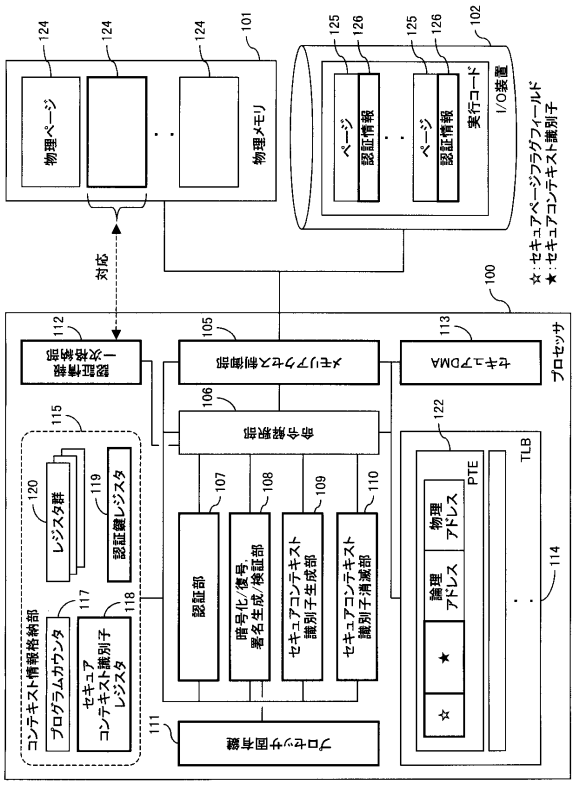
【図４３】

第3、および第4の実施例における
実行ユニットの入出力信号の説明図

項目	方向	図番号 20	21	22	23	24	21+22+23	27	29	30	31	35	36	38	40	41	備考
ロード復号化鍵	出力	○														○	
ロードデータ	入力	○	○		○		○	○	○	○	○	○	○	○	○	○	
ロード鍵番号指示	出力		○		○		○	○	○	○	○	○	○	○	○	○	
命令フェッチデータ	入力			○		○	○	○	○	○	○	○	○	○	○	○	
ストア暗号化鍵	出力	○														○	
ストアデータ	出力	○	○		○		○	○	○	○	○	○	○	○	○	○	
ストア鍵番号指示	出力		○				○									○	
コマンド	出力	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	アクセス種別、アクセスサイズなど
実行状態信号	出力			○		○	○	○	○	○	○	○	○	○	○	○	
レジスタライト信号	出力		○	○	○	○	○	△	○	○	○	○	△	△	△	○	
レジスタリードデータ	入力		○	○	○	○	○	△	○	○	○	○	△	△	△	○	
レジスタライトデータ	出力		○	○	○	○	○	△	○	○	○	○	△	△	△	○	
アクセスアドレス	出力	△	△	△	△	△	△	○	○	○	○	○	○	○	○	○	
ロード・ストア状態信号	出力	△	△	△	△	△	△	△	○	○	○	○	△	△	○	○	
鍵選択指示	出力												○			○	
MMU 状態信号	出力															○	
スーパバイザ・ユーザ状態信号	出力						○	○						○		○	
コンテキストIDデータ	出力							○							○	○	

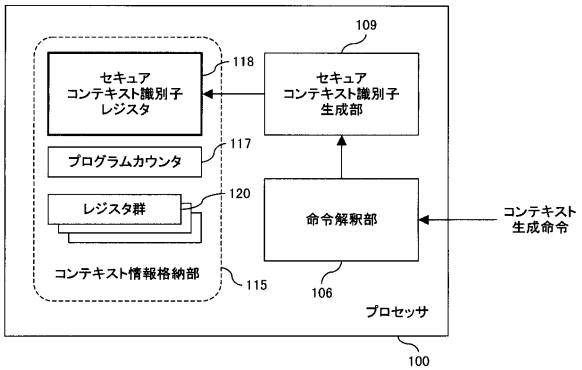
【図４４】

第5の実施例におけるプロセッサシステムの
詳細構成ブロック図



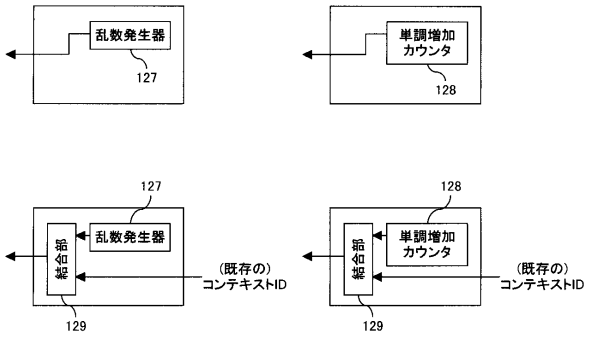
【図４５】

セキュアコンテキスト識別子
生成方式の説明図



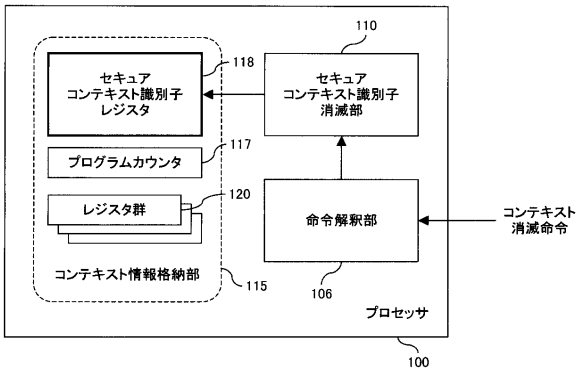
【図４６】

セキュアコンテキスト識別子
生成方法の説明図



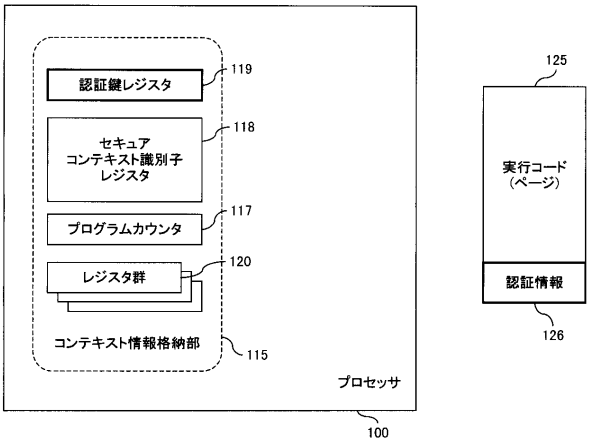
【図４７】

セキュアコンテキスト識別子
消滅方式の説明図



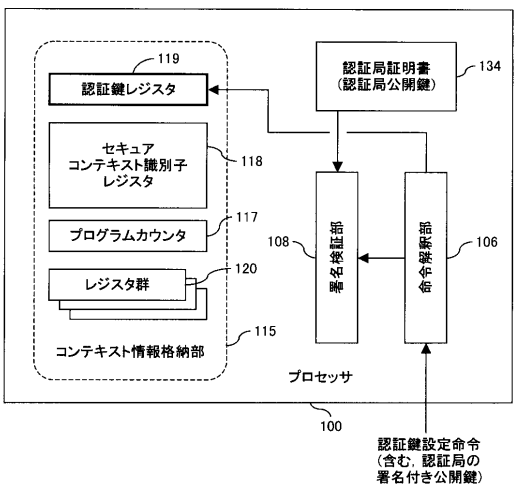
【図４８】

実行コードに付加された
認証情報の説明図



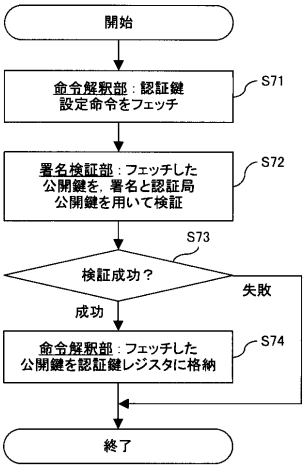
【図４９】

公開鍵の認証鍵レジスタへの
格納方式の説明図



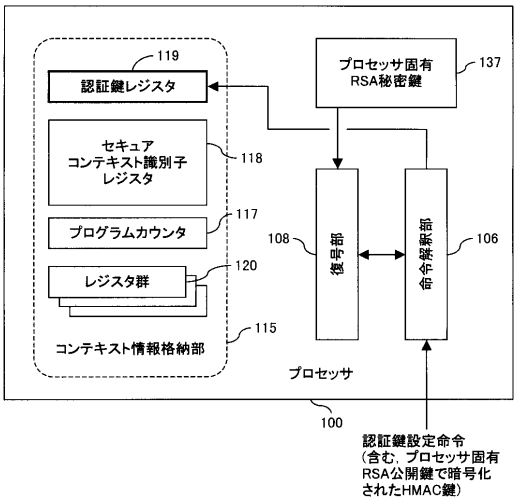
【図５０】

公開鍵の認証鍵レジスタへの
格納処理のフローチャート



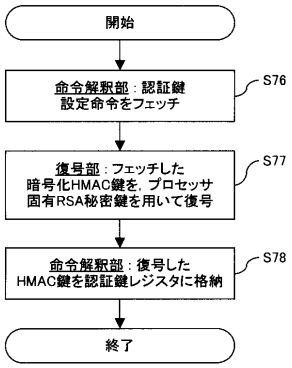
【図５１】

暗号化された共通鍵の認証鍵レジスタへの
格納方式の説明図



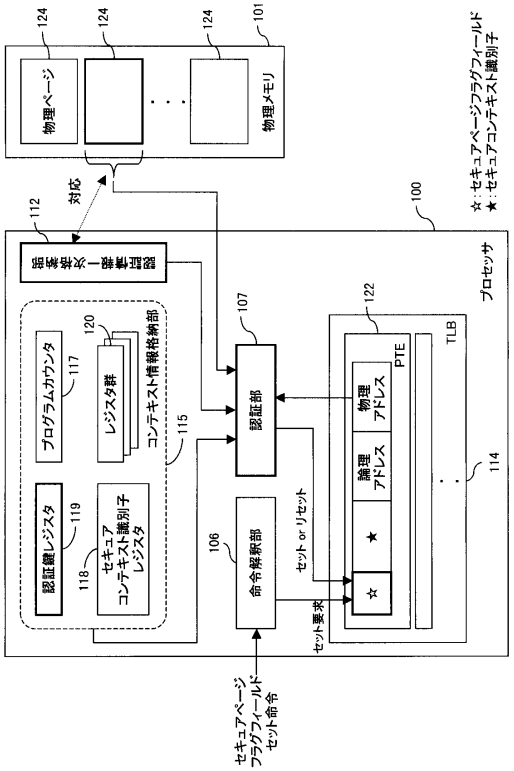
【図５２】

暗号化された共通鍵の認証鍵レジスタへの
格納処理フローチャート



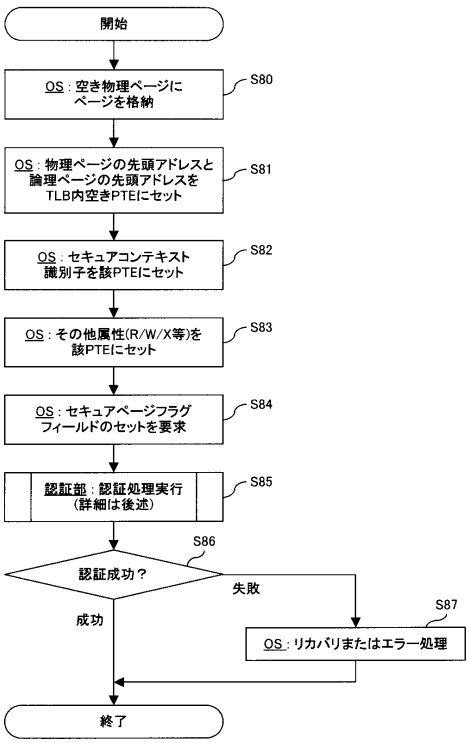
【図５３】

物理メモリへのページイン時の
処理方式の説明図



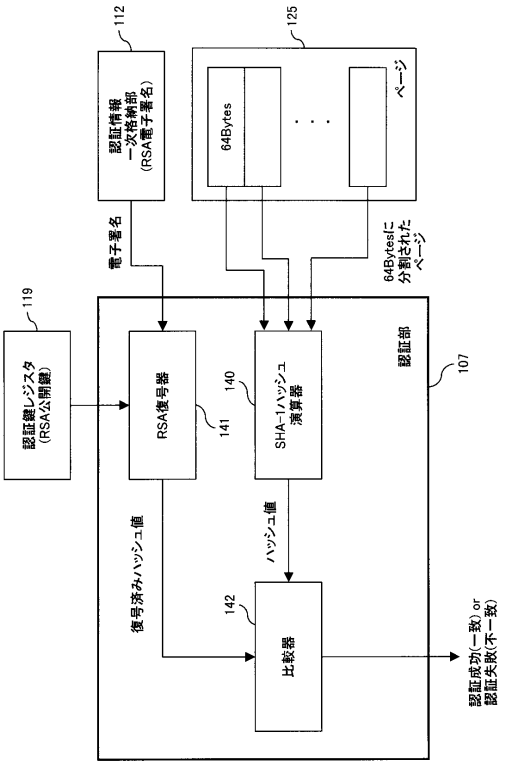
【図５４】

物理メモリへのページイン時の
処理フローチャート



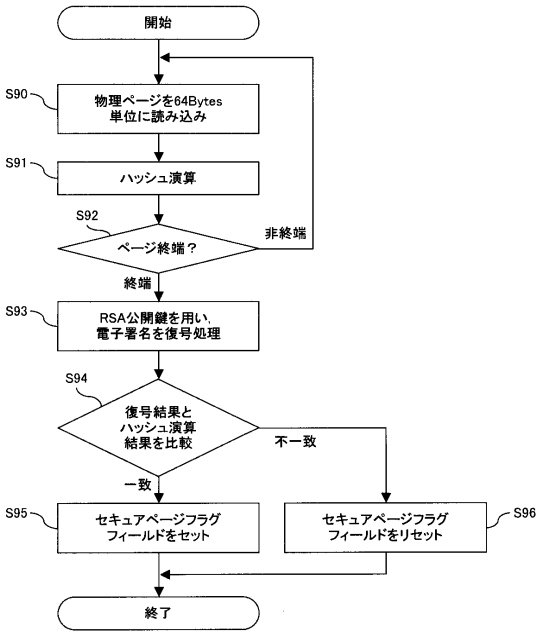
【図５５】

認証部の構成を示すブロック図



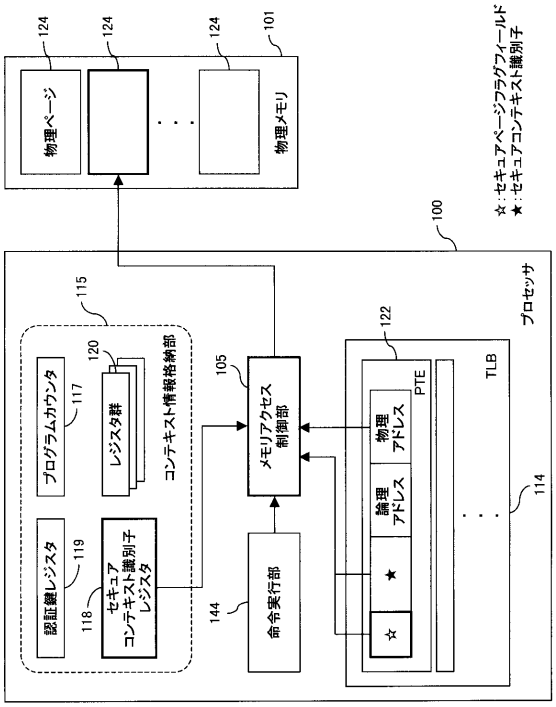
【図５６】

認証部の動作フローチャート



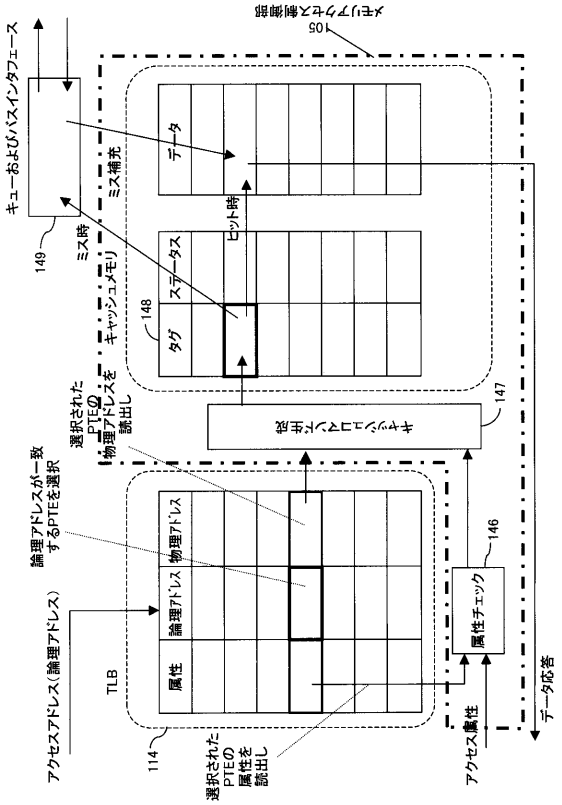
【図５７】

第5の実施例におけるページ利用時の
メモリアクセス制御部による
アクセスチェック方式の説明図



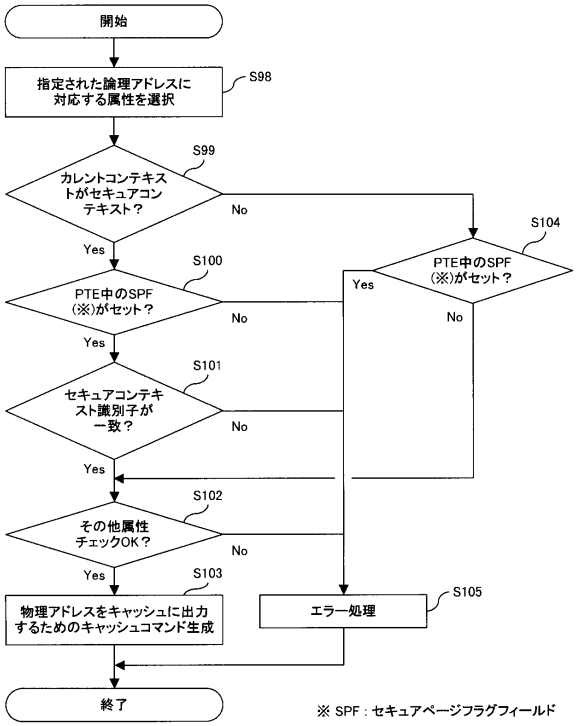
【図５８】

メモリアクセス制御部の動作例を説明する図



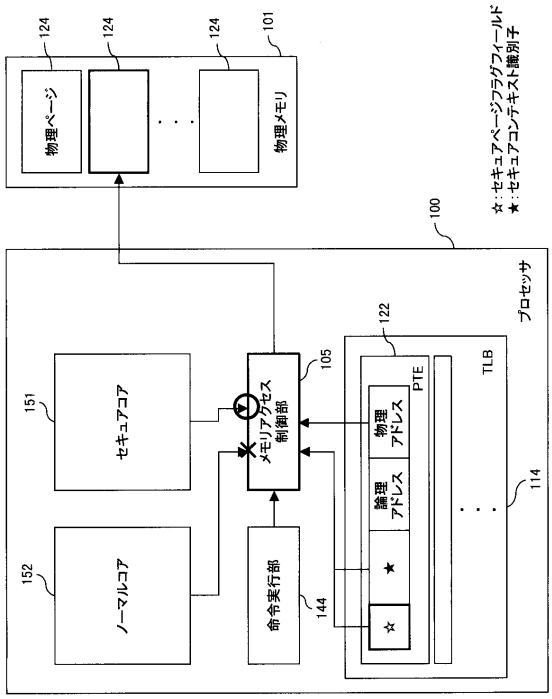
【図５９】

命令フェッチ時のメモリアクセス制御部の処理フローチャート



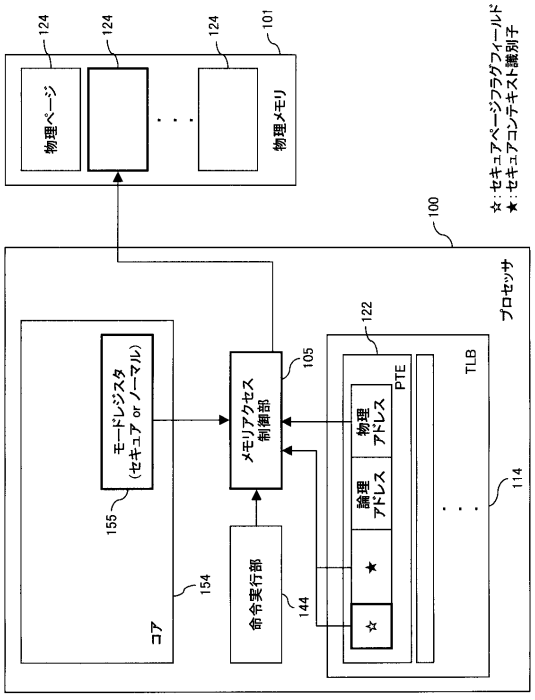
【図６０】

セキュアコアとノーマルコアからのページ利用時のアクセス制御方式を説明する図



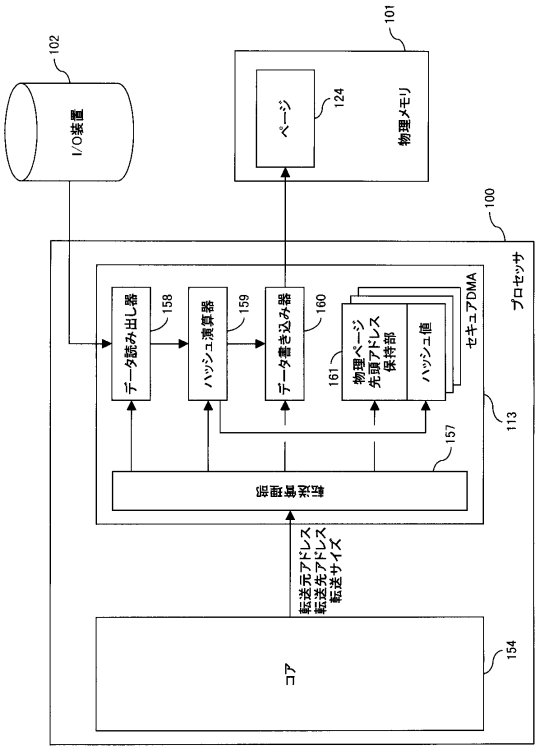
【図６１】

セキュアモードとノーマルモードを切り替えるためのモードレジスタを備えるプロセッサの構成図

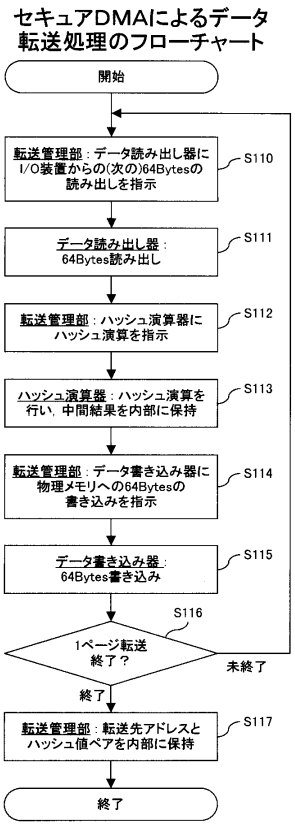


【図６２】

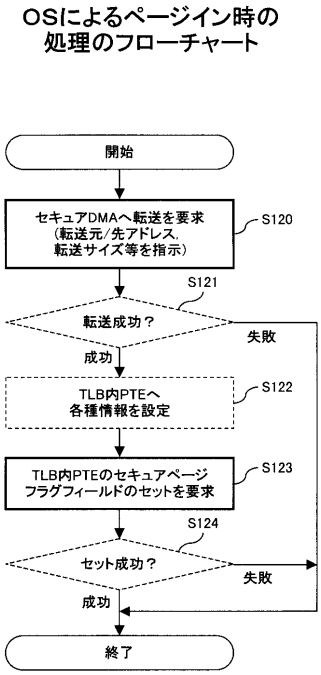
セキュアDMAの構成を示すブロック図



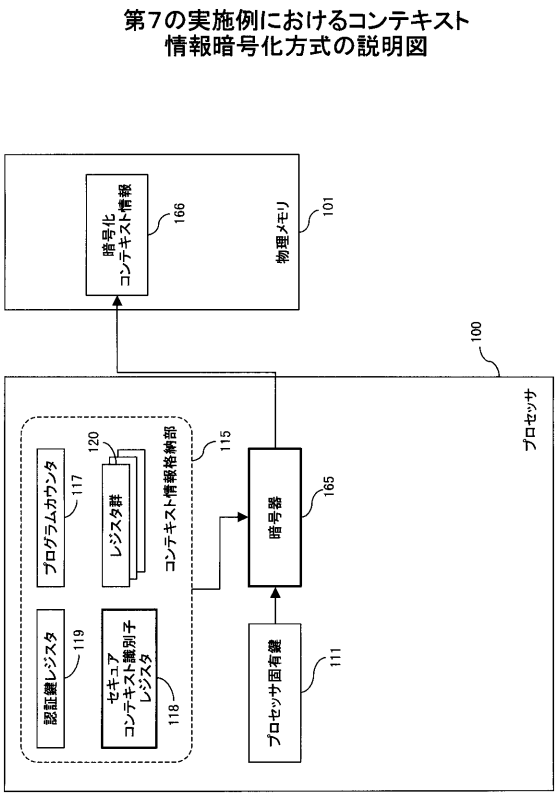
【図６３】



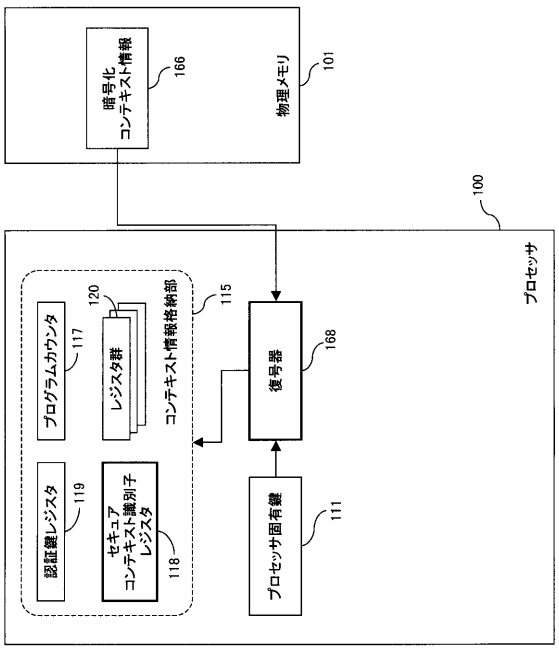
【図６４】



【図６５】

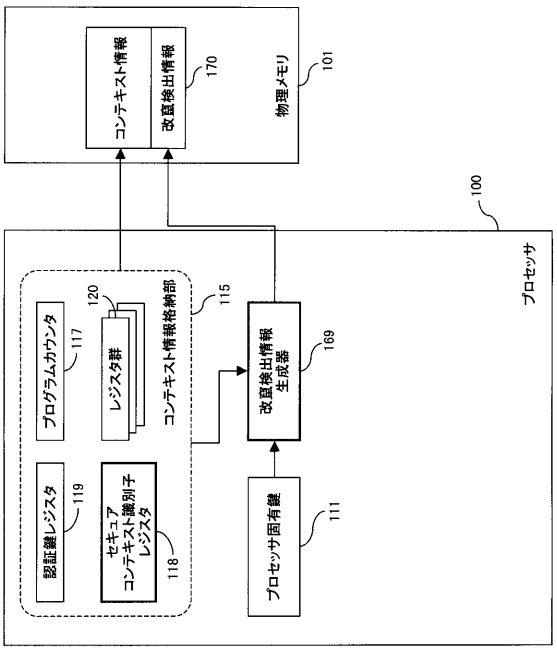


【図６６】



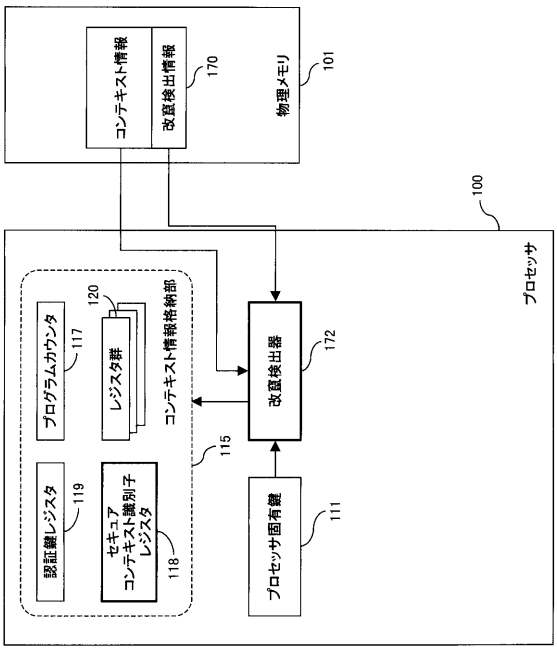
【図６７】

コンテキスト情報に対する改ざん
検出情報付加方式の説明図



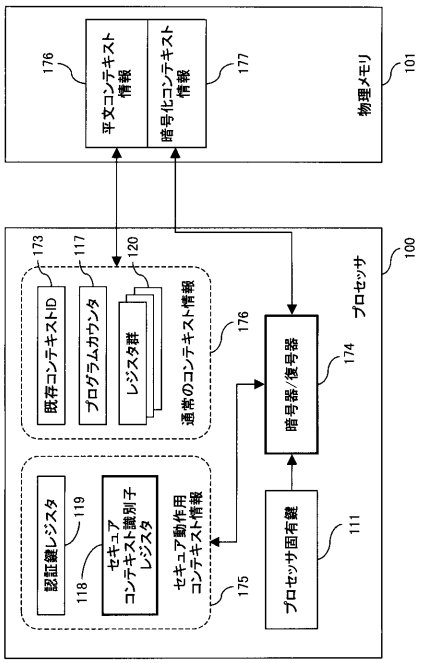
【図６８】

コンテキスト情報に対する改ざん
検出方式の説明図



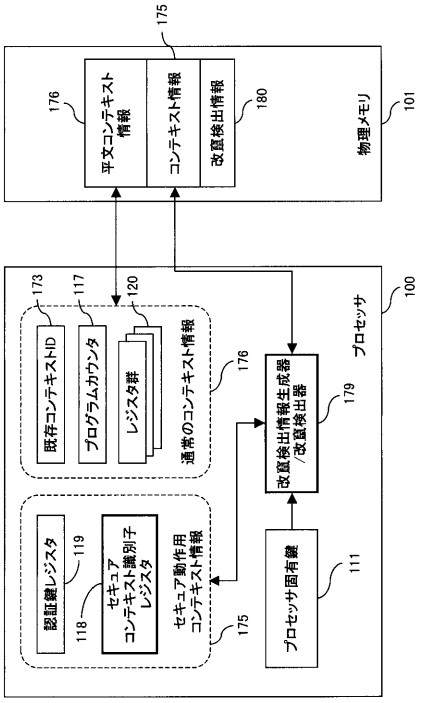
【図６９】

セキュア動作用コンテキスト情報の
暗号化方式の説明図



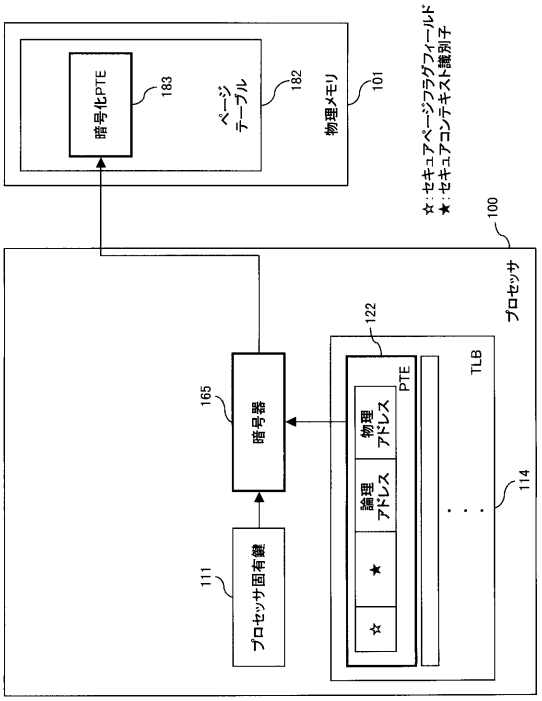
【図７０】

セキュア動作用コンテキスト情報に対する
改ざん検出情報付加方式の説明図



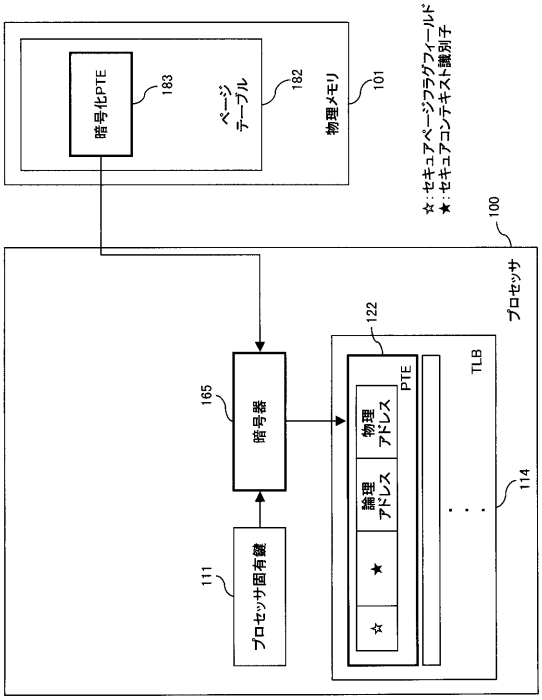
【図７１】

ページ・テーブル・エントリの
暗号化方式の説明図



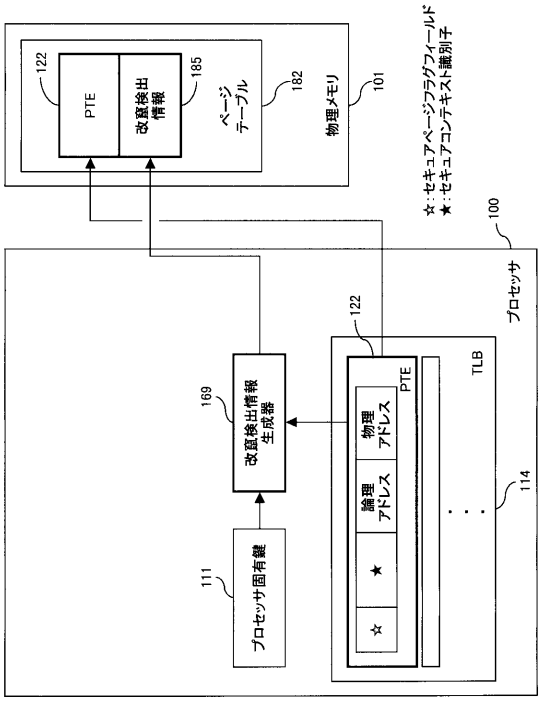
【図７２】

ページ・テーブル・エントリの
復号方式の説明図



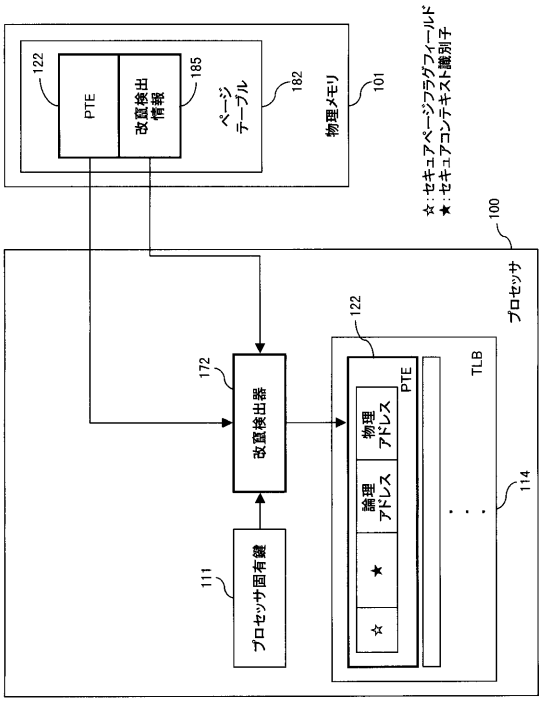
【図７３】

ページ・テーブル・エントリへの
改ざん検出情報付加方式の説明図



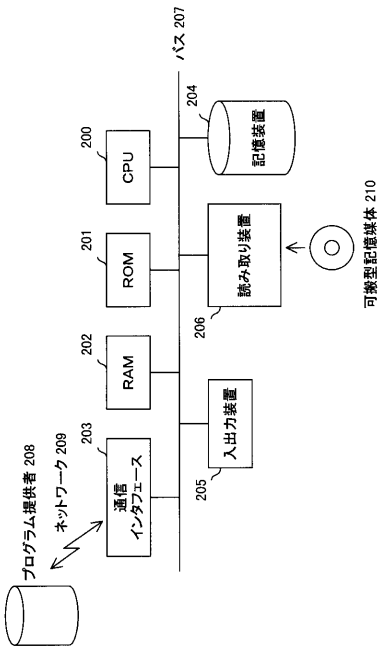
【図７４】

ページ・テーブル・エントリに対する
改ざん検出方式の説明図



【図７５】

本発明を実現するためのプログラムのコンピュータへのローディングを説明する図



フロントページの続き

(72)発明者　田宮　大司
神奈川県川崎市中原区上小田中４丁目１番１号　富士通株式会社内

審査官　和田　財太

(56)参考文献　特表２００１－５０８８９３（ＪＰ，Ａ）
国際公開第２００４／００６０７５（WO，Ａ１）
特開２００３－２８０９８９（ＪＰ，Ａ）
特開２００３－１０８４４２（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｇ０６Ｆ　１２／１４
Ｇ０６Ｆ　２１／６２
Ｈ０４Ｌ　９／３２