(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第5607546号
(P5607546)
(45)発行日　平成26年10月15日(2014.10.15)　(24)登録日　平成26年9月5日(2014.9.5)

(51)Int.Cl.　　F I
G06F 21/74 (2013.01)　　G06F 21/02 174
G06F 21/62 (2013.01)　　G06F 21/24 163C
G06F 15/78 (2006.01)　　G06F 15/78 510G
　　G06F 15/78 510K

請求項の数 21　(全 39 頁)

(21)出願番号　特願2010-545881 (P2010-545881)
(86)(22)出願日　平成21年2月6日(2009.2.6)
(65)公表番号　特表2011-511383 (P2011-511383A)
(43)公表日　平成23年4月7日(2011.4.7)
(86)国際出願番号　PCT/US2009/000768
(87)国際公開番号　WO2009/099647
(87)国際公開日　平成21年8月13日(2009.8.13)
審査請求日　平成24年2月3日(2012.2.3)
(31)優先権主張番号　61/063,925
(32)優先日　平成20年2月7日(2008.2.7)
(33)優先権主張国　米国(US)

(73)特許権者　501144003
アナログ・デバイシズ・インコーポレーテッド
アメリカ合衆国マサチューセッツ州ノーウッド，ワン・テクノロジー・ウェイ（番地なし）
(74)代理人　100102842
弁理士　葛和　清司
(72)発明者　ジョルダノ，フィリップ，ピー.
アメリカ合衆国　マサチューセッツ州　02019、ベリングハム、フラッグ　ドライブ　10

最終頁に続く

(54)【発明の名称】保護された動作モードの間にシステムアクセスを制御するための方法および装置

(57)【特許請求の範囲】
【請求項１】
プロセッサのリソースへのセキュアアクセスを提供するセキュアモードを含む、複数のモードで動作するように構成されるプロセッサであって、
署名者によって署名されるメッセージおよびファームウェアコードを保存するように構成されるメモリ、
第一ステートおよび第二ステートを含む、複数のステートの中の１つのステートを示すように構成された第一レジスタビットであって、プライベートエミュレーション命令が実行される場合は第一ステートを示すように構成され、プライベートエミュレーション命令が無視される場合には第二ステートを示すように構成される、前記第一レジスタビット、
次のセキュアモードへ入る時に、第一レジスタビットが第一ステートを示すか、または第二ステートを示すかを制御する、第二レジスタビット、および
セキュアモード以外でメッセージを認証するためのファームウェアコードを実行し、メッセージの認証が成功すると、第二レジスタビットに従って第一レジスタビットを設定し、セキュアモードに入るように構成される、論理ユニット
を含む、前記プロセッサ。
【請求項２】
第二レジスタビットが、セキュアモードにおいてのみ書き込み可能であって、次のセキュアモードへ入る時に第一レジスタビットが第一ステートを示すように制御するためのものである、請求項１に記載のプロセッサ。

【請求項３】
メッセージがソフトウェアコードであり、論理ユニットがさらに、セキュアモードに入った後にソフトウェアコードを実行するように構成される、請求項２に記載のプロセッサ。
【請求項４】
ソフトウェアコードが、セキュアモードで実行される場合に、次のセキュアモードへ入る時に第一レジスタビットが第一ステートを示すように制御するために、第二レジスタビットをセットする、請求項３に記載のプロセッサ。
【請求項５】
プロセッサがさらに、プライベートエミュレーション命令を受信し、論理ユニットにプライベートエミュレーション命令を提供するように動作可能に構成されるデバッグポートを含み、論理ユニットが、第一レジスタビットに従ってプライベートエミュレーション命令を実行するように構成される、請求項１に記載のプロセッサ。
【請求項６】
デバッグポートおよびプライベートエミュレーション命令がJTAG標準に従う、請求項１に記載のプロセッサ。
【請求項７】
メモリがさらにデジタル署名を保存するように構成され、
ファームウェアコードがコンピュータ実行可能な命令を含み、論理ユニットにより実行される場合に、
メッセージからハッシュ値を決定する、
パブリックキーと共にデジタル署名を復号化する、および
復号化されたデジタル署名をハッシュ値と比較する
ことを含む方法を実行することによって、メッセージを認証する、請求項１に記載のプロセッサ。
【請求項８】
セキュアモードにおいてのみアクセス可能なプライベートメモリ領域をさらに含む、請求項１に記載のプロセッサ。
【請求項９】
プロセッサが、論理ユニットにより実行可能な命令のメモリアドレスを保存するためのプログラムカウンターをさらに含み、論理ユニットが、ファームウェアコードを実行する間に、プログラムカウンターに保存されるメモリアドレスが、ファームウェアコードのメモリアドレスと一致しない場合に、ファームウェアコードの実行を中止するように構成される、請求項１に記載のプロセッサ。
【請求項１０】
セキュアモードを含む、複数のモードで動作可能であるマイクロプロセッサの動作方法であって、
（a）セキュアモード外で、署名者によって署名されるメッセージを認証する、
（b）動作（a）が正常に完了すると、セキュアモードに入り、プライベートエミュレーション命令が実行されるかまたは無視されるかの定義を制御する第一のレジスタから第一のステートを読み込み、第一のステートに基づいて、第二のレジスタに第二のステートを書き込み、第一のレジスタはセキュアモードにおいてのみ第一のステートに書き換え可能であり、第二のステートがエミュレーション命令が実行されることを定義する、
（c）セキュアモードにおいて、エミュレーション命令が、第二のレジスタの読み込みに基づいて実行されることを決定する
動作を含む、前記マイクロプロセッサの動作方法。
【請求項１１】
（d）動作（a）の前のセキュアモードにおいて、第一のステートを第一のレジスタに書き込むように構成された利用者により入力されるセットアップコードを実行し、第一のステートが、次のセキュアモードの期間においてエミュレーション命令が実行されることを

示す、および
（e）セキュアモードを終了する
動作をさらに含む、請求項１０に記載の方法。
【請求項１２】
メッセージがプロセッサにより実行可能なターゲットコードを含み、動作方法が、
（d）セキュアモードにおいて、ターゲットコードを実行する、および
（e）動作（c）の次に、エミュレーション命令を実行することを更に含み、エミュレーション命令は実行される場合にターゲットコードの実行を制御するように構成されているものである、請求項１０に記載の方法。
【請求項１３】
動作（a）が、
メッセージのためのハッシュ値を決定する、
パブリックキーと共にデジタル署名を復号化する、および
復号化されたデジタル署名をハッシュ値と比較する
動作を含む、請求項１０に記載の方法。
【請求項１４】
ハッシュ値がSHA-1ハッシュアルゴリズムを利用して決定され、デジタル署名が楕円曲線暗号を利用して復号化される、請求項１３に記載の方法。
【請求項１５】
動作（b）がさらに、第三のレジスタに保存された値に基づいて、プライベートメモリへのアクセスを選択的に有効にすることを含む、請求項１０に記載の方法。
【請求項１６】
セキュアモードを含む、複数のモードで動作可能であるプロセッサであって、
プライベートエミュレーション命令が実行される場合は第一値を保存し、プライベートエミュレーション命令が無視される場合は第二値を保存するように構成された、第一メモリ、
プロセッサがセキュアモードへ入る時に、第一メモリが第一値を保存するか、または第二値を保存するかを制御するように構成された、第二メモリ、および
プロセッサがセキュアモードへ入る時に、第二メモリに基づいて第一メモリをセットする、論理ユニット
を含む、前記プロセッサ。
【請求項１７】
第一メモリが、プロセッサがセキュアモードで動作している時のみ、第一値に書き込み可能である、請求項１６に記載のプロセッサ。
【請求項１８】
ファームウェアを保存するROMをさらに含み、論理ユニットが、セキュアモードに入る認証を行うファームウェアコードを実行するように構成される、請求項１６に記載のプロセッサ。
【請求項１９】
論理ユニットにより実行される場合に、ファームウェアコードが、署名者によって署名されるメッセージの信頼性を決定し、メッセージが承認されない場合はセキュアモードへ入ることを認めない、請求項１８に記載のプロセッサ。
【請求項２０】
セキュアモード動作において、プロセッサ内のターゲットコードをデバッグする方法であって、プロセッサが、プライベートエミュレーション命令が実行または無視されるかを定義する第一メモリ、および、次のセキュアモードの期間において、第一メモリのプライベートエミュレーション命令が実行または無視されるかの定義を制御する第二メモリを含み、プロセッサが、セキュアモードを含む複数のモードで動作可能であり、
（a）利用者により入力されるセットアップコードを認証し、セキュアモードに入る、
（b）セキュアモードにおいてセットアップコードを実行し、セットアップコードが、

セキュアモードの次の期間において、プライベートエミュレーション命令が実行されることを示すように第二メモリをセットするように構成する、
（c）セキュアモードを終了する、
（d）ターゲットコードを認証する、
（e）動作（d）の次に、第二メモリに基づいて第一メモリをセットし、セキュアモードに入る、および
（f）動作（e）の次に、セキュアモードにおいて、ターゲットコードの実行を、プライベートエミュレーション命令を介して制御する
動作を含む、前記方法。
【請求項２１】
プライベートエミュレーション命令が、プライベートJTAGエミュレーション命令である、請求項１０に記載の方法。
【発明の詳細な説明】
【技術分野】
【０００１】
背景
集積回路およびコンピュータの生産および利用が極端に速い速度で行われることに伴い、またほかのタイプのデジタルモジュール動作において、設計者は、集積チップの内部構造および動作のテストおよび診断のための方法が、有益で重要な役割を果たすことをますます認識するようになってきた。論理回路を集積チップに埋め込むための基準を創設することの望ましさが、IEEE委員会により提案および開発され、IEEE 1149.1として指定されるバウンダリスキャンアーキテクチャを提案した。標準IEEE1149.1または“JTAG”は電子回路のテストのための規格を定義するために開発された。JTAGの頭文字は、もともとは標準IEEEを定義したjoint test access group (JTAG)として指定された委員会団体に由来する。JTAGは、“内部チップ”機能のテストおよびプリント基板上のチップ相互接続のテストを含む、幅広い種類のテスト機能のために利用される。
【０００２】
デジタル署名認証は、電子メッセージの信頼性を決定し、メッセージの発信者を識別するための手段を提供する。この形式の認証を利用する場合、メッセージは、メッセージに関連するデジタル署名と共に送信される。デジタル署名は、パブリックキー暗号（すなわち非対称暗号）技術を利用して、生成および照合される。非対称暗号は、２つの異なる、しかし数学的に関連したキーすなわちパブリックキーおよびプライベートキーを利用したアルゴリズムを採用している。プライベートキーは、デジタル署名の生成、またはデータを一見理解不能な形式に変換するために利用される。パブリックキーは、デジタル署名の照合、またはメッセージをその元の形式に戻す。
【０００３】
概要
マイクロプロセッサは取扱注意および／または秘密情報を保存するための信頼することができるものである。プロセッサは、ソフトウェア開発のデバッキング機能を提供するものとして、一方でそれと同時に、プロセッサに保存された取扱注意および／または秘密情報のためのセキュリティーを提供するものとして述べられる。
いくつかの態様では、プロセッサは、オープンモード,セキュアエントリーモード,セキュアモードのうちの１つで動作する。オープンモードでは、いくつかのレジストリビットへのアクセス防止およびプライベートメモリ領域（例えば、取扱注意情報が保存される場所）へのアクセス防止以外には、セキュリティー対策は実施されない。セキュアエントリーモードは、プロセッサ上でセキュアコードの実行要求が受信され認証された場合に開始される。いくつかの態様では、認証はデジタル署名を利用して行われる。一度認証されると、セキュアコードは、プライベートメモリ領域がアクセス可能であるセキュアモードで実行される。セキュアコードは、プライベートメモリ領域にアクセスし、レジストリへのより高いアクセスおよび制御を持つ。認証が失敗した場合には、状態はオープンモードに

戻る。

【０００４】

いくつかの態様では、本発明は、プロセッサのリソースへのセキュアアクセスを提供するセキュアモードを含む、複数のモードで動作するように構成されたプロセッサに関する。プロセッサはメモリ、第一レジスタビット、第二レジスタビットおよび論理ユニットを含む。メモリは、メッセージおよびファームウェアコードを保存するように構成される。第一レジスタビットは、第一ステートおよび第二ステートを含む、複数のステートの中の１つのステートを示すように構成されており、第一レジスタビットは、プライベートエミュレーション命令が実行される場合は第一ステートを示すように構成され、プライベートエミュレーション命令が無視される場合には第二ステートを示すように構成される。第二レジスタビットは、次のセキュアモードへ入る時に、第一レジスタビットが第一ステートを示しているのか、または第二ステートを示しているのかを示す。論理ユニットは、セキュアモード外でメッセージを認証するためのファームウェアコードを実行し、メッセージの認証が成功すると、第二レジスタビットに従って第一レジスタビットを設定し、セキュアモードに入る。

【０００５】

別の態様では、本発明は、セキュアモードを含む、複数のモードで動作可能なマイクロプロセッサの動作方法に関する。動作方法は、次のものを含む。メッセージを認証するセキュアモード外の動作；メッセージの認証動作が正常に完了すると、セキュアモードに入り、第一レジスタから第一ステートを読み込み、第一ステートに基づいて、第二レジスタに第二ステートを書き込み、第一レジスタはセキュアモードにおいてのみ第一ステートに書き換え可能であり、第二ステートが実行されるエミュレーション命令を示す；セキュアモードにおいて、エミュレーション命令が、第二レジスタの読み込みに基づいて実行されることを決定する。

別の態様では、本発明は、セキュアモードを含む、複数のモードで動作可能なプロセッサに関する。プロセッサは第一メモリ、第二メモリおよび論理ユニットを含む。第一メモリは、プライベートエミュレーション命令が実行される場合には第一値を、プライベートエミュレーション命令が無視されるべき場合には第二値を保存するように構成される。第二メモリは、プロセッサがセキュアモードに入る場合に、第一メモリが第一値を保存するのか、または第二値を保存するのかを示すように構成されている。論理ユニットは、プロセッサがセキュアモードに入っている場合に、第二メモリに基づいて第一メモリを設定するように構成されている。

【０００６】

さらに別の態様では、本発明は、セキュアモード動作中のプロセッサ上のターゲットコードのデバッキング方法に関し、プロセッサは、プライベートエミュレーション命令が実行されるか無視されるかを示す第一メモリ、および次のセキュアモードの期間において、プライベートエミュレーション命令が実行されるか無視されるかを示す第二メモリを含み、プロセッサは、セキュアモードを含む複数のモードで動作可能である。方法は、セットアップコードを認証してセキュアモードに入る動作を含み；セキュアモードにおいてセットアップコードを実行し、セットアップコードは、セキュアモードの次の期間においてプライベートエミュレーション命令が実行されることを示すように第二メモリを設定し；セキュアモードから抜けて；ターゲットコードを認証し；第二メモリに基づいて第一メモリを設定し、セキュアモードに入り；セキュアモードにおけるターゲットコードの実行を、プライベートエミュレーション命令を介して制御する。

【図面の簡単な説明】

【０００７】

本発明およびその態様は、添付の図と併せて以下の詳細な説明を読むことでよりよく理解される。図において、要素は必ずしも縮尺通りに描かれているとは限らない。一般的には、複数の図に現れるような要素は、そのような参照記号により示される。図において：

【図１Ａ】図１Ａは、いくつかの態様における、マイクロプロセッサのブロック図である

。
【図１Ｂ】図１Ｂは、いくつかの態様における、組み込みシステムのブロック図である。
【図１Ｃ】図１Ｃは、いくつかの態様における、マイクロプロセッサに接続されたホストのブロック図である。
【図１Ｄ】図１Ｄは、いくつかの態様における、組み込みシステムに接続されたホストのブロック図である。
【図２】図２は、いくつかの態様における、セキュアステートマシンのステート図である。
【図３Ａ】図３Ａは、デジタル署名生成工程の例を示す、フロー図である。
【図３Ｂ】図３Ｂは、デジタル署名照合工程の例を示す、フロー図である。
【図４】図４は、デジタル署名認証を実行する方法である。
【図５】図５は、いくつかの態様における、マイクロプロセッサのブロック図である。
【図６Ａ】図６Ａは、いくつかの態様における、マイクロプロセッサ上のレジスタのフィールドを示すブロック図である。
【図６Ｂ】図６Ｂは、いくつかの態様における、マイクロプロセッサ上のレジスタのフィールドを示すブロック図である。
【図６Ｃ】図６Ｃは、いくつかの態様における、マイクロプロセッサ上のレジスタのフィールドを示すブロック図である。
【図７】図７は、いくつかの態様における、マイクロプロセッサのブロック図である。
【図８】図８は、いくつかの態様における、セキュアモードおけるコード認証を実行するための方法である。
【発明を実施するための形態】
【０００８】
詳細な説明
今日の世界において多くの用途に広く利用される、プログラム可能なプロセッサの分野において、これらプロセッサの供給者の顧客は、プロセッサを特定の応用に利用するためのソフトウェアを頻繁に開発する。そのようなソフトウェア開発者のために、ソフトウェアが期待通りに実行されない場合に、彼らのソフトウェアのデバッグを可能にすることが望まれる。JTAGのような、ハードウェアデバッグツールは、アプリケーションコードの開発やテストを容易にする。
ソフトウェアデバッキングを可能にするための要求に加えて、大きなセキュリティー上の問題が存在する。取扱注意、秘密、および／または顧客またはプロセッサの利用者の財産である情報が、プロセッサのメモリおよび／またはレジスタに常に保存される場合が多い。ソフトウェアデバッグツールの利用中に、そのようなメモリおよびレジスタの内容は、ツールの利用者にアクセス可能である場合が多く、そのことは不正利用の危険性をもたらす。したがって、セキュリティーを保証するためには、プログラム可能なプロセッサの供給者の多くが、例えばデバッキング機能を有効にするプロセッサチップのピンを接続しないなどして、デバッキング機能を無効にすることを試みる。
マイクロプロセッサは、特定の動作モードの間に、ソフトウェアのデバッグ機能およびセキュリティーのバランスを取るように提供される。このバランスは、取扱注意、秘密および／または財産的情報が安全であることを保証する。
【０００９】
マイクロプロセッサ 100
図１Ａは、マイクロプロセッサ100の一態様を示す。マイクロプロセッサ100は、中央演算処理装置（CPU）110、レジスタ120、入力／出力（I/O）ポート130およびメモリ140を含んでもよい。
CPU110は、マイクロプロセッサ100の命令を実行するための論理ユニットである。CPU110により実行可能な命令は、例えば一連の実行可能な命令から成るソフトウェア（すなわち、プログラム、コード）から生じるものでもよい。
メモリ140は、実行可能なコード、パブリックキー情報、および／または任意の種類の

デジタルデータを保存するために利用されてもよい。各メモリ位置は、メモリアドレスと関連してもよい。メモリ140は、ワンタイムプログラマブル（OTP）メモリ、スタティックランダムアクセスメモリ（SRAM）、リードオンリーメモリ（ROM）、ダイナミックランダムアクセスメモリ（DRAM）、または他のメモリ技術またはメモリ技術の組合せを備えてもよい。

【００１０】

いくつかの態様では、メモリ140は、プライベートメモリ150領域およびパブリックメモリ160領域を含む。プライベートメモリ150は、特定の動作状態下でのみアクセス可能であってもよい。

パブリックメモリ160は、ファームウェア170を保存してもよい。ファームウェア170は、ユーザーおよび／またはコード認証を行うための認証ソフトウェアを含んでもよい。いくつかの態様では、ファームウェア170は、認証ソフトウェア命令の変更を防止するために、ROMに記憶される。

レジスタ120は、情報のビットを記憶してもよい。ビットは、マイクロプロセッサ100の動作ステートを示してもよい。レジスタ120は、それぞれが１つ以上のビットを含む、任意の数の個別のレジスタに分けられていてもよい。いくつかの態様では、レジスタ120は、CPU110により実行される次の命令のメモリアドレスを含む、プログラムカウンター（PC）122レジスタを含む。

【００１１】

マイクロプロセッサのI/Oポート130は、情報（例えば、メッセージおよびデジタル署名）の転送のための入出力機能を提供する。各ポートは、ピン、ジャック、有線または無線の受信機、または任意の他のインターフェース技術によって実現されてもよい。I/Oポート130は、デバッグポート134（例えば、インサーキットエミュレータ（ICE）ポート）、リセットポート132および１つ以上の追加のI/Oポート（図示せず）を含んでもよい。デバッグポート134は、マイクロプロセッサ100により実行されるソフトウェアをデバッキングするために利用されてもよい。例えば、マイクロプロセッサ100の動作は、ブレイクポイントの設定、シングルステップ実行、および他のデバッキング工程により、デバッグポート134を通じて観測されてもよい。

【００１２】

いくつかの態様では、デバッグポート134は、マイクロプロセッサ100へのJTAG接続をサポートする。JTAGは、デバイスの入力／出力（I/O）が制御され観測されるバウンダリスキャンアーキテクチャを定義する。バウンダリスキャンに加えて、JTAGエミュレーション能力もまた、ソフトウェア開発においてデバイス内に設計された非常に複雑な機能の制御を支援する。エミュレーション能力は、プロセッサの制御、RUN、STOP、シングルーステップ、および内部レジスタの検査／変更、およびリアルタイムブレークポイントの実行を含む。IEEE標準によりサポートされる“パブリック”のJTAG命令（例えば、バウンダリスキャンおよびバイパスモード）に加えて、“プライベート”のJTAG命令もまたサポートされてもよい。プライベートの命令は、例えば、製造者により特定のマイクロプロセッサのために定義されてもよい。デバッグポート134を通して、利用者はマイクロプロセッサ100にパブリックおよび／またはプライベートのJTAG命令を送信してもよい。JTAGエミュレーションがサポートされてもよい。

【００１３】

リセットポート132は、マイクロプロセッサ100をリセットするための外部トリガーを提供するために利用されてもよい。

いくつかの態様では、マイクロプロセッサ100は、CPU110を介してメモリを呼び出す必要性を除去するためのダイレクトメモリアクセス（DMA）をサポートしてもよい。いくつかの態様では、DMAは、メモリ140の一部のために選択的に無効にしてもよい。メモリ140のどの部分が、DMA有効／無効にされるかは、例えば、レジスタ120の１つにより制御されてもよい。

マイクロプロセッサ100は、図１Ｂに示される組み込みシステム180の一部であってもよ

い。組み込みシステムは、マイクロプロセッサの出力を受信するため、および／またはマイクロプロセッサに入力を提供するために動作可能に接続される追加のハードウェアから構成されてもよい。組み込みシステム180は、デバッグコネクタ181、フラッシュメモリ182、電源制御装置183および水晶発振子184などの典型的な要素を備えたブロック図として示される。これらの要素は、純粋に典型的なものであって、実施の態様において存在しても、また存在しなくてもよい。マイクロプロセッサ100は、組み込みシステム180を形成するための任意の適切な要素と組み合わせて利用されてもよい。

【００１４】

接続191が、図１Ｃに示されるように、１つ以上のI/Oポート130（例えば、デバッグポート134）を介して、ホスト190とマイクロプロセッサ100が通信するために確立されてもよい。適切ないかなるデバイスも、ホスト190としての役割を果たす。例えば、ホスト190は、パーソナルコンピュータ、ラプトップコンピュータ、PDA、またはフラッシュメモリデバイスであってもよい。

接続192は、マイクロプロセッサ100を含む組み込みシステムと、ホスト190の間に、図１Ｄに示されるような任意の適切なインターフェース193を介して確立されてもよい。

接続191および192は、有線および無線技術を含む、任意の適切な技術を利用して実現してもよい。

【００１５】

セキュアステートマシン200

マイクロプロセッサ100は、動作を管理するためのセキュアステートマシン200を実行してもよい。いくつかの態様によるセキュアステートマシン200のステート図が図２に示される。セキュアステートマシン200は、動作モードおよび動作モード間の遷移経路から成ってもよい。各遷移が異なるモード間の関係を定義する一方、各動作モードは、異なるアクセス権およびセキュリティー機能と関連していてもよい。

セキュアステートマシン200は、レジスタ120、メモリ140、または他の任意の適切な方法を介して、マイクロプロセッサ100に実行されてもよい。図２に示される態様例において、セキュアステートマシン200は、オープンモード210,セキュアエントリーモード220,およびセキュアモード230で動作してもよい。

オープンモード210は、プライベートメモリ150へのアクセスが制限されることを除いては、何も制限が掛からないプロセッサのデフォルト動作ステートである。いくつかの態様では、レジスタ120内の特定のレジスタビットへの読み込みおよび／または書き込みアクセスもまた禁止されてもよい。オープンモード210は、マイクロプロセッサ100の起動時およびリセット（経路201）後のデフォルトステートである。いくつかの態様では、デバッキング機能（例えば、JTAGエミュレーション）は、オープンモード210において有効である。

【００１６】

図２に示される態様例において、オープンモード210で動作するセキュアステートマシン200は、セキュアエントリーモード220（遷移202を介して)にのみ遷移してもよい。オープンモード210からセキュアモード230への直接経路はない。

オープンモード210からセキュアエントリーモード220への遷移は、プロセッサの実行の対象がファームウェア170内の認証ソフトウェアに向けられた場合にトリガーされてもよい。いくつかの態様では、プログラムカウンター122を認証ソフトウェアの第一アドレスへ向けることにより、プロセッサの実行をファームウェア170に向けてもよい。いくつかの態様では、ノンマスカブル割り込み（NMI）もまたアクティブであることが要求される。セキュアエントリーモード220への遷移は、例えば、コード実行、ユーザー入力、または他の任意の適切な手段によりトリガーされてもよい。

セキュアエントリーモード220において、ファームウェア170内の認証ソフトウェアは、CPU110により実行されてもよい。認証ソフトウェアは、セキュアステートマシンが遷移204に従ってセキュアモード230に移行するか、または遷移203に従ってオープンモードに戻るかを決定してもよい。いくつかの態様では、認証ソフトウェアは、決定を行うためのセ

キュアエントリーサービスルーチン（SESR）を含んでもよい。

【００１７】

SESRは、ユーザーの認証（例えば、ユーザーがセキュアモードへのアクセスを許可されていることを確認する）、ユーザーコードの認証（例えば、セキュアモードで実行されるコードが、セキュアモードへのアクセスが許可されたユーザーにより提供されていることを確認する）、および／または、他の任意のセキュリティー工程または組合せまたは複数のセキュリティー工程であってもよい。いくつかの態様では、工程350（図３Ｂ）のようなデジタル署名認証プロセスが、メッセージおよびデジタル署名において行われる。署名済みメッセージの認証のための方法400が、図４に関連して後に示される。

セキュアエントリーモード220において、プライベートメモリ150は、アクセス不可能であってもよい。いくつかの態様では、プログラムカウンター122は、それ自身がファームウェア170に割り当てられたアドレス範囲内に滞在していることを確認するためにハードウェアにより監視されてもよい。いくつかの態様では、プロセッサメモリ140の特定の領域へのDMAアクセスは許可されず、JTAGエミュレーションは無効にされる。

【００１８】

認証が失敗した場合、セキュアエントリーモード220からオープンモード210への遷移203が起こってもよい。認証は、例えば、ユーザー認証ができない、ユーザーコード認証ができない、メッセージおよびデジタル署名の組がローカルパブリックキーと合わない、ファームウェア内でエラーが観測された、または、割り込み処理がされなければならない場合に、失敗するであろう。ハードウェア監視により見つかる任意のエラーもまた、認証失敗という結果になるであろう。エラーの例としては、不正なメモリバウンダリ状態（例えば、プログラムカウンター122が、認証コードのアドレス範囲外に向いている場合）、またはファームウェア領域外へのジャンプ（例えば、割り込みを行って）を含んでもよい。

セキュアステートマシン200は、認証が成功する時にだけ、セキュアエントリーモード220からセキュアモード230へ遷移してもよい。認証が成功した場合、SESRは、セキュアモード230に入る前に、遷移204を介して追加のステップを行ってもよい。いくつかの態様では、割り込みが無効にされる。割り込みは、割り込みレベルがSESRを介してNMIから引き下げられることにより再び有効にしても、または、認証が成功するまで待機して、セキュアモード230に入った後に、割り込みを認証コード内で再び有効にしてもよい。

【００１９】

セキュアモード230は、マイクロプロセッサ100のセキュアな動作ステートである。JTAGエミュレーションは、セキュアモードに入った時のデフォルト設定により無効にされてもよい。いくつかの態様では、認証コードは、プライベートメモリ150、パブリックメモリ160、およびレジスタ120を含む、プロセッサのリソースへの無制限アクセスが許可される。いくつかの態様では、セキュアモード230では、シークレットキーのようなセキュアデータが保存されるプライベートメモリ150へのアクセス（読み込みおよび書き込み）を許可する。プライベートメモリ150は、機密、許可された秘密情報、許可されたユーザー、および／またはアクセスしてもよいコードを保存するために利用されてもよい。

セキュアモード230は、例えば、シークレットキーが利用される、任意の暗号文の暗号法の実行を安全に行うために利用されてもよい（例えば、プライベートキーがプライベートメモリ150に保存されてもよい）。

最終コードをデバッキングする（例えば、JTAGエミュレーションを利用して）ための方法800は、図８に関連して後に示される。

セキュアステートマシン200は、セキュアモード230からオープンモード210へ復帰するように遷移205してもよい。いくつかの態様では、セキュアモード230からセキュアエントリーモード220への直接経路が存在しなくてもよい。

【００２０】

認証

セキュアエントリーモード220の間は、認証プロセスは、セキュアモード230へ遷移する前に行われてもよい。いくつかの態様では、デジタル署名認証は、電子メッセージの信頼

性の決定、およびメッセージ署名者の確認のために利用される。例えば、メッセージおよびデジタル署名は、I/Oポートを介してマイクロプロセッサ100に送信され、メモリ（例えば、メモリ140）に保存されてもよい。この形式の認証を利用する場合は、メッセージは、署名者により生成されるデジタル署名に関連してもよい。デジタル署名は、メッセージおよび署名者に特有であり、よって両方が認証される。

デジタル署名は、パブリックキー暗号（すなわち、非対称暗号）技術を利用して生成および照合される。非対称暗号は、２つの異なる、しかし数学的に関連したキー：パブリックキーおよびプライベートキーを利用したアルゴリズムを採用している。プライベートキーは、デジタル署名の生成、またはデータを一見理解不能な形式に変換するために利用される。パブリックキーは、デジタル署名の照合をおこなうか、またはメッセージをその元の形式に戻す。

【００２１】

パブリックキーが利用可能またはデジタル署名の照合を行うすべてのもの（例えば、マイクロプロセッサ100）に分配されるのに対し、プライベートキーは、署名者のみに知られるであろう。キーの組は数学的に関連しているとはいえ、非対称暗号システムが安全に設計され実行される場合は、パブリックキーの知識からプライベートキーを演算によって導き出すことは不可能である。したがって、たとえ多くの人が或る署名者のパブリックキーを知り、それを署名者の署名を照合するために利用することがあっても、彼らは、署名者のプライベートキーを見つけ出し、それをデジタル署名構築のために利用することはできない。

デジタル署名の利用は、通常２つの工程を含み、１つは署名者によって、もう１つはデジタル署名の受信者によって実行される。いくつかの態様では、デジタル署名は、図３Ａに示される工程300に従って生成される。署名されるメッセージ301の境界（bounds）が一度定義されると、ハッシュ関数310が、入力メッセージ301に特有のハッシュ値303を計算する。ハッシュ値303は、メッセージ301の“デジタル指紋”である。一般的に、ハッシュ値303は、メッセージよりずっと小さい標準的な長さであるが、十分に特有なものである。ハッシュ関数310は、例えばSHA-1（secure
hashing algorithm）のような一方向ハッシュ関数でもよい。“一方向ハッシュ関数”と呼ばれることもある、セキュアなハッシュ関数の場合、そのハッシュ値の知識から元のメッセージ301をコンピュータ的に導き出すことは不可能である。よってハッシュ関数は、デジタル署名を生成するためのソフトウェアが、小さくて予測可能な量のデータ上で動作することを可能にしながら、さらに元のメッセージ内容との強固な相関関係の証拠を提供するため、その結果、メッセージがデジタル的に署名されてから、変更がないことの保証を効果的に提供することができる。

【００２２】

SHA-1は、国家安全保障局（NSA）により策定された５つの暗号法のハッシュ関数のうちの１つであり、国立標準技術研究所（NIST）によって、米国連邦情報処理標準として公開されている。

次に、署名生成ソフトウェア320は、プライベートキー302を利用して、ハッシュ値303をデジタル署名305に変換する。プライベートキー302および対応するパブリックキー304（図３Ｂ）は、例えば、楕円曲線暗号（ECC）を利用して生成されてもよい。デジタル署名305は、メッセージ301およびそれを生成するために利用されるプライベートキー302の両方に特有なものである。ECCが、プライベートキーおよびパブリックキーを生成するために利用される場合は、楕円曲線暗号は、プライベートキー302およびハッシュ値303から、デジタル署名305を生成するために利用されてもよい。

デジタル署名305（デジタル的に署名されたメッセージのハッシュ結果）は、メッセージ301に添付され、メッセージ301と共に保存されるか、または送信されてもよい。しかしながら、それはまた、メッセージ301との信頼できる関係性を維持できる限り、別個のデータ要素として送信または保存されてもよい。

【００２３】

デジタル署名照合工程350は、図１Ｂに示される。工程350は、例えば、マイクロプロセッサ100上で実行されてもよい。いくつかの態様では、工程350を実行するためのソフトウェアは、ファームウェア170または任意の適切なメモリに保存される。例えば、ソフトウェアはSESRの一部であってもよい。工程350は、デジタル署名305および所定のパブリックキー304を参照することにより、受信したメッセージ306をチェックし、それによって、デジタル署名305が、受信したメッセージ306のために、参照されたパブリックキー304に対応するプライベートキー302を利用して生成されたかどうかを決定する。

デジタル署名の照合は、デジタル署名を生成するために利用されたハッシュ関数と同一であるハッシュ関数330を用いて、受信したメッセージ306の新しいハッシュ値308を計算することにより達成される。パブリックキー304および新しいハッシュ値308を利用して、照合ソフトウェア340は、デジタル署名305が、パブリックキー304に関連するプライベートキー302を利用して生成されたかどうか、および新たに計算されたハッシュ値308が、デジタル署名生成工程300の間に、デジタル署名305に変換された元のハッシュ値303に一致するかどうかのチェックを行う。パブリックキー304を利用して、デジタル署名305が、元のハッシュ値303に復号されてもよい。

【００２４】

照合ソフトウェア340は、信頼度307を出力する。信頼度307は、受信したメッセージ306が署名者の元のメッセージ301であること、および元のハッシュ値303と計算されたハッシュ値308が一致する場合に、パブリックキーの所有者の対応するプライベートキー302が、本物のソースであることを確認する。認証が成功すると、セキュアモード230への次の遷移を許可するようにしてもよい。

受信したメッセージ306が元のメッセージ301から変更された場合、変更は必ずハッシュ値308に影響を及ぼし、同じハッシュ関数が利用された場合は、異なる結果を作り出す。メッセージおよびデジタル署名はパブリックキーと一致せず、照合は失敗する。これは、オープンモード210への次の遷移につながってもよい。

【００２５】

セキュア方法400

セキュアステートマシン200、デジタル署名生成工程300、およびデジタル署名照合工程350の概略が提供され、セキュリティー工程を実行する方法400が、図４に関して示される。方法400は、デジタル署名認証のために実行されてもよい。方法400は、例えば、許可されたユーザーがセキュアモード230においてマイクロプロセッサ100のコードを実行したい場合に実行されてもよい。

方法400は、マイクロプロセッサ100の外側（“オフチップ”）で任意に実行されてもよい、ステップ402および404を含む。ステップ408、410および412は、デジタル署名認証のために“オンチップ”で実行されてもよいステップに対応する。

ステップ402において、認証されるメッセージ（例えば、コード）の一方向ハッシュは、任意の適切なハッシュ関数を利用して作成される。例えば、ハッシュ関数は、SHA-1（secure hash algorithm）のような一方向ハッシュ関数であってもよい。ステップ402は、ホスト190（図１Ｃおよび１Ｄ）により任意選択的に実行されてもよい。認証されるメッセージは、実行可能なコードであってもよい。適切なハッシュ関数は、ハッシュ値を出力してもよい。

【００２６】

ステップ404において、ハッシュ値は、プライベートキーと共に暗号化されてもよく、それによってファイルに署名し、デジタル署名の生成を完了する。ハッシュ値は、任意の適切な方法により暗号化されてもよい。例えば、楕円曲線暗号（ECC）アルゴリズムを利用してもよい。

ステップ406において、メッセージおよびデジタル署名は、マイクロプロセッサ100によりアクセス可能なメモリに転送される。例えば、メッセージおよびデジタル署名は、プロセッサメモリ140に保存されてもよい。いくつかの態様では、マイクロプロセッサ100に転送される前に、メッセージおよびデジタル署名は、転送を促進するために、外部ホスト19

0（図１Ｃおよび１Ｄ）またはオンボードメモリデバイス（例えば、図１Ｂのフラッシュメモリ182）に保存されてもよい。いくつかの態様では、ステップ406の完了は、マイクロプロセッサ100のオープンモード210からセキュアエントリーモード220へのスイッチを引き起こすようにしてもよい。

【００２７】

ステップ408において、ステップ406で転送されたメッセージは、任意の適切なハッシュ関数を利用してハッシュされてもよい。ハッシュ関数は、プロセッサメモリ140上に存在していてもよい。いくつかの態様では、ハッシュ関数は、ファームウェア170の一部である。いくつかの態様では、ハッシュ関数は、ROM上に存在していてもよい。ハッシュ関数は、ステップ402で利用されるハッシュ関数と機能的に同じであってもよい。

ステップ410において、デジタル署名は、パブリックキーおよび復号アルゴリズムを利用して復号化されてもよい。復号化されたデジタル署名は、ステップ402において生成されたハッシュ値であってもよい。パブリックキーは、パブリックメモリ150に保存されてもよい。任意の適切な復号アルゴリズムが利用されてもよい。復号アルゴリズムは、ステップ404において利用された暗号アルゴリズムと同じアルゴリズムを基にしてもよい。例えば、楕円曲線暗号が利用されてもよい。

【００２８】

ステップ412において、ステップ408で作成されたハッシュ値、およびステップ410でデジタル署名を復号することにより決定されたハッシュ値が比較されてもよい。復号されたハッシュが、計算されたハッシュと一致する場合に、署名が有効でメッセージが無傷であるとしてもよい。

ステップ414において、一度署名の検証が成功すると、セキュアステートマシン200は、セキュアエントリーモード230に入ってもよい。セキュアモード230に入ると、プライベートメモリ150へのアクセスは選択的に有効にしてもよい。いくつかの態様では、セキュアモード230においてプライベートメモリ150へアクセス可能かどうかの決定は、レジストリフィールドまたは任意の他の適切な指標に基づいて利用可能であってもよい。また、セキュアモード230へ入ると、エミュレーション命令（例えば、プライベートのJTAG命令）の実行は、選択的に無効にしてもよい。いくつかの態様では、セキュアモード230においてエミュレーション命令が実行されるかどうかの決定は、レジストリフィールドまたは任意の他の適切な指標に基づいて行われてもよい。いくつかの態様では、認証されたメッセージは、CPU110において実行可能なコードであってよい。いくつかの態様では、認証されたコードは、セキュアモード230において実行可能であってもよい。

【００２９】

デジタル署名は、オフチップ（例えば、ホストコンピュータ上で）で作成されてもよい。プライベートキーは、マイクロプロセッサ100の外（“オフチップ”）でデジタル署名を作成し、対応するパブリックキーは、マイクロプロセッサ100の中（“オンチップ”）で署名を照合してもよい。プライベートキーは、その所有者にのみ知られていてもよく、マイクロプロセッサ100には保存されないようにしてもよい。パブリックキーは、プライベートキーの所有者からのメッセージを認証するために、誰にでも利用可能であってもよく、また、マイクロプロセッサ100に保存されてもよい。

【００３０】

マイクロプロセッサ 500

図５は、マイクロプロセッサ500の一態様のブロック図である。マイクロプロセッサ500は、マイクロプロセッサ100（図１Ａ）の１つの態様例である。マイクロプロセッサ100の要素と同じ操作記述を共有する要素は、共通の参照番号を共有する。

マイクロプロセッサ500は、中央演算処理装置（CPU）110、レジスタ120、I/Oポート130およびプロセッサメモリ140を備える。

マイクロプロセッサ500のレジスタ120は、PCレジスタ122、システムスイッチレジスタ124、制御レジスタ126、および状態レジスタ128を含む。各レジスタは、それと関係するビットのセットを備えていてもよい。各ビットおよびビットのサブセットは、レジストリフ

ィールドのステートを示していてもよい。プロセッサメモリ140は、ワンタイムプログラマブル（OTP）メモリ510、レベル１（L1）キャッシュ520、およびレベル2（L2）キャッシュ560を含んでもよい。

【００３１】

OTPメモリ510は、一度だけプログラムされてもよい、不揮発性の書込み禁止可能なメモリの配列であってもよい。いくつかの態様では、配列の半分はパブリックメモリ（どのモードでもアクセス可能であってよい、パブリックOTP512）であり、他の半分はプライベートメモリ（セキュアモード230においてのみアクセス可能であってよい、プライベートOTP511）である。マイクロプロセッサ500のプライベートOTP511は、例えば、プライベートメモリ（図１Ａ）の一態様である。

L1キャッシュ520は、L1読み取り専用メモリ（ROM）530、L1データバンクA540、およびL1データバンクB550を含んでもよい。

【００３２】

ファームウェア170は、L1ROM530に保存されてもよい。ファームウェア170は、セキュアエントリーモード220において認証のために利用されるセキュアエントリーサービスルーチン（SESR）アプリケーション・プログラミング・インタフェース（API）171を含んでもよい。ファームウェア170は、SHA-1（secure hash algorithm）172のようなハッシュ関数、および楕円曲線暗号173コードのような非対称暗号コードをさらに含んでもよい。ファームウェア170をROMへの保存することで、ファームウェアコードの悪意のある変更を防止してもよい。

デジタル署名認証が実行される態様において、デジタル署名およびメッセージは、任意の適切なメモリの場所に保存されてもよい。いくつかの態様では、デジタル署名およびメッセージは、格納スペース541および542のL1データバンクA540それぞれに保存されてもよい。メッセージおよび署名もまた、もしくはもう１つの方法として、L2　560または任意の他の適切な場所に保存されてもよい。

システムスイッチレジスタ124、制御レジスタ126、および状態レジスタ128は、図６Ａ、図６Ｂ、および図６Ｃに関連してそれぞれ示される。レジスタ124、126、および128の各フィールドは、2進値を利用してもよい。いくつかの態様では、論理“０”は、“クリア”ステートを示し、一方で、論理“１”は、“セット”ステートを示す。言うまでもなく、任意の適切な論理的な記録、およびステートを保存するための任意の適切な物理的態様が利用されてもよい。

【００３３】

セキュアレジスタ

図６Ａは、システムスイッチレジスタ124“SECURE_S YSSWT”のいくつかの態様に存在する、レジストリフィールドのいくつかを示すブロック図である。システムスイッチレジスタ124は、フィールド641-645“EMUDABL”、“EMUOVR”、“RSTDABL”、“DMAOVR”および“OTPSEN”のそれぞれを含んでもよい。

フィールド621、EMUDABL（“エミュレーション無効”）は、エミュレーションが無効の場合を示す。クリア（例えば、“０”）の場合、EMUDABLは、実行された場合に認識されるエミュレーション命令（例えば、プライベートJTAGエミュレーション命令）を示す。セット（例えば、“１”）の場合、EMUDABLがアサートされ、エミュレーション命令が無視される。オープンモード210に入ると、EMUDABLはクリアされる。セキュアモード230に入ると、EMUDABLは、EMUOVRを基に決定される。

【００３４】

フィールド642、EMUOVR（“エミュレーションオーバーライド”）は、セキュアモードに入ってからエミュレーションが有効または無効にされることを示す。クリアの場合は、EMUDABLは、セキュアモードに入るとセットされる。セットの場合は、EMUDABLは、セキュアモードに入るとクリアされる。EMUOVRは、セキュアモードでのみセットされてもよい。

フィールド643、RSTDABL（“リセット無効”）は、どのように外部リセットが行われるかを決定してもよい。クリアの場合は、リセットは標準的に行われる。セットの場合は、

リセットは、NMIイベントを保存するNMIピンにリダイレクトされる。RSTDABLは、セキュアモードに入るとセットされ、オープンモードに入るとクリアされる。

【００３５】

フィールド644、DMAOVR（“ダイレクトメモリアクセスオーバーライド”）は、DMAが有効（例えば、DMAOVRがセットの場合）、または無効（例えば、DMAOVRがクリアの場合）であるかを示す。システムスイッチレジスタ124内の他のフィールド（図示せず）は、制限されたメモリ領域を指定してもよい。いくつかの態様では、DMAはオープンモードに入ると（例えば、DMAOVRがクリア）無効にされてもよい。

フィールド645、OTPSEN（“シークレット有効”）は、プライベートメモリ150が読み込み可能でプログラム可能（例えば、OTPSENがセットの場合）、またはアクセス不可能（例えば、OTPSENがクリアの場合）であるかを決定する。セキュアモードでのみ書き込み可能である。

【００３６】

図６Ｂは、制御レジスタ126、“SECURE_CONTROL”のいくつかの態様に存在するいくつかのレジストリフィールドを示すブロック図である。制御レジスタ126は、フィールド661-644、それぞれ“SECURE0”、“SECURE1”、“SECURE2”、および“SECURE3”を含んでもよい。

フィールド661、SECURE0は、書き込み専用ビットである。SECURE0は、セキュアエントリーモードにおいてのみセットされてもよい。SECURE0がクリアになった場合、フィールド661-664（例えば、制御レジスタ126内のすべてのSECUREビット）はクリアされ、オープンモードに入る。最初にSECURE0がセットされた場合、SECURE1がセットされる。次のSECURE0のセットは、SECURE2のセットをもたらす。

【００３７】

フィールド662-664、それぞれSECURE1、SECURE2、およびSECURE3は、読み込み専用ビットである。SECURE3をセットすると、セキュアモード230に入る。

図６Ｃは、状態レジスタ128、“SECURE_STATUS”のいくつかの態様に存在する、いくつかのレジストリフィールドを示すブロック図である。状態レジスタ128は、フィールド681-684、それぞれ“SECMODE”、“NMI”、“AFVALID”および“AFEXIT”を含んでもよい。

フィールド681、SECMODE（“セキュアモード制御ステート”）は、セキュアステートマシン200の現在のステートを示す２ビットの読み込み専用フィールドである。いくつかの態様では、“00”は、セキュアステートマシンがオープンモードであることを示し、一方で、“01”および“10”は、それぞれセキュアエントリーおよびセキュアモードを示す（“11”は、リザーブドステートである）。

【００３８】

フィールド682、NMIは、ノンマスカブル割り込みの検出を反映する読み込み専用ビットである。

フィールド683、AFVALID（“認証ファームウェア有効”）は、認証の状態を反映する、読み込み専用ビットである。クリアされた場合、認証が正確に始まらないか、または割り込みが入る。セットされた場合、認証が有効になり、正確に進行し、割り込みはない。

フィールド684、AFEXIT（“認証ファームウェア終了”）は、認証ファームウェアからの不適切な終了が行われた場合にセットされる。例えば、セキュアステートマシン200が、AFEXITのセットを検出すると、セキュアエントリーモードを終了しオープンモードへ戻ってもよい。

【００３９】

マイクロプロセッサ 700

図７は、マイクロプロセッサ700のブロック図である。マイクロプロセッサ700は、マイクロプロセッサ100（図１Ａ）の態様例である。マイクロプロセッサ700は、ハードウェア、ソフトウェア、またはその両方の任意の適切な組み合わせであってもよい要素を含む。マイクロプロセッサ100の要素と同じ操作記述を共有する要素は、共通の参照番号を共有しているであろう。いくつかの態様では、マイクロプロセッサ700の要素は、マイクロプ

ロセッサ200および／またはマイクロプロセッサ500からの要素の任意の適切な組み合わせを利用して実行してもよい。

マイクロプロセッサ700は、CPU110、I/Oポート130、動作モジュール705、実行モジュール710、メッセージストア715、署名ストア720、アクセスモジュール725、ハッシュモジュール730、復号モジュール735、プライベートメモリ745、およびエミュレーション制御モジュール750を備えてもよい。いくつかの態様では、アクセスモジュール725、ハッシュモジュール730、および復号モジュール735は、ファームウェア740の一部である。

【００４０】

動作モジュール705は、現在の動作モードのアクセス権限およびセキュリティー機能を実施する。いくつかの態様では、動作モードは、オープンモード210、セキュアエントリーモード220、およびセキュアモード230を含んでもよい。いくつかの態様では、動作モジュールは、セキュアステートマシン200（図２）に従って、動作モード間を遷移する。いくつかの態様では、動作モジュールは、メモリ（例えば、メモリ140）および／またはレジスタ（例えば、レジスタ120）を利用して実現してもよい。例えば、制御レジスタ126は、セキュアモード230へ入ることを指定するために利用してもよく、一方で、状態レジスタ128内のSECMODEフィールド681が、現在の動作モードを指定するために利用してもよい。

【００４１】

実行モジュール710は、CPU110により実行されるプログラムを指定してもよい。実行モジュールは、例えば、CPU110により実行される次の命令のメモリアドレスを指定してもよい。いくつかの態様では、実行モジュール710は、特定のメモリアドレスを示すように指示されない限り、各連続的な実行と共にインクリメントする。いくつかの態様では、実行モジュール710は、プログラムカウンターレジスタ122として実現される。

メッセージストア715および署名ストア720は、認証されるメッセージおよびメッセージのデジタル署名それぞれを保存してもよい。メッセージストアおよび署名ストアは、メモリ140を介して実行してもよい。いくつかの態様では、メッセージストアおよび署名ストアは、L1　520および／またはL2　560（図５）の一部である。

【００４２】

セキュアアクセスモジュール725は、セキュアエントリーサービスルーチン（SESR）を実行してもよい。セキュアアクセスモジュールは、メッセージおよびデジタル署名の組の信頼性を評価してもよい。

セキュアアクセスモジュール725は、ハッシュモジュール730および／または復号モジュール735を呼び出してもよい。呼び出しは、実行されるモジュールのアドレスと共に実行モジュール710を更新することにより行われてもよい。

ハッシュモジュール730は、メッセージをハッシュし、ハッシュ値を出力してもよい。ハッシュモジュール730は、SHA-1アルゴリズムまたは任意の適切なハッシュアルゴリズムを実装してもよい。

復号モジュール735は、認証されたメッセージ送信者のパブリックキーを利用して、メッセージのハッシュ値とデジタル署名を認証してもよい。いくつかの態様では、復号モジュール735は、楕円曲線暗号を利用して、パブリックキーと、メッセージ／デジタル署名の組を認証してもよい。

【００４３】

実行モジュール710が、CPU110により実行されるプログラムとしてセキュアアクセスモジュール725を特定した場合、動作モジュール705は、セキュアエントリーモード220にスイッチしてもよい。いくつかの態様では、動作モジュール705は、セキュアエントリーモード220にスイッチする前にオープンモード210で動作する。

セキュアアクセスモジュール725が、メッセージ／デジタル署名の組の信頼性を照合する場合、動作モジュール705はセキュアモード230に入ってもよい。

セキュアモードにおいて、プライベートメモリ領域745は、読み込みおよび／または書き込みアクセス可能であってもよい。プライベートメモリ領域745のアクセスのしやすさ

は、例えば、セキュアモード230においてOTPSENフィールド645により決定されてもよい。いくつかの態様では、プライベートメモリ領域745のための読み込みおよび書き込み命令は、オープンモード210およびセキュアエントリーモード220において中断／拒否されてもよい。いくつかの態様において、プライベートメモリ領域745は、ワンタイムプログラマブル（OTP）メモリ配列510（図５）の少なくとも一部であってもよい。

【００４４】

いくつかの態様では、エミュレーション制御モジュール750は、例えば、デバッグポート134により受信されるエミュレーション命令が、実行されるかどうかを決定する。エミュレーション制御モジュール750は、システムスイッチレジスタ124内のEMUDABLフィールド641およびEMUOVRフィールド642を介して実行されてもよい。いくつかの態様では、エミュレーション命令は、JTAGエミュレーション命令であってもよい。

セキュアアクセスモジュールが、メッセージおよびデジタル署名の組が本物でないことを決定、または任意の人物（例えば、割り込みを実行するために）のために認証プロセスを中止する場合、動作モジュール705は、セキュアエントリーモード220からオープンモード210へスイッチしてもよい。いくつかの態様では、セキュアアクセスモジュール725は、認証プロセスが失敗した場合、AFVALID、状態レジスタ128のフィールド683をクリアする。クリアされたAFVALIDは、動作モジュール705に対して、オープンモード210へ戻ることを示してもよい。

セキュアアクセスモジュール725、ハッシュモジュール730および復号モジュール735は、ファームウェア740の一部であってもよい。いくつかの態様では、ファームウェア740は、これらのモジュールと共に改ざんを防止するリードオンリーメモリ（ROM）である。

【００４５】

方法800：最終コードのデバッグのための動作例

利用者は、コードの最終バージョンをセキュアモードでテストすることを望むかもしれない。テストは、実行が利用者によって綿密に観察されるように、エミュレーションが有効になることを要求してもよい。図８に示される方法800は、例えば、セキュアモードでファイナルバージョンのコードのテストをするために実行されてもよい。

ステップ802の最初は、マイクロプロセッサ100は、オープンモード210であると仮定される。いくつかの態様では、エミュレーション（例えば、JTAGエミュレーション）は、セキュアモードへ入ると、デフォルトで無効にされてもよい。エミュレーションがセキュアモードにおいて利用可能であることを保証するために、システムスイッチレジスタ124内のEMUOVRフィールド642がセットされてもよい。

【００４６】

EMUOVRをセットするために、ステップ804、利用者はコードをアップロードして、対応するデジタル署名と認証してもよい。コード（例えば、“JTAG有効コード”）は、EMUOVRをセットするための命令を含む。

一度コードが認証されると、ステップ804において、EMUOVRをセットするコードが実行される。

EMUOVRのセットが行われると、マイクロプロセッサ100は、ステップ808においてオープンモードに戻る。利用者は、対応するデジタル署名にしたがってデバッグされる最終コードをアップロードしてもよい。

ステップ810において、マイクロプロセッサは、セキュアエントリーモードに入り、最終コードおよび対応するデジタル署名を認証する。

【００４７】

ステップ812において、マイクロプロセッサは、セキュアモードに入る。EMUOVRが前もってセットされたため、EMUDABLフィールド641はクリアされる。認証された最終コードは、すぐにセキュアモードで実行されてもよい。利用者は、コードがその最終の形式で実行されることを観察し制御するために、エミュレーション（例えば、JTAGエミュレーション）を利用してもよい。

一方で、最終コードがEMUDABLをクリアするための命令に追加されてもよいことが知ら

れており、したがって、セキュアモードへ特別に入ることを排除し、これが、実際には、最終コードを無効にする。方法800は、セキュアモードにおいて、利用者が実際の最終コードをデバッグすることを可能にする。

【００４８】

さらなる態様

発明の少なくとも１つの実施態様が示されたが、様々な修正、変更、および改良は、当業者により容易になされ得るであろう。

いくつかの態様では、メッセージ（例えば、図３Ａのメッセージ301）は、任意の適切な暗号アルゴリズムを利用して、それ自体が暗号化されてもよい。いくつかの態様では、メッセージの暗号化およびデジタル署名の利用は、プライバシーおよび信頼性の両方を確実にしてもよい。共通鍵暗号（symmetric-key algorithm）が暗号化のために利用されてもよい。暗号化のために利用されてもよい暗号化規格の例は、新暗号規格（ＡＥＳ）、データ暗号化標準（ＤＥＳ）を含む。いくつかの態様では、暗号化されたメッセージは、セキュアモード230（図２）およびプライベートメモリ150（図１Ａ）へのアクセスを提供しながら最初に認証される。プライベートメモリは、復号化のために必要な共通鍵を保存してもよい。

【００４９】

いくつかの態様では、多重パブリックキーがマイクロプロセッサ100（例えば、複数の利用者がセキュアモードにおいて認証コードの実行を許可された場合）に保存されてもよい。マイクロプロセッサ100は、メッセージが認証するまで、または各パブリックキーの試みが失敗するまで、各パブリックキーとの認証プロセスを実行してもよい。いくつかの態様では、メッセージ／署名の組は、どのパブリックキーを利用するかを示してもよい。

マイクロプロセッサ100は、システムオンチップ、コンピュータオンチップ、マイクロコントローラ、および同類のものとして具現化されてもよい。いくつかの態様では、マイクロプロセッサ100は、アナログデバイセズのBlackfin processor（登録商標）である。

マイクロプロセッサ100は、任意のハードウェアおよび／またはソフトウェアデバッグツールと互換性があってもよい。デバッグおよび／またはエミュレーション命令は、デバッグポート134を介して受信してもよい。マイクロプロセッサ100は、IEEE 1149.1 J-TAG標準と互換性があってもよい。いくつかの態様では、JTAG命令は、デバッグポート134を介して受信される。

いくつかの態様では、プライベートメモリ領域のサイズおよび／または場所は、選択可能である。

SECURE_SYSSWTレジスタは、アドレスマップ0xFFC04320のメモリを備える32ビットレジスタであってもよい。表１は、いくつかの態様にしたがった、レジスタ内の各ビットの機能の概要を提供する。

【００５０】

【表１－１】

表1.

ビット位置	ビット名	ビット詳細
		リセット＝0x0000 セキュアエントリーモード＝0x000704d9 セキュアモード＝0x000704db
0	EMUDABL	エミュレーション無効 セキュアエントリーモードでは、EMUDABLの設定は、EMUOVRの前の状態に基づく。オープンモードに再び入ると、EMUDABLはクリアされる。このビットは、常に読み込みアクセス可能である。このビットは、セキュアモードにおいてのみ書き込みアクセス可能である。 0－非公開JTAGエミュレーション指示が、認識および実行される。セキュアモードの間に、このビットが一度クリアされると、それは、セキュアエントリーモードではセットされない。この条件は、これがクリアされるリセットまでは、状態を維持する。この特徴は、セキュリティーデバックにおいて利用される。 1－非公開JTAGエミュレーション指示は、無視される。バイパスのような標準のエミュレーション命令は、許可される。
1	RSTDABL	リセット無効 このビットは、セキュアエントリーモードでは達成されない。このビットは、セキュアモードに入るとセットされる。オープンモードに再び入るとRSTDABLはクリアされる。このビットは常に読み込みアクセス可能である。このビットは、セキュアモードにおいてのみ書き込みアクセス可能である。 0－通常に外部リセットが生成され実行される。 1－外部リセットは、NMIピンにリダイレクトされる。これは、メモリクリーン動作の回避を回避する。
4:2	L11DABL	L1指示メモリ無効 セキュアエントリーモードに入ると、L11DABLは0x6にセットされる。オープンモードに再び入ると、L11DABLはクリアされる。これらのビットは、常に読み込みアクセス可能である。これらのビットは、セキュアモードにおいてのみ書き込みアクセス可能である。

【００５１】

【表１－２】

ビット位置	ビット名	ビット詳細
		DMAアクセスが、制限されたメモリ領域に対して実行される場合、DMAアクセスエラーが起こり、その結果、DMA_ERR割り込みおよびDMA_RUNのクリアが起こる。 000－すべてのDMAアクセスがL1指示領域に対して許可される。 001－1KBのメモリ（0xFFA00000 - 0xFFA003FF）が、制限されたノンコアアクセス（non core access）を備える。 010－2KBのメモリ（0xFFA00000 - 0xFFA007FF）が、制限されたノンコアアクセスを備える。 011－4KBのメモリ（0xFFA00000 - 0xFFA00FFF）が、制限されたノンコアアクセスを備える。 100－8KBのメモリ（0xFFA00000 - 0xFFA01FFF）が、制限されたノンコアアクセスを備える。 101－16KBのメモリ（0xFFA00000 - 0xFFA03FFF）が、制限されたノンコアアクセスを備える。 110－32KBのメモリ（0xFFA00000 - 0xFFA07FFF）が、制限されたノンコアアクセスを備える。これはセキュアエントリーモードに入る時の初期設定である。 111－リザーブ
7:5	L1DADABL	L1データバンクAメモリ無効 セキュアエントリーモードに入ると、L1DADABLは0x6にセットされる。オープンモードに再び入ると、L1DADABLはクリアされる。これらのビットは、オープン、セキュアエントリー、およびセキュアモードにおいて、読み込みアクセス可能である。これらのビットは、セキュアモードにおいてのみ書き込みアクセス可能である。DMAアクセスが、制限されたメモリ領域に対して実行される場合、DMAアクセスエラーが起こり、その結果、DMA_ERR割り込みおよびDMA_RUNがクリアされる。 000－すべてのDMAアクセスが、L1データバンクA領域に対して許可される。 001－1KBのメモリ（0xFF800000 - 0xFF8003FF）が、制限されたノンコアアクセスを備える。 010－2KBのメモリ（0xFF800000 - 0xFF8007FF）が、制限されたノンコアアクセスを備える。

【００５２】

【表１－３】

ビット位置	ビット名	ビット詳細
		011－4KBのメモリ（0xFF800000 - 0xFF800FFF）が、制限されたノンコアアクセスを備える。 100－8KBのメモリ（0xFF800000 - 0xFF801FFF）が、制限されたノンコアアクセスを備える。 101－16KBのメモリ（0xFF800000 - 0xFF803FFF）が、制限されたノンコアアクセスを備える。 110－32KBのメモリ（0xFF800000 - 0xFF807FFF）が、制限されたDMAアクセスを備える。これはセキュアエントリーモードに入る時の初期設定である。 111－リザーブ

【００５３】

【表１－４】

ビット位置	ビット名	ビット詳細
10:8	L1DBDABL	L1データバンクBメモリ無効 セキュアエントリーモードに入ると、L1DBDABLは0x4にセットされ、L1データバンクBの8KBに、制限されたノンコアアクセスを与える。オープンモードに再び入ると、L1DBDABLはクリアされる。これらのビットは、オープン、セキュアエントリー、およびセキュアモードにおいて、読み込みアクセス可能である。これらのビットは、セキュアモードにおいてのみ書き込みアクセス可能である。DMAアクセスが、制限されたメモリ領域に対して実行される場合、DMAアクセスエラーが起こり、その結果、DMA_ERR割り込みおよびDMA_RUNがクリアされる。 000－すべてのDMAアクセスが、L1データバンクB領域に対して許可される。これはセキュアエントリーモードに入る時の初期設定である。 001－1KBのメモリ（0xFF900000 - 0xFF9003FF）が、制限されたノンコアアクセスを備える。 010－2KBのメモリ（0xFF900000 - 0xFF9007FF）が、制限されたノンコアアクセスを備える。 011－4KBのメモリ（0xFF900000 - 0xFF900FFF）が、制限されたノンコアアクセスを備える。 100－8KBのメモリ（0xFF900000 - 0xFF901FFF）が、制限されたノンコアアクセスを備える。これはセキュアエントリーモードに入る時の初期設定である。 101－16KBのメモリ（0xFF900000 - 0xFF903FFF）が、制限されたノンコアアクセスを備える。 110－32KBのメモリ（0xFF900000 - 0xFF907FFF）が、制限されたDMAアクセスを備える。 111－リザーブ
11	DMA0OVR	DMA0メモリアクセスオーバーライド セキュアエントリーモードまたはセキュアモードへ入ることは、このビットに影響を与えない。オープンモードに再び入ると、DMA0OVRはクリアされる。このビットは、オープン、セキュアエントリー、およびセキュアモードにおいて、読み込みアクセス可能である。このビットは、セキュアエントリーモードおよびセキュアモードの両方において書き込みアクセス可能である。 DMA0のL1指示、L1データおよびL2メモリ領域へのアクセスを制御する。

【００５４】

【表１－５】

ビット位置	ビット名	ビット詳細
		クリアされる場合、アクセス制限は、このレジスタ内のメモリ無効設定に基づく。 0－DMA0アクセスは、メモリ無効設定に基づき制限される。 1－無制限のDMA0アクセスは、すべてのメモリ領域に対して許可される。
12	DMA1OVR	DMA1メモリアクセスオーバーライド セキュアエントリーモードまたはセキュアモードへ入っても、このビットに影響を与えない。オープンモードに再び入ると、DMA0OVRはクリアされる。このビットは、オープン、セキュアエントリー、およびセキュアモードにおいて、読み込みアクセス可能である。このビットは、セキュアエントリーモードおよびセキュアモードの両方において書き込みアクセス可能である。 DMA1のL1指示、L1データおよびL2メモリ領域へのアクセスを制御する。クリアされる場合、アクセス制限は、このレジスタ内のメモリ無効設定に基づく。 0－DMA1アクセスは、メモリ無効設定に基づいて制限される。 1－無制限のDMA1アクセスは、すべてのメモリ領域に対して許可される。
13	RESERVED	リザーブビット このリザーブビットは、読み込みアクセスにおいて、常に″０″値に戻る。このビットにどのような値を書き込んでも、影響はない。
14	EMUOVR	エミュレーションオーバーライド このビットは、常に読み込みアクセス可能である。このビットは、セキュアモードにおいてのみ″１″が書き込まれる。このビットは、オープンモード、セキュアエントリーモードおよびセキュアモードにおいてクリアすることができる。セキュアエントリーモードでEMUDABLの値を制御する。 0－セキュアエントリーモードにおいて、EMUDABLビットはセットされる。 1－セキュアエントリーモードにおいて、EMUDABL ビットはクリアされる。このビットは、EMUDABL（bit-0）に″０″が書き込まれ、同時にこのビット（bit-14）に、″１″が書き込まれる場合にのみセットされる。

【００５５】

【表１－６】

ビット位置	ビット名	ビット詳細
15	OTPSEN	OTPシークレット有効 このビットは、すべてのモードにおいて読み込み可能であるが、セキュアモードにおいてのみ書き込みアクセス可能である。 0－OTP領域の読み込みおよびプログラミングアクセスは制限される。アクセスは、アクセスエラー（FERROR）をもたらす。 1－OTP領域の読み込みおよびプログラミングアクセスは許可される。アクセスのための対応するプログラム保護ビットがセットされた場合、このビットの設定に関らず、プログラムアクセスは保護される。
18:16	L2DABL	L2 メモリ無効 セキュアエントリーモードに入ると、L2DABLは0x7にセットされる。オープンモードに再び入ると、L2DABLはクリアされる。これらのビットは、オープンモード、セキュアエントリーモードおよびセキュアモードにおいて、読み込みアクセス可能である。これらのビットは、セキュアモードにおいてのみ書き込みアクセス可能である。DMAアクセスが、制限されたメモリ領域に対して実行される場合、DMAアクセスエラーが起こり、その結果、DMA_ERR割り込みおよびDMA_RUNのクリアが起こる。 000－すべてのDMAアクセスがL2に対して許可される。 001－1KBのメモリ（0xFEB00000 - 0xFEB003FF）が、制限されたノンコアアクセス（non core access）を備える。 010－2KBのメモリ（0xFEB00000 - 0xFEB007FF）が、制限されたノンコアアクセスを備える。 011－4KBのメモリ（0xFEB00000 - 0xFEB00FFF）が、制限されたノンコアアクセスを備える。 100－8KBのメモリ（0xFEB00000 - 0xFEB01FFF）が、制限されたノンコアアクセスを備える。 101－16KBのメモリ（0xFEB00000 - 0xFEB03FFF）が、制限されたノンコアアクセスを備える。 110－32KBのメモリ（0xFEB00000 - 0xFEB07FFF）が、制限されたノンコアアクセスを備える。 111－64KBのメモリ（0xFEB00000 - 0xFEB0FFFF）が、制限されたDMAアクセスを備える。

【表１－７】

ビット位置	ビット名	ビット詳細
		これはセキュアエントリーモードに入る時の初期設定である。

SECURE_CONTROLレジスタは、アドレスマップ0xFFC04324のメモリを備える16ビットであってよい。表２は、いくつかの態様にしたがって、レジスタ内の各ビットの機能の概要を提供する。

【００５６】

【表２－１】

表2.

ビット位置	ビット名	ビット詳細
		リセット＝0x0000
0	SECURE0	SECURE 0 書き込み専用ビット。読み込みは常に"0"を返す。"1"値は、セキュアエントリーモードの場合、SECURE0にのみ書き込むことができる。この制御ビットの目的は、セキュアモードに入るための、"1"値のSECURE0への３つの連続的な書き込みを要求することである。 0－"0"値が書き込まれる場合、このレジスタ内のすべてのSECUREビットはクリアされ、オープンモードに入る。すべてのSYSSWTビットは、EMUOVRを除いてクリアされる。EMUOVRが利用者によりセットされた場合、それはセットを保持する（リセットがアサートされるまで、またはそれに"0"が書き込まれるまで）。 1－最初に"1"値が書き込まれる場合、SECURE1がセットされる。後続の"1"書き込みと共にSECURE2がセットされる。後続の"1"書き込みがSECURE3をセットする。SECURE3がセットされると、セキュアモードに入る。
1	SECURE1	SECURE 1 これは、読み込み専用ビットであり、SECURE0へのデータ値"1"の書き込み成功を示す。 0－SECURE0に、"1"値が書き込まれない。 1－SECURE0に、"1"値が書き込まれる。
2	SECURE2	SECURE 2 これは、読み込み専用ビットであり、SECURE0へのデータ値"1"の書き込みの成功が２回起こったことを示す。 0－SECURE1がセットされる間、SECURE0に"1"値が書き込まれない。

【００５７】

【表２－２】

ビット位置	ビット名	ビット詳細
		1−SECURE0に２回目の"1"値が書き込まれる。
3	SECURE3	SECURE 3 これは、読み込み専用ビットであり、SECURE0へのデータ値"1"の書き込みの成功が３回起こったことを示す。 0−SECURE2がセットされる間、SECURE0に"1"値が書き込まれない。 1−SECURE0に３回目の"1"値が書き込まれる。一部は、セキュアモードであって、SYSSWTレジスタは、認証されたコードで書き込み可能である。

SECURE_STATUSレジスタは、アドレスマップ0xFFC04328のメモリを備える16ビットレジスタであってよい。表３は、いくつかの態様にしたがって、レジスタ内の各ビットの機能の概要を提供する。

【００５８】

【表３－１】

表3.

ビット位置	ビット名	ビット詳細
		リセット＝0x0000
1:0	SECMODE	セキュアモード制御状態 セキュアステートマシンの現在のモードを反映する読み込み専用ビットである。 00−オープンモード 01−セキュアエントリーモード 10−セキュアモード 11−リザーブ
2	NMI	ノンマスカブル割り込み NMIの検出を反映する読み込み専用ビットである

【００５９】

【表３－２】

ビット位置	ビット名	ビット詳細
		0－現在、NMIが検出されていない。 1－現在、NMIが検出されている。
3	AFVALID	認証ファームウェア有効 ハードウェアモニタ論理の状態を反映する読み込み専用ビットである。認証の実行が適切に始まり、割り込み動作があった場合、認証は有効であると見なされる。セキュアエントリーモードおよびセキュアモード動作のためには、有効な認証が要求される。 0－認証が適切に始まらない、または割り込みが起こった。 1－認証が有効で、適切に進行し、割り込みがない。
4	AFEXIT	認証ファームウェア終了 状態ビットをクリアするために1を書き込む。認証は適切に始まったが、完了前に不適切な終了があった場合、このビットはセットされる。これは、セキュアエントリーモードを終了しオープンモードに戻る時にのみ起こる。 0－認証ファームウェアを実行している間に、不適切な終了がなかった。 1－認証ファームウェアからの不適切な終了があった。
7:5	SECSTAT	セキュア状態 認証が失敗した場合に、ハンドラー（handler）に状態を戻す、読み込み／書き込みビットである。

当業者にとって直ちに想到される、変更、修正、および改良は、本発明の範囲内のものである。したがって、先述の詳細は例に過ぎず、制限するものではない。本発明は、以下のクレームおよびそれと同等のものに定義されているものによってのみ限定される。

【図１Ａ】

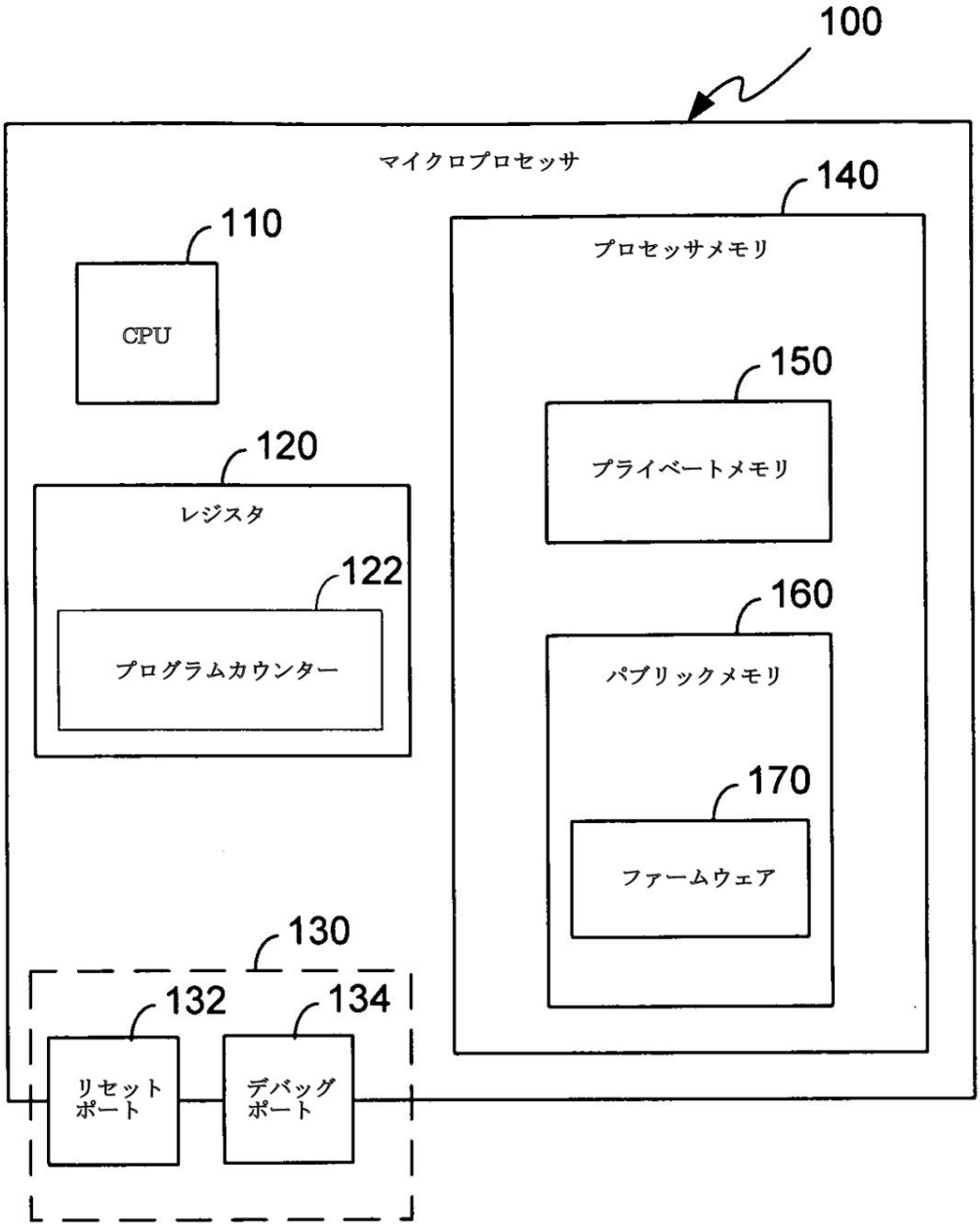

100
マイクロプロセッサ
140
プロセッサメモリ
110
CPU
150
プライベートメモリ
120
レジスタ
122
プログラムカウンター
160
パブリックメモリ
170
ファームウェア
130
132
134
リセット
ポート
デバッグ
ポート

図１Ａ

【図１Ｂ】

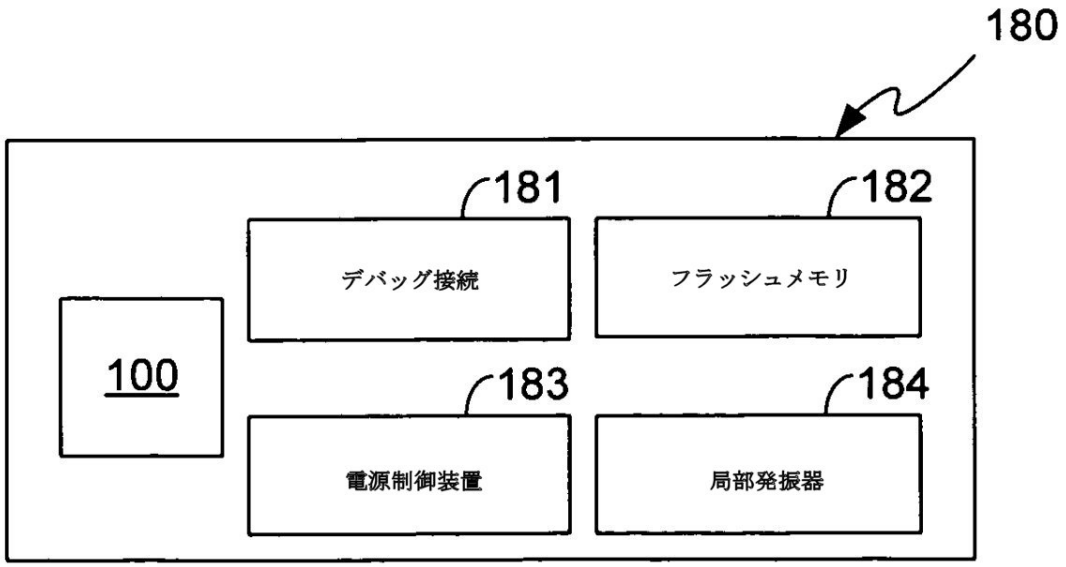


図１Ｂ

【図１Ｃ】

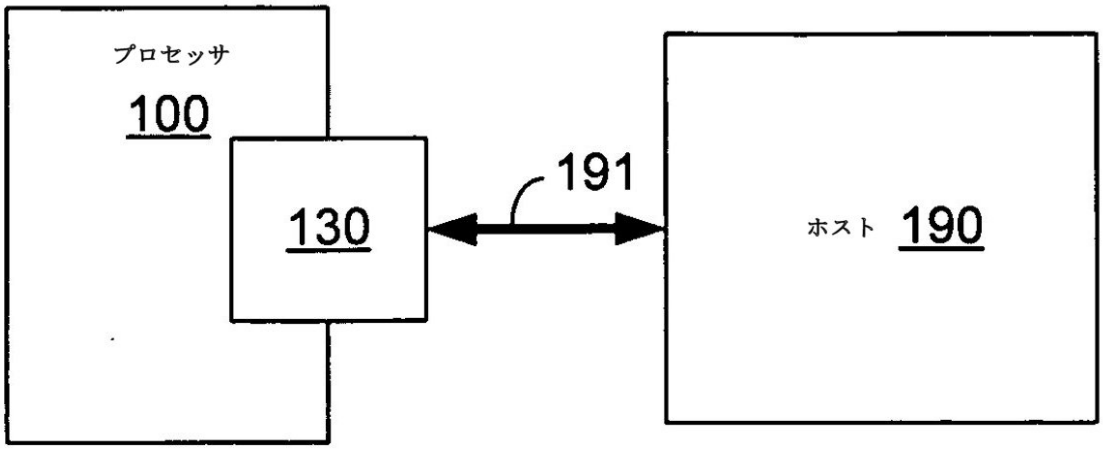


図１Ｃ

【図１Ｄ】

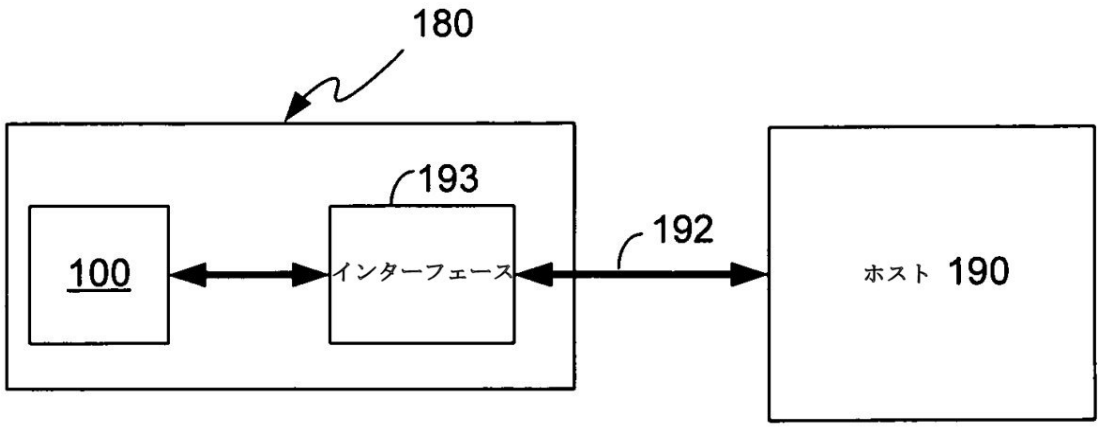


図１Ｄ

【図２】

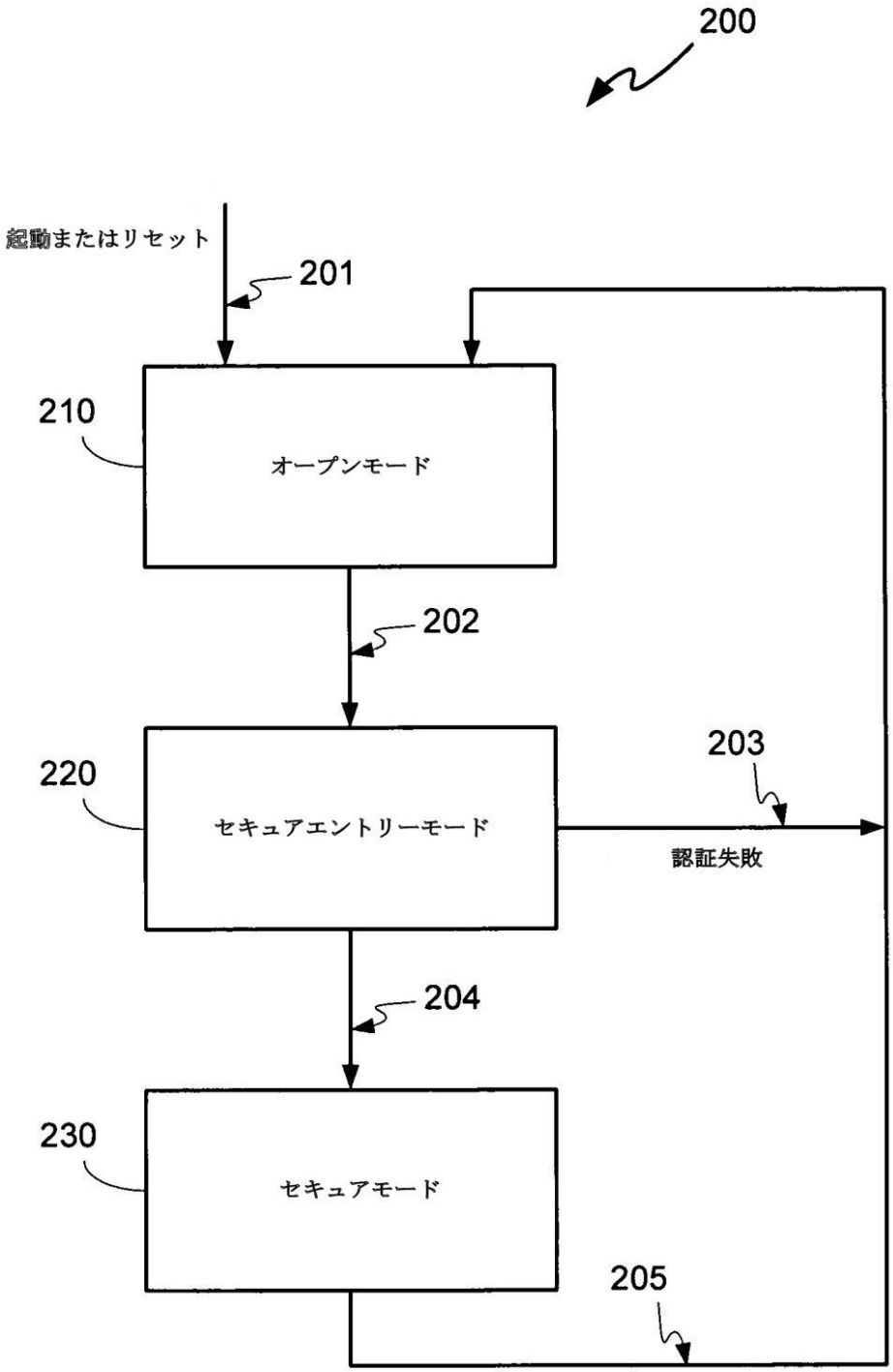

200
起動またはリセット
201
210
オープンモード
202
220
セキュアエントリーモード
203
認証失敗
204
230
セキュアモード
205

図２

【図３Ａ】

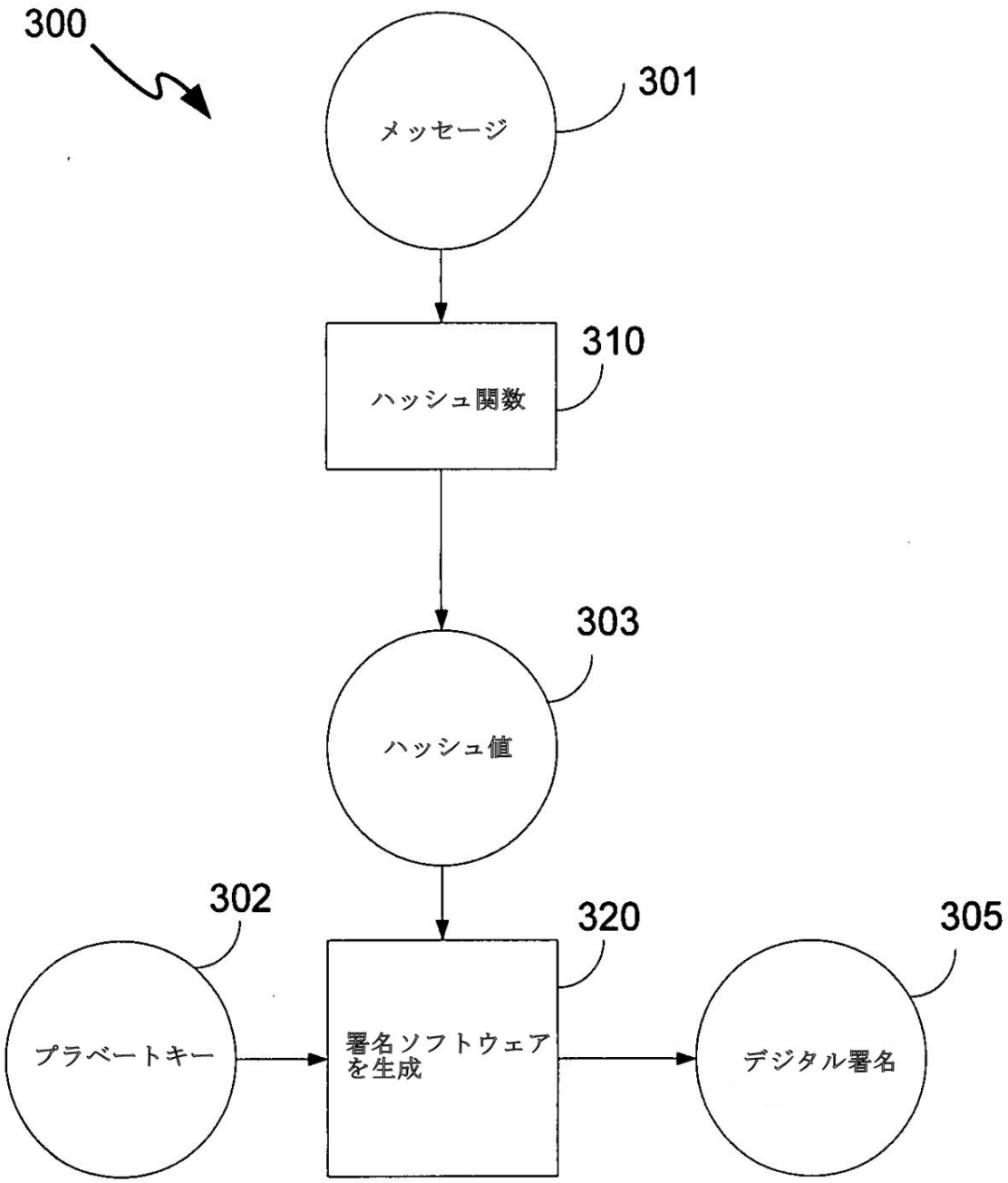

300
メッセージ
301
ハッシュ関数
310
ハッシュ値
303
プラベートキー
302
署名ソフトウェアを生成
320
デジタル署名
305

図３Ａ

【図３Ｂ】

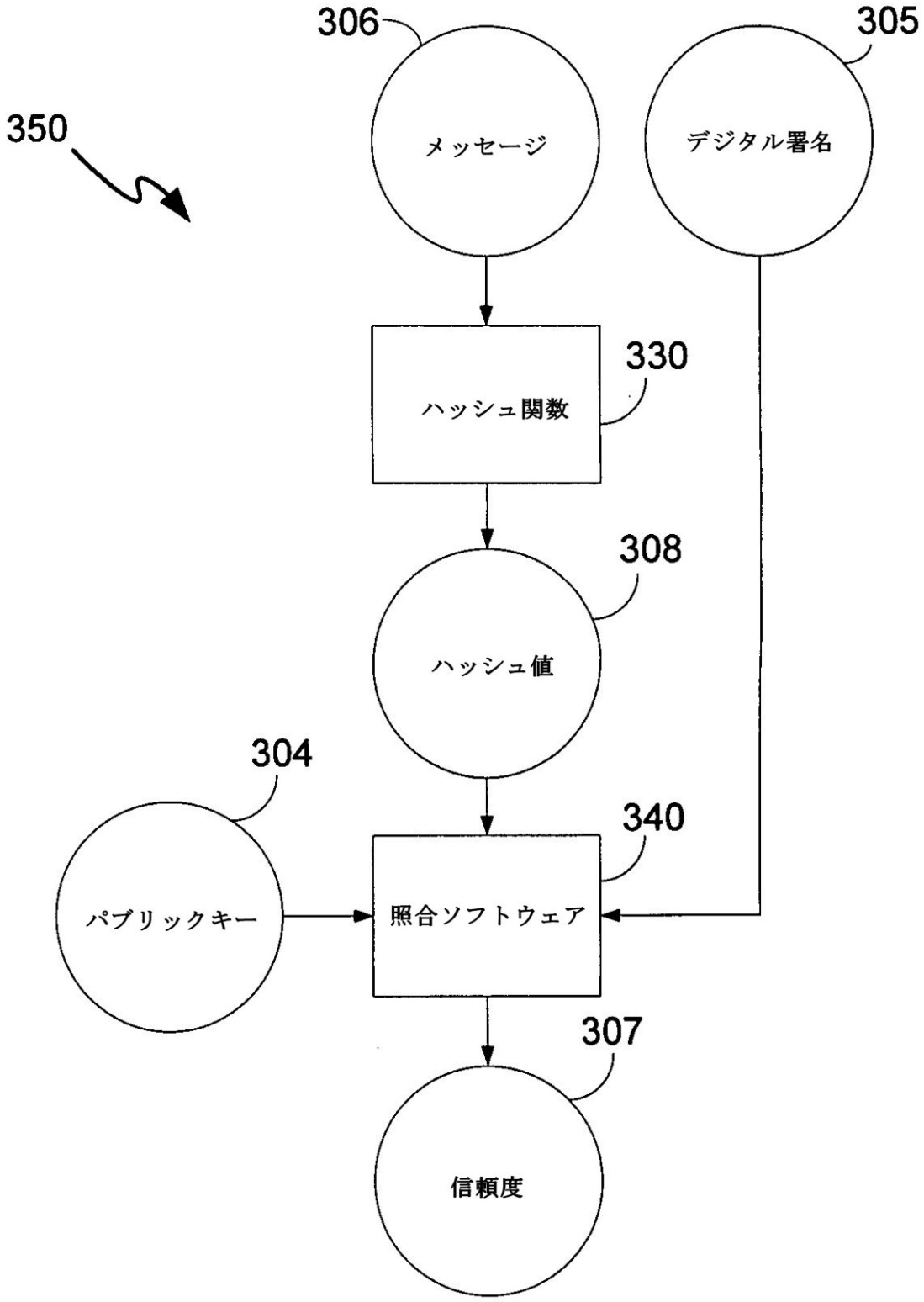

306
305
350
メッセージ
デジタル署名
330
ハッシュ関数
308
ハッシュ値
304
340
パブリックキー
照合ソフトウェア
307
信頼度

図３Ｂ

【図４】

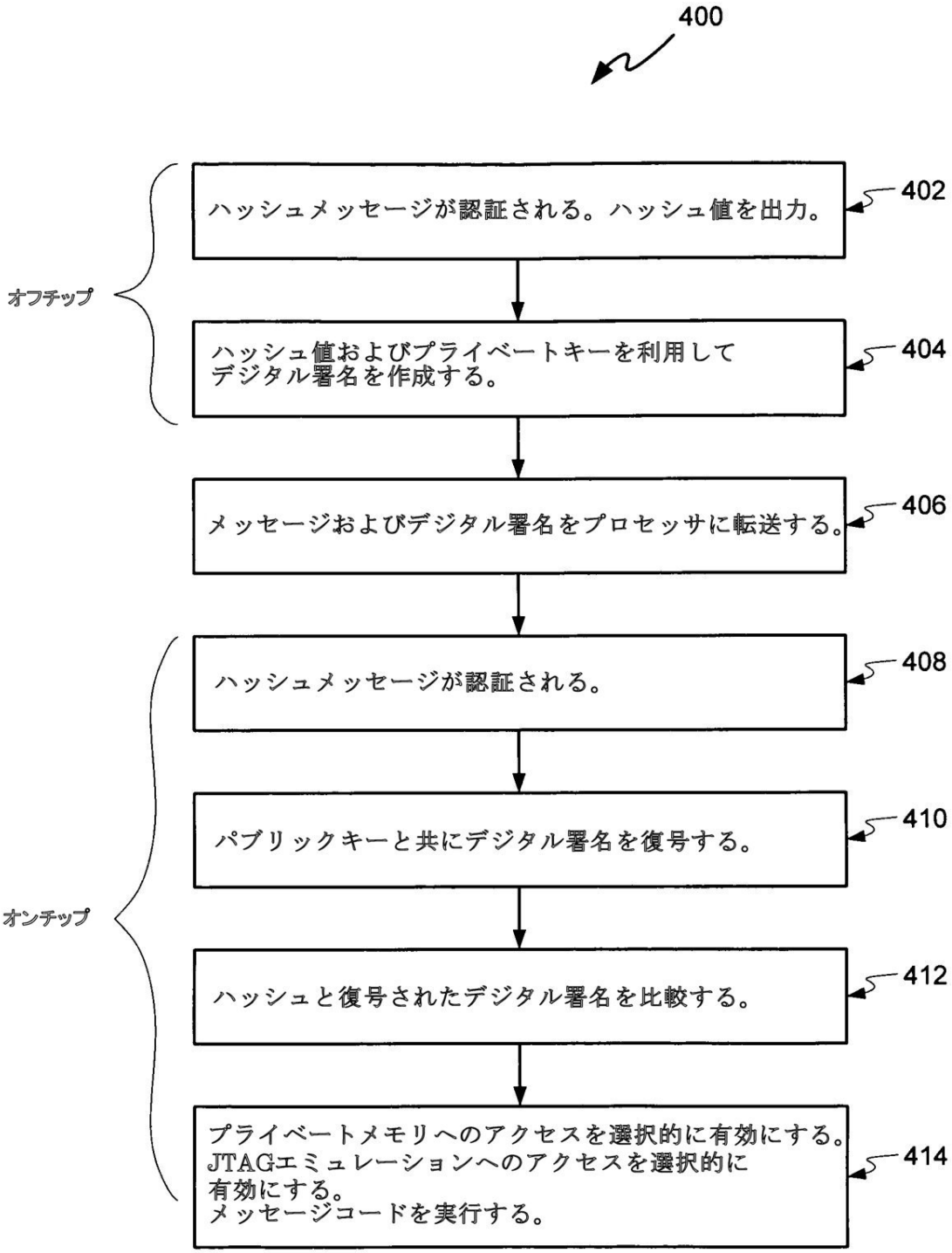

400
ハッシュメッセージが認証される。ハッシュ値を出力。
402
オフチップ
ハッシュ値およびプライベートキーを利用して
デジタル署名を作成する。
404
メッセージおよびデジタル署名をプロセッサに転送する。
406
ハッシュメッセージが認証される。
408
パブリックキーと共にデジタル署名を復号する。
410
オンチップ
ハッシュと復号されたデジタル署名を比較する。
412
プライベートメモリへのアクセスを選択的に有効にする。
JTAGエミュレーションへのアクセスを選択的に
有効にする。
メッセージコードを実行する。
414

図４

【図５】

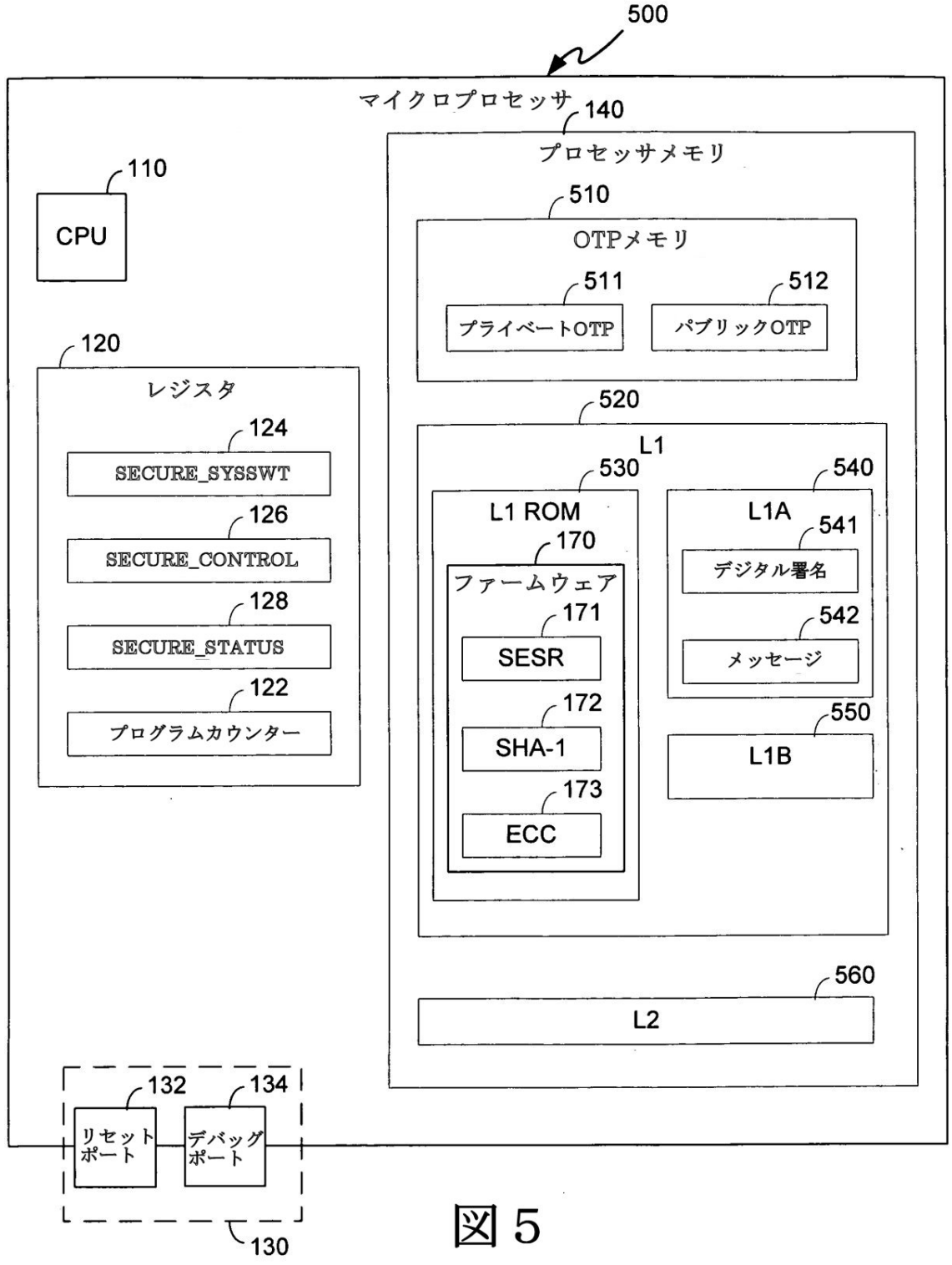

500
マイクロプロセッサ
140
プロセッサメモリ
110
CPU
510
OTPメモリ
511
プライベートOTP
512
パブリックOTP
120
レジスタ
520
L1
124
SECURE_SYSSWT
530
L1 ROM
540
L1A
126
SECURE_CONTROL
170
ファームウェア
541
デジタル署名
128
SECURE_STATUS
171
SESR
542
メッセージ
122
プログラムカウンター
172
SHA-1
550
L1B
173
ECC
560
L2
132
134
リセットポート
デバッグポート
130

図５

【図６Ａ】

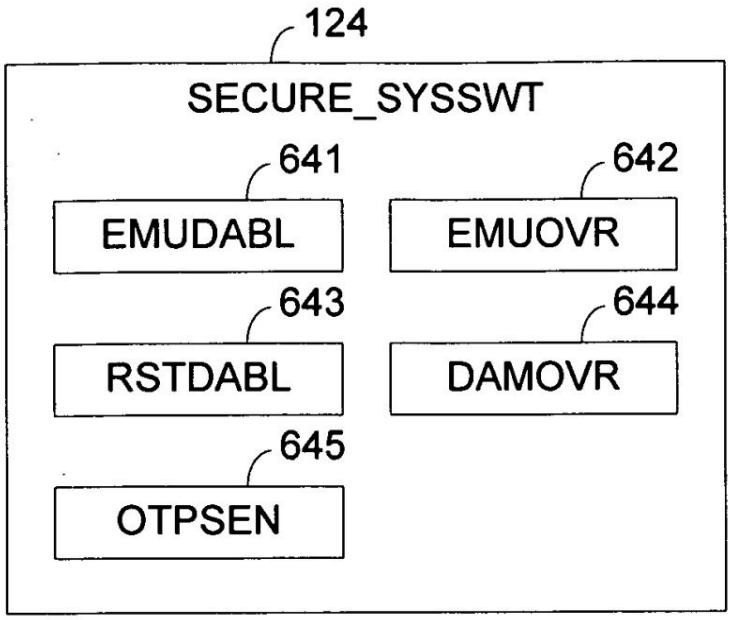
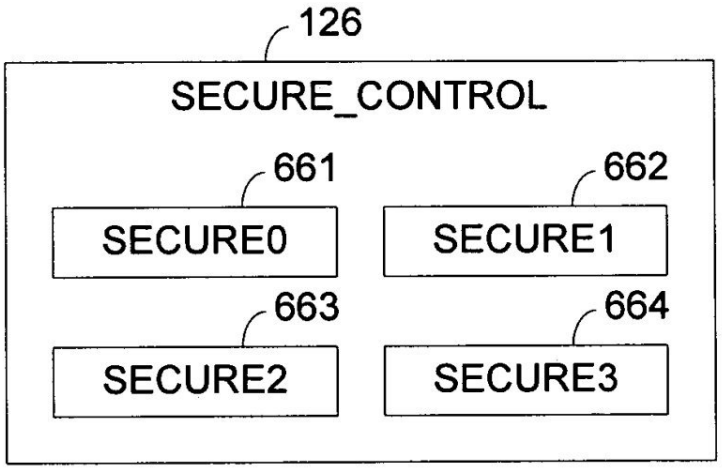


図６Ａ

【図６Ｂ】

図６Ｂ



【図６Ｃ】

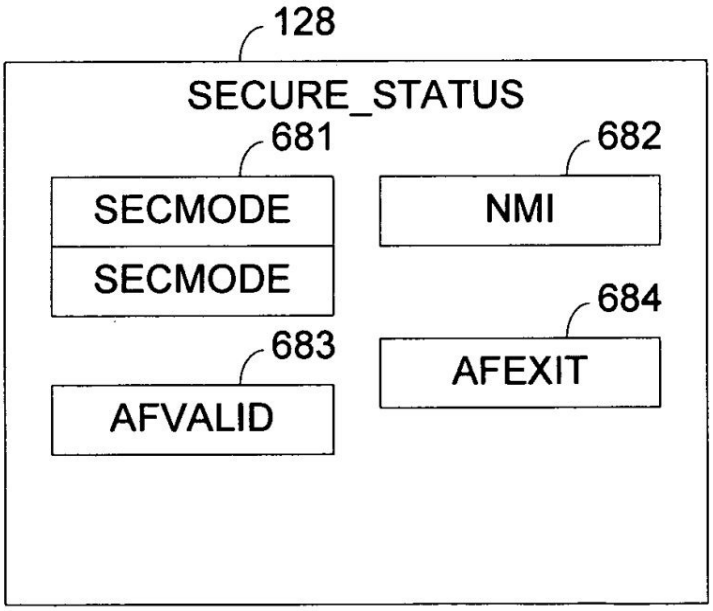

128
SECURE_STATUS
681
682
SECMODE
SECMODE
NMI
684
683
AFEXIT
AFVALID

図６Ｃ

【図７】

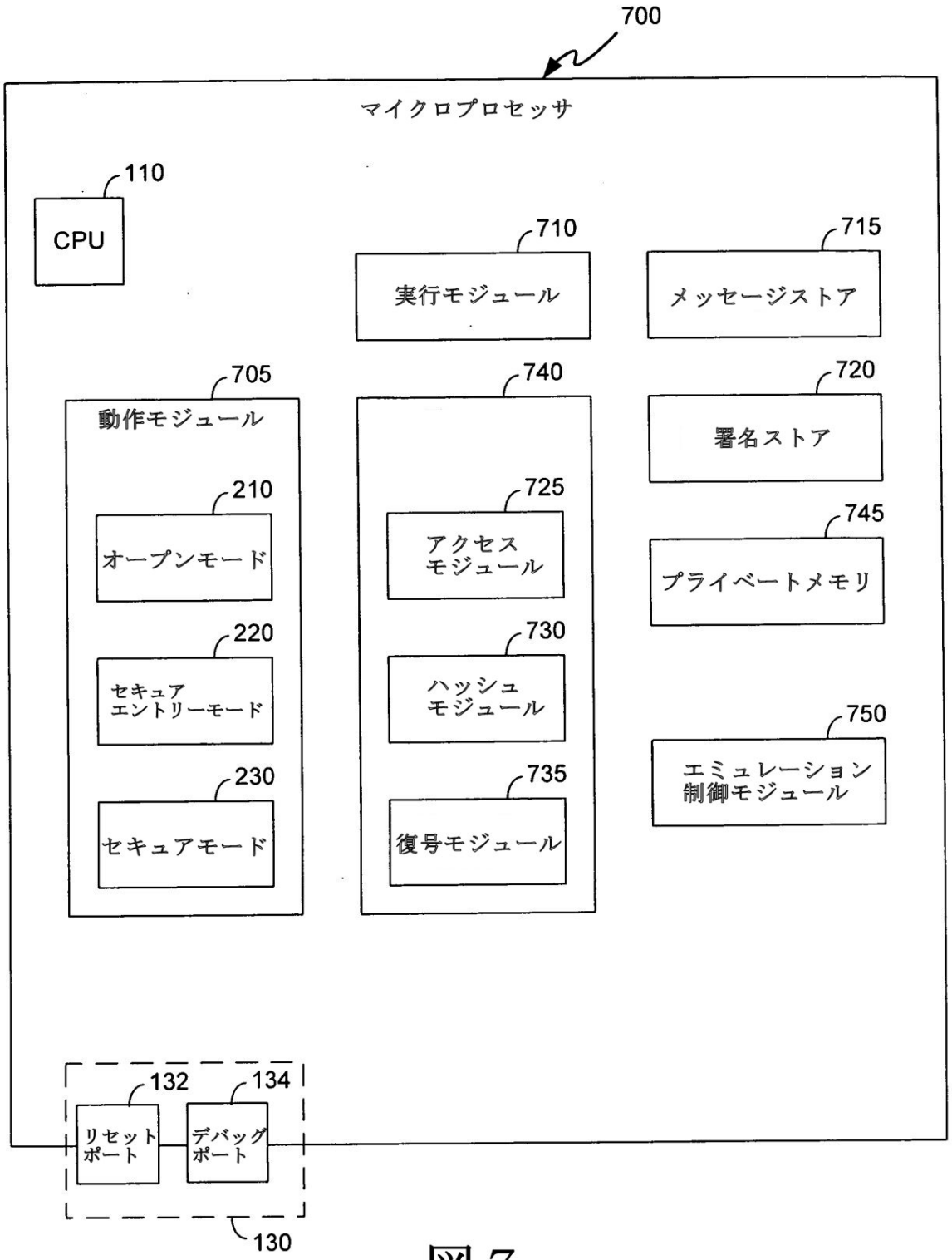

700
マイクロプロセッサ
110
CPU
710
実行モジュール
715
メッセージストア
705
動作モジュール
740
720
署名ストア
210
オープンモード
725
アクセス
モジュール
745
プライベートメモリ
220
セキュア
エントリーモード
730
ハッシュ
モジュール
750
エミュレーション
制御モジュール
230
セキュアモード
735
復号モジュール
132
134
リセット
ポート
デバッグ
ポート
130

図７

【図８】

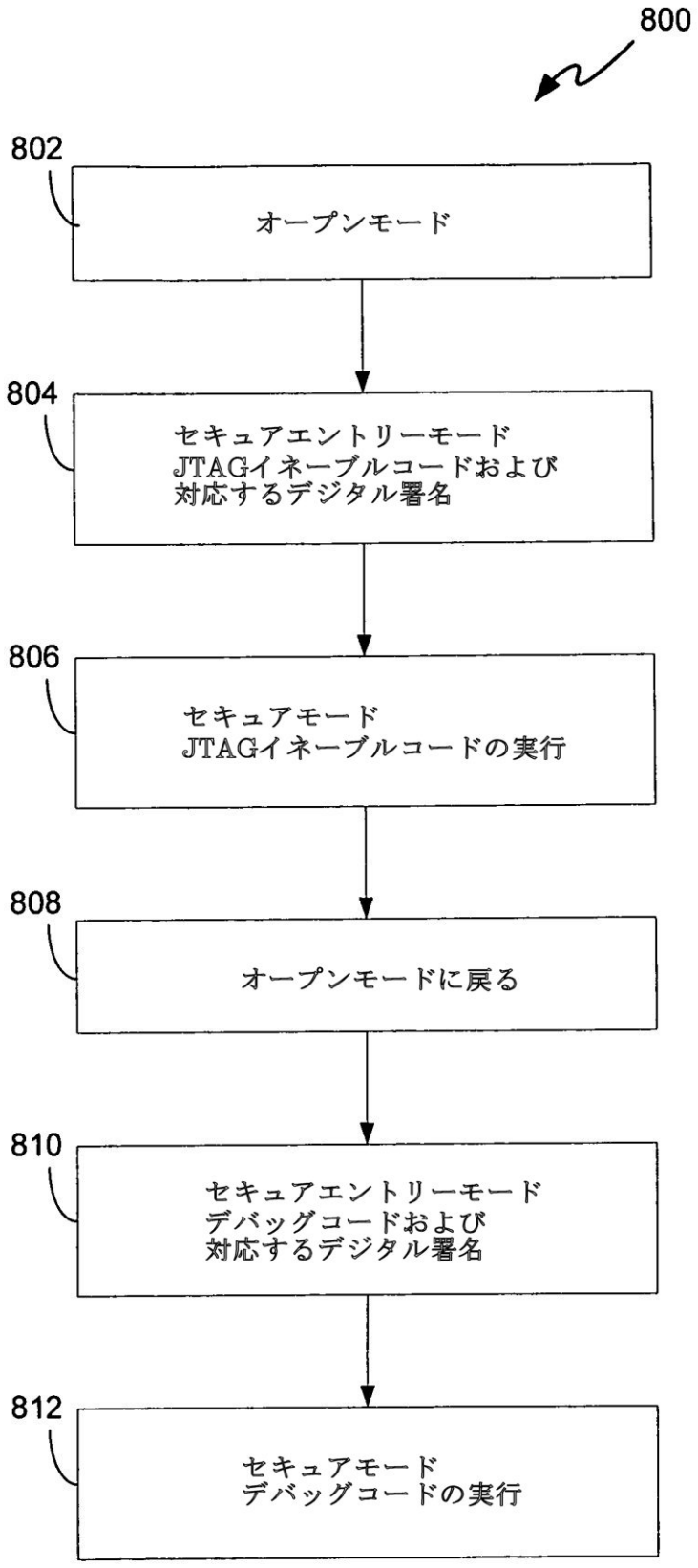

800
802
オープンモード
804
セキュアエントリーモード
JTAGイネーブルコードおよび
対応するデジタル署名
806
セキュアモード
JTAGイネーブルコードの実行
808
オープンモードに戻る
810
セキュアエントリーモード
デバッグコードおよび
対応するデジタル署名
812
セキュアモード
デバッグコードの実行

図８

フロントページの続き

(72)発明者　ビーダーウォルフ，スコット，ディー.
アメリカ合衆国　テキサス州　７８７３３、オースティン、シルバー　パイン　コーブ　９２０２

審査官　脇岡　剛

(56)参考文献　特開２００２－３５８１３７（ＪＰ，Ａ）
米国特許出願公開第２００４／０１５３６７２（ＵＳ，Ａ１）
国際公開第２００５／０２０２８０（WO，Ａ１）
特開２００３－１８６６９３（ＪＰ，Ａ）
特開平０３－００６７５８（ＪＰ，Ａ）
特開２００１－０８３８７４（ＪＰ，Ａ）
米国特許出願公開第２００６／０１３０１３０（ＵＳ，Ａ１）

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｇ０６Ｆ　２１／７４
Ｇ０６Ｆ　１５／７８
Ｇ０６Ｆ　２１／６２