(19)日本国特許庁(JP)　(12)公表特許公報(A)　(11)公表番号
特表2025-514826
(P2025-514826A)
(43)公表日　令和7年5月9日(2025.5.9)

(51)国際特許分類　FI
G06F 21/52 (2013.01)　G06F 21/52
G06F 9/48 (2006.01)　G06F 9/48 100H

審査請求　未請求　予備審査請求　未請求　（全37頁）

(21)出願番号　特願2024-562301(P2024-562301)
(86)(22)出願日　令和5年3月17日(2023.3.17)
(85)翻訳文提出日　令和6年10月22日(2024.10.22)
(86)国際出願番号　PCT/GB2023/050653
(87)国際公開番号　WO2023/209323
(87)国際公開日　令和5年11月2日(2023.11.2)
(31)優先権主張番号　2206164.2
(32)優先日　令和4年4月28日(2022.4.28)
(33)優先権主張国・地域又は機関
英国(GB)
(81)指定国・地域　AP(BW,CV,GH,GM,KE,LR,LS,MW,MZ,NA,RW,SD,SL,ST,SZ,TZ,UG,ZM,ZW),EA(AM,AZ,BY,KG,KZ,RU,TJ,TM),EP(AL,AT,BE,BG,CH,CY,CZ,DE,DK,EE,ES,FI,FR,GB,GR,HR,HU,IE,IS,IT,LT,LU,LV
最終頁に続く

(71)出願人　500395107
アーム・リミテッド
グレート・ブリテン及び北部アイルランド連合王国、イングランド、 シー・ビー１、９エヌ・ジェイ、ケンブリッジ、フルボーン・ロード １１０
(74)代理人　110000855
弁理士法人浅村特許事務所
(72)発明者　クラスケ、サイモン ジョン
グレート・ブリテン及び北部アイルランド連合王国 シービー１ ９エヌジェイ ケンブリッジシャー ケンブリッジ フルボーン ロード １１０ アーム リミテッド宛
(72)発明者　ホーリー、ジョン マイケル
最終頁に続く

(54)【発明の名称】　例外復帰状態ロックパラメータ

(57)【要約】
装置は、例外復帰状態レジスタ記憶装置と、処理回路と、を備える。処理回路は、保護制御スタック（ＧＣＳ）例外復帰状態プッシュ命令に応答して、例外復帰状態レジスタ記憶装置から例外復帰状態情報を取得し、状態情報をＧＣＳデータ構造にプッシュする。処理回路は、ＧＣＳ例外復帰状態ポップ命令に応答して、ＧＣＳ保護例外復帰状態情報をＧＣＳデータ構造から取得する。処理回路は、少なくとも１つの動作状態において、例外復帰状態レジスタ記憶装置に記憶された例外復帰状態情報を修正しようとする試みに応答して、例外復帰状態ロックパラメータがロック状態にあるか、又はロック解除状態にあるかを検出し、ロック状態にあるときにはフォールトをシグナリングする。

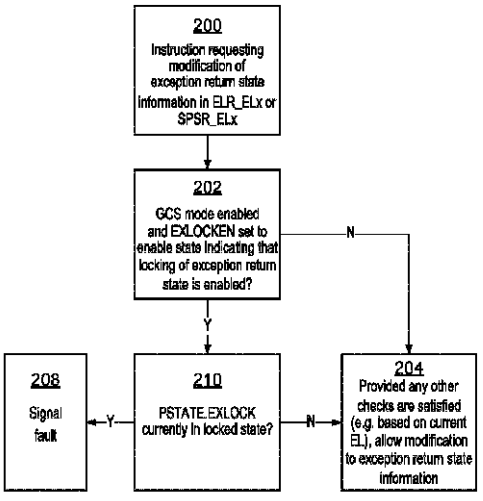


FIG. 13

【特許請求の範囲】
【請求項１】
装置であって、
例外復帰状態情報を記憶するための例外復帰状態レジスタ記憶装置と、
命令に応答して処理動作を実行するための処理回路と、を備え、
前記処理回路は、保護制御スタック（ＧＣＳ）例外復帰状態プッシュ命令に応答して、前記例外復帰状態レジスタ記憶装置から前記例外復帰状態情報を取得し、前記例外復帰状態情報をＧＣＳデータ構造にプッシュするように構成されており、
前記処理回路は、ＧＣＳ例外復帰状態ポップ命令に応答して、ＧＣＳ保護例外復帰状態情報を前記ＧＣＳデータ構造から取得するように構成されており、
前記装置は、ターゲットアドレスを指定する書き込み要求を、前記ターゲットアドレスに対応するメモリ領域が前記ＧＣＳデータ構造を記憶するためのＧＣＳ領域として指定され、前記書き込み要求はＧＣＳアクセスタイプの命令の制限されたサブセットであって、前記ＧＣＳアクセスタイプの命令の制限されたサブセットは、少なくとも前記ＧＣＳ例外復帰状態プッシュ命令を含む、前記ＧＣＳアクセスタイプの命令の制限されたサブセットのうちの１つによってトリガされたＧＣＳメモリアクセス要求以外の要求であると判定したことに応答して拒否するためのメモリ管理回路を備え、
前記処理回路は、前記処理回路の少なくとも１つの動作状態において、前記例外復帰状態レジスタ記憶装置に記憶された前記例外復帰状態情報を修正しようとする試みに応答して、例外復帰状態ロックパラメータがロック状態にあるか、又はロック解除状態にあるかを検出し、前記例外復帰状態ロックパラメータが前記ロック状態にあると検出すると、フォールトをシグナリングするように構成されている、装置。
【請求項２】
前記処理回路は、異なる特権レベルに関連付けられた複数の例外レベルのうちの１つで前記処理動作を実行するように構成されており、
前記処理回路は、前記処理回路の前記少なくとも１つの動作状態において、例外が取られる前の現在の例外レベルと同じ例外レベルで前記例外が取られるべきである前記例外が取られたことに応答して、前記例外復帰状態ロックパラメータを前記ロック状態に設定するように構成されている、請求項１に記載の装置。
【請求項３】
前記処理回路は、前記処理回路の前記少なくとも１つの動作状態において、前記ＧＣＳ例外復帰状態プッシュ命令に応答して、前記例外復帰状態ロックパラメータを前記ロック解除状態に設定するように構成されている、請求項１～２のいずれか一項に記載の装置。
【請求項４】
前記処理回路は、前記処理回路の前記少なくとも１つの動作状態において、前記ＧＣＳ例外復帰状態ポップ命令に応答して、前記例外復帰状態ロックパラメータを前記ロック状態に設定するように構成されている、請求項１～３のいずれか一項に記載の装置。
【請求項５】
前記処理回路は、前記処理回路の前記少なくとも１つの動作状態において、前記例外復帰状態レジスタ記憶装置に記憶された前記例外復帰状態情報に基づいて、例外復帰が前記例外復帰前の現在の例外レベルと同じ例外レベルでの処理への復帰である、前記例外復帰をトリガするための例外復帰命令に応答して、前記例外復帰状態ロックパラメータが前記ロック状態に設定されているか否かをチェックし、前記例外復帰状態ロックパラメータが前記ロック解除状態に設定されていることを検出することに応答してフォールトをシグナリングするように構成されている、請求項１～４のいずれか一項に記載の装置。
【請求項６】
前記処理回路は、前記処理回路の前記少なくとも１つの動作状態において、前記例外復帰状態ロックパラメータが前記ロック状態に設定されているときに、前記例外復帰状態レジスタ記憶装置内の情報に対する修正を要求する命令を実行しようとする試みに応答して、前記例外復帰状態レジスタ記憶装置に書き込まれるべき新しい値が前記例外復帰状態レ

ジスタ記憶装置に以前に記憶された古い値と同じであるか、異なるかにかかわらず、フォールトをシグナリングするように構成されている、請求項１～５のいずれか一項に記載の装置。

【請求項７】

前記処理回路は、前記処理回路の前記少なくとも１つの動作状態において、前記例外復帰状態ロックパラメータが前記ロック状態に設定されているときに、前記例外復帰状態レジスタ記憶装置内の情報に対する修正を要求する命令を実行しようとする試みに応答して、前記例外復帰状態レジスタ記憶装置に書き込まれるべき新しい値が前記例外復帰状態レジスタ記憶装置に以前に記憶された古い値と同じであるか、異なるかをチェックし、前記新しい値が前記古い値とは異なると判定すると、前記フォールトをシグナリングするように構成されている、請求項１～５のいずれか一項に記載の装置。

【請求項８】

前記処理回路は、前記ＧＣＳ例外復帰状態ポップ命令に応答して、前記例外復帰状態レジスタ記憶装置から取得した、意図した例外復帰状態情報と、前記ＧＣＳデータ構造から取得した、前記ＧＣＳ保護例外復帰状態情報との間で不一致が検出されると、フォールトをシグナリングするように構成されている、請求項１～７のいずれか一項に記載の装置。

【請求項９】

前記例外復帰状態レジスタ記憶装置は、例外からの復帰後に実行されるべき命令の例外復帰アドレスを記憶するための例外復帰アドレスレジスタを備える、請求項１～８のいずれか一項に記載の装置。

【請求項１０】

前記例外復帰状態レジスタ記憶装置は、例外からの復帰後に復元されるべきプロセッサ状態を記憶するための保存プロセッサ状態レジスタを備える、請求項１～９のいずれか一項に記載の装置。

【請求項１１】

前記処理回路は、例外が取られることに応答して、プロセッサ状態レジスタから前記保存プロセッサ状態レジスタにプロセッサ状態をコピーするように構成されており、

前記例外復帰状態ロックパラメータは、前記プロセッサ状態レジスタにおいて指定されるパラメータである、請求項１０に記載の装置。

【請求項１２】

前記処理回路は、例外復帰命令に応答して、保存例外復帰状態ロックパラメータを含む保存プロセッサ状態を前記保存プロセッサ状態レジスタから前記プロセッサ状態レジスタにコピーして、前記例外復帰状態ロックパラメータを、前記保存例外復帰状態ロックパラメータによって示される値に復元するように構成されている、請求項１１に記載の装置。

【請求項１３】

前記処理回路の前記少なくとも１つの動作状態は、例外復帰状態ロック有効化制御パラメータが、前記例外復帰状態ロックパラメータを使用して例外復帰状態情報のロックが有効であることを示す有効状態に設定されている動作状態を含む、請求項１～１２のいずれか一項に記載の装置。

【請求項１４】

前記処理回路の前記少なくとも１つの動作状態は、

例外復帰状態ロック有効化制御パラメータが、前記例外復帰状態ロックパラメータを使用して例外復帰状態情報のロックが有効であることを示す有効状態に設定されており、かつ、

ＧＣＳ有効化パラメータが、前記ＧＣＳデータ構造の使用をサポートするためのＧＣＳモードが有効であることを示す有効状態に設定されている動作状態を含む、請求項１～１２のいずれか一項に記載の装置。

【請求項１５】

前記メモリ管理回路は、所与のターゲットアドレスを指定するＧＣＳメモリアクセス要求を、前記所与のターゲットアドレスに対応するメモリ領域が前記ＧＣＳ領域として指定

されていないと判定したことに応答して拒否するように構成されている、請求項１～１４のいずれか一項に記載の装置。
【請求項１６】
方法であって、
例外復帰状態情報を例外復帰状態レジスタ記憶装置に記憶することと、
処理回路を使用して、命令に応答して処理動作を実行することと、
保護制御スタック（ＧＣＳ）例外復帰状態プッシュ命令に応答して、前記例外復帰状態レジスタ記憶装置から前記例外復帰状態情報を取得し、前記例外復帰状態情報をＧＣＳデータ構造にプッシュすることと、
ＧＣＳ例外復帰状態ポップ命令に応答して、ＧＣＳ保護例外復帰状態情報を前記ＧＣＳデータ構造から取得することと、
ターゲットアドレスを指定する書き込み要求を、前記ターゲットアドレスに対応するメモリ領域が前記ＧＣＳデータ構造を記憶するためのＧＣＳ領域として指定され、前記書き込み要求は、ＧＣＳアクセスタイプの命令の制限されたサブセットであって、前記ＧＣＳアクセスタイプの命令の制限されたサブセットは、少なくとも前記ＧＣＳ例外復帰状態プッシュ命令を含む、前記ＧＣＳアクセスタイプの命令の制限されたサブセットのうちの１つによってトリガされたＧＣＳメモリアクセス要求以外の要求であると判定したことに応答して拒否することと、
前記処理回路の少なくとも１つの動作状態において、前記例外復帰状態レジスタ記憶装置に記憶された前記例外復帰状態情報を修正しようとする試みに応答して、例外復帰状態ロックパラメータがロック状態にあるか、又はロック解除状態にあるかを検出し、前記例外復帰状態ロックパラメータが前記ロック状態にあると検出すると、フォールトをシグナリングすることと、を含む方法。
【請求項１７】
ホストデータ処理装置によって実行されると、前記ホストデータ処理装置を制御して、ターゲットコードを実行するための命令実行環境を提供する命令を含むコンピュータプログラムであって、前記コンピュータプログラムは、
前記ホストデータ処理装置を制御して、例外復帰状態情報を記憶するための例外復帰状態レジスタ記憶装置をシミュレートするレジスタエミュレートプログラムロジックと、
前記ホストデータ処理装置を制御して、前記ターゲットコードの命令の実行をシミュレートする処理プログラムロジックと、を備え、
前記処理プログラムロジックは、前記ターゲットコードの保護制御スタック（ＧＣＳ）例外復帰状態プッシュ命令に応答して、前記ホストデータ処理装置を制御して、シミュレートされた前記例外復帰状態レジスタ記憶装置から前記例外復帰状態情報を取得し、前記例外復帰状態情報をＧＣＳデータ構造にプッシュするように構成されており、
前記処理プログラムロジックは、前記ターゲットコードのＧＣＳ例外復帰状態ポップ命令に応答して、前記ＧＣＳデータ構造から取得したＧＣＳ保護例外復帰状態情報を取得するように構成されており、
前記コンピュータプログラムは、ターゲットアドレスを指定する書き込み要求を、前記ターゲットアドレスに対応するメモリ領域が前記ＧＣＳデータ構造を記憶するためのＧＣＳ領域として指定され、前記書き込み要求はＧＣＳアクセスタイプの命令の制限されたサブセットであって、前記ＧＣＳアクセスタイプの命令の制限されたサブセットは、少なくとも前記ＧＣＳ例外復帰状態プッシュ命令を含む、前記ＧＣＳアクセスタイプの命令のうちの１つによってトリガされたＧＣＳメモリアクセス要求以外の要求であると判定したことに応答して拒否するためのメモリ管理プログラムロジックを備え、
前記処理プログラムロジックは、前記処理プログラムロジックの少なくとも１つのシミュレートされた動作状態において、前記ホストデータ処理装置を制御して、シミュレートされた前記例外復帰状態レジスタ記憶装置に記憶された前記例外復帰状態情報を修正しようとする試みに応答して、例外復帰状態ロックパラメータがロック状態にあるか、又はロック解除状態にあるかを検出し、前記例外復帰状態ロックパラメータが前記ロック状態に

あると検出すると、フォールトをシグナリングするように構成されている、コンピュータプログラム。
【請求項１８】
請求項１７に記載のコンピュータプログラムを記憶する記憶媒体。
【発明の詳細な説明】
【技術分野】
【０００１】
本技法は、データ処理の分野に関する。
【０００２】
データ処理装置は、例外を取ったことに応答して例外復帰状態情報を保存することができる。例外復帰状態情報は、例外を取ったときのプロセッサの状態を示し、例外処理が終了すると、プロセッサは、例外を取る前に行われていた処理に復帰することができる。
【０００３】
少なくともいくつかの例は、例外復帰状態情報を記憶するための例外復帰状態レジスタ記憶装置と、命令に応答して処理動作を実行するための処理回路と、を備える装置を提供し、処理回路は、保護制御スタック（guarded control stack、ＧＣＳ）例外復帰状態プッシュ命令に応答して、例外復帰状態レジスタ記憶装置から例外復帰状態情報を取得し、例外復帰状態情報をＧＣＳデータ構造にプッシュするように構成されており、処理回路は、ＧＣＳ例外復帰状態ポップ命令に応答して、ＧＣＳ保護例外復帰状態情報をＧＣＳデータ構造から取得するように構成されており、装置は、ターゲットアドレスに対応するメモリ領域がＧＣＳデータ構造を記憶するためのＧＣＳ領域として指定され、書き込み要求がＧＣＳアクセスタイプの命令の制限されたサブセットであって、ＧＣＳアクセスタイプの命令の制限されたサブセットは、少なくともＧＣＳ例外復帰状態プッシュ命令を含む、ＧＣＳアクセスタイプの命令の制限されたサブセットのうちの１つによってトリガされたＧＣＳメモリアクセス要求以外の要求であると判定したことに応答して、ターゲットアドレスを指定する書き込み要求を拒否するためのメモリ管理回路を備え、処理回路は、処理回路の少なくとも１つの動作状態において、例外復帰状態レジスタ記憶装置に記憶された例外復帰状態情報を修正しようとする試みに応答して、例外復帰状態ロックパラメータがロック状態にあるか、又はロック解除状態にあるかを検出し、例外復帰状態ロックパラメータがロック状態にあると検出した場合にフォールトをシグナリングするように構成されている。
【０００４】
少なくともいくつかの例は、例外復帰状態情報を例外復帰状態レジスタ記憶装置に記憶すること、及び処理回路を使用して、命令に応答して処理動作を実行すること、保護制御スタック（ＧＣＳ）例外復帰状態プッシュ命令に応答して、例外復帰状態レジスタ記憶装置から例外復帰状態情報を取得し、例外復帰状態情報をＧＣＳデータ構造にプッシュすること、ＧＣＳ例外復帰状態ポップ命令に応答して、ＧＣＳ保護例外復帰状態情報をＧＣＳデータ構造から取得すること、ターゲットアドレスに対応するメモリ領域がＧＣＳデータ構造を記憶するためのＧＣＳ領域として指定され、書き込み要求がＧＣＳアクセスタイプの命令の制限されたサブセットであって、ＧＣＳアクセスタイプの命令の制限されたサブセットは、少なくともＧＣＳ例外復帰状態プッシュ命令を含む、ＧＣＳアクセスタイプの命令の制限されたサブセットのうちの１つによってトリガされたＧＣＳメモリアクセス要求以外の要求であると判定したことに応答して、ターゲットアドレスを指定する書き込み要求を拒否すること、及び処理回路の少なくとも１つの動作状態において、例外復帰状態レジスタ記憶装置に記憶された例外復帰状態情報を修正しようとする試みに応答して、例外復帰状態ロックパラメータがロック状態にあるか、又はロック解除状態にあるかを検出し、例外復帰状態ロックパラメータがロック状態にあると検出した場合にフォールトをシグナリングすることを含む。
【０００５】
少なくともいくつかの例は、ホストデータ処理装置によって実行されると、ターゲット

コードを実行するための命令実行環境を提供するようにホストデータ処理装置を制御する命令を含むコンピュータプログラムを提供し、コンピュータプログラムは、例外復帰状態情報を記憶するための例外復帰状態レジスタ記憶装置をシミュレートするようにホストデータ処理装置を制御するレジスタエミュレートプログラムロジックと、ターゲットコードの命令の実行をシミュレートするようにホストデータ処理装置を制御する処理プログラムロジックと、を含み、処理プログラムロジックは、ターゲットコードの保護制御スタック（ＧＣＳ）例外復帰状態プッシュ命令に応答して、シミュレートされた例外復帰状態レジスタ記憶装置から例外復帰状態情報を取得し、例外復帰状態情報をＧＣＳデータ構造にプッシュするように構成されており、処理プログラムロジックは、ターゲットコードのＧＣＳ例外復帰状態ポップ命令に応答して、ＧＣＳデータ構造から取得したＧＣＳ保護例外復帰状態情報を取得するように構成されており、コンピュータプログラムは、ターゲットアドレスに対応するメモリ領域がＧＣＳデータ構造を記憶するためのＧＣＳ領域として指定され、書き込み要求がＧＣＳアクセスタイプの命令の制限されたサブセットであって、ＧＣＳアクセスタイプの命令の制限されたサブセットは、少なくともＧＣＳ例外復帰状態プッシュ命令を含む、ＧＣＳアクセスタイプの命令の制限されたサブセットのうちの１つによってトリガされたＧＣＳメモリアクセス要求以外の要求であると判定したことに応答して、ターゲットアドレスを指定する書き込み要求を拒否するためのメモリ管理プログラムロジックを備え、処理プログラムロジックは、処理プログラムロジックの少なくとも１つのシミュレートされた動作状態において、シミュレートされた例外復帰状態レジスタ記憶装置に記憶された例外復帰状態情報を修正しようとする試みに応答して、例外復帰状態ロックパラメータがロック状態にあるか、又はロック解除状態にあるかを検出し、例外復帰状態ロックパラメータがロック状態にあると検出した場合にフォールトをシグナリングするように構成されている。

【０００６】

コンピュータプログラムは、記憶媒体に記憶されてもよい。記憶媒体は、非一時的記憶媒体であってもよい。

【図面の簡単な説明】

【０００７】

本技法の更なる態様、特徴、及び利点は、添付の図面と併せて読まれる以下の例の説明から明らかとなる。

【図１】処理回路及びメモリ管理回路を有するデータ処理装置の一例を示す。

【図２】処理回路の実行状態の一例を示す。

【図３】２ステージのアドレス変換を示す。

【図４】処理回路のレジスタの一例を示す。

【図５】取られる例外へのエントリ時及び例外からの復帰時に実行される命令の一例を示す。

【図６】例外エントリに応答して取得された復帰状態情報を保護するための保護制御スタック（ＧＣＳ）構造の使用を示す。

【図７】ＧＣＳ構造を使用するときに、取られた例外へのエントリ時及び例外からの復帰時に実行される命令の一例を示す。

【図８】メモリアクセス要求が許可されたか否かをチェックする方法を示すフロー図である。

【図９】例外エントリ時に実行されるステップを示すフロー図である。

【図１０】ＧＣＳ例外復帰状態プッシュ命令に応答して実行されるステップを示すフロー図である。

【図１１】ＧＣＳ例外復帰状態ポップ命令に応答して実行されるステップを示すフロー図である。

【図１２】例外復帰命令に応答して実行されるステップを示すフロー図である。

【図１３】例外復帰状態レジスタ記憶装置に記憶された例外復帰状態情報の修正を要求する命令に応答して実行されるステップの２つの代替例を示すフロー図である。

【図１４】例外復帰状態レジスタ記憶装置に記憶された例外復帰状態情報の修正を要求する命令に応答して実行されるステップの２つの代替例を示すフロー図である。
【図１５】シミュレータの実装形態を示す。
【０００８】
装置は、例外復帰状態情報を記憶するための例外復帰状態レジスタ記憶装置と、命令に応答して処理動作を実行するための処理回路と、を有する。
【０００９】
復帰指向プログラミング（Return-oriented-programming、ＲＯＰ）ベースの攻撃は、データ処理システムに対する一般的な攻撃のクラスである。ＲＯＰ攻撃は、関数呼び出し又は例外からの復帰に使用される復帰状態情報を破壊することによって、プログラムを予期せぬ方法で動作させようとする攻撃である。多くの場合、ソフトウェアは、例えば、関数呼び出し又は例外のネスティングを容易にするために、復帰状態情報をメモリに保存し、（ネスティングされた関数呼び出し又は例外のセットの）外部関数呼び出し又は例外の復帰状態情報は、レジスタ内で内部関数呼び出し又は例外の復帰状態情報で上書きされ得る前に、保存するためにメモリに保存される。ＲＯＰ攻撃が成功すると、関数復帰又は例外復帰は、関数が呼び出されたか又は例外が取られた時点の次の命令以外の命令にプログラムフローを復帰させ得、これにより、攻撃者は、プログラマが意図した一連の動作以外の任意の動作を実行するように処理回路を制御することができる。
【００１０】
例外復帰状態情報を改ざんするＲＯＰベースの攻撃から保護するために、例外から復帰するための復帰状態情報を保護するために使用される保護制御スタック（ＧＣＳ）データ構造を提供することが有用であり得る。処理回路は、保護制御スタック（ＧＣＳ）例外復帰状態プッシュ命令に応答して、例外復帰状態レジスタ記憶装置から例外復帰状態情報を取得し、例外復帰状態情報をＧＣＳデータ構造にプッシュする。処理回路は、ＧＣＳ例外復帰状態ポップ命令に応答して、ＧＣＳ保護例外復帰状態をＧＣＳデータ構造から取得する。ＧＣＳデータ構造の保護は、メモリ領域を、ＧＣＳデータ構造を記憶するように意図されたＧＣＳ領域として指定することをサポートするメモリ管理回路によって提供され得る。メモリ管理回路は、書き込み要求が、ＧＣＳアクセスタイプの命令の制限されたサブセットのうちの１つによってトリガされたＧＣＳメモリアクセス要求以外の要求であるとき（ＧＣＳアクセスタイプの命令の制限されたサブセットは、少なくともＧＣＳ例外復帰状態プッシュ命令を含む）、ターゲットアドレスに対応するメモリ領域がＧＣＳ領域として指定されていると判定したことに応答して、ターゲットアドレスを指定する書き込み要求を拒否する。したがって、いずれのタイプの書き込み要求によっても書き込むことができないメモリの保護領域を定義することができる（例えば、汎用（ＧＣＳ専用ではない）ストア命令を、ＧＣＳアクセスタイプの命令の制限されたサブセットから除外することができる）。これにより、攻撃者がＧＣＳデータ構造に記憶された、ＧＣＳ保護例外復帰状態の破壊を試みるために利用できる攻撃面を制限する。したがって、標準的な読み取り可能／書き込み可能メモリ領域内のソフトウェアによって維持される汎用スタックを使用して可能な保護よりも強力な保護を例外復帰状態に提供することができる。
【００１１】
しかしながら、発明者らは、成功した場合には、ＧＣＳデータ構造に記憶された、保護例外復帰状態情報に基づいたチェックを回避することができる、レジスタにおける改ざんに対して例外復帰状態情報が脆弱であり得るときに、例外エントリ（例外復帰状態が例外復帰状態レジスタ記憶装置で取得されるとき）とＧＣＳ例外復帰状態プッシュ命令の実行との間の窓、及びＧＣＳ例外復帰状態ポップ命令の実行と例外復帰との間の別の時間窓が存在し得ることを認識した。攻撃者が、例外エントリとＧＣＳ例外復帰状態プッシュ命令の実行との間に例外復帰状態レジスタ記憶装置内の例外復帰状態を修正することができる場合、誤った例外復帰状態がＧＣＳデータ構造にプッシュされ、したがってＧＣＳデータ構造で信頼されていると誤ってみなされ得る。同様に、一部の攻撃は、ＧＣＳ例外復帰状態ポップ命令の実行後であるが、例外復帰の実行前に、例外復帰状態レジスタ記憶装置内

の例外復帰状態を修正することに基づくことがある。

【００１２】

この問題に対処するための１つの手法は、これらの時間窓をなくすために、ＧＣＳ例外復帰状態プッシュ命令の動作が、代わりに、例外エントリ自体に応答してハードウェアで実行されること、及びＧＣＳ例外復帰状態ポップ命令の動作が、例外復帰をトリガする同じ命令に応答して実行されることを要求することであり得る。しかしながら、この手法は、単一のイベント（例外エントリ及び復帰）によるトリガを必要とする動作の数を増加させ得る。単一のトリガイベントに応答して必要とされるアクションのそのような過負荷は、プロセッサマイクロアーキテクチャ設計者が、回路タイミング要件を満たしつつ、これらの動作をサポートするための回路ハードウェアを実装することをはるかに複雑にし得る。また、ＧＣＳ例外復帰状態プッシュ命令を例外エントリから分離し、ＧＣＳ例外復帰状態ポップ命令を例外復帰から分離することはまた、ソフトウェアがどの例外エントリ／リターンイベントを保護するか、又は保護しないかを決定するための柔軟性がより高いという利点を有し、したがって、これらの動作を組み合わせることは望ましくない場合がある。

【００１３】

以下に説明する例では、ロック状態又はロック解除状態に設定することができる例外復帰状態ロックパラメータが提供される。処理回路の少なくとも１つの動作状態において、処理回路は、例外復帰状態レジスタ記憶装置に記憶された例外復帰状態情報を修正しようとする試みに応答して、例外復帰状態ロックパラメータがロック状態にあるか、又はロック解除状態にあるかを検出し、例外復帰状態ロックパラメータがロック状態にあると検出すると、フォールトをシグナリングする。例外復帰状態情報をロックして修正を防止するアーキテクチャ機構を設けることによって、例外エントリとＧＣＳデータ構造に対する例外復帰状態のプッシュとの間、及びＧＣＳ保護例外復帰状態情報のポップと後続の例外復帰との間の窓内の例外復帰状態情報の改ざんを防止して、例外復帰状態情報のＧＣＳベースのチェックで提供されるセキュリティを改善することができる。

【００１４】

処理回路は、異なる特権レベルに関連付けられたいくつかの例外レベルのうちの１つでの命令の実行をサポートし得る。処理回路の少なくとも１つの動作状態において、処理回路は、例外が取られる前の現在の例外レベルと同じ例外レベルで例外が取られるべきである例外が取られたことに応答して、例外復帰状態ロックパラメータをロック状態に設定することができる。したがって、例外が取られる前に実行される処理と同じ例外レベルで取られるべき例外について、例外エントリは、取得された例外復帰状態情報がＧＣＳデータ構造にプッシュされ得る前に、処理回路に例外復帰状態情報を修正に対してロックさせて、例外エントリに応答して取得された例外復帰状態情報を改ざんから保護することができる。例外復帰状態ロックパラメータのロック状態への設定は、例外復帰状態ロックパラメータをロック状態に設定する例外エントリ後に実行される明示的なソフトウェア命令を必要とせずに、例外が取られたことに応答して自動的に行われ得る。

【００１５】

例外が取られる前に最新であった例外レベルよりも高い特権を有する例外レベルに対して例外が取られるべき例外エントリに応答して、例外復帰状態ロックパラメータをロック状態に設定することは必須ではない。例外がより高い特権を有するレベルに対して取られる場合、例外復帰を適切に管理するために、より高い特権を有する例外処理コードを信頼することができる。攻撃は、例外が同じ例外レベルに対して取られる場合により起こりやすい。それにもかかわらず、他の実装形態はまた、より高い特権を有する例外レベルに対して例外が取られるときに例外復帰状態情報をロックすることを選択することができる。

【００１６】

処理回路の少なくとも１つの動作状態において、処理回路は、ＧＣＳ例外復帰状態プッシュ命令に応答して、例外復帰状態ロックパラメータをロック解除状態に設定し得る。したがって、例外復帰状態情報がＧＣＳデータ構造にプッシュされると、例外復帰状態レジ

スタ記憶装置内の例外復帰状態情報の修正を防止する必要はもはやなく、したがって、ロックパラメータをロック解除状態に切り替えることによって、例外復帰状態レジスタ記憶装置内の情報を有効に変更する必要があり得る後続の動作を進めることが可能になる。ＧＣＳ例外復帰状態プッシュ命令に応答して例外復帰状態ロックパラメータをロック解除することはまた、例外復帰状態の修正がロックされるべきときに実行される命令（例えば、例外復帰命令）が、例外復帰状態の修正が現在ロック解除されているときに正しくない一連のプログラムフローで実行される場合の検出を可能にすることによってセキュリティを改善するのに役立つ。

【００１７】

処理回路の少なくとも１つの動作状態において、処理回路は、ＧＣＳ例外復帰状態ポップ命令に応答して、例外復帰状態ロックパラメータをロック状態に設定し得る。ＧＣＳ例外復帰状態ポップ命令は、ＧＣＳデータ構造からポップされた例外復帰状態が改ざんに対して脆弱である窓の開始をマークすることができ、したがって、ＧＣＳ例外復帰状態ポップ命令に応答して例外復帰状態をロックすることは、後続の例外復帰において、例外復帰に使用される例外復帰状態情報が依然としてＧＣＳデータ構造から取得した情報と同じであるという確信を与えるのに役立つ。

【００１８】

処理回路の少なくとも１つの動作状態において、処理回路は、例外復帰が例外復帰前の現在の例外レベルと同じ例外レベルでの処理への復帰である、例外復帰状態レジスタ記憶装置に記憶された例外復帰状態情報に基づいて例外復帰をトリガするための例外復帰命令に応答して、例外復帰状態ロックパラメータがロック状態に設定されているか否かをチェックし、例外復帰状態ロックパラメータがロック解除状態に設定されていることを検出することに応答してフォールトをシグナリングする。これは、ＧＣＳ例外復帰状態ポップ命令を使用してＧＣＳ保護例外復帰状態情報を最初に取得することなく例外復帰命令を実行させるようにプログラムフローを制御しようとする場合がある攻撃から保護するのに役立ち得る。

【００１９】

例外復帰状態ロックパラメータがロック状態に設定されているときに例外復帰状態レジスタ記憶装置内の情報を修正させ得る命令が実行されると、フォールトをシグナリングするべきか否かを判定するために、異なる手法を取ることができる。

【００２０】

一例では、処理回路の少なくとも１つの動作状態において、処理回路は、例外復帰状態ロックパラメータがロック状態に設定されているときに、例外復帰状態レジスタ記憶装置に書き込まれるべき新しい値が、例外復帰状態レジスタ記憶装置に以前に記憶された古い値と同じであるか、異なるかにかかわらず、例外復帰状態レジスタ記憶装置内の情報に対する修正を要求する命令を実行しようとする試みに応答して、フォールトをシグナリングすることができる。したがって、この場合、命令が例外復帰状態レジスタ記憶装置の情報を修正しようと試み、そのような修正がロックされていると、命令によって書き込まれるべき新しい値が、例外復帰状態レジスタ記憶装置に以前に記憶された古い値と同じであったとしてもフォールトがシグナリングされる。新しい値が古い値と同じである場合には例外復帰状態情報の修正は存在しないが、古い値と新しい値との間でのいかなる比較にも関係なくフォールトをシグナリングすることは、（例外復帰状態ロックパラメータに応じてフォールトのシグナリングを制御するための制御回路ロジックを実装しようとするハードウェア設計者の観点から）より単純であり得る。これは、例外復帰状態レジスタ記憶装置に現在記憶されている値を読み取って、フォールトをトリガするか否かを判定する必要がないためである。

【００２１】

別の例では、処理回路の少なくとも１つの動作状態において、処理回路は、例外復帰状態ロックパラメータがロック状態に設定されているときに、例外復帰状態レジスタ記憶装置内の情報に対する修正を要求する命令を実行しようとする試みに応答して、例外復帰状

態レジスタ記憶装置に書き込まれるべき新しい値が例外復帰状態レジスタ記憶装置に以前に記憶された古い値と同じであるか、又は異なるかをチェックし、新しい値が古い値とは異なると判定すると、フォールトをシグナリングすることができる。この手法は、例外ハンドラが非常に短いために例外復帰状態をメモリ内のスタックに保存する／スタックから復元する必要がない、一部の高速例外コードパスの処理を高速化するのに有用であり得る。この場合、例外ハンドラは、最初に取得した例外復帰状態が例外ハンドラの実行を通して例外復帰状態レジスタ記憶装置内に残り得るので、いずれのＧＣＳ例外復帰状態プッシュ／ポップ命令も実行しないことを好む場合がある。これは、例外復帰状態ロックパラメータが例外ハンドラにわたってロック状態のままであり得ることを意味し得る。それにもかかわらず、例外ハンドラが例外復帰状態レジスタ記憶装置の修正を試み得る正当な理由がいくつか存在することがある。したがって、ロックパラメータがロック状態にある間に例外復帰状態レジスタ記憶装置に記憶された例外復帰状態情報を修正しようとする試みが存在する場合にはフォールトがトリガされるべきであるが、時には、試みられた修正が、実際には例外復帰状態レジスタ記憶装置に既に記憶された同じ値を書き込むことがあり、その場合、例外復帰状態が破壊されるリスクはない。例えば、これは、上述したように、高速例外コードパス内にあるために例外復帰状態情報を変更する意図はなかったが、例外ハンドラコードが、例外復帰状態情報の修正を必要とする他の使用事例と共有される共通コードパスを含み、例外復帰状態レジスタに書き込む命令を実行することがとにかく好ましい（例えば、条件分岐を回避することによって性能を高速化するために）が、命令が高速例外コードパスと共に使用される場合には、例外復帰状態レジスタ内に以前に記憶された同じ値を書き込むように、当該命令のオペランドを設定する場合に起こり得る。したがって、修正命令によって例外復帰状態レジスタ記憶装置内の値が変更されない場合にフォールトをシグナリングしないことによって、フォールトのシグナリングによって追加される待ち時間を排除することにより性能を改善することができる（別様に現在の処理を中断し、別の例外ハンドラを実行させる）。

【００２２】

ＧＣＳ例外復帰状態ポップ命令は、異なる方法で実装され得る。

【００２３】

いくつかの例では、ＧＣＳ例外復帰状態ポップ命令は、単に、ＧＣＳ保護例外復帰状態情報をレジスタ記憶装置にポップすることができる（例外復帰状態レジスタ記憶装置に直接、又は後続の命令がＧＣＳ保護例外復帰状態情報を例外復帰状態レジスタ記憶装置に移動させることができる汎用レジスタに）。次いで、ＧＣＳ保護例外復帰状態情報を直接使用して、後続の例外復帰命令の実行時に、例外復帰を制御することができる。この場合、ソフトウェアが、通常メモリ内のソフトウェア管理スタック構造への例外復帰状態情報の保存／復元を管理する必要はない。これは、ネスティングされた例外が例外復帰状態レジスタ記憶装置内の例外復帰状態を上書きさせる場合であっても、ＧＣＳ例外復帰状態プッシュ／ポップ命令の実行は、ＧＣＳデータ構造が例外復帰状態を追跡するのに十分であることを意味するためである。

【００２４】

他の例では、ＧＣＳ保護例外復帰状態を使用して例外復帰を直接制御するのではなく、これを使用して、例外復帰状態レジスタ記憶装置に記憶された、意図した例外復帰状態情報が、後続の例外復帰に使用するのに安全であるか否かを検証することができる。この場合、ソフトウェアは、その通常の機構を使用して、例外復帰状態レジスタ記憶装置内の例外復帰状態情報を管理することができる（例えば、例外がネスティングされている場合、ソフトウェアが、例外復帰状態を、非ＧＣＳメモリ内に維持されたスタックデータ構造に保存／復元する）。ＧＣＳ例外復帰状態プッシュ／ポップ命令は更に、例外復帰状態をＧＣＳデータ構造に保存／復元して、ソフトウェアで取得した、意図した例外復帰状態情報が、例外復帰の制御に使用される前にチェックされることを可能にする。

【００２５】

したがって、そのような例では、ＧＣＳ例外復帰状態ポップ命令に応答して、ＧＣＳ保

護例外復帰状態情報をＧＣＳデータ構造からポップさせることに加えて、処理回路はまた、例外復帰状態レジスタ記憶装置から取得した、意図した例外復帰状態情報と、ＧＣＳデータ構造から取得した、ＧＣＳ保護例外復帰状態情報との間で不一致が検出されると、フォールトをシグナリングする。したがって、ＧＣＳ保護例外復帰状態情報は、例外復帰を直接制御するために使用されるのではなく、意図した例外復帰状態情報を検証するために使用される。

【００２６】

別の例では、ＧＣＳ保護例外復帰状態情報は、意図した例外復帰状態情報が正しいか否かを検証するために使用され得るが、この検証は、ＧＣＳ例外復帰状態ポップ命令自体に応答して行われなくてよく、代わりに、後続のＧＣＳ例外復帰状態検証命令が、例外復帰状態レジスタ記憶装置内の意図した例外復帰状態情報が、ＧＣＳ例外復帰状態ポップ命令によって取得された、以前にポップされたＧＣＳ例外復帰状態情報と一致するか否かを検証することができる。

【００２７】

例外復帰状態レジスタ記憶装置に記憶された例外復帰状態情報は、様々な形態であり得る。一般に、例外復帰状態レジスタ記憶装置は、例外復帰状態情報を記憶する１つ以上のレジスタを備え得る。例えば、例外復帰状態レジスタ記憶装置は、例外からの復帰後に実行される命令の例外復帰アドレスを記憶するための例外復帰アドレスレジスタ、及び／又は例外からの復帰時に復元されるべきプロセッサ状態を記憶するための保存プロセッサ状態レジスタを備え得る。これらのレジスタの一方又は両方に記憶された情報は、ＧＣＳ例外復帰状態プッシュ命令に応答してＧＣＳ構造に記憶され得る。

【００２８】

保存されたプロセッサ状態レジスタは、例外を取ったことに応答して、処理回路の現在のプロセッサ状態がプロセッサ状態レジスタから保存プロセッサ状態レジスタにコピーされるレジスタであり得る。例外を取ることによって現在のプロセッサ状態を変更させる可能性があるので、現在のプロセッサ状態は、保存プロセッサ状態レジスタにコピーすることによって取得され得、その結果、処理回路は、例外が取られる状態に従って現在のプロセッサ状態を自由に更新することができる。次いで、ソフトウェアで実行される例外ハンドラは、保存プロセッサ状態レジスタを読み取り、取得した状態をメモリに保存して、以前の例外の処理をプリエンプトする更なる例外が発生した場合でも、保存プロセッサ状態が保持されることを確実にすることができる。ＧＣＳ例外状態プッシュ命令を使用して、当該保存プロセッサ状態をＧＣＳデータ構造にプッシュして、対応する例外復帰における例外復帰状態情報の検証を可能にすることもできる。

【００２９】

例外復帰状態ロックパラメータは、プロセッサ状態レジスタにおいて指定されるパラメータであることが有用であり得る。これは、プロセッサ状態レジスタの他の内容と共に、例外が取られたことに応答して保存プロセッサ状態レジスタにコピーすることができるので、例外が取られたときに例外復帰状態ロックパラメータ自体が自動的に保存されることを意味する。同様に、例外復帰命令に応答して、保存プロセッサ状態を保存プロセッサ状態レジスタからプロセッサ状態レジスタにコピーすることにより、例外復帰状態ロックパラメータの以前の値が復元される。したがって、この手法は、例外エントリとＧＣＳ例外復帰状態プッシュ命令との間の窓中に、又はＧＣＳ例外復帰状態ポップ命令と例外復帰との間の窓中に新しい例外が取られた場合、例外復帰状態ロックパラメータの現在の状態（これらの窓中にロック状態にあるはずである）が取得され、したがって、保存プロセッサ状態の通常の復元の一部として新しい例外からの復帰時に復元されて、新しい例外が取られた時点の例外復帰状態ロックパラメータの前の状態に復帰可能であることを意味する。したがって、攻撃者は、例外復帰状態ロックパラメータによって提供される保護を回避しようと試みる機構として、新しい例外をトリガすることを使用することができない。

【００３０】

例外復帰状態ロックパラメータに基づいた、例外復帰状態レジスタ記憶装置に対する修

正の監視は、処理回路の全動作状態において必ずしも必要ではない。例えば、１つ以上の制御レジスタに記憶された一部の制御状態情報を使用して、上述の例外復帰状態ロックパラメータの様々なチェックを実行すべきか否かを判定することができる。これらのチェックを無効にすることは、例外復帰状態ロックパラメータをサポートしていなかったレガシーシステム向けに書かれたソフトウェアとの後方互換性のために有用であり得る。そのようなレガシーソフトウェアでは、攻撃者がＲＯＰ攻撃を実施するために利用できる抜け穴がないことを確実にする際に、ソフトウェア開発者のより厳重な警戒を必要とし得る。

【００３１】

したがって、一例では、処理回路の少なくとも１つの動作状態は、例外復帰状態ロックパラメータを使用して例外復帰状態情報のロックが有効であることを示す有効状態に例外復帰状態ロック有効化制御パラメータが設定されている動作状態を含む。例外復帰状態ロック有効化制御パラメータが無効状態に設定されている場合、上述したように例外復帰状態ロックパラメータを更新するか、又はチェックする必要はない。

【００３２】

いくつかの例では、ＧＣＳデータ構造の使用をサポートするアーキテクチャ機構（ＧＣＳ例外復帰状態プッシュ／ポップ命令、ＧＣＳデータ構造と相互作用する他の命令、及び無許可アクセスからＧＣＳデータ構造を保護するためにトリガされる様々なフォールトなど）は、ＧＣＳ有効化パラメータの状態に応じて完全に無効化され得る。この場合も、これらのＧＣＳベースのアーキテクチャ機構をサポートしなかったレガシーシステム向けに書かれたソフトウェアにサポートを提供することができる。したがって、（例外復帰状態ロックパラメータに関連して上述の様々な手段が講じられる）処理回路の少なくとも１つの動作状態は、例外復帰状態ロック有効化制御パラメータが、例外復帰状態ロックパラメータを使用して例外復帰状態情報のロックが有効であることを示す有効状態に設定されており、かつＧＣＳ有効化パラメータが、ＧＣＳデータ構造の使用をサポートするためのＧＣＳモードが有効であることを示す有効状態に設定されている、動作状態を含み得る。

【００３３】

上述したように、メモリ管理回路は、ターゲットアドレスに対応するメモリ領域がＧＣＳ領域として指定され、書き込み要求が、ＧＣＳ領域へのアクセスを許可されたＧＣＳアクセスタイプの命令の制限されたサブセットのうちの１つによってトリガされたＧＣＳメモリアクセス要求以外の要求であると判定したことに応答して、ターゲットアドレスを指定する書き込み要求を拒否し得る。メモリ管理回路はまた、メモリアクセス要求がＧＣＳメモリアクセス要求であり、ターゲットアドレスに対応するメモリ領域がＧＣＳ領域ではないと判定したことに応答して、メモリアクセス要求を拒否し得る。これは、ＧＣＳメモリアクセス要求がＧＣＳ領域にアクセスするためにのみ使用され得、他のデータを記憶するために使用される通常のメモリ領域にアクセスするために使用され得ないことを意味するので、攻撃者に利用可能な攻撃面を低減させる。これは、ＧＣＳメモリアクセス要求に非ＧＣＳ領域へアクセスさせるように、ＧＣＳアクセスタイプの命令のアドレスオペランドが攻撃者によって修正された（例えば、誤った一連のプログラムフローを強制する攻撃者によって誤ったものにされた）場合、これがエラーとして検出され、メモリへのアクセス要求が拒絶されることを意味する。これは、攻撃者が、ＧＣＳデータ構造の代わりにメモリの別の領域を代用することでＧＣＳ保護例外復帰状態を偽造できないようにすることによってセキュリティを改善する。

【００３４】

ＧＣＳデータ構造は、ＲＯＰ攻撃から例外復帰状態情報を保護するために使用されるものとして上述されているが、同じデータ構造は、関数復帰アドレスなどの関数復帰状態情報を保護するためにも使用され得る。ＧＣＳデータ構造から取得した情報が例外復帰状態情報であるか、又は関数復帰状態情報であるかを（例えば、有効な例外復帰状態レコード及び有効な関数復帰状態レコードの符号化を区別して、区別できるようにすることによって）検出するためのメカニズムが提供され得、その結果、ＧＣＳ例外復帰状態検証命令は、処理回路に、ＧＣＳ保護例外復帰状態情報として使用される、取得したＧＣＳレコード

が有効な例外復帰状態レコードに対応するか否かをチェックさせ、したがって、関数復帰状態レコードを使用して例外復帰について例外復帰状態情報を検証することによって、誤った挙動を生じさせる試みを監視し得る。

【００３５】

他の例では、ＧＣＳデータ構造は、例外復帰状態情報を保護するために使用され得るが、関数復帰状態情報を保護するためには使用され得ない（例えば、異なる機構が、ＲＯＰ攻撃から関数復帰状態を保護するために使用され得るか、又は関数復帰状態情報については、ＲＯＰ攻撃からのアーキテクチャレベルの保護が提供され得ない（代わりに、関数復帰に対するＲＯＰ攻撃から防御する責任は、ソフトウェア開発者が負い得る））。したがって、ＧＣＳデータ構造が例外復帰状態と関数リターン状態との間で必ずしも共有されるわけではない。

【００３６】

上述の技法は、上述の処理回路及びメモリ管理回路を実装するために設けられたハードウェア回路を有するデータ処理装置内で実装され得る。

【００３７】

しかしながら、同じ技法はまた、ホストデータ処理装置上で実行されてターゲットコードの実行のための命令実行環境を提供するコンピュータプログラムで実装されてもよい。そのようなコンピュータプログラムは、ホストデータ処理装置自体がそのアーキテクチャをサポートしていなくても、特定の命令セットアーキテクチャに従ってターゲットコードを実際にサポートするハードウェア装置上に提供されるアーキテクチャ環境をシミュレートするようにホストデータ処理装置を制御することができる。コンピュータプログラムは、上述したように、ＧＣＳ例外復帰状態プッシュ命令及びＧＣＳ例外復帰状態ポップ命令、及び例外復帰状態ロックパラメータのサポートなど、上述した処理回路、例外復帰状態レジスタ記憶装置、及びメモリ管理回路の機能をエミュレートする、処理プログラムロジック、レジスタエミュレートプログラムロジック、及びメモリ管理プログラムロジックを有し得る。そのようなシミュレーションプログラムは、例えば、ある命令セットアーキテクチャに対して書き込まれたレガシーコードが、異なる命令セットアーキテクチャをサポートするホストプロセッサ上で実行されている場合に有用であり得る。また、シミュレーション実行環境上でソフトウェアを実行することにより、新しいアーキテクチャをサポートするハードウェアデバイスの進行中の開発と並行してソフトウェアのテストを可能にすることができるため、シミュレーションは、新しいアーキテクチャバージョンをサポートするハードウェアを処理する前に、命令セットアーキテクチャの新しいバージョンのソフトウェア開発を開始することを可能にすることができる。シミュレーションプログラムは記憶媒体に記憶されてもよく、この記憶媒体は、非一時的記憶媒体であってもよい。

【００３８】

データ処理装置の具体例

図１は、データ処理装置２の例を概略的に示す。データ処理装置は、いくつかのパイプラインステージを含む処理パイプライン４を有する。この例では、パイプラインステージは、命令キャッシュ８から命令をフェッチするためのフェッチステージ６と、パイプラインの残りのステージによって処理されるマイクロオペレーション（復号命令）を生成するために、フェッチされたプログラム命令を復号するための復号ステージ１０と、マイクロオペレーションに必要なオペランドがレジスタファイル１４内で利用可能であるかどうかをチェックし、所与のマイクロオペレーションに必要なオペランドが利用可能になると、実行するためのマイクロオペレーションを発行する発行ステージ１２と、結果値を生成するためにレジスタファイル１４から読み取られたオペランドを処理することによって、マイクロオペレーションに対応するデータ処理動作を実行するための実行ステージ１６と、処理の結果をレジスタファイル１４に書き戻すための書き戻しステージ１８と、を含む。これは、可能なパイプラインアーキテクチャの単なる一例にすぎず、他のシステムは追加のステージ又は異なる構成のステージを有してもよいことが理解されるであろう。例えば、アウトオブオーダプロセッサでは、プログラム命令又はマイクロオペレーションによっ

て指定されたアーキテクチャレジスタを、レジスタファイル１４内の物理レジスタを識別する物理レジスタ指定子にマッピングするためのレジスタリネームステージが含まれ得る。いくつかの例では、復号ステージ１０によって復号されるプログラム命令と、実行ステージによって処理される対応するマイクロ演算との間に１対１の関係が存在し得る。また、プログラム命令とマイクロオペレーションとの間に１対多又は多対１の関係があってもよく、例えば、単一のプログラム命令を２つ以上のマイクロオペレーションに分割してもよいし、２つ以上のプログラム命令を融合して単一のマイクロオペレーションとして処理してもよい。

【００３９】

実行ステージ１６は、異なるクラスの処理動作を実行するためのいくつかの処理ユニットを含む。例えば、実行ユニットは、レジスタ１４から読み取られたスカラオペランドに対して算術演算又は論理演算を実施するためのスカラ算術／論理ユニット（arithmetic/logic unit、ＡＬＵ）２０と、浮動小数点値に対して演算を実施するための浮動小数点ユニット２２と、分岐動作の結果を評価し、それに応じて現在の実行点を表すプログラムカウンタを調節するための分岐ユニット２４と、メモリシステム８、３０、３２、３４内のデータにアクセスするロード／記憶動作を実施するためのロード／ストアユニット２６と、を含み得る。データアクセス命令のオペランドに基づいてロード／記憶ユニット２６によって指定される仮想アドレスと、メモリシステム内のデータのストレージロケーションを識別する物理アドレスとの間のアドレス変換を実行するための、メモリ管理回路の一例である、メモリ管理ユニット（memory management unit、ＭＭＵ）２８が提供される。ＭＭＵは、メモリシステムに記憶されたページテーブルからのアドレス変換データをキャッシュするための変換ルックアサイドバッファ（translation lookaside buffer、ＴＬＢ）２９を有し、ページテーブルのページテーブルエントリはアドレス変換マッピングを定義し、また、例えば、パイプライン上で実行している所与のプロセスが所与のメモリ領域から命令を読取り、書込み、又は実行することを可能にするかどうかを支配するアクセスパーミッションを指定し得る。

【００４０】

この例では、メモリシステムは、レベル１データキャッシュ３０と、レベル１命令キャッシュ８と、共有のレベル２キャッシュ３２と、メインシステムメモリ３４と、を含む。これは可能なメモリ階層の一例にすぎず、キャッシュの他の配置を提供することができることが理解されるであろう。実行ステージ１６に示される特定のタイプの処理ユニット２０～２６は単なる一例であり、他の実装形態は、処理ユニットの異なるセットを有してもよく、あるいは、同じタイプの複数のマイクロオペレーションを並行して処理できるように同じタイプの処理ユニットの複数のインスタンスを含んでもよい。図１は、可能なプロセッサパイプライン実装形態のいくつかのコンポーネントの簡素化された表現にすぎず、プロセッサは、簡潔にするために図示していない多くの他の要素を含むことができることが理解されるであろう。図１は、メモリ３４へのアクセスを有する単一のプロセッサコアを示しているが、装置２は、各コアがそれぞれのキャッシュ８、３０、３２を有する、メモリ３４へのアクセスを共有する１つ以上の更なるプロセッサコアを有することもできる。

【００４１】

装置はまた、例外の処理を制御するための例外処理回路４０を備える。例外処理回路４０は、例外信号４２に応答して、パイプライン４によって実行されている処理を中断させ、パイプライン４を制御して、例外信号４２によって示される例外のタイプに対応する例外ハンドラの処理に切り替える。例えば、例外は、処理パイプライン４によって実行されているソフトウェアによって直接的に引き起こされない外部イベントによってトリガされた割り込み、例えば、処理装置を備える電子デバイス上のボタンをユーザが押すこと、プロセッサリセットがアサートされること、ネットワークインターフェース又は他の入力／出力デバイスからのメッセージの受信など外部イベントをシグナリングする割り込みであり得る。また、例外は、パイプライン４によって実行されているソフトウェアによって直

接引き起こされた、ソフトウェアによってトリガされた例外であり得る。ソフトウェアによってトリガされた例外は、例えば、行われるべき「スーパーバイザコール」割り込みに、より高い特権を有する実行状態への切り替えをトリガさせる命令を実行することによって、ソフトウェアによって自発的にトリガされ得る。あるいは、ソフトウェアによってトリガされた例外は、ソフトウェアによって非自発的にトリガされた例外であり得、これは、未定義命令エラー又はアドレス障害など、ソフトウェアによって実行される少なくとも１つの命令で発生するエラーに起因して発生する。

【００４２】

図２は、命令を実行するときに処理回路４が動作することができる異なる実行状態（例外レベルとも呼ばれる）を示す図である。この例では、４つの例外レベルＥＬ０、ＥＬ１、ＥＬ２、ＥＬ３が存在し、例外レベルＥＬ０は最低特権例外レベルであり、例外レベルＥＬ３は最高特権例外レベルである。一般に、より特権のある例外レベルで実行するときに、処理回路は、より低い特権の低い例外レベルにアクセスできないいくつかのメモリロケーション又はレジスタ１４にアクセスすることができる。

【００４３】

この例では、例外レベルＥＬ０は、例外レベルＥＬ１で実行される対応するオペレーティングシステム又は仮想マシンによって管理されるアプリケーションを実行するためのものである。複数の仮想マシンが同じ物理プラットフォーム上に共存する場合、それぞれの仮想マシンを管理するために、ＥＬ２で動作するハイパーバイザが提供され得る。図２は、ハイパーバイザが仮想マシンを管理し、仮想マシンがアプリケーションを管理する例を示すが、ハイパーバイザがＥＬ０においてアプリケーションを直接管理することも可能である。

【００４４】

必須ではないが、いくつかの実装形態は、処理回路のための別個のハードウェア区分されたセキュアな動作ドメイン及び非セキュアな動作ドメインを実装し得る。データ処理システム２は、セキュアドメインで動作するソフトウェアプロセスに関連付けられたデータ及びコードが、非セキュアドメインで動作するプロセスによるアクセスから隔離されることを保証するために、プロセッサ及びメモリシステム内に実装されたハードウェア機能を有してもよい。例えば、英国ケンブリッジのＡｒｍ（登録商標）Ｌｉｍｉｔｅｄによって提供されるＴｒｕｓｔＺｏｎｅ（登録商標）アーキテクチャなどのハードウェアアーキテクチャを使用することができる。代替的に、他のハードウェア実施セキュリティ分割アーキテクチャを使用することもできる。セキュアアプリケーション（信頼できるサービス）は、セキュアドメイン内の例外レベルＥＬ０で動作することができ、セキュア（信頼できる）オペレーティングシステム又は仮想マシンは、セキュアドメイン内の例外レベルＥＬ１で動作することができる。いくつかの実装形態では、セキュア状態においてＥＬ２に対するサポートはなく、ハイパーバイザは、非セキュアＥＬ２においてのみ実行することができる。他の実装形態では、図２のアスタリスクによって示されるように、セキュアＥＬ２において実行されるセキュアハイパーバイザに対するサポートが存在し得る。いくつかの例では、非セキュアドメインとセキュアドメインとの間の遷移を管理するためのセキュアモニタプログラムが、例外レベルＥＬ３において実行して提供され得る。他の実装形態は、セキュアモニタプログラムが必要とされない場合があるように、ハードウェア内のセキュリティドメイン間の遷移を監視することができる。

【００４５】

現在の例外レベルＥＬｘでの処理中に例外が取られた場合、当該例外は、同じ例外レベルＥＬｘにおいて、又はより高い特権を有する例外レベルＥＬｙ（ｙ＞ｘである）において実行され得る。例外タイプに関連付けられたプロパティは、例外のターゲット例外レベルを示し得る。

【００４６】

アドレス変換

ＭＭＵ　２８によって実行される１つのタスクは、仮想アドレス（virtual address

、ＶＡ）と物理アドレス（physical address、ＰＡ）との間のアドレス変換である。処理回路４上で実行されるソフトウェアは、仮想アドレスを使用してメモリロケーションを指定するが、これらの仮想アドレスは、ＭＭＵ　２８によって、アクセスすべきメモリシステムロケーションを識別する物理アドレスに変換され得る。仮想アドレスを使用する利点は、オペレーティングシステム（Operating System、ＯＳ）などの管理ソフトウェアが、ソフトウェアに提示されるメモリのビューを制御することを可能にすることである。ＯＳは、どのメモリが可視であるか、そのメモリが可視である仮想アドレス、及びそのメモリにどのアクセスが許可されるかを制御することができる。これは、ＯＳがアプリケーションをサンドボックス化し（１つのアプリケーションのリソースを別のアプリケーションから隠す）、基礎となるハードウェアからの抽象化を提供することを可能にする。仮想アドレスを使用する別の利点は、ＯＳが、メモリの複数の断片化された物理領域を単一の連続仮想アドレス空間としてアプリケーションに提示することができることである。仮想アドレスは、アプリケーションを書くときにシステムの正確なメモリアドレスを知らないソフトウェア開発者にも利益をもたらす。仮想アドレスを用いると、ソフトウェア開発者は、物理メモリに関心を持つ必要がない。アプリケーションは、アドレス変換を実行するために協働するのがＯＳ及びハードウェア次第であることを知っている。

【００４７】

実際には、各アプリケーションは、物理システム内の異なるロケーションにマッピングされる、それ自体の仮想アドレスのセットを使用することができる。オペレーティングシステムは、異なるアプリケーション間で切り替えると、マップを再プログラムする。これは、現在のアプリケーションの仮想アドレスがメモリ内の正しい物理ロケーションにマッピングされることを意味する。

【００４８】

仮想アドレスはマッピングを介して物理アドレスに変換される。仮想アドレスと物理アドレスとの間のマッピングは、変換テーブル（ページテーブルと呼ばれることもある）に記憶される。変換テーブルはメモリに格納され、ソフトウェア、典型的にはＯＳ又はハイパーバイザによって管理される。変換テーブルは静的ではなく、ソフトウェア変更の必要に応じてテーブルを更新することができる。これは、仮想アドレスと物理アドレスとの間のマッピングを変更する。

【００４９】

処理回路４が実行状態の特定のサブセットにあるとき（特に、処理回路４が非セキュアＥＬ０又は非セキュアＥＬ１にあるとき）に実行されるメモリアクセスについては、図３に示されるように２ステージアドレス変換が使用される（他の実行状態については、ステージ１ページテーブルを使用する１つのステージのアドレス変換で十分である）。したがって、非セキュアＥＬ０及び非セキュアＥＬ１からの仮想アドレスは、二組のテーブルを使用して変換される。これらのテーブルは、仮想化をサポートし、ハイパーバイザが、所与の仮想マシン（virtual machine、ＶＭ）（ゲストオペレーティングシステム及びそのゲストオペレーティングシステムによって制御されるアプリケーションに対応する仮想マシン）によって見られる物理メモリのビューを仮想化することを可能にする。ＯＳによって制御される変換のセットをステージ１と呼ぶ。ステージ１テーブルは、仮想アドレスを中間物理アドレス（intermediate physical address、ＩＰＡ）に変換する。ステージ１において、ＯＳは、ＩＰＡが物理アドレス空間であると考える。しかしながら、ハイパーバイザは、ステージ２と呼ばれる変換の第２のセットを制御する。この第２の変換セットは、ＩＰＡを物理アドレスに変換する。

【００５０】

ステージ１変換テーブル及びステージ２変換テーブルは、ステージ１及びステージ２についてそれぞれ変換テーブルのいくつかのレベルを含む階層テーブル構造として実装される。ステージ１テーブル及びステージ２テーブルの両方が、最大４レベルのページテーブル、すなわち、レベル０（level 0、Ｌ０）、レベル１（level 1、Ｌ１）、レベル２（level 2、Ｌ２）、及びレベル３（level 3、Ｌ３）を有することができる。

【００５１】
所与のアドレスに対する物理アドレスマッピングを見つけるために、１つ以上の変換テーブルルックアップを含む変換テーブルウォークが実行される。変換テーブルウォークは、仮想アドレスを物理アドレスに変換するのに必要なルックアップのセットである。非セキュアＥＬ１＆０変換レジームの場合、このセットは、ステージ１変換及びステージ２変換の両方のルックアップを含む。ステージ１及びステージ２ルックアップを使用して成功した変換テーブルウォークによって返される情報は、以下の通りである。
・必要な物理アドレス（中間アドレスへのステージ１マッピング及び物理アドレスへのステージ２マッピングに基づいて変換された）。
・ターゲットメモリ領域の、当該メモリ領域へのアクセスをどのように制御するかについての情報を提供する、アクセス許可及び／又はメモリ属性。これらは、ステージ１テーブル構造において定義されたステージ１アクセス許可及び／又は属性と、ステージ２テーブル構造において定義されたステージ２アクセス許可及び／又は属性とを含み得る。
【００５２】
ステージ１構造及びステージ２構造のうちの所与の１つをトラバースするために、ウォークは、変換テーブルベースアドレスレジスタ（ステージ１についてはＴＴＢＲ、ステージ２についてはＶＴＴＢＲ＿ＥＬ２）において指定されたアドレスに基づいて、初期ルックアップのためのトップレベル（Ｌ０）変換テーブルの読取りから開始する。各変換テーブルルックアップは、以下のうちの１つを示す記述子を返す。
・エントリは、ステージ１構造又はステージ２構造のトラバーサルの最終エントリであり、探索されているアドレスマッピングを提供する。エントリが最後のＬ３にある場合、このエントリはページ記述子（Page descriptor、Ｄ＿Ｐａｇｅ）と呼ばれ、一方、ウォークの最後のエントリを提供するエントリがより高いレベルのうちの１つにある場合、ブロック記述子（Block descriptor、Ｄ＿Ｂｌｏｃｋ）と呼ばれる。トラバーサルの最終エントリは、出力アドレス（ＯＡ、すなわちステージ１のＩＰＡ又はステージ２のＰＡ）と、アクセスの許可及び属性とを含む。ブロック記述子が変換テーブル構造のより高いレベルで見つかる場合、これは、ブロック記述子が、Ｌ３における単一エントリによって表される４ｋＢメモリページよりも大きいサイズのメモリ領域を表すことを意味する（Ｌ１及びＬ２におけるブロック記述子によって表される特定のサイズは、Ｌ１又はＬ２テーブルにインデックスを付けるために使用されるインデックスビットの数に依存し、この例では、Ｌ１及びＬ２ブロック記述子は、それぞれ１ＧＢ及び２ＭＢ領域を表す）。
・追加レベルのルックアップが必要とされる。この場合、エントリは、テーブル記述子（Table descriptor、Ｄ＿Ｔａｂｌｅ）と呼ばれるが、これは、更なるレベルのテーブルにおけるそのルックアップのための変換テーブルベースアドレスを提供するためである。テーブル記述子は、任意選択で、最終変換に適用することができる他の階層属性を提供することもできる。レベル１及びレベル２での変換テーブルエントリの符号化は、ブロック記述子をテーブル記述子から区別する。
・記述子は無効である。この場合、メモリアクセスは、変換失敗を生成する。
【００５３】
レジスタ状態
図４は、データ処理装置のレジスタ１４のサブセットの一例を示しており、例外復帰を処理するために使用されるアーキテクチャ状態を記憶するためのレジスタと、例外復帰状態を改ざんから保護するためのレジスタと、を含む。レジスタ１４は、データ処理命令及びロード／ストア命令の汎用オペランド及び結果を記憶するための１セットの汎用レジスタ（general purpose register、ＧＰＲ）５０を含む。プログラムカウンタ（program counter、ＰＣ）レジスタ５２は、プログラムフローの現在のポイントを表す命令のアドレスを示すプログラムカウンタ値を記憶するために設けられる。プロセッサ状態（processor state、ＰＳＴＡＴＥ）レジスタ５４は、処理回路４の現在の状態を示す１つ以上の値を記憶する。例えば、ＰＳＴＡＴＥレジスタ５４は、現在の例外レベルＥＬの表示５６を含む。ＰＳＴＡＴＥレジスタ５４はまた、復号ステージ１０で命令の復号に使

用されるように設定されるべき現在の命令セットの表示、現在のセキュリティ状態、処理回路４が動作しているプロセッサモードの他の表示、並びに後続の条件付き命令（条件付き命令は、テスト条件が満たされていることを条件とする関数を実行するように処理回路を制御する）についてテスト条件が満たされているか否かを評価するために使用され得る、以前の処理命令の結果を示すために使用される条件コードフラグを含み得る。また、以下で更に説明するように、ＰＳＴＡＴＥレジスタ５４は、例外復帰状態情報をロックして更新を防止するための例外復帰状態ロックパラメータ（ＥＸＬＯＣＫ）５８を含む。

【００５４】

レジスタ１４は、この例では保存プロセッサ状態レジスタ（ＳＰＳＲ）６０及び例外リンクレジスタ（ＥＬＲ）６２などいくつかの例外復帰レジスタを含む。この例では、ＳＰＳＲ６０及びＥＬＲ６２はバンクレジスタであり、すなわち、これらのレジスタのいくつかの別個のインスタンスが、異なる例外レベルＥＬ１～ＥＬ３に対応して提供されることを意味する。所与の例外レベルＥＬｘ（ＥＬｘは、ＥＬ１、ＥＬ２、及びＥＬ３のうちの１つである）において処理されるように例外が取られると、プログラムカウンタレジスタ５２に記憶された現在のプログラムカウンタがＥＬＲ＿ＥＬｘ６２にコピーされ、ＰＳＴＡＴＥレジスタ５４に記憶された現在のプロセッサ状態がＳＰＳＲ＿ＥＬｘ６０にコピーされる（すなわち、所与の例外レベルＥＬｘに対応する、数バージョンのＥＬＲ６２及びＳＰＳＲ６０にコピーされる）。プログラムカウンタのコピーされた値は、例外復帰アドレスと呼ぶことができる。現在のプロセッサ状態のコピーされた値は、保存プロセッサ状態と呼ぶことができる。例外復帰アドレス及び保存プロセッサ状態は、例外復帰状態情報と総称することができる。所与の例外レベルＥＬｘで処理された例外からの例外復帰時に、保存プロセッサ状態がＳＰＳＲ＿ＥＬｘ６０からＰＳＴＡＴＥレジスタ５４にコピーされ、例外復帰アドレスがＥＬＲ＿ＥＬｘ６２からプログラムカウンタレジスタ５２にコピーされて、プロセッサを例外が取られたときの状態に復元する。レジスタサフィックスＥＬｘは、命令をこれらのレジスタに書き込むことが許可される最低の特権を有する例外レベルが例外ＥＬｘであることを示す。したがって、例えば、ＳＰＳＲ＿ＥＬ２は、ＥＬ２又はＥＬ３に書き込み可能であるが、ＥＬ０又はＥＬ１には書き込み可能ではない。

【００５５】

レジスタ１４は、保護制御スタックポインタレジスタ（ＧＣＳＰＲ）６４も含み、この場合もＧＣＳＰＲ６４は、例外レベルＥＬ１、ＥＬ２、ＥＬ３のそれぞれに関連付けられた、あるバージョンのＧＣＳＰＲ＿ＥＬｘが存在するように、例外レベルごとにバンクされる。ＧＣＳポインタレジスタ６４は、例外時にＳＰＳＲ６０及びＥＬＲ６２に保存された例外復帰状態を保護するために使用される保護制御スタック（ＧＣＳ）と呼ばれるメモリ内のデータ構造を指すスタックポインタを記憶するために使用される。これは、以下でより詳細に論じられる。また、レジスタは、ＧＣＳ制御レジスタ（ＧＣＳＣＲ）６６を含み、これも例外レベルごとにバンクされて、例外レベルＥＬ１、ＥＬ２、ＥＬ３に対応するバージョンのＧＣＳＣＲ＿ＥＬｘを提供する。処理回路４が所与の例外レベルＥＬｘで動作しているとき、対応するスタックポインタ、並びにＧＣＳ制御レジスタＧＣＳＰＲ＿Ｅｌｘ及びＧＣＳＣＲ＿ＥＬｘは、ＧＣＳデータ構造の使用を制御するためのスタックポインタ及び制御状態を提供する。ＧＣＳ制御レジスタ６６は、少なくとも、ＧＣＳデータ構造の使用を保護するためのアーキテクチャ手段が有効か、又は無効かを制御するためのＧＣＳ有効化パラメータ６８と、例外復帰レジスタ６０、６２の修正を防止するための例外復帰状態ロックパラメータ（ＥＸＬＯＣＫ）５８の使用が有効か、又は無効かを制御するための例外復帰状態ロック有効化パラメータ（ＥＸＬＯＣＫＥＮ）６９と、を含む。ＧＣＳ制御レジスタ６６はまた、ＧＣＳデータ構造の使用が有効化パラメータ６８によって示されるように有効であるＧＣＳモードにあるとき、処理回路の挙動を制御するための他の状態情報を記憶することができることが理解されよう。

【００５６】

図４は、レジスタ６０、６２、６４、６６がバンクされて、処理回路４の異なる例外状態に対応する、これらのレジスタの複数のバージョンを提供する一例を示すが、これは必

須ではなく、他の例では、これらのレジスタのいずれかの単一のバージョンが提供され、例外状態間で共有され得る。その場合、ソフトウェアは、例外状態間での切り替え時に、共有レジスタに記憶された値を切り替えることができ、したがって、所与の例外状態で共有レジスタに記憶された値が、異なる例外状態で共有レジスタに記憶された値と異なり得る。レジスタ６０、６２、６４、６６のサブセットがバンクされ、レジスタ６０、６２、６４、６６の別のサブセットが例外状態間で共有されることも可能である。

【００５７】

図５は、ＧＣＳの使用をサポートしないシステムにおいて、例外エントリ及び復帰時に実行される命令の一例を示す。図５の左側は、例外エントリシーケンスを示す。例外処理回路４０によって例外が取られたことに応答して、例外処理回路４０は、処理回路４をトリガして、現在のプログラムカウンタ値及び現在のプロセッサ状態を、それぞれプログラムカウンタレジスタ５２及びＰＳＴＡＴＥレジスタ５４から、例外が取られるべき例外状態ＥＬｘに関連付けられた例外リンクレジスタ６２及びＳＰＳＲ６０にコピーする。例外の例外復帰状態をこのようにコピーすることは、ハードウェアでトリガされる。したがって、例外復帰状態をコピーさせるために実行されるべき特定の命令を必要としない。次いで、例外処理回路４０は、例外処理ルーチンを実行するように処理回路を制御する。例えば、例外エントリに応答して最初に実行される例外処理ルーチンは、例外の特定の原因を処理するためのより個別的なハンドラに分岐する前にいくつかの汎用制御機能を実行する、汎用例外ラッパーハンドラであり得る。したがって、個別のハンドラに分岐する前に、更なる例外が発生して新しい例外復帰状態がレジスタ６０、６２にコピーされる場合であっても、保存例外復帰状態が確実に保存されるようにするために、例外処理ルーチンは、命令７２を実行して、レジスタ状態情報（少なくともＳＰＳＲ６０及びＥＬＲ６２からの例外復帰状態情報を含む）をスタックなどメモリ内のソフトウェアで維持されるデータ構造にプッシュする。例外復帰状態を保存するために使用されるデータ構造が編成される特定の方法は、ソフトウェア固有の方法で選択することができ、この例外復帰状態の保存方法のアーキテクチャ上の定義が存在しない場合がある。例外復帰状態をメモリに保存すると、例外処理ソフトウェアコードは、例外の原因を処理するための個別のハンドラコードに分岐する関数呼び出し分岐命令７４を実行する。

【００５８】

例外から復帰すると、例外処理ソフトウェアコードは、メモリ内のデータ構造から以前に保存されたレジスタ状態を読み出す命令７６（例えば、スタックデータ構造からレジスタ状態をポップする）と、レジスタ状態を例外復帰レジスタ６０、６２に復元する命令７７（例えば、これは、システム制御レジスタへの書き込みが許可されたシステムレジスタ修正命令であり得る）と、レジスタ６０、６２に記憶された例外復帰状態に基づいて例外復帰をトリガする例外復帰命令７８と、を実行する。したがって、例外復帰命令７８は、以前に保存されたプロセッサ状態をＳＰＳＲ＿ＥＬｘ６０からＰＳＴＡＴＥレジスタ５４にコピーさせ、以前に保存された例外復帰アドレスをＥＬＲ＿ＥＬｘ６２からプログラムカウンタレジスタ５２にコピーさせる。これにより、以前に保存された例外復帰アドレスによって示される命令への分岐が生じ、この命令は、ＳＰＳＲ＿ＥＬｘ内の以前に保存されたプロセッサ状態値で示されるプロセッサ状態で実行される。

【００５９】

攻撃者は、例外復帰状態情報が命令７２によって書き込まれてから命令７６によって読み出されるまでメモリ内に保持されている間にこれを改ざんすることによって、復帰指向プログラミング（ＲＯＰ）攻撃を実装しようと試みることがある。攻撃者が例外復帰状態の破壊に成功することができると、これによって、例外復帰が、例外がとられたポイントからプログラムの異なるポイントに復帰すること、又はＰＳＴＡＴＥレジスタ５４によって表されるプロセッサ状態を、例外が取られたときとは異なる状態（例えば、より高い特権を有する状態）に復帰させることが可能になり得、いずれも、セキュアな情報又はソフトウェアコードを漏洩させるリスクに潜在的に曝し得る、望ましくない挙動をもたらす可能性がある。

【００６０】
図６は、例外復帰状態に対するそのようなＲＯＰ攻撃から保護するための技法を示す。例外エントリ時に、追加の命令が実行されて、レジスタ６０、６２内の例外について取得された例外復帰状態を、保護制御スタック（ＧＣＳ）データ構造８０と呼ばれる保護されたデータ構造にプッシュする。例外復帰状態がプッシュされるアドレスは、現在の例外状態ＥＬｘに関連付けられたＧＣＳスタックポインタレジスタ６４に基づいて決定される。（アドレス変換を制御するためにＭＭＵ２８によって使用される）ページテーブルによって指定されるアクセス許可は、ＧＣＳアクセス命令の制限されたサブセットによってのみアクセスされるように制限され、制限された当該サブセット内にはない他のタイプの命令を使用してアクセスされることができない、特別な「ＧＣＳ」メモリタイプとしてＧＣＳデータ構造８０を記憶するために使用されるアドレス領域を定義する。したがって、実行中のプログラムの汎用ロード／ストア命令のアドレスオペランドを改ざんする攻撃者は、命令が特別なＧＣＳアクセス命令の１つでない場合に、当該命令にＧＣＳデータ構造８０へアクセスさせることができないことがある。例えば、ＧＣＳメモリタイプは、仮想アドレス空間と中間アドレス空間との間のマッピングを提供するために使用される、対応するステージ１変換テーブルエントリによって直接的又は間接的に指定されるステージ１アクセス許可内で示され得る（例えば、ＧＣＳメモリタイプ許可は、ステージ１変換テーブルエントリ自体の符号化において、又はステージ１変換テーブルエントリによって参照されるレジスタにおいて示され得る）。逆に、ＧＣＳアクセス命令がＧＣＳメモリタイプとして指定されていないメモリの領域にアクセスしようと試みる場合、攻撃者がＧＣＳアクセス命令を欺いてＧＣＳデータ構造８０以外の領域にアクセスすることを防止するために、フォールトがシグナリングされ得る。

【００６１】
例外復帰時に、（例外復帰を制御するために例外復帰状態命令７８によって後で使用されることを意図した）図５で説明したようなメモリ内のソフトウェア固有のデータ構造からの例外復帰状態の任意のソフトウェアで制御された復元とは別に、追加のＧＣＳポップ命令が実行されて、ＧＣＳＰＲ＿ＥＬｘ６４内のＧＣＳスタックポインタに基づいて決定されたＧＣＳデータ構造８０上の位置から保護例外復帰状態をポップし、保護例外復帰状態を意図した例外復帰状態と比較し、不一致が検出された場合にはフォールトをシグナリングする。これは、意図した復帰状態が命令７６によってポップされる前にメモリ内で改ざんされたか否かを検出する手段を提供する。意図した例外復帰状態が、ＧＣＳデータ構造８０から取得された保護例外復帰状態と一致する場合、例外復帰命令７８は、意図した例外復帰状態に基づいて例外復帰をトリガすることに進むことができる。

【００６２】
したがって、図７に示すように、例外エントリコード及例外復帰コードは、例外エントリ時の追加のＧＣＳ例外状態プッシュ命令８１及び例外復帰時のＧＣＳ例外状態検証（ポップ及び比較）命令８２を含み、命令７２によるメモリへの例外復帰状態のプッシュ及び命令７６によるリードバックに対する追加の保護を提供するように拡張されている。図７は、ＧＣＳデータ構造８０から保護例外復帰状態をポップすること、及び保護例外復帰状態に対してレジスタ６０、６２内の意図した例外復帰状態を検証することの両方を行う単一の命令８２を示すが、他の例は、これを２つの別個の命令に分けることができ、１つは、メモリから保護例外復帰状態をポップするための命令であり、もう１つは、意図した例外復帰状態がポップされた例外復帰状態と一致するか否かを検証するための命令である。また、いくつかの例では、ＧＣＳ例外復帰状態ポップ命令８２は、ＧＣＳデータ構造８０から例外復帰状態をポップするだけで、例外復帰状態レジスタ６０、６２内の例外復帰状態に対して検証しなくてもよい（例えば、ＧＣＳ例外復帰状態ポップ命令８２は、ＧＣＳ保護例外復帰状態を例外復帰状態レジスタ６０、６２に直接ポップすることができ、ＧＣＳ保護例外復帰状態は、そこから例外復帰命令７８によって使用されて、例外復帰を直接制御することができる。この場合、ソフトウェアが命令７２、７６を実行して、通常の非ＧＣＳメモリ内の別個のソフトウェア管理スタックに例外復帰状態をプッシュ／ポップす

る必要はない）。
【００６３】
例外復帰状態をＧＣＳ構造８０にプッシュし、ＧＣＳ構造から例外復帰状態をポップするために提供される命令８１、８２は、メモリのＧＣＳ領域であるとしてステージ１アクセス許可において指定されたメモリ領域へのアクセスを許可されたＧＣＳアクセス命令とみなされる。
【００６４】
上述した様々なＧＣＳ関連機能は、ＧＣＳモードが有効であることを示す有効状態にＧＣＳ有効化パラメータ６８が設定されていると、実行することができることが理解されよう。ＧＣＳ有効化パラメータ６８が無効状態に設定されていて、ＧＣＳモードが無効であることを示すとき（例えば、これは、ＧＣＳモードで使用するように設計されていないレガシーコードとの互換性のために有用であり得る）、ＧＣＳアクセス命令は、アーキテクチャ機能を有しない無動作命令として処理され得るか、又は未定義命令として処理され得（ＧＣＳアクセス命令が実行されると、フォールトをシグナリングさせる）、ＧＣＳモードが有効であるときに特定の非ＧＣＳアクセス命令に対して実行され得る追加のチェックが省略され得る（以下で説明する例示的なフロー図を参照されたい）。
【００６５】
ＧＣＳモードは、所与の例外に対して取得された有効な例外復帰状態のコピーを維持するために使用されているメモリ内の保護領域をサポートするのに有用であり得、このコピーは、例外復帰を直接制御するために使用され得るか、又は例外復帰を制御するために使用される「意図した」例外復帰状態を検証するために使用され得る。
【００６６】
しかしながら、ＧＣＳ例外状態プッシュ命令８０は、例外エントリ７０に応答した例外復帰状態のハードウェアでトリガされた取得とは別個であるため、攻撃者が例外復帰状態レジスタ６０、６２の内容を修正しようと試みる場合に時間窓９０（図７において「窓Ａ」とラベル付けされる）が存在し、その結果、命令８１に応答してＧＣＳデータ構造８０にプッシュされる例外復帰状態は、例外が取られたことに応答して取得された例外復帰状態と同じではない。同様に、例外復帰状態情報がレジスタ６０、６２において修正され得る場合には、例外復帰において、ＧＣＳ例外復帰状態ポップ命令８２の実行と例外復帰命令７８の実行との間に、時間窓９２（「窓Ｂ」とラベル付けされる）が存在し、これにより、ＧＣＳ８０から取得されていないか、又はＧＣＳ８０から取得された状態に対して検証されていないプログラムカウンタ値及びプロセッサ状態に対して例外復帰がトリガされることが可能になり得る。
【００６７】
この問題に対処するために、例外復帰状態ロックパラメータ（ＥＸＬＯＣＫ）５８がＰＳＴＡＴＥレジスタ５４に提供される。ＥＸＬＯＣＫ５８は、ロック状態及びロック解除状態に設定され得る。一例では、ロック状態においてＥＸＬＯＣＫ＝１であり、ロック解除状態においてＥＸＬＯＣＫ＝０である。他の例では、逆のマッピングを使用することができ、又はＥＸＬＯＣＫ５８は、例外復帰状態ロックパラメータ及び他の情報の両方を符号化するマルチビットパラメータであり得、この場合、ＥＸＬＯＣＫ５８の状態の第１のサブセットをロック状態とみなすことができ、ＥＸＬＯＣＫ５８の状態の第２のサブセットをロック解除状態とみなすことができる。したがって、ＥＸＬＯＣＫ５８を符号化してロック状態及びロック解除状態を示すことができる様々な方法がある。
【００６８】
ＥＸＬＯＣＫがロック解除状態にあるとき、例外復帰状態レジスタへの修正が許可され得る（現在の例外レベル５６が、修正を試みられる例外復帰状態レジスタ６０、６２の特定のバンクされたバージョンを更新するのに十分な特権であるなど、他のいずれのチェックも満たされる場合）。ＥＸＬＯＣＫ５８がロック状態にあるとき、ＥＸＬＯＣＫ５８がロック解除状態にあるときに実行されると、例外復帰状態情報を修正することが別様に許可される命令を使用して試みられたとしても、レジスタ６０、６２内の例外復帰状態に記

憶された情報に対する修正は禁止される。少なくとも例外が取られる前に最新であった例外レベルと同じレベルで取られた例外について、ＥＸＬＯＣＫパラメータ５８は、図７に示す窓Ａ９０及び窓Ｂ９２中の修正に対して例外復帰状態レジスタ６０、６２をロックして、例外エントリ時に取得されてＧＣＳデータ構造８０にプッシュされる間、及びＧＣＳデータ構造８０から復元された、保護例外復帰状態に対する検証を行い、例外復帰７８に使用する間の例外復帰状態情報の改ざんを防止するように設定することができる。例外が取られた後に実行される、より高い特権を有するコードは、信頼してその復帰状態を適切に管理するように任せられるので、現在の例外レベルよりも高い特権を有する例外レベルで取られる例外は、例外を取られるときにロックパラメータの設定を必要としないと考えられ得る。

【００６９】

したがって、例外復帰状態がＧＣＳデータ構造８０に保存される前及びＧＣＳデータ構造８０から復元された後の例外復帰状態の修正を防止するために、アーキテクチャ機構が設けられる。例外が取られると、状態はロックされ、修正は不可能である。ＰＳＴＡＴＥ．ＥＸＬＯＣＫ５８は、ＥＬＲ６２及びＳＰＳＲ６０への書き込みアクセスをロックすることができ（少なくとも、レジスタ６０又は６２に記憶された値を変更しようとする書き込みに対して）、例外が取られるときに設定される。プッシュ手続き８１を実行して例外復帰状態をＧＣＳバッファ８０に保存すると、ＰＳＴＡＴＥ．ＥＸＬＯＣＫがロック解除され、レジスタ６０、６２への将来の書き込みが可能になる。ＧＣＳバッファ８０から例外復帰状態を復元するためにポップ手続き８２を実行すると、ＰＳＴＡＴＥ．ＥＸＬＯＣＫが再ロックされ、将来の書き込みを防止し、ＧＣＳから復元された状態が不変であることを保証する。ロックされると、状態（ＥＬＲ／ＳＰＳＲ）への書き込みが試みられた場合には、フォールトがシグナリングされる（これにより、更に例外を取らせ得る）。あるいは、このフォールトは、レジスタ６０、６２の一方又は両方に書き込まれている値が当該レジスタ内の現在の値と異なる場合にのみ発生し得る。これにより、復帰状態が変更されないシナリオについての高速コードパスが可能になる。

【００７０】

ＥＸＬＯＣＫパラメータ８５の追加のチェックを実行して、以下のいずれかが発生した場合にフォールトをトリガすることができる。

・ＥＸＬＯＣＫパラメータ８５が現在ロック状態にあるときの例外エントリ７０、

・ＥＸＬＯＣＫパラメータ８５が現在ロック解除状態にあるときのＧＣＳ例外状態プッシュ手続き８１、

・ＥＸＬＯＣＫパラメータ８５が現在ロック状態にあるときのＧＣＳ例外状態ポップ手続き８２、

・ＥＸＬＯＣＫパラメータ８５が現在ロック解除状態にあるときの例外復帰７８。

【００７１】

これらのイベントのいずれかが、攻撃者が制御フローを破壊して、ＧＣＳ保護命令のうちの１つを見落させるか、又はコードが意図した順序とは異なる順序で実行させたことを示し得、これは、ＲＯＰ攻撃が試みられたことを示唆し得る。

【００７２】

例外復帰状態ロック有効化制御パラメータＥＸＬＯＣＫＥＮ６９は、有効状態又は無効状態（これもまた、ＥＸＬＯＣＫ５８について上述した方法と同様の様々な方法で符号化され得る）に設定することができ、有効状態は、例外復帰状態ロックパラメータ５８を使用して上述した例外復帰状態ロック機能が有効であることを示し、無効状態は、これらの例外復帰状態ロック機能が無効であることを示す。以下に示すように、無効状態に設定されたＧＣＳ有効化パラメータ６８を使用してＧＣＳモードが無効であるか、又はＥＸＬＯＣＫＥＮ６９が無効状態に設定されるかのいずれの場合、ＥＸＬＯＣＫパラメータ５８に基づいた様々なチェック（及びＥＸＬＯＣＫパラメータ５８をロック状態又はロック解除状態に設定するアクション）を無効にすることができる。

【００７３】

図８は、現在の例外レベルＥＬｘについてＧＣＳ制御レジスタ６６に記憶されたＧＣＳ有効化パラメータ６８によってＧＣＳモードが有効であるときに発行されるメモリアクセスについて、ＭＭＵ２８によって実行されるアクセス許可チェックを示すフロー図である。ステップ１００において、ターゲットメモリ領域に対応するターゲット仮想アドレスを指定するメモリアクセス要求が発行される。ステップ１０２において、ＭＭＵ２８は、図３に示すようなステージ１アドレス変換構造及びステージ２アドレス変換構造で定義される、関連するステージ１（及び必要であればステージ２）アドレスマッピングに基づいてアドレス変換を開始する。変換用のアドレスマッピングを定義するために、任意の既知のアドレス変換技法を使用することができる。ステップ１０４において、ＭＭＵ２８は、ターゲット仮想アドレスにアドレスマッピングを提供するターゲットステージ１変換テーブルエントリによって直接的又は間接的に指定されたステージ１許可情報が、ターゲットメモリ領域がＧＣＳ領域であることを指定するか否かを判定する。例えば、ステージ１許可情報は、ターゲットステージ１変換テーブルエントリ自体に符号化され得るか、又はターゲットステージ１変換テーブルエントリによって参照されるレジスタフィールドに記憶され得る。他の例は、ターゲットステージ１変換テーブルエントリによってターゲットＶＡにマッピングされた中間アドレスに対応するステージ２変換テーブルエントリによって、（直接的に、又はレジスタへの参照を介して間接的に）指定されたステージ２許可情報を使用してＧＣＳ領域を定義することができる。また、（変換テーブル構造とは別個の）別個のデータ構造が、ＧＣＳ領域であるメモリ領域を定義することも可能である。

【００７４】

ターゲットメモリ領域がＧＣＳ領域である場合、ステップ１０６において、ＭＭＵ２８は、メモリアクセス要求が、ＧＣＳ領域へのアクセスを許可されたＧＣＳアクセスタイプの命令の制限されたサブセットのうちの１つによってトリガされたＧＣＳメモリアクセス要求であるか否かを判定する。そうでない場合、ステップ１０７において、ＭＭＵ２８は、メモリアクセス要求が書き込み要求であるか否かを判定する。メモリアクセス要求が書き込み要求である場合、ステップ１０８において、フォールトがシグナリングされて、他の汎用ロード／ストア命令タイプがＧＣＳ構造８０にアクセスできないようにする。メモリアクセス要求が読み取り要求である場合、フォールトをシグナリングする必要はなく、本方法はステップ１１０に進む（他の例は、ＧＣＳメモリ領域に対する非ＧＣＳ読み取り要求についてフォールトをシグナリングすることを選択することもできるが、読み取り要求はＧＣＳ保護例外復帰状態情報を破壊しないので、これは任意選択である）。

【００７５】

メモリアクセス要求がＧＣＳメモリアクセス要求である場合、ステップ１１０において、ＭＭＵ２８は、メモリアクセス要求について他のいずれのアクセス許可も満たされているか否かを判定する。例えば、これは、ターゲット領域に対応するステージ１変換テーブルエントリ又はステージ２変換テーブルエントリにおいて指定された他のパラメータに基づき得るか、又は例えば、処理回路４のセキュリティ状態、例外レベル、若しくは他の動作モードに基づいた他のセキュリティチェックに基づき得る。他の要求されたアクセス許可がいずれも満たされない場合には、ステップ１０８においてフォールトが再びシグナリングされる。メモリアクセス要求に対して満たす必要がある他のアクセス許可がいずれも満たされていると判定された場合、ステップ１１２において、メモリアクセス要求を許可することができる。

【００７６】

ステップ１０４において、ターゲットメモリ領域がＧＣＳ領域でなかったとＭＭＵ２８が判断した場合、ステップ１１４において、ＭＭＵ２８は、メモリアクセス要求が、ＧＣＳアクセスタイプの命令の制限されたサブセットのうちの１つによってトリガされたＧＣＳメモリアクセス要求であるか否かを判定する。これはステップ１０６におけるチェックと同じであるが、ステップ１０６と比較して反対の結果を伴う。ステップ１１４において、メモリアクセス要求がＧＣＳメモリアクセス要求である場合、ステップ１０８においてフォールトがシグナリングされる。これにより、ＧＣＳメモリアクセス要求を使用してメ

モリの非ＧＣＳ領域にアクセスすることが確実にできなくなり、ソフトウェア開発者が、攻撃者に利用可能な攻撃面を増加させる非ＧＣＳアクセスにＧＣＳアクセスタイプの命令を使用したくなることを回避させ、ＧＣＳ以外の領域を指すように攻撃者によって修正されたＧＣＳアクセスタイプの命令のアドレスオペランドの検出を可能にする。ＧＣＳアクセスタイプの命令をＧＣＳ領域へのアクセスに制限し、非ＧＣＳアクセスタイプの命令を非ＧＣＳ領域へのアクセスに制限することによって、セキュリティを向上させることができる。

【００７７】

ステップ１１４において、メモリアクセス要求がＧＣＳメモリアクセス要求ではないと判定される場合、ステップ１１６において、ＭＭＵ２８は、メモリアクセス要求について他のアクセス許可がいずれも満たされているか否かを判定する。例えば、これは、上述したように、セキュリティ状態、例外レベル又は動作モードに基づいたチェックを含み得る。全ての要求されたアクセス条件が満たされる場合、ステップ１１８において、メモリアクセス要求が許可される。いずれかのアクセス許可に違反があった場合、ステップ１０８においてフォールトがシグナリングされる。

【００７８】

図８は、いくつかの異なるチェックが失敗したことに応答してフォールトをシグナリングする単一のステップ１０８を示すが、いくつかの実装形態は、フォールトの原因に応じて異なるフォールトタイプをシグナリングし得ることが理解されよう。

【００７９】

図８は、現在の例外レベルＥＬｘに対してＧＣＳ制御レジスタ６６内のＧＣＳモード有効化パラメータ６８によってＧＣＳモードが有効である間にメモリアクセス要求が受信された場合に実行されるチェックを示す。ＧＣＳモードが現在無効である場合、ステップ１０４、１０６、１０７、１１４を省略することができ、実行されるチェックは、ステップ１１０、１１６において説明したように、ＧＣＳデータ構造８０の使用に関連しない他のアクセス許可に基づくチェックであり得る。

【００８０】

図８に示すように、ＧＣＳデータ構造８０の保護は、ＧＣＳアクセスタイプの命令の限定されたサブセットによってアクセスされ、他のロード／ストア命令によってアクセスされないことが要求される際に、当該領域内のアドレスが特別な保護を受けることを示す専用ＧＣＳメモリ領域タイプの定義を（例えば、変換テーブルを使用して）サポートするＭＭＵ２８によってアーキテクチャレベルで提供される。ＧＣＳアクセスタイプの命令は、少なくとも、図７で上述したＧＣＳ例外復帰状態プッシュ命令８１及びＧＣＳ例外復帰状態ポップ／検証命令８２を含み得る。他のタイプのＧＣＳアクセスタイプの命令、例えば、意図した例外復帰状態に対して検証することなく、ＧＣＳデータ構造８０から例外復帰状態をポップする、スタンドアロンＧＣＳ例外復帰状態ポップ命令も提供され得る。他のＧＣＳアクセスタイプの命令は、例えば、関数復帰に使用される復帰アドレスを保護するために、関数復帰状態をプッシュし、復元するために提供され得る。

【００８１】

図９は、例外エントリに応答して実行されるステップを示すフロー図である。ステップ１２０において、例外処理回路４０は、例外信号４２に基づいて、例外が取られるべきであることを検出し、例外エントリ手続きをトリガする。例外エントリに応答して、ステップ１２２において、プログラムカウンタ及び現在のプロセッサ状態が、レジスタ５２、５４から、例外が実行されるべき例外レベルＥＬｘにそれぞれ対応する、例外リンクレジスタ６２及び保存されたプロセッサ状態レジスタ６０のバージョンにコピーされる。ステップ１２４において、処理回路４又は例外処理回路４０は、プログラムカウンタレジスタ５２を更新して例外エントリプログラムコードから実行されるべき第１の命令のアドレスを示し、ＰＳＴＡＴＥレジスタ５４を更新して例外が取られるべきプロセッサ状態を示して、ＰＳＴＡＴＥレジスタ５４によって示される新しい状態で実行される例外エントリコードへの分岐を生じさせる。ステップ１２６において、処理回路４は、（ＧＣＳ有効化パラ

メータ６８に基づいて）ＧＣＳモードが現在有効であるか否かを判定し、（ＧＣＳ例外復帰状態ロック有効化パラメータＥＸＬＯＣＫＥＮ６９に基づいて）例外復帰ステータスのロックが有効か否かを判定する（使用するパラメータ６８、６９のバージョンは、現在の例外レベルＥＬｘに対応するＧＣＳ制御レジスタ６６のバンクされたバージョンのうちの１つに記憶されたバージョンである）。また、処理回路４は、例外が取られる前の現在の例外レベル（ＥＬｘ）と同じＥＬｘで例外が取られるべきか否かを判定する。ＧＣＳモードが有効であり、例外復帰状態のロックが有効であり、かつ例外が取られる前の現在の例外レベルと同じ例外レベルで例外が取られる場合、ステップ１２８において、処理回路４は、例外復帰状態ロックパラメータＥＸＬＯＣＫ５８をロック状態に設定する。そうでなければ、ＥＸＬＯＣＫ５８はステップ１２９においてロック解除状態に設定される。図９は、ＥＸＬＯＣＫ５８が同じ例外レベルＥＬで取られた例外についてのみロック状態に設定される例を示しているが、他の例は、より高い特権を有する例外レベルに取られた例外についてＥＸＬＯＣＫ５８をロック状態に設定することもできる。

【００８２】

図１０は、保護例外復帰状態をＧＣＳデータ構造８０にプッシュするための例外復帰状態プッシュ手続きを示すフロー図である。ステップ１３０において、命令デコーダ１０は、ＧＣＳ有効化パラメータ６８がＧＣＳモードは有効であることを示す有効状態である間に、ＧＣＳ例外復帰状態プッシュ命令を復号する（ＧＣＳモードが無効であった場合、ＧＣＳ例外復帰状態プッシュ命令は、無動作命令又は未定義命令として解釈され得、図１０の残りのステップは実行されない）。ステップ１３２において、処理回路４は、命令デコーダ１０によって制御されて、ＥＸＬＯＣＫＥＮ６９が、例外復帰状態のロックが有効であることを示すか否かをチェックする（ＧＣＳ制御レジスタ６６のバンクされたバージョンが異なる例外レベルに提供される場合、ステップ１３２におけるチェックに使用するパラメータ６９のバージョンは、現在の例外レベルに対応するバージョンである）。そうである場合、ステップ１３４において、処理回路４は、例外復帰状態ロックパラメータ（ＥＸＬＯＣＫ５８）が現在ロック解除状態にあるか否かをチェックするように制御される。ＥＸＬＯＣＫ５８がロック解除状態にある場合、ステップ１３６においてフォールトがシグナリングされる。これは、図７に示すように、例外復帰状態ロックが有効であるときに有効な例外エントリシーケンスについて、例外復帰状態がＧＣＳデータ構造８０にプッシュされているときには、例外復帰状態ロックパラメータ５８が現在ロックされているはずだからである（図９に示すように、例外が取られるときにＥＸＬＯＣＫ５８がロックされていない、より高い特権を有する例外レベルに取られた例外については、より高い特権を有する例外ハンドラを信頼して例外復帰状態の管理を任せることができ、したがって、ＧＣＳ例外復帰状態プッシュ命令の実行時にＥＸＬＯＣＫ５８がロックされるための要件は、これらのより高い特権を有する例外ハンドラについては問題を引き起こさないため、ＧＣＳ例外復帰状態プッシュ命令を実行する必要が全くない）。ＥＸＬＯＣＫ５８が現在ロック解除状態にある場合にフォールトをシグナリングすることは、例外エントリの後以外にＧＣＳプッシュ手続きを実行させるプログラムフローの改ざんに基づいた攻撃に対する対策を提供する。したがって、ステップ１３４におけるチェックは、攻撃者によって制御される任意の例外復帰状態（例外エントリに応答して取得される例外復帰状態ではない）がＧＣＳデータ構造８０にプッシュされるリスクを低減する。

【００８３】

ステップ１３２において例外復帰状態ロックが無効であることが判明したか（ＥＸＬＯＣＫＥＮ６９が無効状態に設定された）、又はステップ１３４においてＥＸＬＯＣＫ５８がロック状態である場合、ステップ１３８において、処理回路４は例外復帰状態レジスタ６０、６２から例外復帰状態情報を取得し、ステップ１４０において、処理回路４は、１つ以上のメモリアクセス要求を発行して、現在の例外レベルＥＬｘに対応するＧＣＳポインタレジスタＧＣＳＰＲ＿ＥＬｘ６４に記憶された現在のスタックポインタに基づいて決定されたＧＣＳデータ構造８０上の位置に取得した例外復帰状態情報をプッシュする。

【００８４】

ステップ１４０において、処理回路４は、ＥＸＬＯＣＫＥＮ６９が、例外復帰状態のロックが有効であることを示すか否かをチェックし、例外復帰状態のロックが有効である場合、ステップ１４４に進む（あるいは、ステップ１３２において実行される同じチェックが、例外復帰状態のロックが有効であるか無効であるかに応じてステップ１４４を条件付きで実行させ得、したがって、ＥＸＬＯＣＫＥＮパラメータ６９を再びチェックする必要がない場合がある）。例外復帰状態のロックが有効の場合、ステップ１４４において、例外復帰状態ロックパラメータ５８はロック解除状態に設定される（例外復帰状態のロックが無効の場合、このステップはスキップされる）。例外復帰状態のロックが有効であるか無効であるかにかかわらず、ステップ１４６において、処理回路４は、現在の実行状態（例外レベル）ＥＬｘに対応するＧＣＳポインタレジスタ６４内のスタックポインタを更新して、例外復帰状態のスタックへのプッシュを説明する（例えば、スタックポインタは、レジスタ６０、６２の内容に基づいて形成された例外復帰状態レコードのサイズに対応する量だけデクリメント又はインクリメントされ得る）。

【００８５】

図１１は、ＧＣＳデータ構造８０から例外復帰状態をポップするために実行されるステップを示すフロー図である。ステップ１５０において、命令デコーダ１０は、ＧＣＳモードがＧＣＳ有効化パラメータ６８によって有効であると示されている間に、ＧＣＳ例外復帰状態ポップ命令を復号する（この場合も、ＧＣＳモードが無効である場合、ＧＣＳ例外復帰状態ポップ命令は、ＮＯＰ（無動作命令）又は未定義命令として処理され得、図１１の残りのステップは実行され得ない）。

【００８６】

ＧＣＳ例外復帰状態ポップ命令に応答して、ステップ１５２において、処理回路４は、ＥＸＬＯＣＫＥＮ６９が、例外復帰状態のロックが有効であることを示す有効状態に設定されているか否かをチェックする（ＧＣＳ制御レジス６６のバンクされたバージョンが異なる例外レベルに提供される場合、ステップ１５２におけるチェックに使用するパラメータ６９のバージョンは、現在の例外レベルに対応するバージョンである）。例外復帰状態のロックが有効である場合、ステップ１５４において、処理回路４は、ＥＸＬＯＣＫパラメータ５８がロック状態に設定されているか否かをチェックする。ＥＸＬＯＣＫ５８が現在ロック状態にある場合には、ステップ１５６においてフォールトがシグナリングされる。これは、ＥＸＬＯＣＫがロックされているときに例外復帰状態ポップ命令が実行されるべきではなく、予想される使用法からの逸脱を示すためである。このフォールトは、プログラムフローを破壊して、ＧＣＳデータ構造８０からの保護例外復帰状態の不正確な検証を生じさせようと試みる潜在的な攻撃をシグナリングすることができる。

【００８７】

ステップ１５２において例外復帰状態のロックが無効であると判定されたか、又はステップ１５４においてＥＸＬＯＣＫ５８が現在ロック解除状態にある場合、ステップ１５８において、処理回路４は１つ以上のメモリアクセス要求を発行して、現在の例外レベルＥＬｘに対応するＧＣＳポインタレジスタ６６内のスタックポインタに基づいて決定されるＧＣＳデータ構造８０上の位置からＧＣＳ保護復帰状態情報をポップする。ステップ１６０において、レジスタＧＣＳＰＲ＿ＥＬｘ６６内のＧＣＳスタックポインタは、ＧＣＳデータ構造８０からの例外復帰状態のポップを説明するために更新される（例えば、スタックポインタは、図１０のステップ１４６で行われた更新とは反対方向に（ただし、対応する量だけ）インクリメント又はデクリメントされる）。

【００８８】

ステップ１６２、１６４、及び１６６は、ＧＣＳ例外復帰状態ポップ命令が処理回路４も制御して、例外復帰レジスタ６０、６２に現在記憶されている、意図した例外復帰状態情報が、ステップ１５８においてＧＣＳデータ構造８０からポップされたＧＣＳ保護例外復帰状態情報と一致するか否かを検証させる実装形態において、ＧＣＳ例外復帰状態ポップ命令に応答して任意選択的に実行され得るステップである。ステップ１６２において、処理回路４は、現在の例外レベルＥＬｘに対応する例外リンクレジスタ６２及びＳＰＳＲ

６０内の現在の値によって表される、意図した例外復帰状態情報を読み取る。「意図した」例外復帰状態情報という用語は、ＧＣＳ例外復帰状態ポップ命令の実行時のレジスタ６２、６０内の例外復帰状態の現在の値（対応する例外エントリが行われたときにそれらのレジスタに書き込まれた値と同じであってもなくてもよい）を指すために使用される。ステップ１６４において、処理回路４は、レジスタ６０、６２から取得した、意図した例外復帰状態情報が、メモリの保護ＧＣＳ領域に記憶されたＧＣＳデータ構造８０から読み取られたＧＣＳ保護例外復帰状態情報と一致するか否かを判定し、不一致がある場合には、ステップ１６６においてフォールトがシグナリングされる。

【００８９】

ステップ１６４において、意図した例外復帰状態情報がＧＣＳ保護例外復帰状態情報と一致する場合、又はＧＣＳ例外復帰状態ポップ命令が検証をトリガしない（ただし、その代わりに、例えば、ＧＣＳ保護復帰状態情報を例外復帰状態レジスタ６０、６２に直接ポップする）ために、ステップ１６２、１６４、１６６が実施されない場合、ステップ１６８において、処理回路４は、ＥＸＬＯＣＫＥＮ６９が例外復帰状態のロックが有効であることを示すか否かをチェックし、例外復帰状態のロックが有効である場合、ステップ１７０に進む（あるいは、ステップ１５２において実行される同じチェックは、例外復帰状態のロックが有効であるか、又は無効であるかに応じて、ステップ１７０を条件付きで実行させ得る）。例外復帰状態のロックが有効の場合、ステップ１７０において、例外復帰状態ロックパラメータ５８はロック状態に設定される（例外復帰状態のロックが無効の場合、このステップはスキップされる）。

【００９０】

図１２は、例外復帰命令に応答して実行されるステップを示すフロー図である。ステップ１８０において、命令デコーダ１０は、例外復帰命令を復号し、それに応答して、後続のステップを実行するように処理回路４（具体的には実行ステージ１６）を制御するための制御信号を生成する。ステップ１８２において、処理回路４は、ＧＣＳ有効化パラメータ６８が、ＧＣＳモードは現在有効であることを示すか否か、ＥＸＬＯＣＫＥＮパラメータ６９は、例外復帰状態のロックも有効であることを示すか否か（この場合も、ＧＣＳ制御レジスタ６６のバンクされたバージョンが異なる例外レベルに提供される場合、ステップ１８２においてチェックに使用するパラメータ６８、６９のバージョンは、現在の例外レベルに対応するバージョンである）、及び例外復帰が現在の例外レベルと同じ例外レベルにおける処理への復帰であるか否かを判定する。ＧＣＳモードと例外復帰状態のロックの両方が有効であり、例外復帰によって現在の例外レベルで実行される処理に復帰する場合、ステップ１８４において、処理回路４は、ＥＸＬＯＣＫパラメータ５８が現在ロック解除状態にあるか否かをチェックする。ＥＸＬＯＣＫ５８が現在ロック解除状態にある場合、ステップ１８６においてフォールトがシグナリングされる。これは、例外復帰が先行する例外復帰状態ポップ命令８２に従わなかった可能性があり、したがって、レジスタ６０、６２において指定された、潜在的に安全でない例外復帰状態を使用して例外復帰を実行することが安全でない可能性があることを示すためである。

【００９１】

ＧＣＳモードが現在無効であるか、又はＥＸＬＯＣＫＥＮパラメータ６９が例外復帰状態のロックを無効にするように設定されているか、又は例外復帰が現在の例外レベルよりも低い特権を有する例外レベルに対してであるか、又はＥＸＬＯＣＫパラメータ５８が現在ロック状態にある場合、例外復帰を進めることができ、次いでステップ１８８において、例外復帰アドレスがＥＬＲ＿ＥＬｘ６２からプログラムカウンタレジスタ５２にコピーされ、保存されたプロセッサ状態がＳＰＳＲ＿ＥＬｘ６０からＰＳＴＡＴＥレジスタ５４にコピーされる。ＥＸＬＯＣＫパラメータ５８は、例外時に保存され、例外復帰時に復元されるプロセッサ状態の一部であるので、ステップ１８８は、例外復帰状態ロックパラメータ５８を、対応する例外がとられるときにＥＸＬＯＣＫ５８によって示される同じ状態に変更することができることに留意されたい。したがって、例外復帰後にＥＸＬＯＣＫ５８がロック状態にあるか、又はロック解除状態にあるかは、対応する例外が取られたとき

のＥＸＬＯＣＫの状態に依存する。

【００９２】

図１３は、レジスタ６０、６２内の例外復帰状態情報の修正が許可され得るか否かを制御するステップを示すフロー図である。ステップ２００において、命令デコーダ１０は、ターゲット例外レベルに関連付けられた例外リンクレジスタ６２又は保存プロセッサ状態レジスタ６０の一方又は両方における例外復帰状態情報（例えば、例外復帰アドレス又は保存プロセッサ状態）の修正を要求する命令を復号する（より高い特権を有する例外レベルで実行される命令が、より低い特権を有する例外レベルに関連付けられた例外復帰状態を修正することができるため、ターゲット例外レベルは現在の例外レベル５６であってもなくてもよい）。ステップ２０２において、処理回路４は、有効化パラメータ６８及びＥＸＬＯＣＫＥＮパラメータ６９によって有効と示されるＧＣＳモードが、例外復帰状態のロックが有効であることを示すか否かを判定する。ＧＣＳモードが無効であるか、又は例外復帰ステータスのロックが無効である場合、ステップ２０４において、他のいずれのチェック（現在の例外レベルが指定されたレジスタの修正を可能にするのに十分な特権を有するか否かのチェックなど）も満たされていれば、例外復帰状態情報への修正が許可され得る。ステップ２０２において、ＧＣＳモード及び例外復帰状態のロックの両方が有効であると判定された場合、ステップ２０６において、処理回路４は、例外復帰状態ロックパラメータＥＸＬＯＣＫ５８が現在ロック状態にあるか否かを判定する。ＥＸＬＯＣＫ５８がロック解除状態にある場合、この場合もステップ２０４において、他のいずれのチェックも満たされていれば修正が許可され得る。ステップ２０６において、ＥＸＬＯＣＫパラメータ５８がロック状態にある場合、ステップ２０８において、フォールトがシグナリングされて、例外復帰状態情報の修正を防止する（したがって、図７に示すように、窓Ａ９０又は窓Ｂ９２において例外復帰状態を改ざんしようとする試みを監視する）。

【００９３】

図１３の例では、ステップ２０８におけるフォールトのシグナリングは、レジスタ６０、６２のうちの１つで例外復帰状態情報に書き込まれるべき値が、当該レジスタに記憶された前の値と同じであるか、又は異なるかとは無関係である。したがって、ステップ２００で遭遇した命令の実行が実際には例外復帰状態を変更させなかった場合であっても、フォールトがシグナリングされる。この手法は、レジスタ６０、６２への読み出し、及び書き込まれるべき新しい値との比較に依存してフォールトのシグナリングを行うための回路ロジックを設ける必要がないので、ハードウェアでの実施がより簡単であり得る。

【００９４】

代替例を図１４に示す。図１４のステップ２００、２０２、２０４、２０６及び２０８は、図１３の対応するステップと同じである。図１４は、ステップ２１０において、レジスタ６０、６２のうちの１つに書き込まれるべき新しい値が、当該レジスタ内の例外復帰状態情報の古い値と異なるか否かをチェックする、追加のチェックを含む。新しい値が古い値と異なる場合には、ステップ２０８においてフォールトがシグナリングされる。新しい値が古い値と同じである場合、ステップ２０４において、試みられた「修正」が例外復帰状態情報を変更しないので許可され得る。この手法は、いくつかのコードパス、例えば、非常に短い例外処理ルーチン（例外を取ったほぼ直後に例外から復帰する）を使用する高速例外パスを高速化することができ、したがって、例外復帰状態をメモリに保存／復元する必要がなく、攻撃者による改ざんのリスクを低減する。そのような高速例外コードパスの場合、ＧＣＳ例外復帰状態プッシュ／ポップ命令が使用されないことがあり、したがって、ＥＸＬＯＣＫは例外ルーチンを通して設定されたままであり得る。それにもかかわらず、例外処理ルーチンが、レジスタ６０、６２内の例外復帰状態を修正しようとするシステムレジスタ修正命令を実行する場合（可能性は低いが、可能である）、ＥＸＬＯＣＫがロックされている間にトリガされることになる、対応するフォールトは、システムレジスタ修正命令によって書き込まれた新しい値が実際に古い値と同じであり、したがって例外復帰状態を破壊するリスクはない場合に回避することができる。これは、そのようなコードパスの性能を改善するのに役立ち得る。

【００９５】
上述したフロー図は、特定の順序で実行されるステップを示しているが、同じ機能を異なる順序で、又は少なくとも部分的に並行して実行することもできることが理解されよう。
【００９６】
図１５は、使用され得るシミュレータ実装を例示する。上記の実施形態は、当該技法をサポートする特定の処理ハードウェアを動作させる装置及び方法の点において本発明を実装しているが、コンピュータプログラムの使用によって実装される本明細書に記載の実施形態による命令実行環境を提供することも可能である。このようなコンピュータプログラムは、コンピュータプログラムがハードウェアアーキテクチャのソフトウェアベースの実装を提供する限り、シミュレータと称されることがよくある。様々なシミュレータコンピュータプログラムは、エミュレータ、仮想マシン、モデル、及び動的バイナリトランスレータを含むバイナリトランスレータを含む。典型的に、シミュレータ実装は、ホストプロセッサ３３０で実行され得、任意選択的にホストオペレーティングシステム３２０を実行し、シミュレータプログラム３１０をサポートする。いくつかの配置では、ハードウェアと提供されている命令実行環境との間に複数のレイヤのシミュレーションがあってもよく、及び／又は、同じホストプロセッサ上に提供されている複数の異なる命令実行環境があってもよい。歴史的に、強力なプロセッサが、合理的な速度で実行されるシミュレータ実装を提供するのに必要とされてきたが、このような手法は、互換性又は再使用の理由から別のプロセッサにネイティブなコードを実行することが望まれるようなときなど一定の状況では、正当化され得る。例えば、シミュレータ実装は、ホストプロセッサハードウェアによってサポートされていない追加の機能性を有する命令実行環境を提供し得るか、又は典型的には異なるハードウェアアーキテクチャに関連付けられた命令実行環境を提供し得る。シミュレーションの概要は、「Ｓｏｍｅ　Ｅｆｆｉｃｉｅｎｔ　Ａｒｃｈｉｔｅｃｔｕｒｅ　Ｓｉｍｕｌａｔｉｏｎ　Ｔｅｃｈｎｉｑｕｅｓ」、Ｒｏｂｅｒｔ　Ｂｅｄｉｃｈｅｋ、１９９０年冬　ＵＳＥＮＩＸ　Ｃｏｎｆｅｒｅｎｃｅ、５３～６３頁に記載されている。
【００９７】
これまで、特定のハードウェア構成物又は特徴を参照して実施形態を説明してきたが、シミュレーションされた実施形態では、好適なソフトウェア構成物又は特徴によって同等の機能性を提供することができる。例えば、特定の回路は、シミュレーションされた実施形態で、コンピュータプログラムロジックとして実装されてもよい。同様に、レジスタ又はキャッシュなどのメモリハードウェアは、シミュレーションされた実施形態で、ソフトウェアのデータ構造として実装されてもよい。前述の実施形態で参照されているハードウェア要素のうちの１つ以上がホストハードウェア（例えば、ホストプロセッサ３３０）に存在する配置では、いくつかのシミュレーションされた実施形態は、好適な場合、ホストハードウェアを使用し得る。
【００９８】
シミュレータプログラム３１０は、コンピュータ可読記憶媒体（非一時的媒体であり得る）に記憶されてもよく、シミュレータプログラム３１０によってモデル化されているハードウェアアーキテクチャのインターフェースと同じであるターゲットコード３００（アプリケーション、オペレーティングシステム、及びハイパーバイザを含み得る）にプログラムインターフェース（命令実行環境）を提供する。よって、上述のＧＣＳアクセス命令を含む、ターゲットコード３００のプログラム命令は、シミュレータプログラム３１０を使用する命令実行環境内から実行され得、これによって、上述の装置２のハードウェア機能を実際には有していないホストコンピュータ３３０が、これらの機能をエミュレートできるようになる。同様に、ＥＸＬＯＣＫパラメータ５８に基づいたチェックは、シミュレータプログラム３１０を使用して（例えば、処理プログラムロジック３１２又は命令復号プログラムロジック３１４の一部を使用して）エミュレートすることができる。
【００９９】

したがって、シミュレータプログラム３１０は、上述した処理回路４の状態をシミュレートする処理プログラムロジック３１２を有し得る。例えば、処理プログラムロジック３１２は、ターゲットコード３００のシミュレートされた実行中に発生するイベントに応答して、実行状態ＥＬ０～ＥＬ３の遷移を制御することができる。命令復号プログラムロジック３１４は、ターゲットコード３００の命令を復号し、ホスト装置３３０のネイティブ命令セットで対応する命令セットにマッピングする。レジスタエミュレートプログラムロジック３１６は、ターゲットコードによって要求されたレジスタアクセスを、ホスト装置３３０のレジスタ又はメモリ３３２内のデータにアクセスすることなどによって、ホスト装置３３０のホストハードウェア上に維持された対応するデータ構造へのアクセスにマッピングする。メモリ管理プログラムロジック３１８は、アドレス変換、ページテーブルウォーク、及びアクセス制御チェックを、上記のハードウェア実装実施形態で説明したＭＭＵ２８に対応する方法で実装するが、ターゲットコード３００に対して定義されたページテーブルに基づくアドレス変換におけるステージ２マッピングによって得られたシミュレートされた物理アドレスを、ホストメモリ３３２にアクセスするために使用されるホスト仮想アドレスにマッピングする追加機能も有する。これらのホスト仮想アドレス自体は、ホストによってサポートされる標準アドレス変換機構を使用してホスト物理アドレスに変換することができる（ホスト仮想アドレスのホスト物理アドレスへの変換は、シミュレータプログラム３１０によって制御されるものの範囲外である）。例外処理プログラムロジック３２０は、ターゲットコード３００の実行中に発生するシミュレートされた例外（上述のＥＸＬＯＣＫベースのチェックによってトリガされたフォールトなど）の処理を制御する。

【０１００】

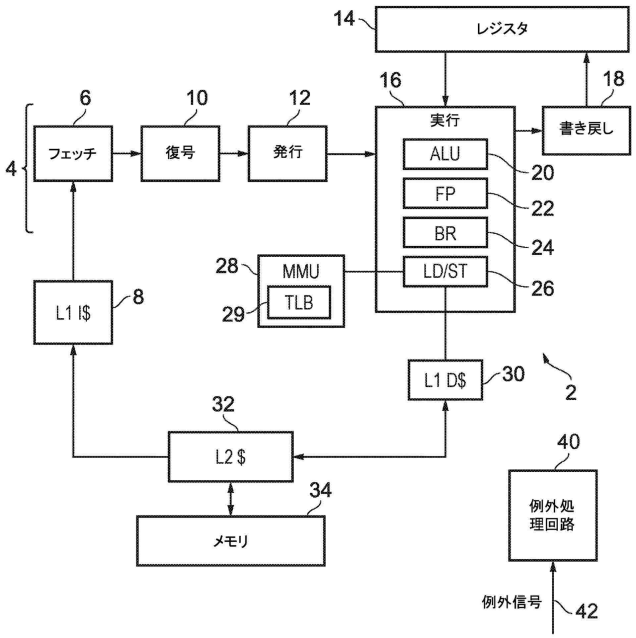
本出願において、「～ように構成された（configured to...）」という用語は、装置の要素が、定義された動作を行うことが可能である構成を有することを意味するために使用される。この文脈において、「構成」とは、ハードウェア又はソフトウェアの配置又は相互接続の方法を意味する。例えば、装置は、定義された動作を提供する専用ハードウェアを有し得るか、又はプロセッサ若しくは他の処理デバイスが、機能を実施するようにプログラムされ得る。「ように構成された」は、装置要素が、定義された動作を提供するために何らかの変更がなされる必要があることを意味しない。

【０１０１】

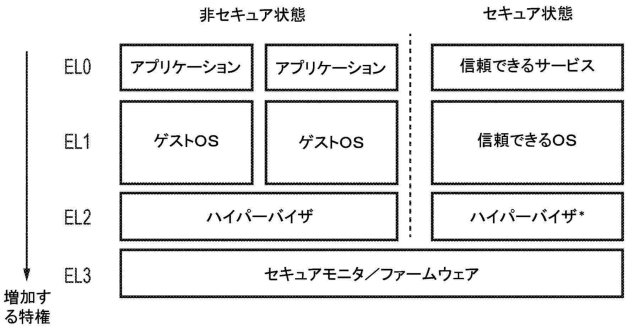
本発明の例解的な実施形態が添付の図面を参照して本明細書で詳細に説明されているが、本発明はこれらの正確な実施形態に限定されないこと、及び様々な変更及び修正が、当業者によって、添付の特許請求の範囲によって定義されている本発明の範囲から逸脱することなく、実施形態に行われ得ることが理解されよう。

【図面】

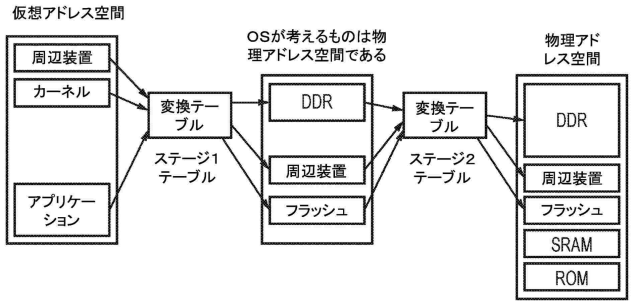
【図１】


14
レジスタ
16
実行
18
書き戻し
6
10
12
4
フェッチ
復号
発行
ALU
20
FP
22
BR
24
LD/ST
26
28
MMU
29
TLB
L1 I$
8
L1 D$
30
2
32
L2 $
40
例外処理回路
34
メモリ
例外信号
42

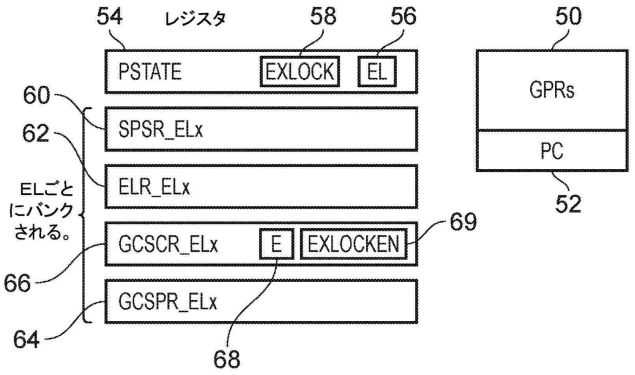
【図２】


非セキュア状態
セキュア状態
EL0
アプリケーション
アプリケーション
信頼できるサービス
EL1
ゲストOS
ゲストOS
信頼できるOS
EL2
ハイパーバイザ
ハイパーバイザ*
EL3
セキュアモニタ／ファームウェア
増加する特権

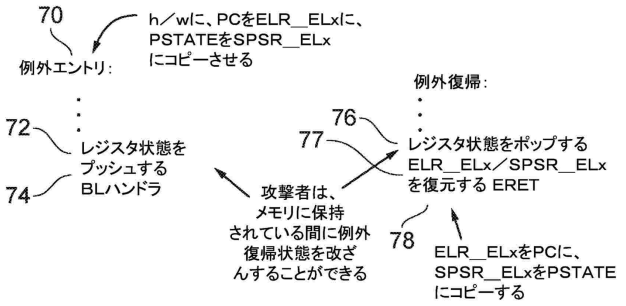
【図３】


仮想アドレス空間
周辺装置
カーネル
アプリケーション
変換テーブル
ステージ1テーブル
OSが考えるものは物理アドレス空間である
DDR
周辺装置
フラッシュ
変換テーブル
ステージ2テーブル
物理アドレス空間
DDR
周辺装置
フラッシュ
SRAM
ROM

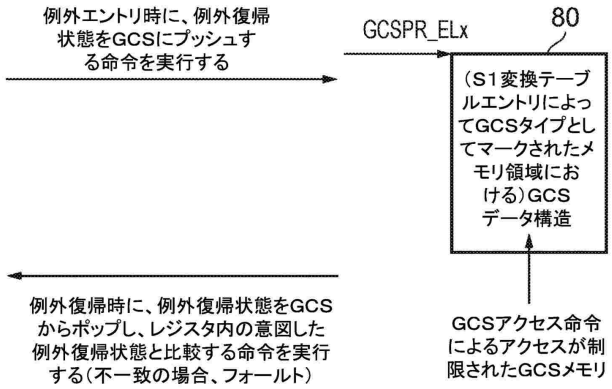
【図４】


54
レジスタ
58
56
50
PSTATE
EXLOCK
EL
GPRs
PC
60
SPSR_ELx
62
ELR_ELx
ELごとにバンクされる。
GCSCR_ELx
E
EXLOCKEN
69
66
GCSPR_ELx
64
68
52

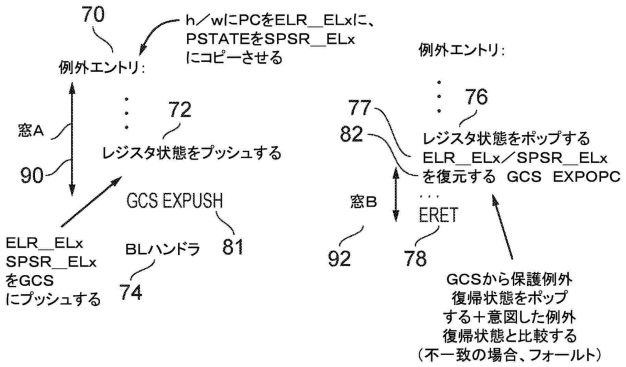
【図５】


70
h／wに、PCをELR_ELxに、PSTATEをSPSR_ELxにコピーさせる
例外エントリ：
72
レジスタ状態をプッシュする
74
BLハンドラ
攻撃者は、メモリに保持されている間に例外復帰状態を改ざんすることができる
例外復帰：
76
77
レジスタ状態をポップするELR_ELx／SPSR_ELxを復元するERET
78
ELR_ELxをPCに、SPSR_ELxをPSTATEにコピーする

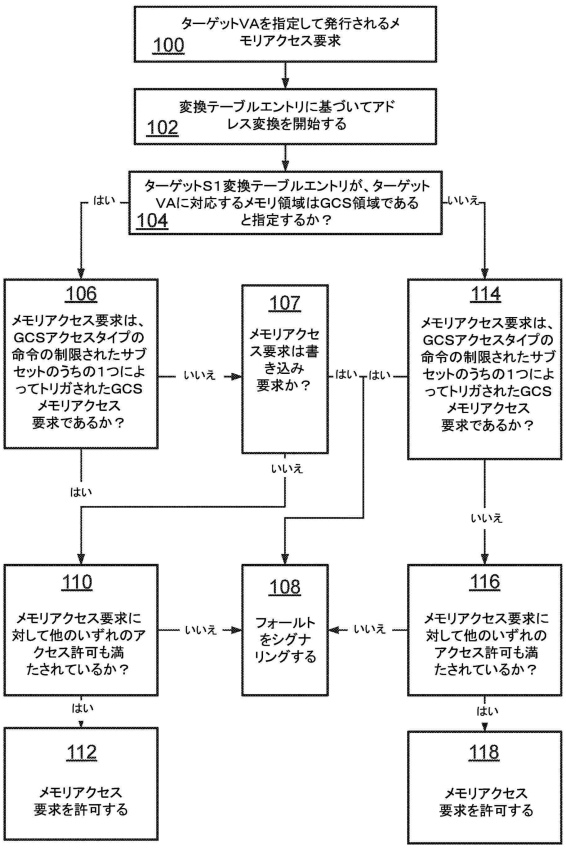
【図６】


例外エントリ時に、例外復帰状態をGCSにプッシュする命令を実行する
GCSPR_ELx
80
（S1変換テーブルエントリによってGCSタイプとしてマークされたメモリ領域における）GCSデータ構造
例外復帰時に、例外復帰状態をGCSからポップし、レジスタ内の意図した例外復帰状態と比較する命令を実行する（不一致の場合、フォールト）
GCSアクセス命令によるアクセスが制限されたGCSメモリ

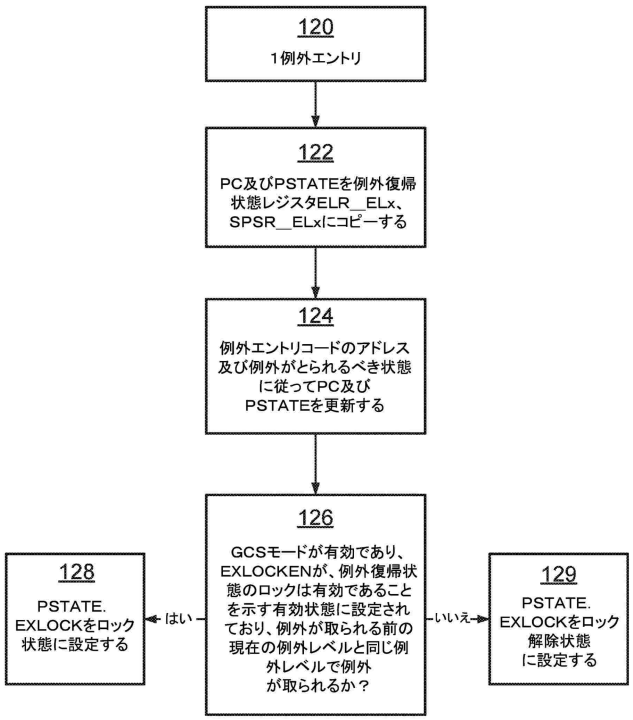
【図７】


70
h／wにPCをELR_ELxに、PSTATEをSPSR_ELxにコピーさせる
例外エントリ：
窓A
72
レジスタ状態をプッシュする
90
GCS EXPUSH
81
ELR_ELx SPSR_ELxをGCSにプッシュする
BLハンドラ
74
例外エントリ：
77
76
82
レジスタ状態をポップするELR_ELx／SPSR_ELxを復元する GCS EXPOPC
窓B
ERET
92
78
GCSから保護例外復帰状態をポップする＋意図した例外復帰状態と比較する（不一致の場合、フォールト）

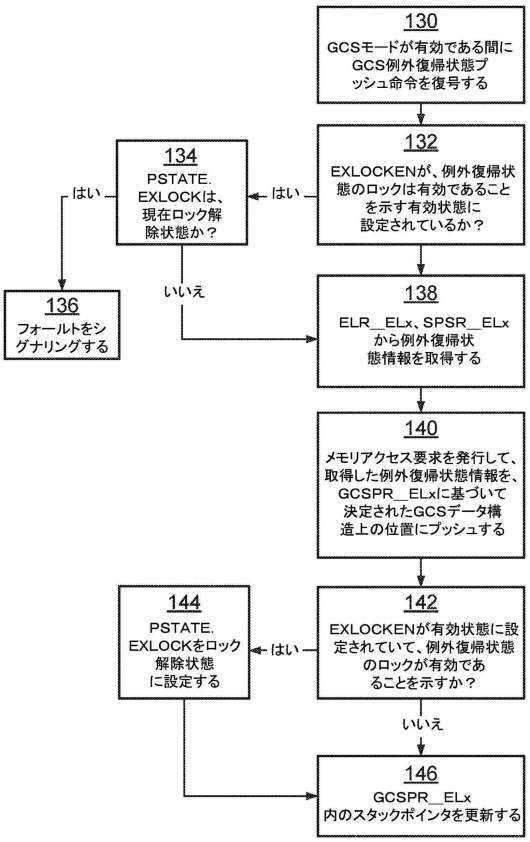
【図８】


ターゲットVAを指定して発行されるメモリアクセス要求
100
変換テーブルエントリに基づいてアドレス変換を開始する
102
ターゲットS1変換テーブルエントリが、ターゲットVAに対応するメモリ領域はGCS領域であると指定するか？
104
はい
いいえ
106
メモリアクセス要求は、GCSアクセスタイプの命令の制限されたサブセットのうちの1つによってトリガされたGCSメモリアクセス要求であるか？
107
メモリアクセス要求は書き込み要求か？
114
メモリアクセス要求は、GCSアクセスタイプの命令の制限されたサブセットのうちの1つによってトリガされたGCSメモリアクセス要求であるか？
110
メモリアクセス要求に対して他のいずれのアクセス許可も満たされているか？
108
フォールトをシグナリングする
116
メモリアクセス要求に対して他のいずれのアクセス許可も満たされているか？
112
メモリアクセス要求を許可する
118
メモリアクセス要求を許可する

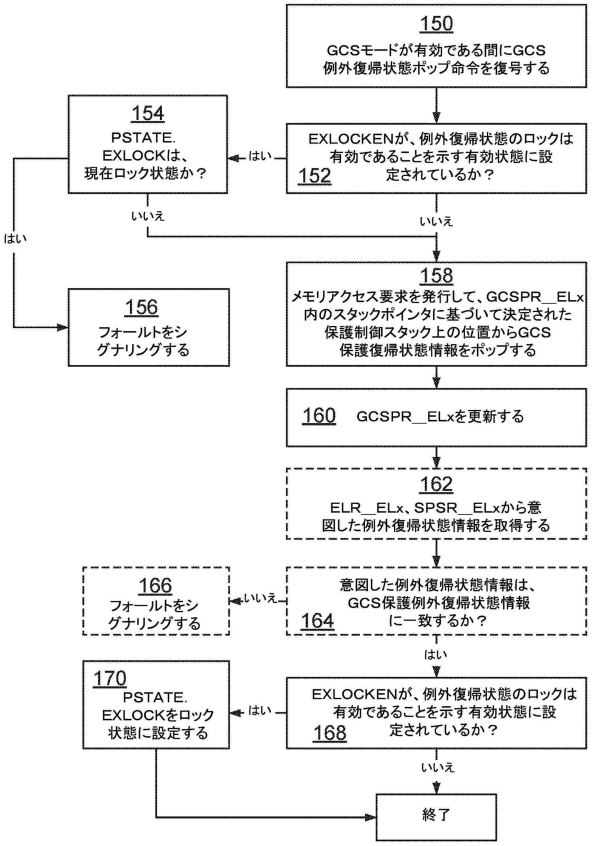
【図９】



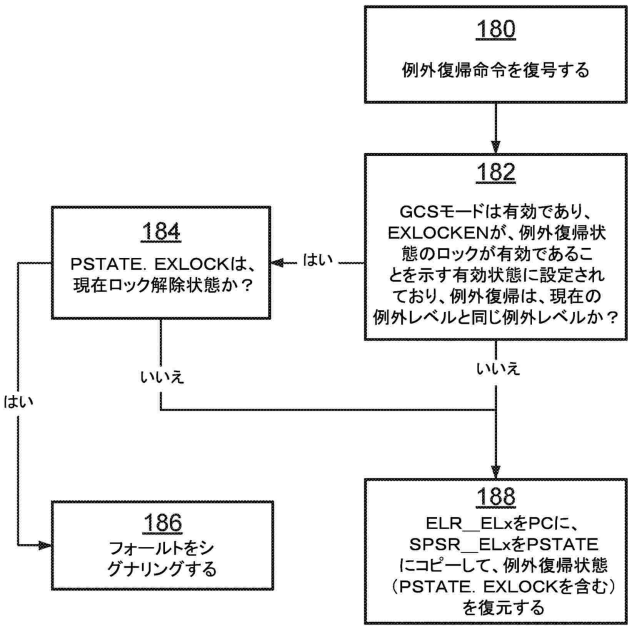
【図１０】



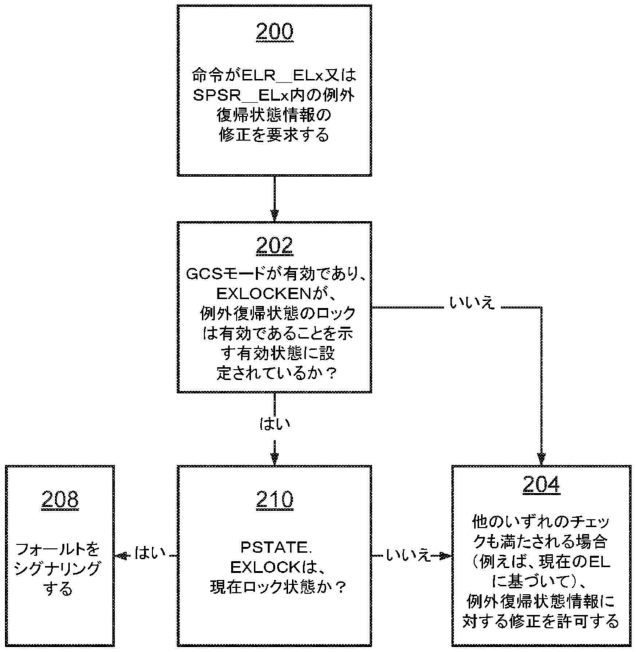
【図１１】



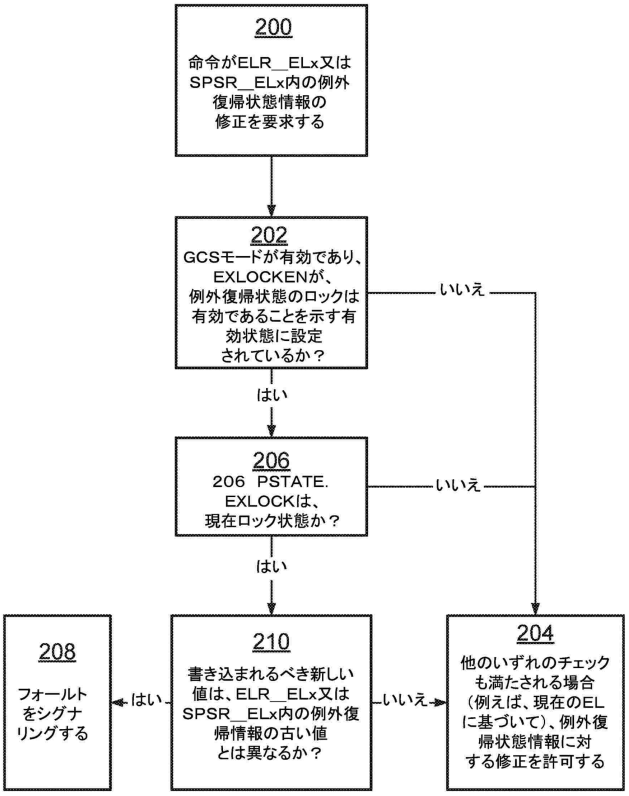
【図１２】



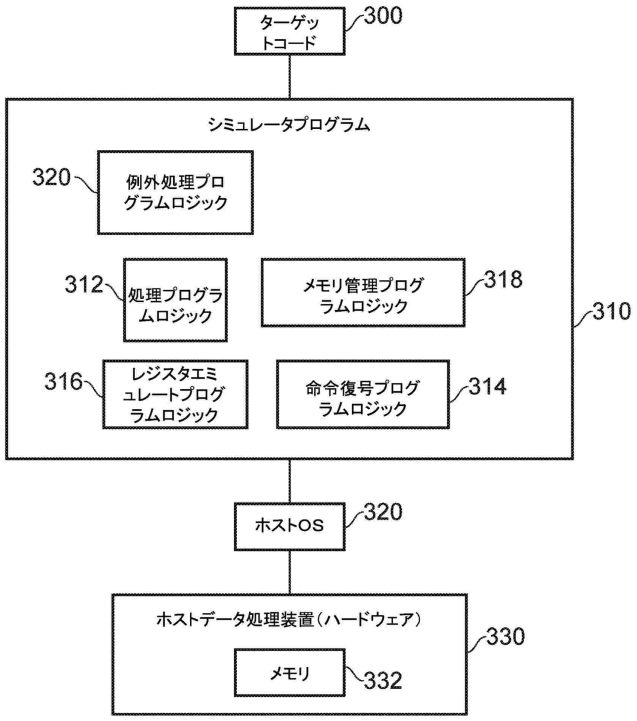
【図１３】


200
命令がELR_ELx又はSPSR_ELx内の例外復帰状態情報の修正を要求する
202
GCSモードが有効であり、EXLOCKENが、例外復帰状態のロックは有効であることを示す有効状態に設定されているか？
いいえ
はい
208
フォールトをシグナリングする
はい
210
PSTATE.EXLOCKは、現在ロック状態か？
いいえ
204
他のいずれのチェックも満たされる場合（例えば、現在のELに基づいて）、例外復帰状態情報に対する修正を許可する

【図１４】


200
命令がELR_ELx又はSPSR_ELx内の例外復帰状態情報の修正を要求する
202
GCSモードが有効であり、EXLOCKENが、例外復帰状態のロックは有効であることを示す有効状態に設定されているか？
いいえ
はい
206
206 PSTATE.EXLOCKは、現在ロック状態か？
いいえ
はい
208
フォールトをシグナリングする
はい
210
書き込まれるべき新しい値は、ELR_ELx又はSPSR_ELx内の例外復帰情報の古い値とは異なるか？
いいえ
204
他のいずれのチェックも満たされる場合（例えば、現在のELに基づいて）、例外復帰状態情報に対する修正を許可する

【図１５】


ターゲットコード
300
シミュレータプログラム
320
例外処理プログラムロジック
312
処理プログラムロジック
メモリ管理プログラムロジック
318
310
316
レジスタエミュレートプログラムロジック
命令復号プログラムロジック
314
ホストOS
320
ホストデータ処理装置（ハードウェア）
330
メモリ
332

【国際調査報告】

INTERNATIONAL SEARCH REPORT

International application No
PCT/GB2023/050653

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06F12/14 G06F9/30 G06F12/1009 G06F9/38
ADD.

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)
G06F

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 practicable, search terms used)
EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2018/088949 A1 (DUVALSAINT KARL J [US] ET AL) 29 March 2018 (2018-03-29) paragraph [0076] - paragraph [0104] paragraph [0202] - paragraph [0224]	1-18
A	US 2010/088705 A1 (ATTINELLA JOHN E [US] ET AL) 8 April 2010 (2010-04-08) paragraph [0030] - paragraph [0036] paragraph [0043] - paragraph [0052]	1-18
A	US 2020/371966 A1 (PARKER JASON [GB] ET AL) 26 November 2020 (2020-11-26) paragraph [0203] - paragraph [0212] paragraph [0071] - paragraph [0083]	1-18

☐ Further documents are listed in the continuation of Box C. ☒ See patent family 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 application or patent 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
14 June 2023	22/06/2023

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, Fax: (+31-70) 340-3016	Toader, Elena Lidia

Form PCT/ISA/210 (second sheet) (April 2005)

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/GB2023/050653

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2018088949 A1	29-03-2018	CN 108475223 A	31-08-2018
		DE 112016005571 T5	16-08-2018
		GB 2562666 A	21-11-2018
		JP 6899614 B2	07-07-2021
		JP 2019506666 A	07-03-2019
		US 9606855 B1	28-03-2017
		US 2017192836 A1	06-07-2017
		US 2018088949 A1	29-03-2018
		WO 2017118893 A1	13-07-2017
US 2010088705 A1	08-04-2010	NONE	
US 2020371966 A1	26-11-2020	GB 2563888 A	02-01-2019
		TW 201905699 A	01-02-2019
		US 2020371966 A1	26-11-2020
		WO 2019002813 A1	03-01-2019

Form PCT/ISA/210 (patent family annex) (April 2005)

フロントページの続き

,MC,ME,MK,MT,NL,NO,PL,PT,RO,RS,SE,SI,SK,SM,TR),OA(BF,BJ,CF,CG,CI,CM,GA,GN,GQ,GW,KM,ML,MR,NE,SN,TD,TG),AE,AG,AL,AM,AO,AT,AU,AZ,BA,BB,BG,BH,BN,BR,BW,BY,BZ,CA,CH,CL,CN,CO,CR,CU,CV,CZ,DE,DJ,DK,DM,DO,DZ,EC,EE,EG,ES,FI,GB,GD,GE,GH,GM,GT,HN,HR,HU,ID,IL,IN,IQ,IR,IS,IT,JM,JO,JP,KE,KG,KH,KN,KP,KR,KW,KZ,LA,LC,LK,LR,LS,LU,LY,MA,MD,MG,MK,MN,MU,MW,MX,MY,MZ,NA,NG,NI,NO,NZ,OM,PA,PE,PG,PH,PL,PT,QA,RO,RS,RU,RW,SA,SC,SC,SD,SE,SG,SK,SL,ST,SV,SY,TH,TJ,TM,TN,TR,TT,TZ,UA,UG,US,UZ,VC,VN,WS,ZA,ZM,ZW

グレート・ブリテン及び北部アイルランド連合王国 シービー１ ９エヌジェイ ケンブリッジシャー ケンブリッジ フルボーン ロード １１０ アーム リミテッド宛