(19)日本国特許庁(JP)　(12)公表特許公報(A)　(11)特許出願公表番号
特表2005-501313
(P2005-501313A)
(43)公表日　平成17年1月13日(2005.1.13)

(51)Int.Cl.[7]	F I		テーマコード（参考）
G06F 12/14	G06F 12/14	510C	5B017
G06F 9/30	G06F 9/30	380Z	5B033
G06F 11/22	G06F 11/22	310D	5B048

審査請求 未請求　予備審査請求 有　（全 42 頁）

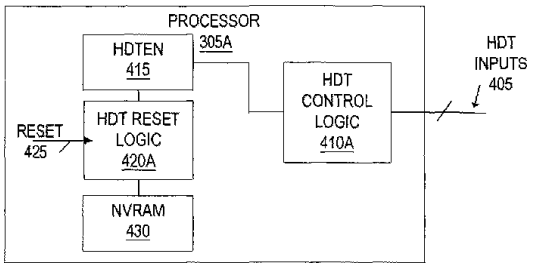
(21)出願番号　特願2002-589946(P2002-589946)
(86)(22)出願日　平成14年4月17日(2002.4.17)
(85)翻訳文提出日　平成15年11月10日(2003.11.10)
(86)国際出願番号　PCT/US2002/011935
(87)国際公開番号　WO2002/093336
(87)国際公開日　平成14年11月21日(2002.11.21)
(31)優先権主張番号　09/852,942
(32)優先日　平成13年5月10日(2001.5.10)
(33)優先権主張国　米国(US)
(31)優先権主張番号　09/852,372
(32)優先日　平成13年5月10日(2001.5.10)
(33)優先権主張国　米国(US)
(31)優先権主張番号　09/853,226
(32)優先日　平成13年5月11日(2001.5.11)
(33)優先権主張国　米国(US)

(71)出願人　591016172
アドバンスト・マイクロ・ディバイシズ・インコーポレイテッド
ＡＤＶＡＮＣＥＤ　ＭＩＣＲＯ　ＤＥＶＩＣＥＳ　ＩＮＣＯＲＰＯＲＡＴＥＤ
アメリカ合衆国、９４０８８－３４５３　カリフォルニア州、サニィベイル、ピィ・オゥ・ボックス・３４５３、ワン・エイ・エム・ディ・プレイス、メイル・ストップ・６８（番地なし）
(74)代理人　100099324
弁理士　鈴木　正剛
(74)代理人　100111615
弁理士　佐野　良太

最終頁に続く

(54)【発明の名称】パーソナルコンピュータシステムにおいて裏口アクセス機構を閉鎖するための機構

(57)【要約】
裏口アクセス機構を閉鎖するための方法、デバイスおよびシステム。１以上のハードウェアデバッグテスト（HDT）イネーブルビットを格納するように構成された第１のレジスタと、複数のHDT入力信号を受信すべく接続された第１の制御ロジックと、第１のレジスタに接続された第２の制御ロジックとが、プロセッサに含まれる。第１の制御ロジックは第１のレジスタにアクセスすべく接続されている。第２の制御ロジックはプロセッサのリセットに応答して１以上のデフォルト値を第１のレジスタに格納するように構成されている。もうひとつのプロセッサには、複数のマイクロコード入力を受信すべく接続された第１の制御ロジックと、第１の制御ロジックに接続された第１のレジスタと、第１のレジスタに接続された第２の制御ロジックとが含まれる。第１のレジスタは、１以上のマイクロコードローダーイネーブルビットを格納するように構成されている。第２の制御ロジックは、プロセッサのリセットに応答して１以上のデフォルト値を第１のレジスタに格納するように構成されている。



【特許請求の範囲】
【請求項１】
１以上のハードウェアデバッグテスト（HDT）イネーブルビットを格納するように構成された第１のレジスタと、
複数のHDT入力信号を受信すべく接続された第１の制御ロジックであって、第１のレジスタにアクセスすべく接続された第１の制御ロジックと、
第１のレジスタに接続され、プロセッサのリセットに応答して１以上のデフォルト値を第１のレジスタに格納するように構成された第２の制御ロジックと、を含むプロセッサ。
【請求項２】
第１の制御ロジックがさらに、HDTモードに入る要求を受信するように構成され、第１の制御ロジックがさらに、HDTモードに入る要求に応答して、第１のレジスタに格納された１以上のHDTイネーブルビットの選択されたエントリを読み出すように構成され、第１の制御ロジックがさらに、１以上のHDTイネーブルビットの選択されたエントリに基づいて、HDTモードに入る要求を受け入れるまたは拒否するように構成されている、請求項１に記載のプロセッサ。
【請求項３】
１以上のHDTイネーブルビットの１以上のデフォルト値を格納するように構成された１以上の不揮発性メモリセルをさらに含み、第２の制御ロジックがさらに、プロセッサのリセットに応答して、１以上のHDTイネーブルビットの１以上のデフォルト値を１以上の不揮発性メモリセルから読み出し、かつ、１以上のHDTイネーブルビットの１以上のデフォルト値を第１のレジスタに書き込むべく接続されている、請求項１に記載のプロセッサ。
【請求項４】
第２の制御ロジックがさらに、プロセッサのリセットに応答して、１以上のHDTイネーブルビットの１以上のデフォルト値を示す信号を受信し、かつ、１以上のHDTイネーブルビットの１以上のデフォルト値を第１のレジスタに書き込みすべく接続されている、請求項１に記載のプロセッサ。
【請求項５】
１以上のマイクロコードローダーイネーブルビットを格納するように構成された第３のレジスタと、
複数のマイクロコード入力を受信すべく接続された第３の制御ロジックであって、第３のレジスタにアクセスすべく接続されている第３の制御ロジックと、
第３のレジスタに接続され、プロセッサのリセットに応答して１以上のデフォルト値を第３のレジスタに格納するように構成された第４の制御ロジックと、をさらに含む、請求項１に記載のプロセッサ。
【請求項６】
第３の制御ロジックがさらに、マイクロコード修正要求を受信するように構成され、第３の制御ロジックがさらに、マイクロコード修正要求に応答して、第３のレジスタに格納された１以上のマイクロコードローダーイネーブルビットの選択されたエントリを読み出すように構成され、第３の制御ロジックがさらに、１以上のマイクロコードローダーイネーブルビットの選択されたエントリに基づいて、マイクロコード修正要求を受け入れるまたは拒否するように構成されている、請求項５に記載のプロセッサ。
【請求項７】
第１の制御ロジックに接続された第２のレジスタをさらに含み、第２のレジスタが１以上のHDTイネーブルロックビットを格納するように構成されている、請求項１に記載のプロセッサ。
【請求項８】
第１の制御ロジックがさらに、HDTモードのステータスの修正要求を受信するように構成され、第１の制御ロジックがさらに、HDTモードのステータスの修正要求に応答して、第２のレジスタに格納された１以上のHDTイネーブルロックビットの選択されたエントリを読み出すように構成され、第１の制御ロジックがさらに、１以上のHDTイネーブルロック

ビットの選択されたエントリに基づいて、HDTモード修正要求を受け入れるまたは拒否するように構成されている、請求項７に記載のプロセッサ。
【請求項９】
複数のマイクロコード入力を受信すべく接続された第１の制御ロジックと、
第１の制御ロジックに接続され、１以上のマイクロコードローダーイネーブルビットを格納するように構成された第１のレジスタと、
第１のレジスタに接続され、プロセッサのリセットに応答して１以上のデフォルト値を第１のレジスタに格納するように構成された第２の制御ロジックと、を含むプロセッサ。
【請求項１０】
第１の制御ロジックがさらに、マイクロコード修正要求を受信するように構成され、第１の制御ロジックがさらに、マイクロコード修正要求に応答して、第１のレジスタに格納された１以上のマイクロコードローダーイネーブルビットの選択されたエントリを読み出すように構成され、第１の制御ロジックがさらに、１以上のマイクロコードローダーイネーブルビットの選択されたエントリに基づいて、マイクロコード修正要求を受け入れるまたは拒否するように構成されている、請求項９に記載のプロセッサ。
【請求項１１】
１以上のマイクロコードローダーイネーブルビットの１以上のデフォルト値を格納するように構成された１以上の不揮発性メモリセルをさらに含み、第２の制御ロジックがさらに、プロセッサのリセットに応答して、１以上のマイクロコードローダーイネーブルビットの１以上のデフォルト値を１以上の不揮発性メモリセルから読み出し、かつ、１以上のマイクロコードローダーイネーブルビットの１以上のデフォルト値をマイクロコードローダーのレジスタに書き込むべく接続されている、請求項９に記載のプロセッサ。
【請求項１２】
第２の制御ロジックがさらに、プロセッサのリセットに応答して、１以上のマイクロコードローダーイネーブルビットの１以上のデフォルト値を示す信号を受信し、かつ、１以上のマイクロコードローダーイネーブルビットの１以上のデフォルト値を第１のレジスタに書き込みすべく接続されている、請求項９に記載のプロセッサ。
【請求項１３】
第１の制御ロジックに接続された第２のレジスタをさらに含み、第２のレジスタが１以上のマイクロコードローダーイネーブルロックビットを格納するように構成されている、請求項９に記載のプロセッサ。
【請求項１４】
第１の制御ロジックがさらに、マイクロコードローダーのロックステータス修正要求を受信するように構成され、第１の制御ロジックがさらに、マイクロコードローダーのロックステータス修正要求に応答して、第２のレジスタに格納された１以上のマイクロコードローダーイネーブルロックビットの選択されたエントリを読み出すように構成され、第１の制御ロジックがさらに、１以上のマイクロコードローダーイネーブルロックビットの選択されたエントリに基づいて、マイクロコードローダーのロックステータス修正要求を受け入れるまたは拒否するように構成されている、請求項１３に記載のプロセッサ。
【請求項１５】
HDTモードの開始要求を受信し、
HDTモードのイネーブルステータスを判断し、
HDTモードのイネーブルステータスがイネーブルに設定されている場合にHDTモードを開始することを含む、HDTモードのイネーブルステータスを判断するための方法。
【請求項１６】
HDTモードのイネーブルステータスを判断することが、１以上のHDTイネーブルビットに対応する１以上のエントリをレジスタから読み出すことを含む、請求項１５に記載の方法。
【請求項１７】
マイクロコード修正要求を受信し、
マイクロコードローダーのイネーブルステータスを判断し、

マイクロコードローダーのイネーブルステータスがイネーブルに設定されている場合にマイクロコードを修正することを含む、マイクロコードを修正するための方法。
【請求項１８】
マイクロコードローダーのイネーブルステータスを判断することが、１以上のマイクロコードローダーイネーブルビットに対応する１以上のエントリをレジスタから読み出すことを含む、請求項１７に記載の方法。
【請求項１９】
HDTモードのステータスの変更要求を受信し、
HDTモードのイネーブルロックステータスを判断し、
HDTモードのイネーブルロックステータスがアンロックに設定されている場合にHDTモードのステータスを修正することを含む、HDTモードのステータスの変更方法。
【請求項２０】
HDTモードのイネーブルロックステータスを判断することが、１以上のHDTイネーブルロックビットに対応する１以上のエントリをレジスタから読み出すことを含む、請求項１９に記載の方法。
【請求項２１】
HDTモードのステータスを修正することが、１以上のHDTイネーブルビットに対応する１以上のエントリをレジスタに書き込むことを含む、請求項１９に記載の方法。
【請求項２２】
マイクロコードローダーのイネーブルステータスの変更要求を受信し、
マイクロコードローダーのイネーブルロックステータスを判断し、
マイクロコードローダーのイネーブルロックステータスがアンロックに設定されている場合にマイクロコードローダーのイネーブルステータスを修正することを含む、マイクロコードローダーのイネーブルステータスの変更方法。
【請求項２３】
マイクロコードローダーのイネーブルロックステータスを判断することが、１以上のマイクロコードローダーイネーブルロックビットに対応する１以上のエントリをレジスタから読み出すことを含む、請求項２２に記載の方法。
【請求項２４】
マイクロコードローダーのイネーブルステータスを修正することが、１以上のマイクロコードローダーイネーブルビットに対応する１以上のエントリをレジスタに書き込むことを含む、請求項２２に記載の方法。
【請求項２５】
１以上の不揮発性メモリセルから１以上のデフォルト値を読み出すことと、プルアップレジスタまたはプルダウンレジスタによって１以上のデフォルト値をストラップされた値として受信することと、からなる群から選択される、１以上のデフォルト値を取得することと、
プロセッサのリセットに応答して、１以上のデフォルト値を１以上のさまざまなエントリとして１以上のレジスタに書き込むことと、を含み、１以上のさまざまなエントリは、
１以上のHDTイネーブルビットと、
１以上のHDTイネーブルロックビットと、
マイクロコードローダーの１以上のイネーブルビットと、
マイクロコードローダーの１以上のイネーブルロックビットと、からなる群から選択される、プロセッサを動作させる方法。
【発明の詳細な説明】
【関連出願の表示】
【０００１】
本件出願は、発明者デール・イー・ギューリック（Dale E. Gulick）およびジェフリー・エス・ストロンギン（Geoffrey S. Strongin）、発明の名称「Secure Execution Box and Method（セキュア実行ボックスおよび方法）」である、２００１年５月１０日に

出願の係属中の米国特許出願第０９／８５２，３７２号の一部継続出願である。また、本件出願は、発明者ジェフリー・エス・ストロンギンおよびデール・イー・ギューリック、２００１年５月１０日に出願の発明の名称「Computer System Architecture for Enhanced Security and Manageability（セキュリティを高め、かつ管理の容易性を高めるためのコンピュータシステムアーキテクチャ）」である、係属中の米国特許出願第０９／８５２，９４２号の一部継続出願でもある。

【技術分野】

【０００２】

本発明は、広義にはコンピュータシステムに関し、特に、パーソナルコンピュータシステムを不正な裏口アクセスから保護するための方法および装置に関する。

【背景技術】

【０００３】

代表的なコンピュータシステム１００を図１に示す。このコンピュータシステム１００は、プロセッサ１０２と、ノースブリッジ１０４と、メモリ１０６と、AGP（Advanced Graphics Port）メモリ１０８と、PCI（Peripheral Component Interconnect）バス１１０と、サウスブリッジ１１２と、バッテリと、ATアタッチメント（ATA）インタフェース１１４[より一般的にはIDE（Integrated Drive Electronics）インタフェースとして知られる]と、ユニバーサルシリアルバス（USB）インタフェース１１６と、Low Pin Count（LPC）バス１１８と、入出力コントローラチップ（Super I／O™）１２０と、BIOSメモリ１２２とを含む。ノースブリッジ１０４およびサウスブリッジ１１２は、チップをひとつしか含まないものであってもよいし、全体を総称して「チップセット」と呼ぶ複数のチップを含むものであってもよい点に注意されたい。また、コンピュータシステム１００には、キャッシュ、モデム、パラレルインタフェースまたはシリアルインタフェース、SCSIインタフェース、ネットワークインタフェースカード［「Super I／O」はカリフォルニア州サンタクララのナショナルセミコンダクターコーポレーションの商標である]など、必要に応じて他のバス、デバイスおよび／またはサブシステムを含み得る点にも注意されたい。

【０００４】

プロセッサ１０２はノースブリッジ１０４に接続されている。ノースブリッジ１０４は、プロセッサ１０２、メモリ１０６、AGPメモリ１０８、PCIバス１１０それぞれの相互インタフェースを提供する。サウスブリッジ１１２は、PCIバス１１０、IDEインタフェース１１４やUSBインタフェース１１６、LPCバス１１８に接続された周辺機器やデバイス、サブシステムそれぞれの相互インタフェースを提供する。バッテリ１１３はサウスブリッジ１１２に接続された状態で示されている。Super I／O™チップ１２０はLPCバス１１８に接続されている。

【０００５】

ノースブリッジ１０４は、プロセッサ１０２、メモリ１０６、AGPメモリ１０８、PCIバス１１０に接続されたデバイス、サウスブリッジ１１２に接続されたデバイスおよびサブシステム同士および／またはこれらの間での通信アクセスを提供する。一般に、コンピュータシステム１００への接続用のPCIバス１１０とつながるPCI「スロット」（図示せず）に、着脱自在の周辺デバイスが挿入される。あるいは、マザーボード上にあるデバイスをPCIバス１１０に直接接続してもよい。

【０００６】

サウスブリッジ１１２は、PCIバス１１０と、LPCバス１１８（あるいはX－バスまたはISAバスなどの旧世代のバス）を介してコンピュータシステム１００に接続されているのが普通であるモデムやプリンタ、キーボード、マウスなどのさまざまなデバイスやサブシステムとの間のインタフェースを提供する。サウスブリッジ１１２には、デバイスとコンピュータシステム１００の残りの部分とを、IDEインタフェース１１４、USBインタフェース１１６、LPCバス１１８を介してインタフェースするロジックが含まれる。

【発明の開示】

【発明が解決しようとする課題】
【０００７】
ハードウェア的な観点からみると、ｘ８６の動作環境では、ユーザプライバシーの保護、企業秘密および法人資産に対するセキュリティの提供、あるいはコンテンツプロバイダの所有者の権利の保護という点ではほとんど何もしていない。これらの目標すなわちプライバシー、セキュリティ、所有権（Privacy、Security、Ownership：以下、PSOと総称する）はいずれもコンピュータがインターネットに接続される時代にあって次第に重要なものとなりつつあるが、初期のパーソナルコンピュータはPSOの需要を見越して設計されたものではなかった。
【０００８】
ソフトウェア的な観点からみても、ｘ８６の動作環境がPSOの点で粗末なことに変わりはない。ソフトウェア経由あるいは単にパーソナルコンピュータのカバーを開けるだけでハードウェアを直接いじることが容易であるため、セキュリティソフトウェアおよびデバイスの大半が侵入者または泥棒による危機にさらされる。パーソナルコンピュータはものの見事に使いやすいものであるが、この点はPSOの問題を大きくするにすぎないのである。
【課題を解決するための手段】
【０００９】
発明の開示
本発明によれば、プロセッサが得られる。このプロセッサは、１以上のハードウェアデバッグテスト（HDT）イネーブルビットを格納するように構成された第１のレジスタと、複数のHDT入力信号を受信すべく接続された第１の制御ロジックと、第１のレジスタに接続された第２の制御ロジックと、を含む。第１の制御ロジックは第１のレジスタにアクセスすべく接続されている。第２の制御ロジックは、プロセッサのリセットに応答して１以上のデフォルト値を第１のレジスタに格納するように構成されている。
【００１０】
本発明によれば、HDTモードのイネーブルステータスを判断するための方法が得られる。この方法は、HDTモードの開始要求を受信し、HDTモードのイネーブルステータスを判断し、HDTモードのイネーブルステータスがイネーブルに設定されている場合にHDTモードを開始することを含む。
【００１１】
本発明によれば、マイクロコードを修正するための方法が得られる。この方法は、マイクロコード修正要求を受信し、マイクロコードローダーのイネーブルステータスを判断し、マイクロコードローダーのイネーブルステータスがイネーブルに設定されている場合にマイクロコードを修正することを含む。
【００１２】
本発明によれば、HDTモードのステータスの変更方法が得られる。この方法は、HDTモードのステータスの変更要求を受信し、HDTモードのイネーブルロックステータスを判断し、HDTモードのイネーブルロックステータスがアンロックに設定されている場合にHDTモードのステータスを修正することを含む。
【００１３】
本発明によれば、マイクロコードローダーのイネーブルステータスの変更方法が得られる。この方法は、マイクロコードローダーのイネーブルステータスの変更要求を受信し、マイクロコードローダーのイネーブルロックステータスを判断し、マイクロコードローダーのイネーブルロックステータスがアンロックに設定されている場合にマイクロコードローダーのイネーブルステータスを修正することを含む。
【００１４】
本発明によれば、プロセッサを動作させる方法が得られる。この方法は、１以上の不揮発性メモリセルから１以上のデフォルト値を読み出すことと、プルアップレジスタまたはプルダウンレジスタによって１以上のデフォルト値をストラップされた値（strapped value）として受信することと、からなる群から選択される、１以上のデフォルト値を得るこ

とを含む。この方法は、プロセッサのリセットに応答して、１以上のデフォルト値を１以上のさまざまなエントリとして１以上のレジスタに書き込むことを含む。１以上のさまざまなエントリは、１以上のHDTイネーブルビットと、１以上のHDTイネーブルロックビットと、マイクロコードローダーの１以上のイネーブルビットと、マイクロコードローダーの１以上のイネーブルロックビットと、からなる群から選択される。

【００１５】

添付の図面を参照しての以下の説明を参照することで、本発明を理解することができよう。図中、同様の構成要素には同様の参照符号を付してある。

【００１６】

本発明はさまざまな改変や別の形態が可能なものであるが、その具体的な実施形態を一例として図示し、本願明細書中にて詳細に説明する。しかしながら、本願明細書に記載の具体的な実施形態についての説明は、ここに開示の特定の形態に本発明を限定することを意図したものではなく、添付の特許請求の範囲に定義した本発明の範囲に含まれる改変例、等価物および別の形をすべて包含することを意図している点を理解されたい。

【発明を実施するための最良の形態】

【００１７】

以下、実例としての本発明の実施形態について説明する。説明を明確にするため、本願明細書では実際の実施例における特徴をすべて説明したわけではない。もちろん、このような実際の実施形態を開発するにあたっては、いずれの場合もシステム関連の制約やビジネス関連の制約を踏まえるなど、実施例ごとにさまざまな判断を行って開発者らの意図する目的を達成していく必要があり、これは実施例ごとに異なるものであることは理解できよう。さらに、このような開発作業は複雑かつ時間を要するものになる場合があるが、それでも本願の開示内容を利用できる当業者らにとっては日常業務の一環であろうことも理解できよう。参照符号にアルファベットも加えてあるのは、それぞれの参照符号を付した項目の別の実施形態または例を示すためである。

【００１８】

低消費電力設計されたコンピュータシステムの動作モードのひとつにシステム管理モード（system management mode:SMM）がある。SMMは第四世代のx８６プロセッサ用に作られたものである。新しい世代のx８６プロセッサが登場するにつれて、SMMはオペレーティングシステムにとって相対的にトランスペアレントなものとなっていった。すなわち、コンピュータシステムはオペレーティングシステムに対してほとんどまたは全く影響を与えることなくSMMに入ったりこのモードから抜けたりするのである。

【００１９】

本発明の一態様による、セキュア実行ボックス２６０を有するコンピュータシステムにおけるデータ・コマンドフローのフローチャートの実施形態のブロック図を図２に示す。ユーザ入出力（I／O）データおよび／またはコマンド２０５が１以上のアプリケーション２１０との間で受け渡しされる。アプリケーション２１０は、コンピュータシステム１００または他の何らかのコンピュータシステムなどのコンピュータシステム内の暗号サービスプロバイダ２１５とデータやコマンドをやり取りする。暗号サービスプロバイダ２１５は、ハードウェア２３０に対するアクセスを提供するドライバ２２５と、API（アプリケーションプログラミングインタフェース）コール２２０を使って通信するものであってもよい。

【００２０】

本発明の一態様によれば、ドライバ２２５およびハードウェア２３０は、セキュア実行モード（SEM）２６０で動作するように構成されたセキュア実行ボックスの一部である。単にセキュリティ動作とも呼ぶ、プライバシー、セキュリティ、所有権（PSO）についての信頼された動作を、コンピュータシステムがSEM２６０にある間に実行することができる。ユーザI／０２０５および／またはアプリケーション２１０からのソフトウェアコールについてはSMM２６０にあるセキュア実行ボックスに対して行うことができる。また、ソフトウェアコールによってSEM２６０にあるセキュア実行ボックスに対してアクセスし、詳

細については後述するものなどのさまざまなセキュリティハードウェア資源を利用できる。

【００２１】

本発明のさまざまな態様による、SMM MSRを有するコンピュータシステム３００の実施形態をブロック図の形で図３に示す。図３は、プロセッサ３０５と、ノースブリッジ３１０と、メモリ３０６と、サウスブリッジ３３０とを含む。プロセッサは１以上のSMM MSR（マシン固有レジスタ）３０７を含む。ノースブリッジ３１０はメモリコントローラ３１５を含む。ノースブリッジ３１０は、プロセッサ３０５とサウスブリッジ３３０との間で、ローカルバス３０８を介してプロセッサ３０５に接続され、PCIバス１１０を介してサウスブリッジ３３０に接続されている。ノースブリッジ３１０はプロセッサ３０５からSMIACT#信号を受信すべく接続されている。

【００２２】

図３の実施形態では、コンピュータシステム３００は、標準的なプロセッサ信号（ノースブリッジ３１０へのSMIACT#など）および／またはローカルバス３０８およびPCIバス１１０でのバスサイクルを利用して、プロセッサ３０５がSMMにある旨を伝達する。

【００２３】

本発明のさまざまな実施形態では、SMMのセキュリティが前提である。弱点を突かれてSMMのセキュリティを危うくする可能性のある１以上のいわゆる「裏口」が存在し得る点に注意されたい。企図される問題としては、プロセッサ３０５のハードウェアデバッグテスト（HDT）モードの誤用ならびにプロセッサ３０５がマイクロコードをロードして差し替える機能があげられる。図４A～図４Dに示されているのは、プロセッサ３０５のさまざまな実施形態３０５A、３０５B、３０５C、３０５Dであり、その各々が１以上の裏口からの攻撃に対するさまざまなセキュリティプロテクションを含む。

【００２４】

図４Aでは、プロセッサ３０５Aに、HDT制御ロジック４１０Aと、HDTリセットロジック４２０Aと、HDTイネーブルレジスタ４１５および不揮発性ランダムアクセスメモリ（NVRAM）４３０を含む１以上のレジスタとが含まれる。図示のように、HDT制御ロジック４１０Aは、複数のHDTピン４０５を介して複数の入力信号を受信すべく接続されている。HDT制御ロジック４１０Aはさらに、HDTイネーブルレジスタ４１５に接続されている。HDTリセットロジック４２０Aは、ライン４２５でRESET信号を受信し、HDTイネーブルレジスタ４１５およびNVRAM４３０にアクセス（すなわち読み出しおよび書き込み）すべく接続されている。

【００２５】

図４Bでは、図４Bのプロセッサ３０５Bに、HDT制御ロジック４１０Bと、HDTリセットロジック４２０Bと、HDTイネーブルレジスタ４１５およびHDTイネーブルロックレジスタ４３５を含む２つのレジスタと、が含まれる。図示のように、HDT制御ロジック４１０Bは、複数のHDTピン４０５を介して複数の入力信号を受信すべく接続されている。HDT制御ロジック４１０Bはさらに、HDTイネーブルレジスタ４１５およびHDTイネーブルロックレジスタ４３５に接続されている。HDTリセットロジック４２０Bは、プルアップ（またはプルダウン）抵抗器４４５を介して、ライン４２５でRESET信号を受信し、ライン４４０などで信号を受信すべく接続されている。

【００２６】

図４Cでは、プロセッサ３０５Cに、マイクロコード制御ロジック４５５と、マイクロコードローダーのイネーブルリセットロジック４６５と、マイクロコードローダーのイネーブルレジスタ４６０を含む１以上のレジスタと、が含まれる。図示のように、マイクロコード制御ロジック４５５は、複数のマイクロコード入力ピン４５０を介して複数の入力信号を受信すべく接続されている。マイクロコード制御ロジック４５５はさらに、マイクロコードローダーのイネーブルレジスタ４６０に接続されている。マイクロコードローダーのイネーブルリセットロジック４６５は、RESET信号を受信し、マイクロコードローダーのイネーブルレジスタ４６０にアクセスすべく接続されている。

【００２７】
図４Dでは、プロセッサ３０５Dに、マイクロコード制御ロジック４５５と統合されたHDT制御ロジック４１０と、HDTリセットロジック４２０と、制御／リセットロジック４７５を形成するためのMLEリセットロジック４６５とが含まれる。HDTイネーブルレジスタ４１５およびマイクロコードローダーのイネーブルレジスタ４６０は、マルチビットロックレジスタ４８０に統合されている。制御／リセットロジック４７５への複数の入力４７０が図示されている。これらの複数の入力４７０には、HDT入力４０５、マイクロコード入力４５０および／またはリセット信号伝達手段を含み得る。他の実施形態（図示せず）では、HDT制御ロジック４１０およびマイクロコード制御ロジック４５５のみを統合するか、あるいはHDTリセットロジック４２０およびMLEリセットロジック４６５だけを統合する。

【００２８】
本発明のさまざまな実施形態によれば、レジスタ４１５、４３５および４６０ならびにNVRAM４３０に、１以上のビット用の格納空間が含まれる。一実施形態では、各レジスタが１ビットを格納するように構成されている。イネーブルレジスタ４１５および４６０を単一のロックレジスタに統合してもよく、HDTイネーブルロックレジスタ４３５をマイクロコードのイネーブルロックレジスタとして利用してもよい点に注意されたい。レジスタ４１５、４３５、４６０および／または４８０をSMM MSR３０７に含み得る点も企図される。

【００２９】
さまざまな実施形態において、HDTイネーブルレジスタ４１５は、HDTモードがイネーブルであるかデセーブルであるかを示す１以上のHDTイネーブルビットを格納するように構成されている。HDTリセットロジック４２０は、プロセッサ３０５のリセット時に１以上のHDTイネーブルビットをデフォルトの状態に設定するように構成されている。

【００３０】
図４Aおよび図４Bに示すものなどのHDTモードを制御するための複数の実施形態が企図される。一実施形態では、エンジニアリングと試験とに用いられる、量産品ではないプロセッサ３０５では、HDTモードがデフォルトでイネーブルである。このHDTモードについては、量産される標準的なプロセッサ３０５のデフォルトでデセーブルにしてもよい。図４Aに示すもうひとつの実施形態では、デフォルトの状態をNVRAM４３０に格納し、ここから読み出すようにすることができる。この実施形態ではデフォルトの状態を変更可能にしてもよいが、図示の実施形態ではデフォルトの状態はデセーブルに設定されている。図４Bに示すさらに他の実施形態では、ストラッピングオプションを使ってデフォルトの状態を設定する。このデフォルト値がプルアップ（またはプルダウン）抵抗器４４５を介してHDTリセットロジック４２０Bに提供される。

【００３１】
図４Cおよび図４Dに示すものなどのマイクロコードローダーのモードを制御するための複数の実施形態も企図される。図示しない一実施形態では、マイクロコード更新モードを、エンジニアリングと試験とに用いられる量産品ではないプロセッサ３０５上のデフォルトとしてイネーブルにする。このマイクロコード更新モードについては、量産される標準的なプロセッサ３０５のデフォルトとしてデセーブルにすることができる。図４Aに示す実施形態に類似の別の実施形態では、デフォルトの状態をNVRAM４３０に格納し、ここから読み出すようにすることができる。この実施形態ではデフォルトの状態を変更可能にしてもよいが、図示の実施形態ではデフォルトの状態はデセーブルに設定されている。図４Bに示すさらに他の実施形態では、ストラッピングオプションを使うことがデフォルトの状態である。このデフォルト値がプルアップ（またはプルダウン）抵抗器４４５を介してMLEリセットロジック４６５に提供される。

【００３２】
ここで図５に移ると、HDTモードを開始するための方法５００が図示されている。HDTモードに入る要求を受信（ステップ５０５）したことに応答して、HDT制御ロジック４１０が１以上のHDTイネーブルビットのステータスをチェックし、HDTモードがイネーブルである

かデセーブルであるかを確認する（ステップ５１０）。HDTモードがイネーブルである（ステップ５１５）場合、HDT制御ロジック４１０はHDTモードを開始する（ステップ５２０）。HDTモードがデセーブルである（ステップ５１５）場合、HDT制御ロジック４１０がHDTモードを開始することはない。

【００３３】

ここで図６に移ると、HDTモードのロックを含むHDTモードのイネーブルステータスを変更するための方法６００が図示されている。HDTモードに入る要求を受信（ステップ６０５）したことに応答して、HDT制御ロジック４１０が１以上のHDTイネーブルロックビットのステータスをチェックし、HDTロックモードがロック状態にあるかアンロック状態にあるかを判断する（ステップ６１０）。HDTロックモードがアンロック状態にある（ステップ６１５）場合、HDT制御ロジック４１０はHDTモードを開始する（ステップ６３５）。HDTロックモードがロック状態にある（ステップ６１５）場合、HDT制御ロジック４１０はHDTロックモードのステータスを変更するための承認を要求する（ステップ６２０）。変更が承認された（ステップ６２５）場合、HDT制御ロジック４１０がHDTモードのロックビットを変更してアンロック状態にする（ステップ６３０）。変更が承認されなかった（ステップ６２５）場合、HDT制御ロジック４１０によってHDTモードのロックビットが変更されることはない。

【００３４】

さまざまな実施形態において、HDTイネーブルステータスの変更には１以上のHDTイネーブルステータスビットを設定またはリセットすればよい。たとえば、HDTモードをデセーブルにすることができるが、SMM内では、HDT制御ロジック４１０へのあらかじめ定められた入力によってHDT制御ロジック４１０にHDTモードのステータスを変更してイネーブルにするように伝えることができる。図４Aの実施形態では、たとえば、一旦信号を受けると、HDT制御ロジック４１０はHDTイネーブルビットのステータスをデセーブルからイネーブルに変更する。

【００３５】

図４Bの実施形態に戻ると、たとえば、HDTモードのステータス変更の要求を受信したことに応答して、HDT制御ロジック４１０が１以上のHDTイネーブルロックビットのステータスをチェックし、HDTロックモードがイネーブルであるかデセーブルであるかを確認する。HDTロックモードがデセーブルである場合、HDT制御ロジック４１０はHDTモードのステータスを変更することができる。HDTロックモードがイネーブルである場合、HDT制御ロジック４１０がHDTモードのステータスを変更することはない。

【００３６】

HDTロックモードがロック状態にある（ステップ６１５）場合に、HDTロックモードのステータスを変更するための承認を要求する（ステップ６２０）のではなく方法６００を終了させてもよい点に注意されたい。方法６００には、方法６００で承認を要求する（ステップ６２０）前にHDTロックモードのステータス変更の要求を受け取る（図示せず）ことも含み得る。

【００３７】

ここで図７に移ると、マイクロコードローダーを開始するための方法７００が図示されている。マイクロコード更新モードの開始要求を受信（ステップ７０５）したことに応答して、マイクロコード制御ロジック４５５が１以上のマイクロコードイネーブルビットのステータスをチェックし、マイクロコード更新モードがイネーブルであるかデセーブルであるかを確認する（ステップ７１０）。マイクロコード更新モードがイネーブルである（ステップ７１５）場合、マイクロコード制御ロジック４５５がマイクロコード更新モードを開始する（ステップ７２０）。マイクロコード更新モードがデセーブルである（ステップ７１５）場合、マイクロコード制御ロジック４５５がマイクロコード更新モードを開始することはない。

【００３８】

ここで図８に移ると、マイクロコードモードのロックを含むマイクロコード更新モードの

イネーブルステータスを変更するための方法８００が図示されている。マイクロコードモードに入る要求を受信（ステップ８０５）したことに応答して、マイクロコード制御ロジック４５５が１以上のマイクロコードイネーブルロックビットのステータスをチェックし、マイクロコードモードがロック状態であるかアンロック状態であるかを確認する（ステップ８１０）。マイクロコードロックモードがアンロック状態にある（ステップ８１５）場合、マイクロコード制御ロジック４５５がマイクロコードモードを開始する（ステップ８３５）。マイクロコードロックモードがロック状態にある（ステップ８１５）場合は、マイクロコード制御ロジック４５５がマイクロコードモードのロックステータスを変更するための承認を要求する（ステップ８２０）。変更が承認された（ステップ８２５）場合、マイクロコード制御ロジック４５５がマイクロコードモードのロックビットを変更してアンロック状態にする（ステップ８３０）。変更が承認されなかった（ステップ８２５）場合、マイクロコード制御ロジック４５５によってマイクロコードモードのロックビットが変更されることはない。

【００３９】

さまざまな実施形態において、マイクロコードイネーブルステータスの変更には１以上のマイクロコードイネーブルステータスビットを設定またはリセットすればよい。たとえば、マイクロコードモードをデセーブルにすることができるが、SMM内では、マイクロコード制御ロジック４５５へのあらかじめ定められた入力によってマイクロコード制御ロジック４５５にマイクロコードモードのステータスを変更してイネーブルにするように伝えることができる。図４Cの実施形態では、たとえば、一旦信号を受けると、マイクロコード制御ロジック４５５は１以上のマイクロコードイネーブルビットのステータスをデセーブルからイネーブルに変更する。

【００４０】

マイクロコードモードのステータス変更の要求を受信したことに応答して、マイクロコード制御ロジック４５５が１以上のマイクロコードイネーブルロックビットのステータスをチェックし、マイクロコードロックモードがイネーブルであるかデセーブルであるかを判断することができる。マイクロコードロックモードがデセーブルである場合、マイクロコード制御ロジック４５５はマイクロコードモードのステータスを変更することができる。マイクロコードロックモードがイネーブルである場合、マイクロコード制御ロジック４５５がマイクロコードモードのステータスを変更することはない。

【００４１】

マイクロコード更新ロックステータスがロック状態にある（ステップ８１５）場合に、マイクロコード更新ロックのステータスを変更するための承認を要求する（ステップ８２０）のではなく方法８００を終了させてもよい点に注意されたい。方法８００には、方法３５００で承認を要求する（ステップ８２０）前にマイクロコード更新ロックステータス変更の要求を受け取る（図示せず）ことも含み得る。

【００４２】

本願開示の目的において、ROMについて説明した内容は、フラッシュメモリや他の実質的に不揮発性のメモリタイプにも適用されるものとする。以上、本願明細書に開示の本発明の方法をフローチャートの形で説明してきたが、さまざまな実施形態においてこれらのフローチャートのさまざまな要素を省略または異なる順序で実行してもよい点に注意されたい。また、本願に開示の本発明の方法では実施例のバリエーションを認めている点にも注意されたい。

【００４３】

上記に開示したような本発明のいくつかの態様は、ハードウェアまたはソフトウェアでインプリメントできるものである。したがって、結果として本願明細書に記載の詳細な説明の内容については、ハードウェアで実現できるプロセスに関するものもあれば、コンピュータシステムまたはコンピューティングデバイスのメモリ内のデータビットについての動作を記号的に表現した内容に関係のある、ソフトウェアで実現できるプロセスに関するものもある。これらの説明および表現は、当業者らが自分たちの行っている作業の内容を、

ハードウェアとソフトウェアの両方を使用している他の当業者に最も効率よく伝達するのに使用する手段である。両方のプロセスおよび動作には物理的な量を物理的に操作する必要がある。ソフトウェアでは、必ずしも必要というわけではないが、これらの量は格納や転送、結合、比較、あるいは操作が可能な電気信号や磁気信号、あるいは光信号の形態をとるのが普通である。主に一般利用の観点から、場合によってはこれらの信号をビットや値、エレメント、シンボル、文字、項、数などで参照すると都合がよいことが明らかになっている。

【００４４】

しかしながら、上記の用語および同様の用語はすべてしかるべき物理的な量に関連付けられるものであり、これらの量に付される都合のよいラベルにすぎない点に留意すべきである。特に明記する場合または自明であろうと思われる場合を除き、本願開示全体を通して、これらの説明は、いくつかの電子デバイスのストレージ内で物理的な（電子的、磁気的または光学的）量として表されるデータを操作し、ストレージ内あるいは送信デバイスまたはディスプレイデバイスで同様に物理的な量として表される他のデータに変換する電子デバイスの行為およびプロセスについてのものである。このような説明を示す代表的な用語としては、「processing（処理する／処理／プロセッシング）」、「computing（コンピューティング）」、「calculating（計算する／計算）」、「determining（判断する／判定する／判断／判定）」、「displaying（表示する／表示）」などの用語があるが、これに限定されるものではない。

【００４５】

また、ソフトウェアで実現される本発明の態様は一般に、何らかの形のプログラム格納媒体でエンコードされるか、何らかのタイプの送信媒体を使って実現される点に注意されたい。プログラム格納媒体としては、磁気（フロッピーディスクまたはハードドライブなど）または光（コンパクトディスク読み出し専用メモリすなわち「CD ROM」など）があげられ、読み出し専用メモリであってもランダムアクセスメモリであってもよい。同様に、送信媒体はツイストペア線であってもよいし、同軸ケーブル、光ファイバまたは従来技術において周知の他の何らかの好適な送信媒体であってもよい。本発明は特定の実施例のいずれの態様にも限定されるものではない。

【００４６】

本発明は本願明細書の教示内容の利益を享受する当業者らに自明の上記とは異なるが等価な方法で改変および実施することのできるものであるため、上記にて開示した個々の実施形態は一例にすぎない。さらに、添付の請求の範囲に記載したものを除き、本願明細書に図示した構成または設計の詳細に対する限定を何ら意図するものではない。したがって、上記にて開示した個々の実施形態を変更または改変してもよく、そのような変形例はいずれも本発明の範囲に包含されるとみなせることは明白である。よって、本願明細書で求める保護は添付の請求の範囲に記載のとおりである。

【図面の簡単な説明】

【００４７】

【図１】従来技術のコンピュータシステムのブロック図を示す。

【図２】本発明の一態様による、セキュア実行ボックスを有するコンピュータシステムにおけるデータ・コマンドフローの実施形態のフローチャートを示す。

【図３】本発明の一態様による、SMM MSRを有するコンピュータシステムの実施形態のブロック図を示す。

【図４Ａ】本発明のさまざまな態様による、ロックレジスタとロジックとを含むプロセッサの実施形態のブロック図を示す。

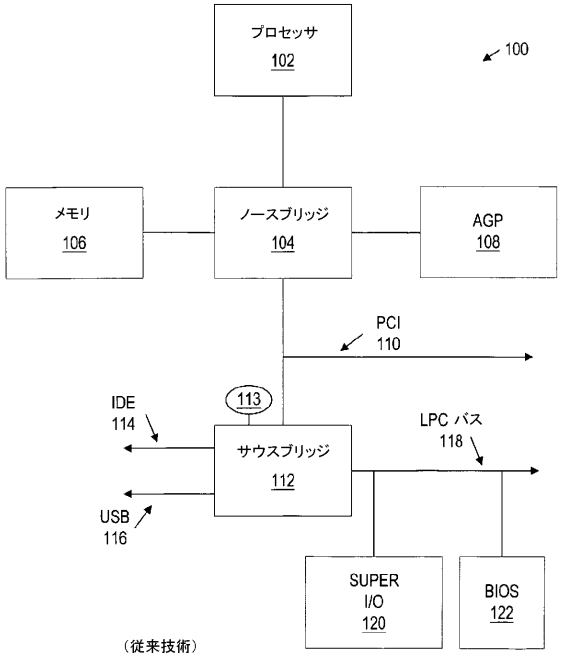
【図４Ｂ】本発明のさまざまな態様による、ロックレジスタとロジックとを含むプロセッサの実施形態のブロック図を示す。

【図４Ｃ】本発明のさまざまな態様による、ロックレジスタとロジックとを含むプロセッサの実施形態のブロック図を示す。

【図４Ｄ】本発明のさまざまな態様による、ロックレジスタとロジックとを含むプロセッ

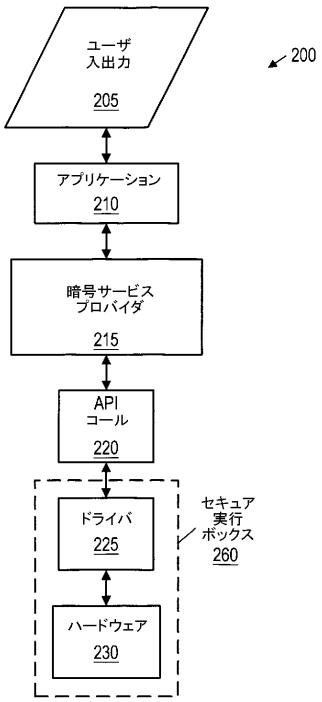
サの実施形態のブロック図を示す。
【図５】本発明の一態様による、HDTモードを開始するための方法の実施形態のフローチャートを示す。
【図６】本発明の一態様による、HDTイネーブルステータスを変更するための方法の実施形態のフローチャートを示す。
【図７】本発明の一態様による、マイクロコードローダーを開始するための方法の実施形態のフローチャートを示す。
【図８】本発明の一態様による、マイクロコードローダーのイネーブルステータスを変更するための方法の実施形態のフローチャートを示す。

【図１】

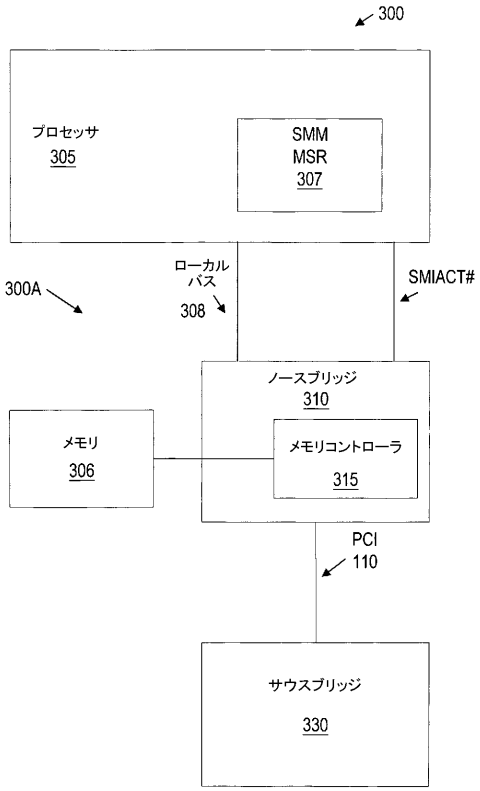


（従来技術）

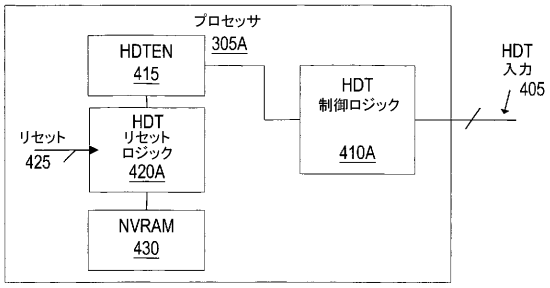
【図２】



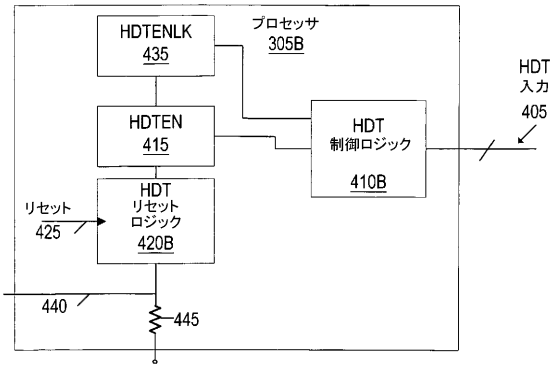
【図３】



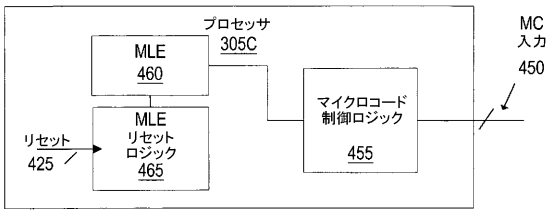
【図４Ａ】



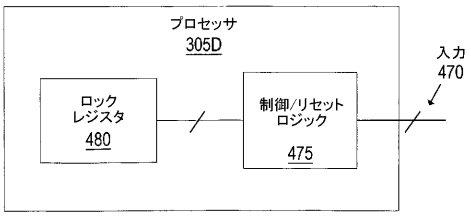
【図４Ｂ】



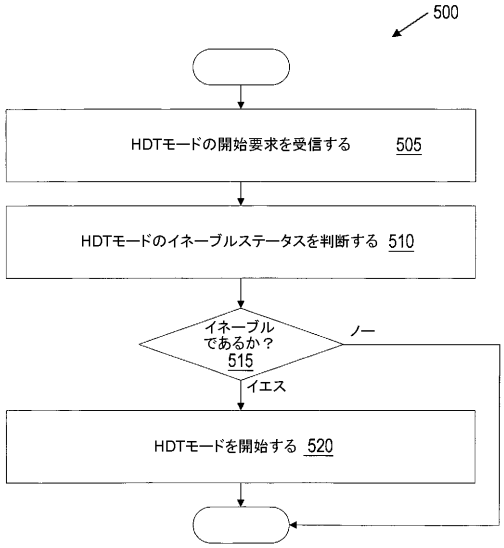
【図４Ｃ】



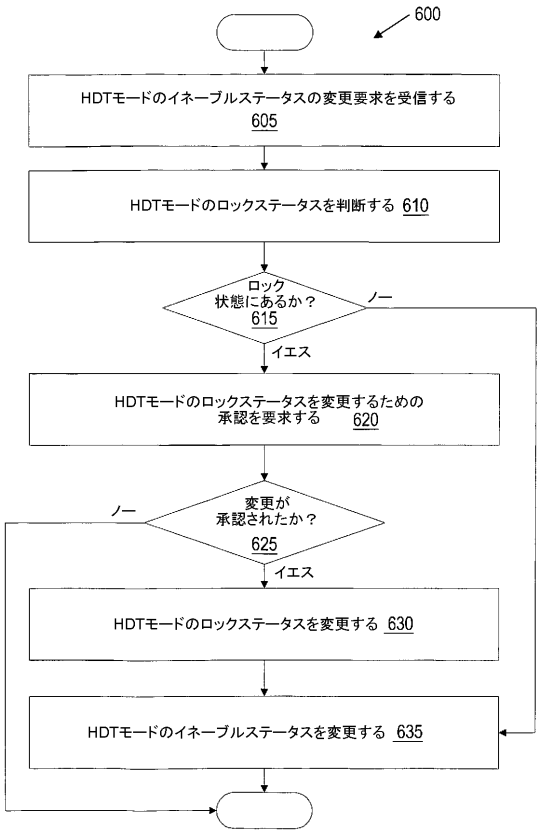
【図４Ｄ】



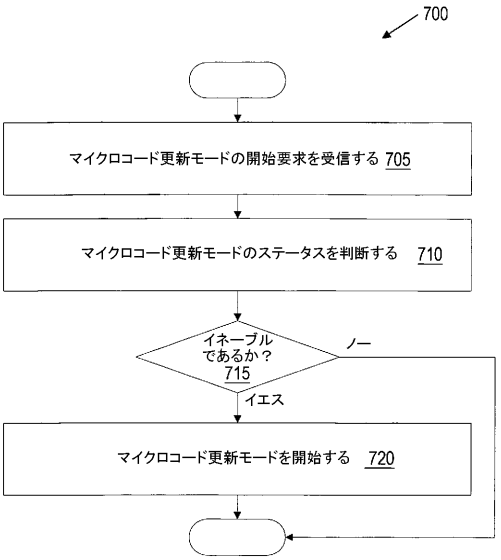
【図５】



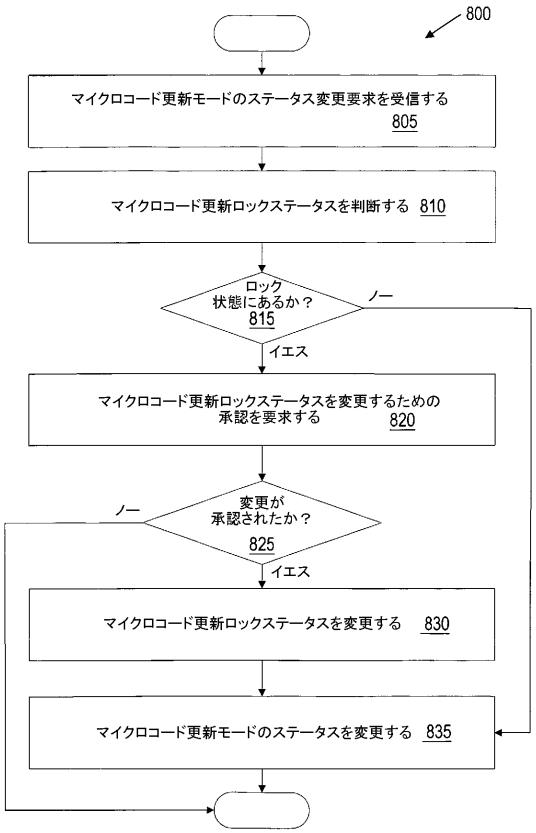
【図６】


600
HDTモードのイネーブルステータスの変更要求を受信する 605
HDTモードのロックステータスを判断する 610
ロック状態にあるか？ 615
ノー
イエス
HDTモードのロックステータスを変更するための承認を要求する 620
変更が承認されたか？ 625
ノー
イエス
HDTモードのロックステータスを変更する 630
HDTモードのイネーブルステータスを変更する 635

【図７】


700
マイクロコード更新モードの開始要求を受信する 705
マイクロコード更新モードのステータスを判断する 710
イネーブルであるか？ 715
ノー
イエス
マイクロコード更新モードを開始する 720

【図８】


800
マイクロコード更新モードのステータス変更要求を受信する 805
マイクロコード更新ロックステータスを判断する 810
ロック状態にあるか？ 815
ノー
イエス
マイクロコード更新ロックステータスを変更するための承認を要求する 820
変更が承認されたか？ 825
ノー
イエス
マイクロコード更新ロックステータスを変更する 830
マイクロコード更新モードのステータスを変更する 835

【国際公開パンフレット】

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
21 November 2002 (21.11.2002)

PCT

(10) International Publication Number
WO 02/093336 A2

(51) International Patent Classification[7]: G06F 1/00

(21) International Application Number: PCT/US02/11935

(22) International Filing Date: 17 April 2002 (17.04.2002)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
09/852,942 10 May 2001 (10.05.2001) US
09/852,372 10 May 2001 (10.05.2001) US
09/853,226 11 May 2001 (11.05.2001) US

(71) Applicant: ADVANCED MICRO DEVICES, INC. [US/US]; One AMD Place, P.O. Box 9453, Sunnyvale, CA 94088-3453 (US).

(72) Inventor: STRONGIN, Geoffrey, S.; 7210 Montana Norte, Austin, TX 78731 (US).

(74) Agent: DRAKE, Paul, S.; Advanced Micro Devices, Inc., 5204 East Ben White Boulevard, M/S 562, Austin, TX 78741 (US).

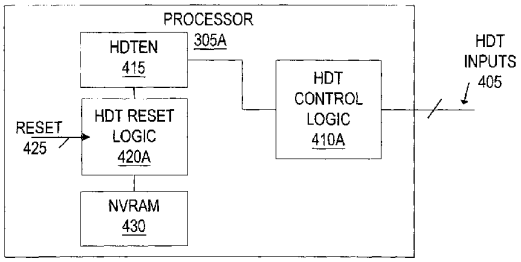
(81) Designated States (national): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, OM, PH, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZM, ZW.

(84) Designated States (regional): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— without international search report and to be republished upon receipt of that report

[Continued on next page]

(54) Title: MECHANISM FOR CLOSING BACK DOOR ACCESS MECHANISMS IN PERSONAL COMPUTER SYSTEMS



(57) Abstract: Methods, devices, and systems for closing back door access mechanisms. A processor includes a first register configured to store one or more hardware-debug-test (HDT) enable bits, a first control logic coupled to receive a plurality of HDT input signals, and a second control logic coupled to the first register. The first control logic is coupled to access the first register. The second control logic is configured to store one or more default values in the first register in response to a reset of the processor. Another processor includes a first control logic coupled to receive a plurality of microcode inputs, a first register coupled to the first control logic, and a second control logic coupled to the first register. The first register is configured to store one or more microcode loader enable bits. The second control logic is configured to store one or more default values in the first register in response to a reset of the processor.

WO 02/093336 A2

WO 02/093336 A2

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

MECHANISM FOR CLOSING BACK DOOR ACCESS MECHANISMS IN PERSONAL COMPUTER SYSTEMS

This Application is a continuation-in-part of co-pending U.S. Patent Application No. 09/852,372, entitled, "Secure Execution Box and Method," filed on May 10, 2001, whose inventors are Dale E. Gulick and Geoffrey S. Strongin. This Application is also a continuation-in-part of co-pending U.S. Patent Application No. 09/852,942, entitled, "Computer System Architecture for Enhanced Security and Manageability," filed on May 10, 2001, whose inventors are Geoffrey S. Strongin and Dale E. Gulick.

TECHNICAL FIELD

This invention relates generally to computing systems, and, more particularly, to a method and apparatus for protecting personal computer systems from unauthorized back door accesses.

BACKGROUND ART

Fig. 1 illustrates an exemplary computer system 100. The computer system 100 includes a processor 102, a north bridge 104, memory 106, Advanced Graphics Port (AGP) memory 108, a Peripheral Component Interconnect (PCI) bus 110, a south bridge 112, a battery, an AT Attachment (ATA) interface 114 (more commonly known as an Integrated Drive Electronics (IDE) interface), a universal serial bus (USB) interface 116, a Low Pin Count (LPC) bus 118, an input/output controller chip (SuperI/O™) 120, and BIOS memory 122. It is noted that the north bridge 104 and the south bridge 112 may include only a single chip or a plurality of chips, leading to the collective term "chipset." It is also noted that other buses, devices, and/or subsystems may be included in the computer system 100 as desired, e.g. caches, modems, parallel or serial interfaces, SCSI interfaces, network interface cards, etc. ["SuperI/O" is a trademark of National Semiconductor Corporation of Santa Clara, Calif.]

The processor 102 is coupled to the north bridge 104. The north bridge 104 provides an interface between the processor 102, the memory 106, the AGP memory 108, and the PCI bus 110. The south bridge 112 provides an interface between the PCI bus 110 and the peripherals, devices, and subsystems coupled to the IDE interface 114, the USB interface 116, and the LPC bus 118. The battery 113 is shown coupled to the south bridge 112. The Super I/O™ chip 120 is coupled to the LPC bus 118.

The north bridge 104 provides communications access between and/or among the processor 102, memory 106, the AGP memory 108, devices coupled to the PCI bus 110, and devices and subsystems coupled to the south bridge 112. Typically, removable peripheral devices are inserted into PCI "slots" (not shown) that connect to the PCI bus 110 to couple to the computer system 100. Alternatively, devices located on a motherboard may be directly connected to the PCI bus 110.

The south bridge 112 provides an interface between the PCI bus 110 and various devices and subsystems, such as a modem, a printer, keyboard, mouse, etc., which are generally coupled to the computer system 100 through the LPC bus 118 (or its predecessors, such as an X-bus or an ISA bus). The south bridge 112 includes the logic used to interface the devices to the rest of computer system 100 through the IDE interface 114, the USB interface 116, and the LPC bus 118.

From a hardware point of view, an x86 operating environment provides little for protecting user privacy, providing security for corporate secrets and assets, or protecting the ownership rights of content providers. All of these goals, privacy, security, and ownership (collectively, PSO) are becoming critical in an

age of Internet-connected computers. The original personal computers were not designed in anticipation of PSO needs.

From a software point of view, the x86 operating environment is equally poor for PSO. The ease of direct access to the hardware through software or simply by opening the cover of the personal computer allows an intruder or thief to compromise most security software and devices. The personal computer's exemplary ease of use only adds to the problems for PSO.

DISCLOSURE OF INVENTION

In accordance with the present invention, a processor is provided. The processor includes a first register configured to store one or more hardware-debug-test (HDT) enable bits, a first control logic coupled to receive a plurality of HDT input signals, and a second control logic coupled to the first register. The first control logic is coupled to access the first register. The second control logic is configured to store one or more default values in the first register in response to a reset of the processor.

In accordance with the present invention, a method for determining an HDT mode enable status is provided. The method includes receiving a request to initiate the HDT mode, determining HDT mode enable status, and initiating the HDT mode if the HDT mode enable status is set to enabled.

In accordance with the present invention, a method for modifying microcode is provided. The method includes receiving a request to modify microcode, determining microcode loader enable status, and modifying microcode if the microcode loader enable status is set to enabled.

In accordance with the present invention, a method of changing HDT mode status is provided. The method includes receiving a request to change HDT mode status, determining HDT mode enable lock status, modifying HDT mode status if the HDT mode enable lock status is set to unlocked.

In accordance with the present invention, a method of changing microcode loader enable status is provided. The method includes receiving a request to change microcode loader enable status, determining microcode loader enable lock status, and modifying microcode loader enable status if the microcode loader enable lock status is set to unlocked.

In accordance with the present invention, a method of operating a processor is provided. The method includes obtaining one or more default values selected from the group consisting of reading the one or more default values from one or more non-volatile memory cells and receiving the one or more default values as a strapped value through a pull-up or pull-down resistor. The method includes writing the one or more default values as one or more various entries in one or more registers in response to a reset of the processor. The one or more various entries are selected from the group consisting of one or more HDT enable bits, one or more HDT enable lock bits, one or more microcode loader enable bits, and one or more microcode loader enable lock bits.

BRIEF DESCRIPTION OF THE DRAWINGS

The invention may be understood by reference to the following description taken in conjunction with the accompanying drawings, in which like reference numerals identify similar elements, and in which:

Fig. 1 illustrates a block diagram of a prior art computer system;

Fig. 2 illustrates a flowchart of an embodiment of data and command flow in a computer system having a secure execution box, according to one aspect of the present invention;

Fig. 3 illustrates a block diagram of an embodiment of a computer system having SMM MSRs, according to one aspect of the present invention;

Figs. 4A-4D illustrate block diagrams of embodiments of processors including lock registers and logic, according to various aspects of the present invention;

Fig. 5 illustrates a flowchart of an embodiment of a method for initiating HDT mode, according to one aspect of the present invention;

Fig. 6 illustrates a flowchart of an embodiment of a method for changing the HDT enable status, according to one aspect of the present invention;

Fig. 7 illustrates a flowchart of an embodiment of a method for initiating the microcode loader, according to one aspect of the present invention; and

Fig. 8 illustrates a flowchart of an embodiment of a method for changing the microcode loader enable status, according to one aspect of the present invention.

While the invention is susceptible to various modifications and alternative forms, specific embodiments thereof have been shown by way of example in the drawings and are herein described in detail. It should be understood, however, that the description herein of specific embodiments is not intended to limit the invention to the particular forms disclosed, but on the contrary, the intention is to cover all modifications, equivalents, and alternatives falling within the scope of the invention as defined by the appended claims.

MODE(S) FOR CARRYING OUT THE INVENTION

Illustrative embodiments of the invention are described below. In the interest of clarity, not all features of an actual implementation are described in this specification. It will, of course, be appreciated that in the development of any such actual embodiment, numerous implementation-specific decisions must be made to achieve the developers' specific goals, such as compliance with system-related and business-related constraints, which will vary from one implementation to another. Moreover, it will be appreciated that such a development effort might be complex and time-consuming, but would nevertheless be a routine undertaking for those of ordinary skill in the art having the benefit of this disclosure. The use of a letter in association with a reference number is intended to show alternative embodiments or examples of the item to which the reference number is connected.

System Management Mode (SMM) is a mode of operation in the computer system that was implemented to conserve power. The SMM was created for the fourth generation x86 processors. As newer x86 generation processors have appeared, the SMM has become relatively transparent to the operating system. That is, computer systems enter and leave the SMM with little or no impact on the operating system.

Fig. 2 illustrates a block diagram of an embodiment of a flowchart showing data and command flow in a computer system having a secure execution box 260, according to one aspect of the present invention. User input and output (I/O) data and/or commands 205 are provided to and received from one or more applications 210. The applications 210 exchange data and commands with cryptography service providers 215 within the computer system, such as the computer system 100 or any other computer system. The cryptography service providers 215 may use API (Application Programming Interface) calls 220 to interact with drivers 225 that provide access to hardware 230.

According to one aspect of the present invention, the drivers 225 and the hardware 230 are part of a secure execution box configured to operate in a secure execution mode (SEM) 260. Trusted privacy, security and ownership (PSO) operations, also referred to simply as security operations, may take place while the computer system is in SEM 260. Software calls propagated from the user I/O 205 and/or the applications 210

may be placed into the secure execution box in SMM 260. The software calls have access to the secure execution box in SEM 260 to various security hardware resources.

Fig. 3 illustrates a block diagram of an embodiment of a computer systems 300 with SMM MSRs, according to various aspects of the present invention. Fig. 3 includes a processor 305, a north bridge 310, memory 306, and a south bridge 330. The processor includes one or more SMM MSRs (machine specific registers) 307. The north bridge 310 includes a memory controller 315. The north bridge 310 is coupled between the processor 305 and the south bridge 330, to the processor 305 through a local bus 308 and to the south bridge 330 through the PCI bus 110. The north bridge 310 is coupled to receive the SMIACT# signal from the processor 305.

In the embodiment of Fig. 3, the computer system 300 signals that the processor 305 is in SMM using standard processor signals (e.g. SMIACT# to the north bridge 310) and/or bus cycles on the local bus 308 and PCI bus 110.

In various embodiments of the present invention, the security of SMM is assumed. It is noted that one or more so-called "backdoors" may exist that could be exploited to compromise the security of SMM. The issues contemplated include misuse of the hardware debug test (HDT) mode of the processor 305 as well as the ability of the processor 305 to load and replace microcode. Illustrated in Figs. 4A-D are various embodiments 305A, 305B, 305C, 305D of the processor 305, each of which includes various security protections against one or more backdoor attacks.

In Fig. 4A, the processor 305A includes HDT control logic 410A, HDT reset logic 420A, and one or more registers, including an HDT enable register 415 and non-volatile random access memory (NVRAM) 430. As shown, the HDT control logic 410A is coupled to receive a plurality of input signals through a plurality of HDT pins 405. The HDT control logic 410A is further coupled to the HDT enable register 415. The HDT reset logic 420A is coupled to receive a RESET signal over a line 425 and to access (i.e. read and write) the HDT enable register 415 and the NVRAM 430.

In Fig. 4B, the processor 305B of Fig. 4B includes HDT control logic 410B, HDT reset logic 420B, and two registers, including the HDT enable register 415 and an HDT enable lock register 435. As shown, the HDT control logic 410B is coupled to receive a plurality of input signals through the plurality of HDT pins 405. The HDT control logic 410B is further coupled to the HDT enable register 415 and the HDT enable lock register 435. The HDT reset logic 420B is coupled to receive the RESET signal over the line 425 and a signal, such as over a line 440, through a pull-up (or pull-down) resistor 445.

In Fig. 4C, the processor 305C includes microcode control logic 455, microcode loader enable reset logic 465, and one or more registers, including a microcode loader enable register 460. As shown, the microcode control logic 455 is coupled to receive a plurality of input signals through a plurality of microcode input pins 450. The microcode control logic 455 is further coupled to the microcode loader enable register 460. The microcode loader enable reset logic 465 is coupled to receive the RESET signal and to access the microcode loader enable register 460.

In Fig. 4D, the processor 305D includes HDT control logic 410 integrated with the microcode control logic 455, the HDT reset logic 420, and the MLE reset logic 465 to form control/reset logic 475. The HDT enable register 415 and the microcode loader enable register 460 are integrated into a multibit lock register 480. A plurality of inputs 470 are shown to the control/reset logic 475. The plurality of inputs 470 may include the

HDT inputs 405, the microcode inputs 450, and/or the reset signaling means. Other embodiments (not shown) integrate only the HDT control logic 410 and the microcode control logic 455, or just the HDT reset logic 420 and the MLE reset logic 465.

According to various embodiments of the present invention, the registers 415, 435, and 460, as well as the NVRAM 430 include storage space for one or more bits. In one embodiment, each register is configured to store a single bit. It is noted that the enable registers 415 and 460 may also be integrated into a single lock register, and the HDT enable lock register 435 may be used as a microcode enable lock register. It is contemplated that the registers 415, 435, 460, and/or 480 could be included in the SMM MSRs 307.

In various embodiments, the HDT enable register 415 is configured to store one or more HDT enable bits signifying whether HDT mode is enabled or disabled. The HDT reset logic 420 is configured to set the one or more HDT enable bits to a default state upon a reset of the processor 305.

Multiple embodiments for controlling the HDT modes are contemplated, such as those illustrated in Figs. 4A and 4B. In one embodiment, the HDT mode is enabled as the default on non-production processors 305 used for engineering and testing. The HDT mode may be disabled as the default in standard production processors 305. In another embodiment, illustrated in Fig. 4A, the default state may be stored in and read from the NVRAM 430. In this embodiment, the default state may be changeable, but in the illustrated embodiment, the default state is set to disabled. In still another embodiment, illustrated in Fig. 4B, the default state is set using a strapping option. The default value is provided to the HDT reset logic 420B through the pull-up (or pull-down) resistor 445.

Multiple embodiments for controlling the microcode loader modes are also contemplated, such as those illustrated in Figs. 4C and 4D. In one embodiment, not illustrated, the microcode update mode is enabled as the default on non-production processors 305 used for engineering and testing. The microcode update mode may be disabled as the default in standard production processors 305. In another embodiment, similar to that illustrated in Fig. 4A, the default state may be stored in and read from the NVRAM 430. In this embodiment, the default state may be changeable, but in the illustrated embodiment the default state is set to disabled. In still another embodiment, illustrated in Fig. 4B, the default state is using a strapping option. The default value is provided to the MLE reset logic 465 through the pull-up (or pull-down) resistor 445.

Turning now to Fig. 5, a method 500 for initiating the HDT mode is shown. In response to receiving a request to enter the HDT mode (step 505), the HDT control logic 410 checks the status of the one or more HDT enable bits to see if the HDT mode is enabled or disabled (step 510). If the HDT mode is enabled (step 515), then the HDT control logic 410 initiates the HDT mode (step 520). If the HDT mode is disabled (step 515), then the HDT control logic 410 will not initiate the HDT mode.

Turning now to Fig. 6, a method 600 for changing the HDT mode enable status, which includes an HDT mode lock, is shown. In response to receiving a request to enter the HDT mode (step 605), the HDT control logic 410 checks the status of the one or more HDT enable lock bits to determine if the HDT lock mode is locked or unlocked (step 610). If the HDT lock mode is unlocked (step 615), then the HDT control logic 410 initiates HDT mode (step 635). If the HDT lock mode is locked (step 615), then the HDT control logic 410 requests authorization to change the HDT lock mode status (step 620). If the change is authorized (step 625), then the HDT control logic 410 changes the HDT mode lock bit to unlocked (step 630). If the change is not authorized (step 625), then the HDT control logic 410 does not change the HDT mode lock bit.

In various embodiments, the HDT enable status may be changed by setting or resetting the one or more HDT enable status bits. For example, the HDT mode may be disabled, but inside SMM, a predetermined input to the HDT control logic 410 may signal the HDT control logic 410 to change the HDT mode status to enabled. In the embodiment of Fig. 4A, for example, once signaled, the HDT control logic 410 would change the status of the HDT enable bit from disabled to enabled.

Referring back to the embodiment of Fig. 4B, for example, in response to receiving a request to change the HDT mode status, the HDT control logic 410 checks the status of the one or more HDT enable lock bits to see if the HDT lock mode is enabled or disabled. If the HDT lock mode is disabled, then the HDT control logic 410 may change the HDT mode status. If the HDT lock mode is enabled, then the HDT control logic 410 will not change the HDT mode status.

It is noted that the method 600 may alternatively terminate if the HDT lock mode is locked (step 615), instead of requesting authorization to change the HDT lock mode status (step 620). The method 600 may also include receiving a request to change the HDT lock mode status (not shown) prior to the method 600 requesting authorization (step 620).

Turning now to Fig. 7, a method 700 for initiating the microcode loader is shown. In response to receiving a request to initiate the microcode update mode (step 705), the microcode control logic 455 checks the status of the one or more microcode enable bits to see if microcode update mode is enabled or disabled (step 710). If the microcode update mode is enabled (step 715), then the microcode control logic 455 initiates the microcode update mode (step 720). If the microcode update mode is disabled (step 715), then the microcode control logic 455 will not initiate the microcode update mode.

Turning now to Fig. 8, a method 800 for changing the microcode update mode enable status, which includes a microcode mode lock, is shown. In response to receiving a request to enter the microcode mode (step 805), the microcode control logic 455 checks the status of the one or more microcode enable lock bits to see if the microcode mode is locked or unlocked (step 810). If the microcode lock mode is unlocked (step 815), then the microcode control logic 455 initiates the microcode mode (step 835). If the microcode lock mode is locked (step 815), then the microcode control logic 455 requests authorization to change the microcode mode lock status (step 820). If the change is authorized (step 825), then the microcode control logic 455 changes the microcode mode lock bit to unlocked (step 830). If the change is not authorized (step 825), then the microcode control logic 455 does not change the microcode mode lock bit.

In various embodiments, the microcode enable status may be changed by setting or resetting the one or more microcode enable status bits. For example, the microcode mode may be disabled, but inside SMM, a predetermined input to the microcode control logic 455 may signal the microcode control logic 455 to change the microcode mode status to enabled. In the embodiment of Fig. 4C, for example, once signaled, the microcode control logic 455 will change the status of the one or more microcode enable bits from disabled to enabled.

In response to receiving a request to change the microcode mode status, the microcode control logic 455 may check the status of the one or more microcode enable lock bits to determine if the microcode lock mode is enabled or disabled. If the microcode lock mode is disabled, then the microcode control logic 455 may change the microcode mode status. If the microcode lock mode is enabled, then the microcode control logic 455 will not change the microcode mode status.

It is noted that the method 800 may alternatively terminate if the microcode update lock status is locked (step 815), instead of requesting authorization to change the microcode update lock status (step 820). The method 800 may also include receiving a request to change the microcode update lock status (not shown) prior to the method 3500 requesting authorization (step 820).

For the purposes of this disclosure, references to ROM are to be construed as also applying to flash memory and other substantially non-volatile memory types. Note that while the methods of the present invention disclosed herein have been illustrated as flowcharts, various elements of the flowcharts may be omitted or performed in different order in various embodiments. Note also that the methods of the present invention disclosed herein admit to variations in implementation.

Some aspects of the invention as disclosed above may be implemented in hardware or software. Thus, some portions of the detailed descriptions herein are consequently presented in terms of a hardware implemented process and some portions of the detailed descriptions herein are consequently presented in terms of a software-implemented process involving symbolic representations of operations on data bits within a memory of a computing system or computing device. These descriptions and representations are the means used by those in the art to convey most effectively the substance of their work to others skilled in the art using both hardware and software. The process and operation of both require physical manipulations of physical quantities. In software, usually, though not necessarily, these quantities take the form of electrical, magnetic, or optical signals capable of being stored, transferred, combined, compared, and otherwise manipulated. It has proven convenient at times, principally for reasons of common usage, to refer to these signals as bits, values, elements, symbols, characters, terms, numbers, or the like.

It should be borne in mind, however, that all of these and similar terms are to be associated with the appropriate physical quantities and are merely convenient labels applied to these quantifies. Unless specifically stated or otherwise as may be apparent, throughout the present disclosure, these descriptions refer to the action and processes of an electronic device, that manipulates and transforms data represented as physical (electronic, magnetic, or optical) quantities within some electronic device's storage into other data similarly represented as physical quantities within the storage, or in transmission or display devices. Exemplary of the terms denoting such a description are, without limitation, the terms "processing," "computing," "calculating," "determining," "displaying," and the like.

Note also that the software-implemented aspects of the invention are typically encoded on some form of program storage medium or implemented over some type of transmission medium. The program storage medium may be magnetic (*e.g.*, a floppy disk or a hard drive) or optical (*e.g.*, a compact disk read only memory, or "CD ROM"), and may be read only or random access. Similarly, the transmission medium may be twisted wire pairs, coaxial cable, optical fiber, or some other suitable transmission medium known to the art. The invention is not limited by these aspects of any given implementation.

The particular embodiments disclosed above are illustrative only, as the invention may be modified and practiced in different but equivalent manners apparent to those skilled in the art having the benefit of the teachings herein. Furthermore, no limitations are intended to the details of construction or design herein shown, other than as described in the claims below. It is therefore evident that the particular embodiments disclosed above may be altered or modified and all such variations are considered within the scope of the invention. Accordingly, the protection sought herein is as set forth in the claims below.

CLAIMS

1. A processor, comprising:

a first register configured to store one or more hardware-debug-test (HDT) enable bits;

a first control logic coupled to receive a plurality of HDT input signals, wherein the first control logic is coupled to access the first register; and

a second control logic coupled to the first register, wherein the second control logic is configured to store one or more default values in the first register in response to a reset of the processor.

2. The processor of claim 1, wherein the first control logic is further configured to receive a request to enter an HDT mode, wherein the first control logic is further configured to read selected entries of the one or more HDT enable bits stored in the first register in response to the request to enter HDT mode, and wherein the first control logic is further configured to grant or deny the request to enter HDT mode based on the selected entries of the one or more HDT enable bits.

3. The processor of claim 1, further comprising:

one or more non-volatile memory cells configured to store the one or more default values for the one or more HDT enable bits, wherein the second control logic is further coupled to read the one or more default values for the one or more HDT enable bits from the one or more non-volatile memory cells and to write the one or more default values for the one or more HDT enable bits into the first register in response to the reset of the processor.

4. The processor of claim 1, wherein the second control logic is further coupled to receive a signal indicative of the one or more default values for the one or more HDT enable bits and to write the one or more default values for the one or more HDT enable bits into the first register in response to the reset of the processor.

5. The processor of claim 1, further comprising:

a third register configured to store one or more microcode loader enable bits;

a third control logic coupled to receive a plurality of microcode inputs, wherein the third control logic is coupled to access the third register; and

a fourth control logic coupled to the third register, wherein the fourth control logic is configured to store one or more default values in the third register in response to a reset of the processor.

6. The processor of claim 5, wherein the third control logic is further configured to receive a request to modify microcode, wherein the third control logic is further configured to read selected entries of the one or more microcode loader enable bits stored in the third register in response to the request to modify microcode, and wherein the third control logic is further configured to grant or deny the request to modify microcode based on the selected entries of the one or more microcode loader enable bits.

7. The processor of claim 1, further comprising:

a second register coupled to the first control logic, wherein the second register is configured to store one or more HDT enable lock bits.

8. The processor of claim 7, wherein the first control logic is further configured to receive a request to modify HDT mode status, wherein the first control logic is further configured to read selected entries in the one or more HDT enable lock bits stored in the second register in response to the request to modify HDT mode status, and wherein the first control logic is further configured to grant or deny the request to modify HDT mode based on the selected entries in the one or more HDT enable lock bits.

9. A processor, comprising:

a first control logic coupled to receive a plurality of microcode inputs;

a first register coupled to the first control logic, wherein the first register is configured to store one or more microcode loader enable bits; and

a second control logic coupled to the first register, wherein the second control logic is configured to store one or more default values in the first register in response to a reset of the processor.

10. The processor of claim 9, wherein the first control logic is further configured to receive a request to modify microcode, wherein the first control logic is further configured to read selected entries of the one or more microcode loader enable bits stored in the first register in response to the request to modify microcode, and wherein the first control logic is further configured to grant or deny the request to modify microcode based on the selected entries of the one or more microcode loader enable bits.

11. The processor of claim 9, further comprising:

one or more non-volatile memory cells configured to store the one or more default values for the one or more microcode loader enable bits, wherein the second control logic is further coupled to read the one or more default values for the one or more microcode loader enable bits from the one or more non-volatile memory cells and to write the one or more default values for the one or more microcode loader enable bits into the microcode loader register in response to the reset of the processor.

12. The processor of claim 9, wherein the second control logic is further coupled to receive a signal indicative of the one or more default values for the one or more microcode loader enable bits and to write the one or more default values for the one or more microcode loader enable bits into the first register in response to the reset of the processor.

13. The processor of claim 9, further comprising:

a second register coupled to the first control logic, wherein the second register is configured to store one or more microcode loader enable lock bits.

14. The processor of claim 13, wherein the first control logic is further configured to receive a request to modify microcode loader lock status, wherein the first control logic is further configured to read

selected entries in the one or more microcode loader enable lock bits stored in the second register in response to the request to modify microcode loader lock status, and wherein the first control logic is further configured to grant or deny the request to modify microcode loader lock status based on the selected entries in the one or more microcode loader enable lock bits.

15. A method for determining an HDT mode enable status, the method comprising:
receiving a request to initiate the HDT mode;
determining HDT mode enable status;
initiating the HDT mode if the HDT mode enable status is set to enabled.

16. The method of claim 15, wherein determining HDT mode enable status comprises reading one or more entries corresponding to one or more HDT enable bits from a register.

17. A method for modifying microcode, the method comprising:
receiving a request to modify microcode;
determining microcode loader enable status;
modifying microcode if the microcode loader enable status is set to enabled.

18. The method of claim 17, wherein determining microcode loader enable status comprises reading one or more entries corresponding to one or more microcode loader enable bits from a register.

19. A method of changing HDT mode status, the method comprising:
receiving a request to change HDT mode status;
determining HDT mode enable lock status; and
modifying HDT mode status if the HDT mode enable lock status is set to unlocked.

20. The method of claim 19, wherein determining HDT mode enable lock status comprises reading one or more entries corresponding to one or more HDT enable lock bits from a register.

21. The method of claim 19, wherein modifying HDT mode status comprises writing one or more entries corresponding to one or more HDT enable bits to a register.

22. A method of changing microcode loader enable status, the method comprising:
receiving a request to change microcode loader enable status;
determining microcode loader enable lock status; and
modifying microcode loader enable status if the microcode loader enable lock status is set to unlocked.

23. The method of claim 22, wherein determining microcode loader enable lock status comprises reading one or more entries corresponding to one or more microcode loader enable lock bits from a register.

24. The method of claim 22, wherein modifying microcode loader enable status comprises writing one or more entries corresponding to one or more microcode loader enable bits to a register.

25. A method of operating a processor, the method comprising:

obtaining one or more default values, wherein obtaining the one or more default values is selected from the group consisting of:

reading the one or more default values from one or more non-volatile memory cells, and

receiving the one or more default values as a strapped value through a pull-up or pull-down resistor; and

writing the one or more default values as one or more various entries in one or more registers in response to a reset of the processor, wherein the one or more various entries are selected from the group consisting of:

one or more HDT enable bits,

one or more HDT enable lock bits,

one or more microcode loader enable bits, and

one or more microcode loader enable lock bits.

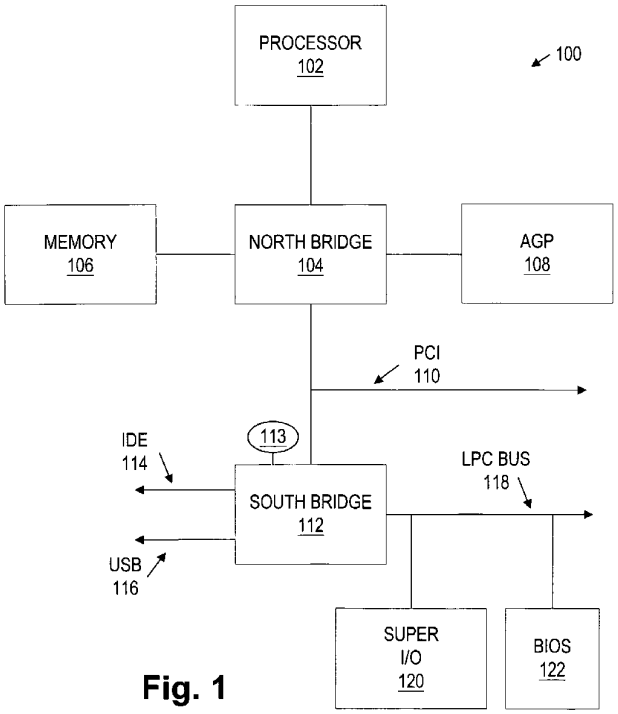


Fig. 1
(Prior Art)

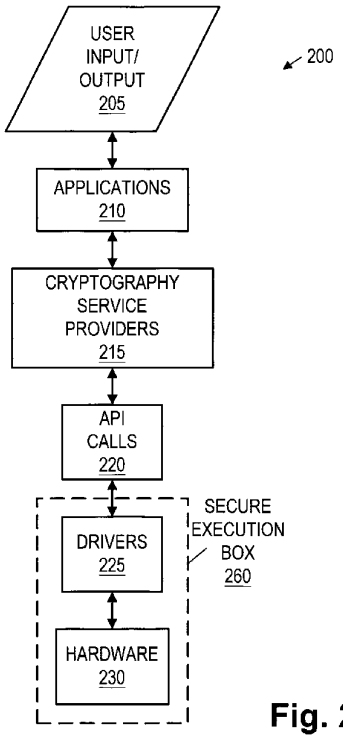

USER INPUT/ OUTPUT 205
200
APPLICATIONS 210
CRYPTOGRAPHY SERVICE PROVIDERS 215
API CALLS 220
SECURE EXECUTION BOX 260
DRIVERS 225
HARDWARE 230

Fig. 2

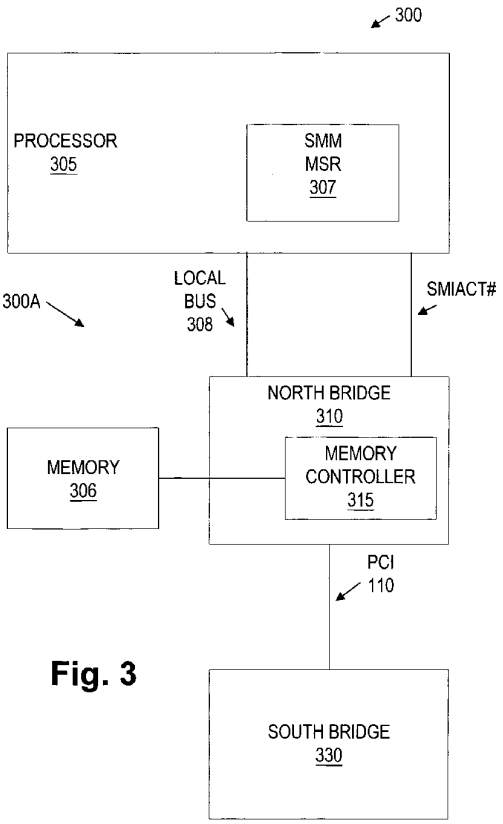

300
PROCESSOR
305
SMM
MSR
307
300A
LOCAL
BUS
308
SMIACT#
NORTH BRIDGE
310
MEMORY
306
MEMORY
CONTROLLER
315
PCI
110
SOUTH BRIDGE
330

Fig. 3

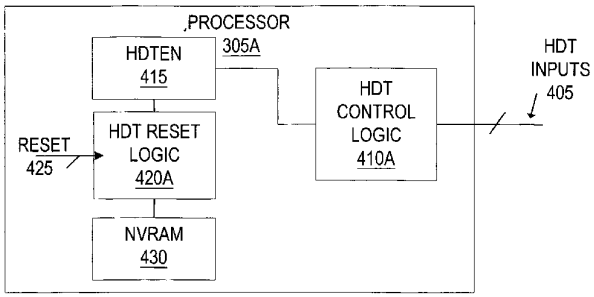

PROCESSOR
305A
HDTEN
415
HDT RESET
LOGIC
420A
NVRAM
430
RESET
425
HDT
CONTROL
LOGIC
410A
HDT
INPUTS
405

Fig. 4A

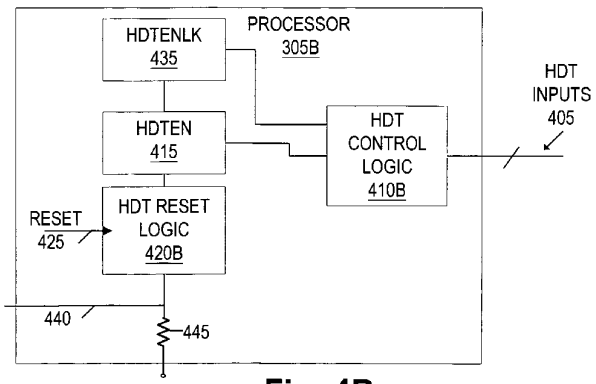

HDTENLK
435
PROCESSOR
305B
HDTEN
415
HDT RESET
LOGIC
420B
RESET
425
HDT
CONTROL
LOGIC
410B
HDT
INPUTS
405
440
445

Fig. 4B

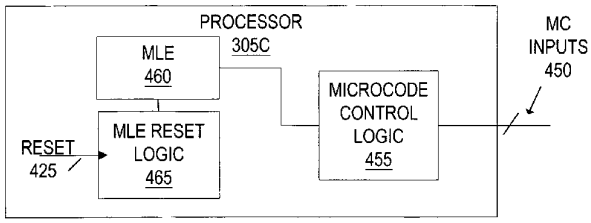

PROCESSOR
305C
MLE
460
MLE RESET
LOGIC
465
RESET
425
MICROCODE
CONTROL
LOGIC
455
MC
INPUTS
450

Fig. 4C

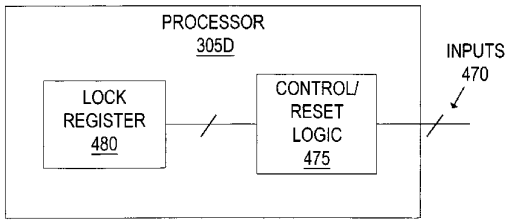

PROCESSOR
305D
LOCK
REGISTER
480
CONTROL/
RESET
LOGIC
475
INPUTS
470

Fig. 4D

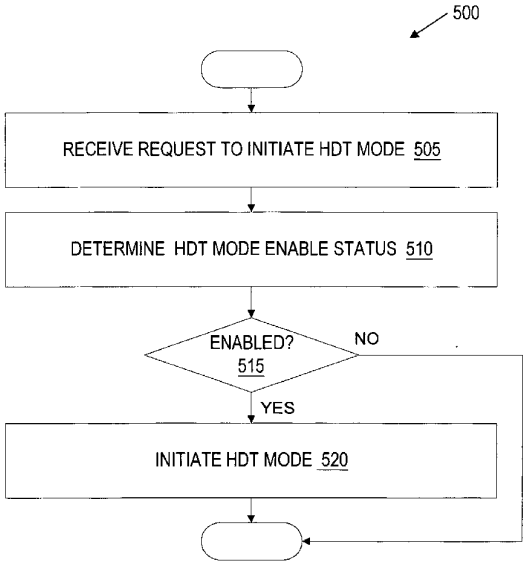

500
RECEIVE REQUEST TO INITIATE HDT MODE 505
DETERMINE HDT MODE ENABLE STATUS 510
ENABLED? 515
NO
YES
INITIATE HDT MODE 520

Fig. 5

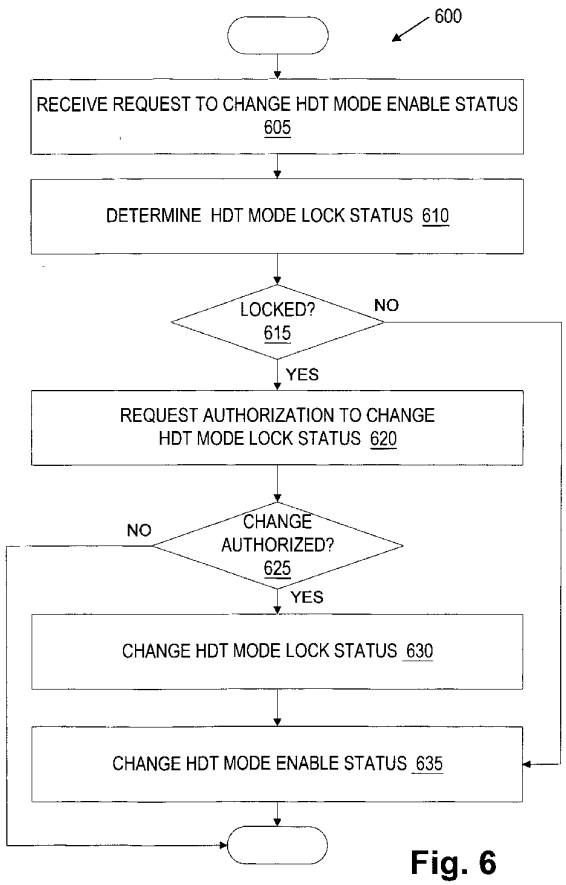

600
RECEIVE REQUEST TO CHANGE HDT MODE ENABLE STATUS 605
DETERMINE HDT MODE LOCK STATUS 610
LOCKED? 615
NO
YES
REQUEST AUTHORIZATION TO CHANGE HDT MODE LOCK STATUS 620
NO
CHANGE AUTHORIZED? 625
YES
CHANGE HDT MODE LOCK STATUS 630
CHANGE HDT MODE ENABLE STATUS 635

Fig. 6

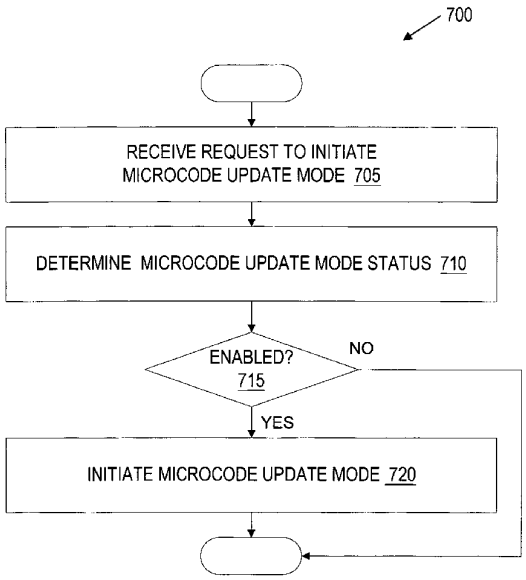

700
RECEIVE REQUEST TO INITIATE MICROCODE UPDATE MODE 705
DETERMINE MICROCODE UPDATE MODE STATUS 710
ENABLED? 715
NO
YES
INITIATE MICROCODE UPDATE MODE 720

Fig. 7

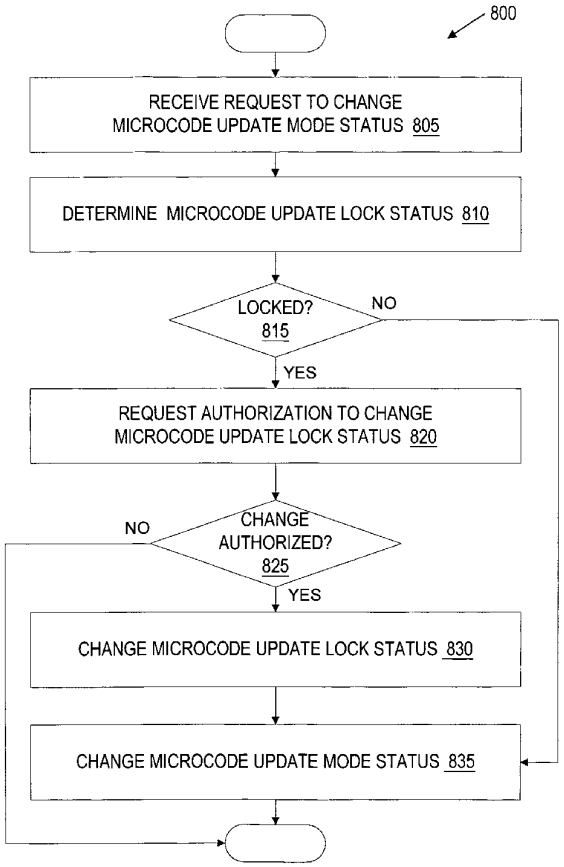

800
RECEIVE REQUEST TO CHANGE MICROCODE UPDATE MODE STATUS 805
DETERMINE MICROCODE UPDATE LOCK STATUS 810
LOCKED? 815
NO
YES
REQUEST AUTHORIZATION TO CHANGE MICROCODE UPDATE LOCK STATUS 820
NO
CHANGE AUTHORIZED? 825
YES
CHANGE MICROCODE UPDATE LOCK STATUS 830
CHANGE MICROCODE UPDATE MODE STATUS 835

Fig. 8

【国際公開パンフレット（コレクトバージョン）】

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
21 November 2002 (21.11.2002)

PCT

(10) International Publication Number
WO 2002/093336 A3

(51) International Patent Classification[7]: G06F 1/00

(21) International Application Number:
PCT/US2002/011935

(22) International Filing Date: 17 April 2002 (17.04.2002)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:

09/852,942	10 May 2001 (10.05.2001)	US
09/852,372	10 May 2001 (10.05.2001)	US
09/853,226	11 May 2001 (11.05.2001)	US

(71) Applicant: ADVANCED MICRO DEVICES, INC. [US/US]; One AMD Place, P.O. Box 9453, Sunnyvale, CA 94088-3453 (US).

(72) Inventor: STRONGIN, Geoffrey, S.; 7210 Montana Norte, Austin, TX 78731 (US).

(74) Agent: DRAKE, Paul, S.; Advanced Micro Devices, Inc., 5204 East Ben White Boulevard, M/S 562, Austin, TX 78741 (US).

(81) Designated States (national): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, OM, PH, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZM, ZW.

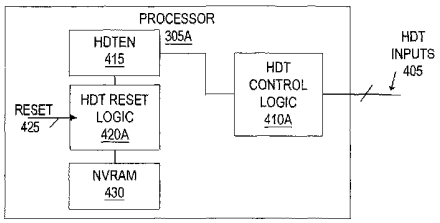
(84) Designated States (regional): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— with international search report

(88) Date of publication of the international search report:
12 February 2004

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: PROCESSOR WITH PROTECTED TEST AND DEBUG MODE



(57) Abstract: Methods, devices, and systems for closing back door access mechanisms. A processor includes a first register configured to store one or more hardware-debug-test (HDT) enable bits, a first control logic coupled to receive a plurality of HDT input signals, and a second control logic coupled to the first register. The first control logic is coupled to access the first register. The second control logic is configured to store one or more default values in the first register in response to a reset of the processor. Another processor includes a first control logic coupled to receive a plurality of microcode inputs, a first register coupled to the first control logic, and a second control logic coupled to the first register. The first register is configured to store one or more microcode loader enable bits. The second control logic is configured to store one or more default values in the first register in response to a reset of the processor.

WO 2002/093336 A3

【国際調査報告】

INTERNATIONAL SEARCH REPORT

Interna[illegible]l Application No
PCT/US 02/11935

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 G06F1/00

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 G06F G11C

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)
EPO-Internal

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	INTEL: "Intel 82802AB/82802AC Firmware Hub" DATASHEET INTEL, 'Online! 30 November 2000 (2000-11-30), page 1-6,17-27 XP002257561 Retrieved from the Internet: <URL:http://www.intel.com/design/chipsets/datashts/29065804.pdf> 'retrieved on 2003-10-13!	1,2, 7-10, 13-24
A	page 17, paragraphs 3.4,-,RESET page 23, paragraph 4.9 -page 26, paragraph 4.9.2	3-6,11, 12,25

-/--

[X] Further documents are listed in the continuation of box C. [X] Patent family members are listed in annex.

* Special categories of cited documents :

'A' document defining the general state of the art which is not considered to be of particular relevance
'E' earlier document but published on or after the international filing date
'L' document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
'O' document referring to an oral disclosure, use, exhibition or other means
'P' document published prior to the international filing date but later than the priority date claimed

'T' later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
'X' document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
'Y' document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.
'&' document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
13 October 2003	03/11/2003

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL – 2280 HV Rijswijk Tel. (+31–70) 340–2040, Tx. 31 651 epo nl, Fax: (+31–70) 340–3016	Alecu, M

Form PCT/ISA/210 (second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

Interna / Application No
PCT/ US 02/11935

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication,where appropriate, of the relevant passages	Relevant to claim No.
X	US 5 826 007 A (MORI KIMIO ET AL) 20 October 1998 (1998-10-20) column 1, line 26 - line 36 column 1, line 49 - line 63 column 2, line 42 -column 3, line 6 column 4, line 42 - line 61 column 5, line 13 - line 33 column 5, line 63 -column 6, line 13	15,16
X. A	US 6 154 819 A (BROWN CHARLES ET AL). 28 November 2000 (2000-11-28) column 4, line 48 - line 63 column 7, line 37 - line 54 column 10, line 1 - line 41 column 6, line 12 - line 48	17,18, 22-24 25
A	US 6 026 016 A (GAFKEN ANDREW H) 15 February 2000 (2000-02-15) the whole document	1-25

Form PCT/ISA/210 (continuation of second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

tion on patent family members

Interna Application No
PCT/US 02/11935

Patent document cited in search report		Publication date	Patent family member(s)		Publication date
US 5826007	A	20-10-1998	JP 9198316 A		31-07-1997
			CN 1162150 A ,B		15-10-1997
			KR 246873 B1		15-03-2000
			TW 464804 B		21-11-2001
US 6154819	A	28-11-2000	NONE		
US 6026016	A	15-02-2000	AU 3672999 A		29-11-1999
			GB 2353617 A ,B		28-02-2001
			TW 440848 B		16-06-2001
			WO 9959288 A1		18-11-1999

Form PCT/ISA/210 (patent family annex) (July 1992)

フロントページの続き

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

（特許庁注：以下のものは登録商標）
フロッピー

(74)代理人 100108604
弁理士 村松 義人
(72)発明者 ジェフリー エス. ストロンジン
アメリカ合衆国、テキサス州 ７８７３１、オースティン、モンタナ ノート ７２１０

Ｆターム(参考) 5B017 AA01 BA04 CA15 CA16
5B033 FA18 FA22 FA27
5B048 AA04 FF01