(19)日本国特許庁(JP)　(12)公表特許公報(A)　(11)公表番号
特表2024-527283
(P2024-527283A)
(43)公表日　令和6年7月24日(2024.7.24)

(51)国際特許分類		FI		テーマコード（参考）
Ｇ０６Ｆ　12/1027(2016.01)		Ｇ０６Ｆ	12/1027	５Ｂ２０５
Ｇ０６Ｆ　12/1009(2016.01)		Ｇ０６Ｆ	12/1009	

審査請求　未請求　予備審査請求　未請求　（全34頁）

(21)出願番号　特願2023-579107(P2023-579107)
(86)(22)出願日　令和4年6月10日(2022.6.10)
(85)翻訳文提出日　令和6年1月30日(2024.1.30)
(86)国際出願番号　PCT/GB2022/051454
(87)国際公開番号　WO2023/285774
(87)国際公開日　令和5年1月19日(2023.1.19)
(31)優先権主張番号　2110258.7
(32)優先日　令和3年7月16日(2021.7.16)
(33)優先権主張国・地域又は機関
英国(GB)
(81)指定国・地域　AP(BW,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,MC,
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　エヴァンス、マシュー ルシアン
イギリス国 ケンブリッジシャー 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，アーム リミテッド宛
(72)発明者　ディモンド、ロバート グウィリム
イギリス国 ケンブリッジシャー シービ
最終頁に続く

(54)【発明の名称】　ページアクセス頻度追跡

(57)【要約】
装置が、メモリシステム（１０、３２）へのアクセスを要求するメモリアクセス要求を処理するメモリアクセス回路（１１）と、アクセス頻度追跡回路（４０）とを備える。メモリアドレス空間の所与のページへのアクセスを要求する所与のメモリアクセス要求に応答して、アクセス頻度追跡回路（４０）は、偶然依存テストの結果を決定し、偶然依存テストの結果は偶然に依存する。偶然依存テストの結果が第１の結果である場合、所与のページに対応するアクセス頻度追跡インジケータが、アクセス頻度追跡構造内で更新される。偶然依存テストが第１の結果以外の結果を有する場合、アクセス頻度追跡回路４０は、所与のページに対応するアクセス頻度追跡インジケータの更新を省略する。

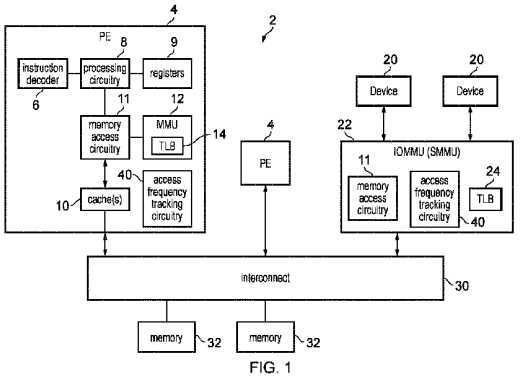


FIG. 1

【特許請求の範囲】
【請求項１】
装置であって、
メモリシステムへのアクセスを要求するメモリアクセス要求を処理するメモリアクセス回路と、
メモリアドレス空間の所与のページへのアクセスを要求する所与のメモリアクセス要求に応答するアクセス頻度追跡回路であって、
偶然依存テストの結果が、前記所与のメモリアクセス要求について決定された場合に偶然に依存する、前記偶然依存テストの結果を決定し、
前記偶然依存テストの結果が第１の結果である場合、アクセス頻度追跡構造内の前記所与のページに対応するアクセス頻度追跡インジケータを更新し、
前記偶然依存テストの結果が前記第１の結果以外の結果である場合、前記所与のページに対応する前記アクセス頻度追跡インジケータの更新を省略するためのアクセス頻度追跡回路とを備える、装置。
【請求項２】
前記偶然依存テストが、前記第１の結果を提供するソフトウェア構成可能な確率を有する、請求項１に記載の装置。
【請求項３】
前記第１の結果を提供する前記ソフトウェア構成可能な確率が、
全てのメモリアクセス要求の間で共有されるソフトウェア構成されるグローバルパラメータと、
前記所与のメモリアクセス要求を含むメモリアクセス要求の特定のサブセットに特有のソフトウェア構成されたローカルパラメータとのうちの少なくとも１つに依存する、請求項２に記載の装置。
【請求項４】
前記ソフトウェア構成可能な確率が、ソフトウェア構成可能なレジスタにおいて指定される、請求項２又は３のいずれか一項に記載の装置。
【請求項５】
前記アクセス頻度追跡回路が、前記メモリアドレス空間の異なるページに対して別々に前記第１の結果を提供する前記偶然依存テストの確率を設定するように構成され、
前記所与のページへのアクセスの頻度が所定の閾値に達したか又はそれを超えたという判定に応答して、前記アクセス頻度追跡回路が、前記所与のページへのアクセスを要求するメモリアクセス要求の後続のインスタンスに対して前記偶然依存テストが前記第１の結果を提供する確率を低減するように構成される、請求項１～４のいずれか一項に記載の装置。
【請求項６】
前記偶然依存テストが、メモリアクセス要求をカウントするためのテスト評価カウンタが所定の条件を満たすかどうかを判定することを含む、請求項１～５のいずれか一項に記載の装置。
【請求項７】
前記テスト評価カウンタが、全てのメモリアクセス要求間で共有されるグローバルカウンタである、請求項６に記載の装置。
【請求項８】
前記テスト評価カウンタが、特有のサブセットのメモリアクセス要求をそれぞれカウントする複数のローカルカウンタのうちの１つであり、前記偶然依存テストが、前記所与のメモリアクセス要求を含むメモリアクセス要求の特有のサブセットに対応する前記ローカルカウンタに基づく、請求項６に記載の装置。
【請求項９】
前記複数のローカルカウンタが、異なる動作モード又は実行状態に関連付けられたメモリアクセス要求の特有のサブセットに対応する、請求項８に記載の装置。

【請求項１０】
前記複数のローカルカウンタが、ロードメモリアクセス要求をカウントするためのロードカウンタと、ストアメモリアクセス要求をカウントするためのストアカウンタとを含む、請求項８又は９のいずれか一項に記載の装置。
【請求項１１】
前記複数のローカルカウンタが、前記メモリアドレス空間の異なる領域又はページへのアクセスを要求するメモリアクセス要求の特有のサブセットに対応する、請求項８～１０のいずれか一項に記載の装置。
【請求項１２】
前記複数のローカルカウンタが、前記メモリアドレス空間の異なるページへのアクセスを要求するメモリアクセス要求の特有のサブセットに対応する、請求項８～１０のいずれか一項に記載の装置。
【請求項１３】
前記偶然依存テストが、ビットの一部が所定の値を有するかどうかを判定することを含み、前記ビットの一部が、前記装置の内部制御情報の１つ以上の項目、又は前記装置の１つ以上のアナログ若しくはデジタル信号から導出されるシリアルパルス列からサンプリングされたビットを含む、請求項１～５のいずれか一項に記載の装置。
【請求項１４】
前記偶然依存テストが、乱数又は擬似乱数が所定の値を有するかどうかを判定することを含む、請求項１～５のいずれか一項に記載の装置。
【請求項１５】
前記アクセス頻度追跡インジケータが、追跡フラグを含み、
前記偶然依存テストの結果が前記第１の結果であり、前記所与のページに対応する前記追跡フラグが第１の値を有するとき、前記アクセス頻度追跡回路が、第２の値を指定するために前記所与のページに対応する前記追跡フラグを設定するように構成される、請求項１～１４のいずれか一項に記載の装置。
【請求項１６】
前記アクセス頻度追跡インジケータが、マルチビットアクセス頻度追跡カウンタを含み、
前記偶然依存テストの結果が前記第１の結果である場合、前記アクセス頻度追跡回路が、前記所与のページに対応する前記マルチビットアクセス頻度追跡カウンタをインクリメントするように構成される、請求項１～１４のいずれか一項に記載の装置。
【請求項１７】
前記アクセス頻度追跡構造が、複数のエントリを含むアクセス頻度追跡ハードウェア構造を含み、各エントリが、前記メモリアドレス空間の対応するページに対する前記アクセス頻度追跡インジケータを記憶する、請求項１～１６のいずれか一項に記載の装置。
【請求項１８】
前記アクセス頻度追跡ハードウェア構造が、変換索引バッファであり、前記複数のエントリが、それぞれが前記対応するページに関連付けられた前記アクセス頻度追跡情報及びアドレス変換情報を記憶するための変換索引バッファエントリである、請求項１７に記載の装置。
【請求項１９】
前記アクセス頻度追跡構造が、前記メモリシステムに記憶されたメモリベースのデータ構造を含む、請求項１～１８のいずれか一項に記載の装置。
【請求項２０】
前記アクセス頻度追跡構造が、メモリアドレス空間の連続ブロック内のアドレスを有する複数のエントリを含む線形データ構造を含み、
前記アクセス頻度追跡回路が、前記所与のページのアドレスに基づいて決定されたオフセットに基づいて、前記所与のページに対応する前記アクセス頻度追跡インジケータを提供する前記線形データ構造の前記エントリを識別するように構成される、請求項１９に記

載の装置。
【請求項２１】
前記アクセス頻度追跡構造がツリー構造を含む、請求項１～１９のいずれか一項に記載の装置。
【請求項２２】
前記ツリー構造が、前記メモリアドレス空間の対応するページに対するアドレス変換情報を指定するための複数のページテーブルエントリを含むページテーブル構造であり、前記所与のページに対応する前記ページテーブルエントリが、前記所与のページに対応する前記アクセス頻度追跡インジケータを含む、請求項２１に記載の装置。
【請求項２３】
命令セットアーキテクチャに従って定義された命令に応答してデータ処理を実行する処理回路と、
前記メモリアクセス回路と、
前記アクセス頻度追跡回路とを備える処理要素を備える、請求項１～２２のいずれか一項に記載の装置。
【請求項２４】
前記メモリシステムへのダイレクトメモリアクセスを有する少なくとも１つのデバイスの代わりにメモリ管理動作を実行するための入出力メモリ管理ユニットを備え、
前記入出力メモリ管理ユニットが、前記メモリアクセス回路と前記アクセス頻度追跡回路とを備える、請求項１～２３のいずれか一項に記載の装置。
【請求項２５】
メモリアドレス空間の１つ以上のページへのアクセス頻度を追跡するための方法であって、
前記メモリアドレス空間の所与のページへのアクセスを要求する所与のメモリアクセス要求に応答して、
偶然依存テストの結果が、前記所与のメモリアクセス要求について決定された場合に偶然に依存する、前記偶然依存テストの結果を決定することと、
前記偶然依存テストの結果が第１の結果である場合、アクセス頻度追跡構造内の前記所与のページに対応するアクセス頻度追跡インジケータを更新することと、
前記偶然依存テストの結果が前記第１の結果以外の結果である場合、前記所与のページに対応する前記アクセス頻度追跡インジケータの更新を省略することとを含む、方法。
【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】
データ処理システムにおいて、メモリアドレス空間のそれぞれのページがアクセスされる相対頻度を追跡するための機構を提供することが有用であり得る。これは、例えば、仮想アドレスを物理アドレスにどのようにマッピングすべきかを決定するのに有用であり得る。なぜならアクセス頻度追跡情報を使用して、より高速にアクセスすることができるメモリシステムの部分にマッピングすることができるアドレス空間のより頻繁にアクセスされるページを識別することができ、より頻繁にアクセスされないページは、アクセスがより遅いメモリシステムの部分に割り振られるからである。
【０００３】
少なくともいくつかの例は、メモリシステムへのアクセスを要求するメモリアクセス要求を処理するメモリアクセス回路と、メモリアドレス空間の所与のページへのアクセスを要求する所与のメモリアクセス要求に応答するアクセス頻度追跡回路であって、偶然依存テストの結果が所与のメモリアクセス要求について決定される場合に偶然に依存する、偶然依存テストの結果を決定し、偶然依存テストの結果が第１の結果である場合、アクセス頻度追跡構造内の所与のページに対応するアクセス頻度追跡インジケータを更新し、偶然

依存テストの結果が第１の結果以外の結果である場合、所与のページに対応するアクセス頻度追跡インジケータの更新を省略するためのアクセス頻度追跡回路とを備える装置を提供する。

【０００４】

少なくともいくつかの例は、メモリアドレス空間の１つ以上のページへのアクセスの頻度を追跡するための方法を提供し、方法は、メモリアドレス空間の所与のページへのアクセスを要求する所与のメモリアクセス要求に応答して、偶然依存テストの結果が所与のメモリアクセス要求について評価される場合に偶然に依存する、偶然依存テストの結果を決定することと、偶然依存テストの結果が第１の結果である場合、アクセス頻度追跡構造内の所与のページに対応するアクセス頻度追跡インジケータを更新することと、偶然依存テストの結果が第１の結果以外の結果である場合、所与のページに対応するアクセス頻度追跡インジケータの更新を省略することとを含む。

【図面の簡単な説明】

【０００５】

本技術の更なる態様、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】データ処理システムの一例を概略的に示している。

【図２】アクセス頻度追跡回路の一例を示す図である。

【図３】メモリアドレス空間の所与のページへのアクセス頻度を追跡する方法を示すフロー図である。

【図４】偶然依存テストを実施するいくつかの例を示す。

【図５】偶然依存テストを実施するいくつかの例を示す。

【図６】偶然依存テストを実施するいくつかの例を示す。

【図７】偶然依存テストを実施するいくつかの例を示す。

【図８】偶然依存テストを実施するいくつかの例を示す。

【図９】アクセス頻度追跡構造を実施するいくつかの例を示す。

【図１０】アクセス頻度追跡構造を実施するいくつかの例を示す。

【図１１】アクセス頻度追跡構造を実施するいくつかの例を示す。

【図１２】所与のページに対するアクセス頻度追跡インジケータがアクセス頻度追跡ハードウェア構造（例えば、変換索引バッファ）に記憶される一例を示す。及び

【図１３】所与のページに対するアクセス頻度追跡インジケータが、所与のページへのアクセスの推定頻度が或る特定の閾値に達したか又はそれを超えたことを示すときに、偶然依存テストが第１の結果を提供する確率を低減することを概略的に示す。

【０００６】

アドレス空間のページへのアクセス頻度を追跡するための１つの手法は、ページがアクセス又は修正されるときに、所与のページに関連付けられたページテーブルエントリ内にフラグを設定することであり得る。しかしながら、これは、一度もアクセスされていないページを、少なくとも一度アクセスされたページと区別するだけである。フラグはカウンタになるように拡張され得るが、典型的には、ページテーブルエントリ内に多くのビットの余裕はない。本発明者らは、或る目的のために、非常に多くの回数アクセスされたページを、より少ない回数アクセスされたページから区別できること（例えば、毎秒１００～１０００回程度アクセスされたページを、毎秒約１００万回程度アクセスされたページから区別すること）が有用であり得ることを認識した。しかしながら、ページ当たりのこのアクセス数を追跡するのに十分な数のビットを有するカウンタを維持することは、性能、メモリオーバーヘッド及び／又は回路面積の点でコストがかかる可能性がある。

【０００７】

以下に説明する例では、装置は、メモリシステムへのアクセスを要求するメモリアクセス要求を処理するためのメモリアクセス回路と、メモリアドレス空間の所与のページへのアクセスを要求する所与のメモリアクセス要求に応答して、偶然依存テストの結果を決定するためのアクセス頻度追跡回路とを有する。偶然依存テストの結果は、所与のメモリア

クセス要求に対して決定されるとき、偶然に依存する。偶然依存テストの結果が第１の結果である場合、所与のページに対応するアクセス頻度追跡インジケータが、アクセス頻度追跡構造内で更新される。しかしながら、偶然依存テストが第１の結果以外の結果を提供するとき、所与のページに対応するアクセス頻度追跡インジケータの更新は省略される。

【０００８】

したがって、対応するページへのアクセスがあるたびにアクセス頻度追跡インジケータを更新するのではなく、アクセス頻度追跡インジケータへの更新は、偶然依存テストが第１の結果を提供するか別の結果を提供するかに応じて、或る時間だけ更新される。偶然依存テストは、サイコロの振り又は抽選に類似していてもよく、メモリアクセス要求の任意の所与のインスタンスに対して第１の結果が発生するかどうかが偶然の問題である場合に、第１の結果を有する或る特定の確率を提供する。２つのメモリアクセス要求が、同一のメモリアクセス制御パラメータ（例えば、同じターゲットアドレス、同じタイプのメモリアクセス、ページテーブルエントリに設定された同じ許可、メモリアクセス要求を発行するときのプロセッサの同じ実行モード又は動作状態、及びメモリアクセス要求が処理される方法を構成するための任意の制御レジスタに設定された同じ制御設定等）に基づいて制御される場合であっても、偶然依存テストは、それにもかかわらず、それらの要求のうちの１つに対して第１の結果を提供し、それらの要求のうちの別のものに対して異なる結果を提供し得る。

【０００９】

したがって、偶然依存テストが第１の結果を提供するかどうか、したがって所与のページに対応するアクセス頻度追跡インジケータが更新されるかどうかは、偶然に依存する。これは、偶然依存テストが、以前にアクセスされたことがない所与のページへのまさに最初のアクセスに対して第１の結果をもたらすが、より頻繁にアクセスされるページがアクセスされるたびに別の結果を提供することが可能であるため、アクセス頻度追跡における不正確さの増加につながり得るので、直感に反していると考えることができ、したがって、アクセス頻度追跡構造が、いくつかのページを、実際よりも頻繁にアクセスされるものとして識別し得ることを意味し得る。しかしながら、本発明者らは、統計的に、より頻繁にアクセスされるページは、偶然依存テストがより頻繁にアクセスされないページよりも第１の結果をもたらす場合に、インスタンスに遭遇する可能性が高いことを認識した。事実上、より頻繁にアクセスされるページには、より頻繁にアクセスされないページと比較して、より多くの数の「サイコロの振り」が与えられ、したがって、最終的に、より頻繁にアクセスされるページがそのアクセス頻度追跡インジケータを更新させる確率がより高くなる。したがって、時折の誤検出は、アクセス頻度追跡構造によって提供される追跡情報をかなりの量だけ歪める可能性が低い。

【００１０】

アクセス頻度追跡インジケータがアクセスされたページに対して更新されるかどうかを制御するために偶然依存テストを使用する利点は、目的がアクセス数の広範囲にわたってページアクセス頻度を追跡することである場合であっても、これが回路面積、電力及び性能コストを低減できることである。追跡インジケータが更新される確率が、所与のページへの任意の特有のメモリアクセスに対して低減されるように、偶然依存テストを実施することによって、これは、アクセス頻度のより大きい絶対範囲にわたってカウントをシミュレートしながらより小さいカウンタ又は他の追跡インジケータを採用することができることを意味する。これにより、異なるページに対するアクセスのおおよその相対頻度に関する有用な情報を、回路面積、電力及び性能に関してはるかに少ないコストで追跡することが可能になる。

【００１１】

偶然依存テストは、第１の結果を提供するソフトウェア構成可能な確率を有し得る。したがって、ソフトウェアは、対応するページへのメモリアクセス要求に対してアクセス頻度追跡インジケータが更新される確率を設定することができる。例えば、テストが第１の結果を提供する確率は、１／Ｎとして定義することができ、ここで、Ｎの値は、ソフトウ

ェアによって選択される構成値に応じて可変である。
【００１２】
場合によっては、第１の結果を提供する偶然依存テストの確率を構成する権利は、或る特定のレベル以上の特権を有するソフトウェアに制限されてもよい。特権の閾値レベル未満の特権で動作するソフトウェアは、確率を構成することを許可されない場合がある。
【００１３】
１つの例では、第１の結果を提供するソフトウェア構成可能な確率は、ソフトウェア構成可能なレジスタ内で指定され得る。したがって、ソフトウェアは、ソフトウェア構成可能なレジスタを更新することによって確率を調整することができる。
【００１４】
他の例は、偶然依存テストが第１の結果を提供する確率をソフトウェアが制御するように更新することができる、ページテーブル又は別のメモリベースの構造内に制御状態を提供することができる。
【００１５】
いくつかの例では、偶然依存テストにおいて第１の結果を提供するソフトウェア構成可能な確率は、全てのメモリアクセス要求間で共有されるソフトウェア構成されたグローバルパラメータに依存し得る。
【００１６】
また、所与のメモリアクセス要求を含むメモリアクセス要求の特定のサブセットに特有であり、メモリアクセス要求のその特定のサブセットに対する第１の結果を提供する偶然依存テストの確率を制御する、少なくとも１つのソフトウェア構成されたローカルパラメータを提供することが可能である。例えば、ローカルパラメータは、特定のタイプのアクセス（例えば、アクセスが特権アクセスであるか非特権アクセスであるか、又はアクセスがロードであるかストアであるか）に特有であってもよく、又は特定の実行モード又は状態で発行されるメモリアクセス要求に特有であってもよい。また、アドレス空間の特定の領域に特有のソフトウェア構成されたローカルパラメータを提供することもできる。したがって、第１の結果を提供する偶然依存テストの確率は、メモリアクセスの異なるサブセットに対して独立して調整することができる。ローカルパラメータが確率を制御するために定義される場合、これらはソフトウェア構成可能なレジスタ又はメモリ内のデータ構造に設定され得る。
【００１７】
メモリアクセス要求の異なるサブセットに対する独立した確率を設定することは、追跡情報が（例えば、アドレスマッピングを決定するために）ソフトウェアによって使用されるときにメモリアクセス要求のいくつかのサブセットが他のサブセットよりも優先することを可能にするので、有用であり得る。例えば、ソフトウェアは、アクセス頻度追跡構造によって表されるアクセスの見かけの数が、アクセスの第１／第２のサブセットに対するアクセスの実際の数と比較して、第１のサブセットを優先して歪められるように、第２のサブセット（例えば、より低いセキュア／特権アクセス）と比較して、アクセスの第１のサブセット（例えば、セキュア／特権アクセス）に対して生じる第１の結果のより高い確率を設定することを望む場合がある。アクセス頻度追跡構造が後にソフトウェアによって使用されて、より頻繁にアクセスされるページがメモリシステムのより速い部分に割り振られ得るようにメモリマップを制御するとき、これは、ソフトウェアの好ましい優先順位付け方式を考慮に入れるために、アクセスの第１のサブセットがより高い性能で行われ得る見込みを高めることができる。これは、アクセス頻度追跡構造が更新されるときに優先順位付けが既に考慮されているかのように、アクセス頻度追跡情報のソフトウェア分析を簡略化することができ、分析ソフトウェアアルゴリズムがアクセス頻度追跡情報をメモリアクセスのタイプに関する他の情報と組み合わせる必要性がより少ない。したがって、ソフトウェアがメモリアクセスの異なるサブセットについて独立した確率を設定する能力をハードウェアでサポートすることによって、これは、アクセス頻度追跡情報を分析するためのソフトウェア開発者のコードを簡略化する際に、ソフトウェア開発者にとって実際的

な利益をもたらすことができる。

【００１８】

いくつかの例は、ソフトウェア構成されたグローバルパラメータと１つ以上のソフトウェア構成されたローカルパラメータとの組み合わせを使用して、第１の結果を提供する偶然依存テストの確率を制御することができる。例えば、グローバルパラメータはデフォルト確率を設定することができるが、１つ以上のソフトウェア構成されたローカルパラメータは、メモリアクセス要求の或る特定のサブセットに対して異なる確率を指定するか、又はアクセス要求のそのサブセットに対して使用されるデフォルト確率に対する調整を指定することができる。

【００１９】

いくつかの例では、メモリアクセス要求の特有のサブセットに対する第１の結果を提供する偶然依存テストの確率は、追跡が開始してから以前に遭遇したアクセスの数にかかわらず、一定のままであり得る。

【００２０】

しかしながら、他の手法では、第１の結果を提供する偶然依存テストの確率は、所与のページへのアクセスの数に応じて経時的に可変であり得る。例えば、アクセス頻度追跡回路は、メモリアドレス空間の異なるページに対して別々に第１の結果を提供する偶然依存テストの確率を設定してもよく、所与のページへのアクセスの頻度が所定の閾値に達したか又は超えたという判定に応答して、アクセス頻度追跡回路は、所与のページへのアクセスを要求するメモリアクセス要求の後続のインスタンスに対して第１の結果を提供する偶然依存テストの確率を低減してもよい。アクセス頻度追跡情報は、上述したような偶然依存テストに基づいて統計的に設定され得るので、アクセス頻度が閾値に達した／超えたという判定は正確ではない場合があり、推定されたアクセス頻度が閾値に達した／超えたという判定である可能性があり、その推定は、平均して妥当な推定であり得る相対頻度の近似指標を提供し得るが、その推定は、各特有のインスタンスにおいて正確ではない場合があることに留意されたい。

【００２１】

一般に、所与のページへのアクセス数の増加に伴って第１の結果を与えるテストの確率を低減することは、アクセス頻度追跡インジケータによって表されるアクセスの推定数と、そのアクセス数を表すために使用されるアクセス頻度追跡インジケータの実際の値との間の非線形スケーリングを実装するために有用であり得る。これはまた、より広い範囲の大きさにわたってアクセスを追跡するために必要とされるカウンタの全体的なサイズを低減するのに役立ち得る。例えば、いくつかの使用事例では、アクセスされたことがないページを少なくとも１回アクセスされたページから区別できることが有用である場合があるが、ページが少なくとも１回アクセスされると、アクセス数がはるかに大きい値に達するまで、１回、２回、３回等アクセスされたかどうかを区別できることは有用でない場合がある。同様に、ページが或る特定の回数（例えば、１００回又は１０００回）アクセスされると、その回数を超えてアクセスされたページを識別することが興味深い次の境界は、ページアクセスカウントが数十万回又は数百万回に達するまで存在しない可能性がある。したがって、いくつかの閾値関心点が存在し得るが、それらは均等に分散されない場合がある。所与のページへのアクセスの数が増加するにつれて、そのページへのアクセスについて第１の結果を提供する偶然依存テストの確率を低減することによって、これは、頻度追跡インジケータの限られた粒度がより効率的に使用され得るように、より小さいカウンタがより広い範囲を追跡することを可能にするのに有用であり得る。

【００２２】

偶然依存テストは、様々な方法で実施することができる。偶然依存テストは、第１の結果を提供する確率が１未満であってもよい。偶然依存テストは、所与のメモリアクセス要求を発行させるソフトウェアによって制御可能でない情報又は信号に依存し得る。場合によっては、メモリアクセス要求の全てのプロパティと、メモリアクセス要求が許可され得るかどうかを制御するためのソフトウェア定義パラメータとが同一である場合であっても

、異なる機会に全く同じメモリアクセス要求を発行することは、偶然依存テストの異なる結果につながり得る。

【００２３】

１つの具体例（唯一の例ではない）では、偶然依存テストは、メモリアクセス要求をカウントするために設けられたテスト評価カウンタが所定の条件を満たすかどうかを判定することを含む。テスト評価カウンタは、いくつかの状態を有し、カウンタを進めるための少なくとも１つの基準を満たす各メモリアクセス要求に応答して次の状態に進められ、テスト評価カウンタが初期状態にリセットされた後に或る特定の回数だけ状態が進められると所定の条件を満たすと判定され得る、任意の情報であり得る。例えば、テスト評価カウンタは、関連するメモリアクセス要求に遭遇するたびにインクリメント又はデクリメントされ、テスト評価カウンタが所定の値に達するか、又は所与の閾値を通過すると、所定の条件を満たすとみなされ得るバイナリ整数であり得る。しかしながら、カウンタがバイナリ整数として表されることは必須ではなく、別の例は、カウンタを進めるための少なくとも１つの基準を満たす各メモリアクセス要求に応答して次の状態にシフトされるシフトレジスタを提供することができ、その結果、シフトレジスタは、或る特定の一連の状態の間を循環し、そのサイクル内で所定の状態に達すると所定の条件を満たす。例えば、シフトレジスタは、第１の値（０又は１）を有する或る特定の数のビットと、第２の値（０及び１の他方）を有する１つのビットとを備えることができ、他の全てのビットとは異なる値を有する単一のビットがシフトレジスタ内の特定の場所にシフトされると、所定の条件を満たすとみなすことができ、シフトレジスタは、前進基準を満たすメモリアクセス要求に遭遇するたびに１つの場所だけシフトされる。もちろん、これらはテスト評価カウンタを実装するいくつかの特有の方法にすぎず、他のオプションも使用することができる。

【００２４】

一般に、偶然依存テストを評価するためのテスト評価カウンタを設けることにより、これは偶然依存テストの評価を実施する簡単な方法を提供することができる。所与のメモリアクセス要求が処理されるときにテスト評価カウンタが所定の条件を満たすかどうかは、その特定のメモリアクセス要求の特性だけでなく、以前に遭遇した要求の数及びメモリアクセス要求の相対的順序等の明らかにランダムな発生にも依存するので、偶然の問題とみなすことができ、これは通常、メモリアクセス要求を発行するソフトウェアの制御外であり、その結果、ソフトウェアの観点から、テスト評価カウンタが所定の条件を満たすかどうかは、第１の結果を提供する１／Ｎの可能性を提供する本質的にランダムなイベントとみなすことができ、ここで、Ｎは、カウンタが初期化と条件が満たされる状態に達するまでの間にトラバースする状態の数である。

【００２５】

いくつかの例では、テスト評価カウンタは、どのページがメモリアクセス要求によってアクセスされるかにかかわらず、全てのメモリアクセス要求間で共有されるグローバルカウンタであり得る。これにより、回路面積及び電力の点で実装コストがより低くなり得る。

【００２６】

代替的に、テスト評価カウンタは、各々が特有のサブセットのメモリアクセス要求をカウントするためのいくつかのローカルカウンタのうちの１つとすることができ、所与のメモリアクセス要求が処理されるときに実行される偶然依存テストは、所与のメモリアクセス要求を含むメモリアクセス要求の特有のサブセットに対応するローカルカウンタに基づき得る。例えば、異なるローカルカウンタは、異なる動作モード又は実行状態に関連付けられたメモリアクセス要求の特有のサブセットに対応することができる（例えば、特権アクセスをより低い特権アクセスから区別する、又はセキュアアクセスをより低いセキュアアクセスから区別する）。また、ロード要求（メモリからレジスタへのデータの転送を要求する）及びストア要求（レジスタからメモリへのデータの転送を要求する）が異なるカウンタに基づいて評価される偶然依存テストを有するように、それぞれロード及びストアに対応する異なるローカルカウンタを提供することが可能である。別のオプションでは、

１つの領域又はページへのメモリアクセス要求が、異なる領域又はページを対象とするメモリアクセス要求に対する異なるカウンタに基づいて評価された偶然依存テストを有することができるように、メモリアドレス空間の特有の領域又はページに対応するローカルカウンタを提供することができる。ページ特有のローカルカウンタの場合、ローカルカウンタを追跡するための便利な構造は、それぞれのページに対応するアドレス変換エントリを記憶する変換索引バッファ（ＴＬＢ）内にあることができ、所与のページに対するアドレス変換エントリは、所与のページに対応するローカルカウンタを含み得る。これは、既存のＴＬＢルックアップが現在アクセスされているページのためのローカルテスト評価カウンタを識別するために使用され得るので、別個の構造への更なるルックアップの必要性を回避することによって、回路面積及び電力消費を低減するのに役立ち得る。それにもかかわらず、領域／ページ特有のローカルテスト評価カウンタを提供するためにＴＬＢとは別個の構造を実装することも可能である。

【００２７】

メモリアドレス空間の個々の領域／ページに特有の別個のローカルテスト評価カウンタを提供することは、アクセス頻度追跡の精度を高めることができ、その結果、所与のページに対するアクセス頻度追跡インジケータによって表される推定アクセス頻度は、そのページへの実際のアクセス数とより密接に相関させることができ、これは、以前にアクセスされていない領域／ページへのアクセスによって引き起こされた誤検出が、偶然依存テストの第１の結果を引き起こす見込みを低減するためである。

【００２８】

テスト評価カウンタは、偶然依存テストを実装する唯一の方法ではない。別の例では、偶然依存テストは、ビットの一部が所定の値を有するかどうかを判定することを含んでもよく、ビットの一部は、装置の内部制御情報の１つ以上の項目から、又は装置の１つ以上のアナログ若しくはデジタル信号から導出されたシリアルパルス列からサンプリングされたビットを含む。場合によっては、或る確率に従ってほぼランダムに変動し得る値を提供するためにサンプリング又は再使用することができ、したがって、テストが満たされるか否かを判定するためにサンプリングされ、所定の値と比較され得る、（アクセス頻度追跡以外の他の目的を有する）データ処理装置内の既存の信号又は情報が存在し得るので、特に、偶然依存テストの実装のために、任意の特有の状態（カウンタ等）を維持する必要がない場合がある。例えば、処理装置は、経過したクロックサイクル、キャッシュミス、又は性能監視のための他のイベント等のイベントをカウントするための内部カウンタを有してもよく、それらのカウンタのいくつかのビットは、特定の確率を有する等間隔のイベントのパルス列として解釈され得る。他の例では、シリアルパルス列は、互いに実質的に独立であり得るが、結果として生じるパルス列（経時的な状態／信号の連続値に基づいてサンプリングされたビットのシーケンス）が効果的にランダムであるとみなされ得るように組み合わせられ得る複数の内部状態又は物理信号から導出することができ、次いで、アクセス頻度追跡回路は、それらのビットが或る特定の条件（或る特定の数の１を或る行内に含む等）を満たすかどうかを評価し得る。デジタル信号がパルス列のソースとして使用されることは必須ではない。ダイオードからのランダムビットストリーム又は無関係な信号間の干渉パターン等のアナログ信号も使用することができる。例えば、２つのクロック信号の混合（ＸＯＲ）は、それぞれのクロック周波数の共通因数でビート周波数を与えることができ、複数のそのようなソースを混合して、ほぼランダムであるパルス列を与えることができる。したがって、例えば、処理装置内で利用可能な既存のアナログ若しくはデジタル信号又は情報を使用して、偶然依存テストを評価するために使用されるサンプリングされたビットを取得することができる多くのオプションが存在し得る。

【００２９】

別の例では、偶然依存テストは、乱数又は擬似乱数が所定の値を有するかどうかを判定することを含み得る。いくつかのシステムは、他の目的（暗号化等）のために乱数発生器又は擬似乱数発生器を既に有している場合があり、そのため、ランダム性の既存のソースを利用していくつかのビットを提供することができ、このビットを何らかの所定の値と比

較して、偶然依存テストが第１の結果を提供するか、異なる結果を提供するかを判定することができる。

【００３０】

偶然依存テストが所与のメモリアクセス要求に対して第１の結果を有すると判定されたときに更新されるアクセス頻度追跡構造は、いくつかの異なる方法で実装されてもよい。

【００３１】

いくつかの例では、アクセス頻度追跡インジケータは、第１の値又は第２の値のうちの１つを有することができる追跡フラグ（例えば、単一ビットフラグ）を含み得る。偶然依存テストの結果が第１の結果であり、所与のページに対応する追跡フラグが第１の値を有するとき、アクセス頻度追跡回路は、第２の値を指定するように所与のページに対応する追跡フラグを更新し得る。したがって、この場合、１ビットフラグは、アクセス頻度追跡インジケータを表すのに十分である場合があり、これは、アクセス頻度追跡構造を実装するコストを低減することができる。偶然依存テストは、第１の結果を提供する或る特定の確率１／ｐを提供するので、第２の値に設定されている追跡フラグは、例えば、或る特定のアクセス数ｐより多くがそのページに対して行われた確率が１／２より大きいことを示すことができる。

【００３２】

他の例では、アクセス頻度追跡インジケータは、マルチビットアクセス頻度追跡カウンタを含み得る。これは、偶然依存テストが第１の結果を提供したか否かを判定するために使用される前述のテスト評価カウンタとは異なるカウンタであることに留意されたい。アクセス頻度追跡カウンタは、偶然依存テストの結果が第１の結果である場合に更新されるアクセス頻度追跡構造内のインジケータである。したがって、偶然依存テストの結果が第１の結果である場合、アクセス頻度追跡回路は、所与のページに対応するマルチビットアクセス頻度追跡カウンタをインクリメントし得る。カウンタは所与の値で飽和し得るので、カウンタが飽和値に達すると、偶然依存テストが第１の結果を提供するときにページが再びアクセスされる場合であっても、更なるインクリメントは実行されなくてよいことが理解されよう。アクセス頻度追跡インジケータとしてマルチビットカウンタを提供することによって、これは、より広い範囲のアクセス頻度が追跡されることを可能にすることができる。それにもかかわらず、カウンタを更新することは、偶然に依存するテスト結果に依存するので、カウンタのサイズは、ページがアクセスされるたびにカウンタがインクリメントされる実施態様よりもはるかに小さくすることができる。

【００３３】

アクセス頻度追跡構造は、異なる方法で実装することができる。

【００３４】

いくつかの実施態様では、アクセス頻度追跡構造は、アドレス空間の異なるページに対して異なるアクセス頻度追跡インジケータが提供されるように、個々のページの粒度でアクセス頻度追跡インジケータを備えることができる。アクセス頻度追跡構造がページごとに対応するアクセス頻度追跡インジケータを有することは必須ではない。例えば、アドレス空間の現在マッピングされていないページは、構造の実装に応じて、対応する追跡インジケータを有する必要はない。ページごとにアクセス頻度追跡インジケータを提供することによって、これは、アクセス頻度のよりきめ細かい追跡を可能にし、ページマッピングに関するより多くの情報に基づく決定を可能にすることができる。

【００３５】

しかしながら、他の例では、アクセス頻度追跡インジケータは、個々のページよりも粗い粒度で定義され得る。例えば、各アクセス頻度追跡インジケータは、複数のページのブロックに対応することができ、したがって、所与のページに対応するアクセス頻度追跡インジケータは、所与のページを含むページのブロックに対応するアクセス頻度追跡インジケータであってもよい。個々のページの粒度ではなく、領域又はブロックベースでアクセス頻度追跡インジケータを管理することは、アクセス頻度追跡構造のサイズを低減することができ、メモリの異なるブロック／領域へのアクセスの相対頻度に関する有用な情報を

依然として提供することができる。

【００３６】

１つの例では、アクセス頻度追跡構造は、メモリアドレス空間の対応するページについてのアクセス頻度追跡インジケータを各々が記憶する複数のエントリを含むアクセス頻度追跡ハードウェア構造を含み得る。したがって、アクセス頻度追跡インジケータは、対応するアクセス頻度追跡インジケータを識別するために、メモリアクセスのアドレスに基づいて（キャッシュと同様に）ルックアップされ得る、メモリとは別個のハードウェア記憶構造内に維持され得る。

【００３７】

ソフトウェアが、アクセス頻度追跡ハードウェア構造から所与のアドレスに対応するアクセス頻度追跡インジケータを読み取ることを可能にする機構を提供することができる。

【００３８】

同様に、又はアクセス頻度追跡ハードウェア構造内のアクセス頻度追跡インジケータへの直接ソフトウェアアクセスを可能にする代わりに、アクセス頻度追跡ハードウェア構造は、アドレス空間のそれぞれのページに対応するアクセス頻度追跡インジケータを提供するメモリベースのバッキングデータ構造のキャッシュとして機能することができる。メモリベースのバッキングデータ構造は、アクセス頻度追跡ハードウェア構造よりも多数のページに対する追跡インジケータを記憶することができる。所与のページへの所与のメモリアクセスに応答して、アクセス頻度追跡回路は、偶然依存テストを実行し、結果が第１の結果である場合、所与のページについてのアクセス頻度追跡ハードウェア構造の対応するエントリ内のアクセス頻度追跡インジケータを更新することができる。所与のページに対応するエントリがまだ存在しない場合、新しいエントリをアクセス頻度追跡ハードウェア構造に割り振ることができる。所与のアクセス頻度追跡インジケータに関連付けられたライトバックイベント時に、その追跡インジケータの値を使用して、メモリ内のバッキングデータ構造内の対応するインジケータを更新することができる。例えば、ライトバックイベントは、所与のアクセス頻度追跡インジケータ、又は所与の閾値に達する所与のアクセス頻度追跡インジケータのカウンタオーバーフローであり得る。また、ライトバックイベントは、アクセス頻度追跡ハードウェア構造からのエントリの追い出し（例えば、アクセス頻度追跡ハードウェア構造内に対応するエントリを以前に有していないページに新しいエントリが割り振られるときの容量競合による）であり得る。したがって、アクセス頻度追跡ハードウェア構造がメモリ内のバッキングデータ構造のキャッシュとして機能する実施態様では、ソフトウェアが、特定のページに関連付けられたアクセス頻度追跡インジケータを読み取るために、メモリ内の基礎となるバッキングデータ構造にアクセスすることが可能であり得るので、アクセス頻度追跡ハードウェア構造への直接ソフトウェアアクセスの必要がない場合がある。

【００３９】

アクセス頻度追跡ハードウェア構造は、アクセス頻度追跡インジケータを記憶するために特に提供される専用ハードウェア記憶構造であってもよい。

【００４０】

代替的に、アクセス頻度追跡ハードウェア構造のエントリは、アクセス頻度の追跡以外の目的のために使用することもできる。例えば、アクセス頻度追跡ハードウェア構造は、変換索引バッファ（ＴＬＢ）であってもよく、複数のエントリは、それぞれ、対応するページに関連付けられたアクセス頻度追跡情報及びアドレス変換情報を記憶するための変換索引バッファエントリであってもよい。これは、電力及び回路面積の点で効率的な手法とすることができ、これは、通常、ＴＬＢがアクセス変換を高速化するために既に提供されていてもよく、メモリ内でアクセスされるページの情報をルックアップするためのルックアップ機構を既に提供しているためであり、したがって所与のページのアクセス頻度追跡インジケータも含むようにＴＬＢエントリを拡張することによって、これは、専用構造をルックアップするためのキャッシュルックアップ論理の第２のセットを提供するコストを回避する。

【００４１】
他の例では、アクセス頻度追跡構造（特定のメモリアクセスについて第１の結果を与える偶然依存テストに応答して更新される）は、メモリシステムに記憶されたメモリベースのデータ構造であってもよい。この場合、（汎用データストレージのために既に提供されている汎用メモリシステム、及びメモリシステムからの汎用データをキャッシュする任意のデータキャッシュ以外の）アクセス頻度追跡インジケータのための特有のハードウェアストレージは存在しなくてもよい。この手法は、アクセス頻度追跡を実装する回路面積コストを低減する。偶然依存テストは、所与のページについてのアクセス頻度追跡インジケータがアクセスの一部でのみ更新されることを意味するので、メモリへの追加アクセスの性能コストは、偶然依存テストを使用する統計的手法を使用しない代替手法と比較して制限され得る。アクセス頻度追跡構造自体がメモリベースの構造である実施態様では、これは、アクセス頻度追跡インジケータのハードウェアキャッシュのためのバッキング構造である上述のメモリベースのバッキングデータ構造とは異なる。なぜなら、メモリベースの構造がハードウェアキャッシュのためのバッキングデータ構造として機能する場合、メモリベースのバッキングデータ構造は、偶然依存テストが第１の結果を提供するたびにメモリベースのバッキングデータ構造を更新するのではなく、キャッシュからの追い出し又はメモリへのキャッシュライトバックをトリガする他のフラッシュイベントに応答して更新されるからである。対照的に、アクセス頻度追跡構造自体がメモリに記憶されたデータ構造である場合、その構造は、偶然依存テストが第１の結果を提供するたびに更新され得る。

【００４２】
したがって、メモリベースの構造がアクセス頻度追跡構造として使用されるとき、アクセス頻度追跡構造は、メモリ空間内の或る特定のアドレスを割り振られてもよく、アクセス頻度追跡構造に対する更新は、所与のページに対応するアクセス頻度追跡インジケータがメモリ内でインクリメント又は更新されることを要求する１つ以上のメモリアクセス要求の発行を制御することによって、アクセス頻度追跡回路によって実行されてもよい。アクセス頻度追跡構造の全て又は一部は、メモリシステムの少なくとも１つのキャッシュ内にキャッシュされてもよく、したがって、アクセス頻度追跡構造を更新するためのメモリアクセス要求がメモリ階層内のより早いキャッシュ内でヒットする場合、メインメモリ内のバッキングストア内のアクセス頻度追跡構造に対応する位置を更新することは必ずしも必要ではない場合がある。

【００４３】
１つの例では、メモリベースのアクセス頻度追跡構造は、メモリアドレス空間の連続ブロック内にアドレスを有するいくつかのエントリを含む線形データ構造として実装することができ、アクセス頻度追跡回路は、所与のページのアドレスに基づいて決定されたオフセットに基づいて、所与のページに対応するアクセス頻度追跡インジケータを提供する線形データ構造のエントリを識別し得る。そのような線形データ構造は、どのページが他のページよりも頻繁にアクセスされたかを識別するためにソフトウェアがデータ構造をトラバースするとき、解析するのがより簡単であり得る。

【００４４】
代替として、メモリベースのアクセス頻度追跡構造は、ツリー構造として実装することができ、所与のページに対するアクセス頻度追跡インジケータにアクセスすることは、ツリーのリーフノードにおけるアクセス頻度追跡インジケータを識別するために、ツリーのルートノードからツリーの様々な分岐を通してポインタを辿るために、ツリー構造の複数のレベルをトラバースすることに依存し得る。場合によっては、ツリー構造は、メモリアドレス空間の対応するページに対するアドレス変換情報を指定するためのページテーブルエントリを追跡するために使用される任意のツリー構造から独立していてもよい。

【００４５】
しかしながら、多くの場合、ページテーブル構造は既に利用可能である場合があり、アクセス頻度追跡インジケータを提供するために再使用され得るページテーブルエントリに

おいて利用可能な或る特定の数のビットが存在し得るので、場合によっては、メモリベースのアクセス頻度追跡構造は、ページテーブル構造自体を含むことができ、その結果、所与のページに対応するページテーブルエントリは、そのページに対応するアクセス頻度追跡インジケータを含む。これにより、別個の構造を定義する必要がなくなり、ページテーブル構造をトラバースするための既存の機構をアクセス頻度追跡に再使用できるようになる。偶然依存テストに依存してアクセス頻度追跡インジケータを更新する確率的手法をサポートすることによって、これにより、アクセス頻度追跡インジケータを指定するために比較的少数のビットしか余裕がない場合でも、ページテーブルエントリを使用して、より広い範囲のアクセス頻度を追跡することが可能になる。

【００４６】

メモリベースのアクセス頻度追跡構造が線形データ構造として実装されるか、ツリー構造として実装されるかにかかわらず、アクセス頻度追跡構造のインデックス化は、所与のページに対応するアドレスに基づいて実行され得る。これは、様々なイベントが発生するにつれて多数のレコードがログデータストアに順次割り振られるイベントの順次ログとは異なり、レコードはそれぞれのイベントが発生した時間によって順序付けられ、各イベントレコードは特定のイベントに関する情報を提供する。所与のページのアドレスに基づいてインデックス化され得る構造を提供することによって、これは、ソフトウェアが特定のページに対応するアクセス頻度追跡インジケータを識別することをより効率的にする。

【００４７】

アクセス頻度追跡回路は、データ処理システム内の異なる位置に実装することができる。場合によっては、アクセス頻度追跡回路及びメモリアクセス回路は、命令セットアーキテクチャ（ＩＳＡ）に従って定義された命令に応答してデータ処理を実行する処理回路も備える処理要素内に設けられてもよい。例えば、処理要素は、中央処理装置（ＣＰＵ）又はグラフィック処理装置（ＧＰＵ）であってもよく、これらはアクセス頻度追跡回路を備えてもよい。

【００４８】

別の例では、装置は、メモリシステムへのダイレクトメモリアクセスを有するデバイスの代わりにメモリ管理動作を実行し得る入出力メモリ管理ユニット（ＩＯＭＭＵ）を備え得る。ＩＯＭＭＵは、「システムメモリ管理ユニット」（ＳＭＭＵ）と呼ばれ得る。ＩＯＭＭＵは、処理要素内のメモリ管理ユニット（ＭＭＵ）によって実行される動作に対応する動作を実行してもよいが、メモリシステムへのダイレクトメモリアクセスを有するデバイスの代わりにこれを実行してもよく、これらのデバイス自体は、ＩＳＡ定義のプログラム命令を実行することができる処理回路を有していなくてもよい。ＩＯＭＭＵは、そのようなデバイスから行われるメモリアクセス要求のアクセスの頻度も追跡され得るように、上述のようなメモリアクセス回路及びアクセス頻度追跡回路を備えてもよい。

【００４９】

場合によっては、データ処理システムは、各々がアクセス頻度追跡回路を有するメモリアクセス要求の複数のソース（例えば、複数の処理要素、又は少なくとも１つの処理要素及び少なくとも１つのＩＯＭＭＵ）を備え得る。その場合、処理要素及びＩＯＭＭＵ内のそれぞれのアクセス頻度追跡回路は、同じアクセス頻度追跡構造を更新するように構成され得る（又は、メモリに記憶された共有アクセス頻度追跡データ構造にバックアップされる異なるローカルアクセス頻度追跡構造を更新し得る）。代替的に、メモリアクセスの異なるソースはそれぞれ、デバイスによって行われるアクセスの頻度が処理要素によって行われるアクセスの頻度とは別に追跡されるように、それら自体の専用アクセス頻度追跡構造を更新することができる。

【００５０】

いくつかのシステムは、ＩＯＭＭＵを全く有さない場合があり、或いは、いくつかのシステムは、処理要素及びＩＯＭＭＵの両方を有する場合であっても、処理要素及びＩＯＭＭＵの一方においてアクセス頻度追跡回路のみを実装することができ、したがって、両方がアクセス頻度追跡回路を備えることは必須ではないことが理解されるであろう。

【００５１】
図１は、プログラム命令をデコードするための命令デコーダ６と、命令デコーダ６によってデコードされた命令に応答して処理動作を実行するように制御される処理回路８とを有する、少なくとも１つの処理要素（ＰＥ）４、例えばＣＰＵ（中央処理装置）又はＧＰＵ（グラフィックス処理装置）を有するデータ処理システム２の一例を概略的に示す。レジスタ９は、処理回路８によって実行される命令によってオペランドとして使用され得るデータを記憶するために提供される。処理回路８によって実行される演算の結果は、レジスタ９にライトバックされてもよい。ＰＥ４は、ＣＰＵによるより高速なアクセスのためにメモリからのデータをキャッシュする少なくとも１つのキャッシュ１０と、処理回路８によって実行されるメモリアクセス命令に応答してメモリシステム（キャッシュ（複数の場合もある）１０及びメインメモリ３２を含む）にアクセスするためのメモリアクセス要求を発行するメモリアクセス回路１１と、ＰＥ４によって実行される命令によって指定される仮想アドレスをメモリシステム内の位置を識別する物理アドレスに変換するアドレス変換回路として機能するメモリ管理ユニット（ＭＭＵ）１２とを有する。ＭＭＵ１２は、メモリシステムに記憶されたページテーブル構造からのページテーブルデータに依存する変換エントリを記憶するための少なくとも１つの変換索引バッファ（ＴＬＢ）１４を有し得る。ページテーブル構造は、仮想アドレスと物理アドレスとの間のアドレスマッピングを定義し、ＰＥ４上で実行される或る特定のソフトウェアプロセスが或る特定のアドレスにアクセスすることを許可されるかどうかを定義することができるメモリアクセス許可も定義することができる。ＭＭＵ１２は、仮想アドレスの物理アドレスへのマッピングが、仮想アドレス（ＶＡ）を中間物理アドレス（ＩＰＡ）にマッピングするステージ１アドレス変換データと、ＩＰＡを物理アドレス（ＰＡ）にマッピングするステージ２変換データとの両方に依存する、２ステージアドレス変換をサポートし得る。ステージ１アドレス変換データは、オペレーティングシステム又は仮想マシンによって設定され得る。ステージ２変換データは、ハイパーバイザによって設定することができる。

【００５２】
ＰＥ４は、プログラム命令を実行可能な要求元デバイスの一例である。図１は、２つのそのようなＰＥ４を有するシステムの一例を示すが、明らかにＰＥの数は変化し得る。いくつかのシステムは単一のＰＥのみを有し得るが、他のシステムは２つより多くのＰＥを有し得る。簡潔にするために、第２のＰＥ４の内部構成要素は図１に示されていないが、第２のＰＥ４は、命令デコーダ６、処理回路８、レジスタ９、キャッシュ（複数の場合もある）１０、メモリアクセス回路１１、及びＭＭＵ１２も含み得ることが理解されよう。場合によっては、ＰＥ４は対称であってもよく、同じ設計を有してもよい。異なる構成要素を有するか、又は処理回路８、キャッシュ（複数の場合もある）１０等の構成要素が異なるマイクロアーキテクチャ実装を有し得る異種又は非対称のＰＥ４を含むことも可能である。

【００５３】
ＰＥ４に加えて、システムはまた、内部ＭＭＵを有していなくてもよい要求元デバイス２０を含んでもよく、したがって、メモリにアクセスし、アドレス変換機能を提供するために、そのようなデバイスは、システムメモリ管理ユニット（ＳＭＭＵ）２２としても知られる入出力メモリ管理ユニット（ＩＯＭＭＵ）を介してシステムの残りの部分と通信してもよい。ＩＯＭＭＵ２２は、メモリ内のページテーブル構造において定義された変換データに基づいてアドレス変換及びメモリ許可を制御するアドレス変換回路を備える。ＩＯＭＭＵ２２は、ＣＰＵ４のＭＭＵ１２内のＴＬＢ１４と同様の機能を有する１つ以上のＴＬＢ２４を有してもよい。例えば、デバイス２０は、ディスプレイ上の画像フレームの表示を制御するためのディスプレイコントローラ、ネットワークを介してデータの入力又は出力を制御するためのネットワークコントローラ、ＣＰＵ４等の汎用プロセッサ上で実行されるソフトウェアを使用して達成することができるよりも効率的な方法で或る特定の専用処理機能を実行するためのハードウェアアクセラレータ等を含むことができる。デバイス２０は、メモリ３２へのダイレクトメモリアクセスを有し、これは、ＰＥ４によってそ

うするように明示的にプログラムされる必要なしに、記憶されたデータへのアクセスを直接要求するアクセス要求を発行することができることを意味する。

【００５４】

要求元デバイス４、２０は相互接続３０を介して互いに通信し、相互接続３０は、要求元デバイスとメモリ３２との間、及びそれぞれの要求元デバイス間のトランザクションをルーティングする役割を果たす。相互接続３０はまた、システムのそれぞれのキャッシュ１０にキャッシュされたデータ間のコヒーレンシを管理する役割を果たすことができる。図１は簡略化された図であり、システム２は、簡潔にするために図１に示されていない多くの他の構成要素を有し得ることが理解されよう。

【００５５】

図１に示されるように、処理要素４及び／又はＩＯＭＭＵ２２は、メモリアクセス回路１１によって行われるメモリアクセス要求を監視し、メモリアドレス空間のそれぞれのページに対するアクセスの相対頻度の指標を提供するアクセス頻度追跡構造を更新するためのアクセス頻度追跡回路４０を有してもよい。

【００５６】

ページアクセス頻度追跡のための１つの手法は、ＰＥ４又はＩＯＭＭＵ２２が、ページごとにアクセス／ダーティフラグを維持し、ページがアクセス又は修正されるときに変換記述子（ページテーブルエントリ）内にフラグを設定することである。アクセスフラグは、ページを「コールド」又は「ホット」ページに分類するためにオペレーティングシステムによって使用することができる。例えば、メモリが逼迫しているときにビクティムページがスワップアウトのために選択される必要がある場合、オペレーティングシステム（ＯＳ）は、すぐには再び使用されないページを対象にすることを好む場合があり、そのページが最近使用されたかどうかに基づいて、そのページが再び使用されるかどうかを予測することができる。

【００５７】

この面積は、メモリシステムにおける新たな不均一性のために特に注目されている。メモリのいくつかの部分は、メモリ技術又は処理要素若しくはアクセスエージェントへの物理的近接性に起因して、他の部分よりも高速（より高い帯域幅又はより低いレイテンシ）であり得る。ＮＵＭＡ（不均一メモリアクセス）移行又は多段ページ移行は、例えば、頻繁にアクセスされるデータをアクセスエージェントに近いページにコピーできるようにアクセスパターンを監視する等、積極的に研究されている。ページが頻繁にアクセスされることを検出することは、これの一部である（データのコピー／移動は別の問題である）。

【００５８】

頻繁にアクセスされるページを検出するための２つの可能な手法は、ページ保護及びページごとのアクセスフラグである。一般的な手順として、定期的なソフトウェアプロセスは、ページがアクセスされたかどうかをサンプリングし、この情報を、いくつかのサンプリング期間にわたってカウントを維持する（メモリに記憶され、ソフトウェアによって維持される）長期データ構造に統合する。次いで、このデータは、どのページが他のページよりも頻繁にアクセスされたかのビューを与える（例えば、ソフトウェアによって維持されるヒストグラム／ページごとのカウンタは、ページ間のサンプルの相対数を示す）。このソフトウェア手順は、以下のようにページアクセスフラグ又はページ保護とともに使用することができる。

【００５９】

オプション１－アクセスフラグは、サンプリング期間の開始時に全てのページにおいてクリアされる。遅延（周期的な時間、又は何らかの興味深いＯＳイベント）の後、ページテーブルがトラバースされ、各ページのアクセスフラグが観察される。アクセスフラグが設定されている場合、ソフトウェア手順は、長期データ構造を更新し（例えば、ページアドレスに対応するカウンタをインクリメントし）、アクセスフラグをクリアする。このプロセスを繰り返す。

【００６０】

オプション２－又は、ページの許可属性は、サンプリング期間の開始時にページをアクセス不可能にするように修正される。そのページがアクセスされた場合、ページフォルトが発生する。ページフォルトハンドラは、長期データ構造を更新し、ページを或る期間にわたってアクセス可能にする。（このイベントは、保護／サンプリングされるべき異なるページをトリガするために使用され得る。）

【００６１】

オプション１は、無指定であり、ページがアクセスされたとマークされていることを発見する可能性がない場合に、ソフトウェアがページテーブル全体をトラバースする必要があるという欠点を有し、このタスクは、アプリケーションとは異なるＣＰＵ上で実行することができるが、大きなページテーブルをトラバースするのに非常に長い時間がかかり、バックグラウンドＣＰＵ時間を浪費する。

【００６２】

オプション２は、アプリケーションがページフォルトによって頻繁に中断される場合にはアプリケーションにとって高オーバーヘッドであり、又はアプリケーションがまれにしか中断されない場合には不正確であるという欠点を有し、この方法は「インライン」でＣＰＵ時間を浪費する。

【００６３】

したがって、ページのアクセス状態をキャプチャすることは、（メカニズムに依存して）コストがかかるだけでなく、実行不可能なほど高いレートでサンプリングされない限り、アクセス状態は、進行中のページアクセス頻度に対する不十分なプロキシである。ページごとのカウンタを維持することなく、非常にホットなページから「時々」アクセスされるページを区別することは困難になり、これも空間的にコストがかかる。一例として、１００万回／秒でアクセスされたページと比較して、１００～１０００回／秒でアクセスされたページは、「コールド」とみなされ得る。

【００６４】

以下に説明する例では、アドレス空間内のそれぞれのページに対応するエントリを有するアクセス頻度追跡構造が（例えば、メモリ内又はハードウェア記憶構造内に）設けられる。追跡構造は、ロード又はストアごとに更新される大きなページごとのカウンタを有するように近似することができるが、その手法の天文学的コストはない。「ページがホットである」という決定に対しては、ページが閾値よりも頻繁にアクセスされた場合にホットであるとみなされるような閾値が採用されてもよいことが観察される（しかし、その時点の後は、相対値はそれほど重要ではない）。本出願で説明する技法では、アクセス頻度追跡回路４０は、ロード命令又はストア命令の実行に、そのロード／ストアのアクセスされたアドレスに対応する追跡構造内のビット又はアクセス頻度追跡カウンタを更新する構成可能な確率を与える。確率はレジスタ内に構成され、使用中のページ移行アルゴリズムに応じて調整することができる。１つの具体例では、これは、各ロード／ストア命令の実行時にテスト評価カウンタをインクリメントすることによって実施することができる。テスト評価カウンタが構成された値に達すると、ロード／ストア命令の実行は、追跡構造の更新とテスト評価カウンタ値のリセットとの両方を引き起こす。構成可能な確率を有する偶然依存テストを実装する他の例も説明される。この技法は、ＩＯ（入力／出力）アクセスにも適用することができ、それによって、ＩＯＭＭＵ２２は、クライアントデバイス２０からの読み取り／書き込みトランザクションに応答して二次データ構造を更新する。

【００６５】

この特徴が装置のハードウェア回路に実装される場合、これは、「ホット」ページのアドレス空間をプロファイリングするソフトウェアが、「ホット」ページとみなすものに適切な確率レジスタを（確率１／Ｐを定義するために）設定することを可能にする。１つの特定の例では、二次データ構造内の情報は、以下を意味することができる。

１－対応するページはＰ／２回を超えてアクセスされた可能性が高い

０－対応するページはＰ／２回を超えてアクセスされた可能性が低い

【００６６】

この考え方の変形形態には以下のものが含まれる。
－二次データ構造の代わりに、対応するＰＴＥ内のフィールドに対して確率的更新を行うことができる。
－ストアの代わりにロードについて異なる確率値を使用する。
－ロード対ストアに基づいて異なる二次構造を更新する。
－アドレス空間の異なる面積に対して異なる確率値を有する。
－データ構造においてページ当たり１ビット（「ページがホット」であることを意味する）の代わりに、小さい（例えば２～３ビット）飽和カウンタを更新する。これにより、「ページがホットである」と「ページが非常にホットである」とを区別することができ、移行アルゴリズムがそれらの間で優先順位を付けることができる。これはまた、誤検出のノイズを或る程度フィルタリングすることを可能にする。

【００６７】

誤検出が発生することになり（例えば、以前にアクセスされたことがないアドレスにアクセスするロードが、更新をトリガし、正しく「サイコロを振る」ことになる）、ソフトウェアアルゴリズムは、或る期間にわたって更新をフィルタリングし得る。

【００６８】

アクセスの実際の頻度と記録された追跡情報との間の改善された相関を提供することができる（誤検出の可能性が低減される）変形形態は、以下のものであり得る。
－ＴＬＢエントリを、そのエントリにヒットしたＴＬＢルックアップに対して、そのルックアップについてサイコロ振り確率テストがパスされた場合にインクリメントされるカウンタで拡張する。
オーバーフロー時に、ＴＬＢエントリのアドレスに対応する構造ビットが更新される。

【００６９】

（これは、キャッシュ内のＬＲＵカウンタと同様に実装され得る）

【００７０】

図２は、処理要素４内又はＩＯＭＭＵ２２内のいずれかに含まれ得るアクセス頻度追跡回路４０の一例を概略的に示す。アクセス頻度追跡回路４０は、所与のメモリアクセス要求に対して実行された偶然依存テストが第１の結果を有するか、又は第１の結果以外の別の結果を有するかを判定するための偶然依存テスト評価回路４２を有する。偶然依存テストのいくつかの異なる例が、図４～図８に関して以下で説明される。アクセス頻度追跡回路４０は、確率を制御することができる１つ以上の構成可能な値を定義する少なくとも１つのソフトウェア構成可能な確率レジスタ４４を有し、偶然依存テスト評価回路が第１の結果を提供する。例えば、確率レジスタは、偶然依存テストが第１の結果を提供するかどうかを評価するために使用されるカウンタ又は他の入力値５４内のビット数を構成する値を記憶することができる。

【００７１】

図２に示されるように、いくつかの例では、ソフトウェア構成可能な確率レジスタ４４は、メモリアクセス要求の異なるサブセットに対応する２つ以上の別個の確率構成値４６を含んでもよく、それにより、第１の結果を提供する偶然依存テストの確率は、メモリアクセス要求の異なるサブセットに対して独立して設定され得る。例えば、異なるサブセットは、それぞれロード及びストアを含むことができ、又は処理要素４の異なる動作状態若しくは実行モードから発行された要求を含むことができ、又はメモリアドレス空間の異なる領域を対象とする要求を含むことができ、又はこれらのプロパティのうちの２つ以上の異なる組み合わせに対応することができる。ソフトウェア構成可能な確率レジスタ４４内の確率制御値４６を更新する能力は、或る特定の例外レベル以上で実行される命令等、或る特定のレベルの特権を有する処理要素によって実行される命令に制限することができる。より低い例外レベルで実行される命令、又は特権が与えられていない命令は、ソフトウェア構成可能な確率レジスタ４４内の値を変更することを許可されない場合がある。場合によっては、ソフトウェア構成可能な確率レジスタは、命令デコーダ６及び処理回路８によってサポートされる命令セットアーキテクチャ内の制御レジスタとして実装され得るが

、他の例では、ソフトウェア構成可能な確率レジスタ４４は、ソフトウェア構成可能な確率レジスタ４４にマッピングされたターゲットメモリアドレスを指定するロード／ストア命令を実行する処理回路８によってアクセスされるメモリマッピングされたレジスタであり得る。

【００７２】

アクセス頻度追跡回路４０はまた、テストが第１の結果を提供すると偶然依存テスト評価回路４２が判定したときにアクセス頻度追跡構造５０への更新をトリガするためのアクセス頻度追跡構造更新回路４８を有する。偶然依存テストが第１の結果以外の結果を提供する場合、アクセス頻度追跡構造５０の更新は必要ない。場合によっては、アクセス頻度追跡構造は、アクセス頻度追跡ハードウェア構造等の専用ハードウェアデータ構造、又はアクセス頻度追跡構造の記憶専用のレジスタのセット内に記憶されてもよい。例えば、アクセス頻度追跡ハードウェア構造はＴＬＢ１４であってもよい。

【００７３】

しかしながら、他の例では、偶然依存テストが第１の結果を与えるときに更新されるアクセス頻度追跡構造は、メモリ３２内のデータ構造として実装することができ、そのいくつかの部分は汎用データキャッシュ１０内にキャッシュすることができる。メモリベースの構造は、アクセス頻度追跡構造５０内の位置に対応するアドレスをターゲットアドレスとして指定する１つ以上のメモリアクセス要求を発行するアクセス頻度追跡構造更新回路４８によって更新することができる。これは、アクセス頻度追跡インジケータを保持するためのＰＥ４又はＩＯＭＭＵ２２内の追加の記憶論理を必要としないので、ＰＥ４又はＩＯＭＭＵ２２内の回路面積コストを低減することができる。

【００７４】

アクセス頻度追跡構造は、アドレス空間の１つ以上のページのブロックにそれぞれ対応し、そのページのブロックへのアクセスの相対頻度の指標を提供する、いくつかのアクセス頻度追跡インジケータ５２を含むことができる。場合によっては、アクセス頻度追跡インジケータ５２は、ページごとに提供され得るが、これは必須ではなく、いくつかのページは、それらの追跡構造内に対応するインジケータを有する必要はない（例えば、仮想アドレスから物理アドレスへのマッピングが定義されていない現在マッピングされていないページに対応するインジケータを含む必要はない場合がある）。また、いくつかの例では、各アクセス頻度追跡インジケータは、それぞれのページが異なる追跡インジケータを有するように単一のページに対応することができるが、他の場合には、追跡の粒度はあまり細かくなくてもよく、単一のインジケータ５２が隣接ページのブロック間で共有されてもよい。特定のページに対応する関連インジケータ５２は、そのページのアドレスに基づいて識別することができ、例えば、アドレスを使用して、構造５０のベースアドレスに対するオフセットを導出することができ、又はページアドレスのインデックスビットの様々なセットを使用して、ツリー構造を通じた横断を制御することができる。アドレス変換が使用される場合、対応する物理アドレスを取得する必要はなく、ソフトウェアが仮想アドレスに基づいて構造５０を解析することができるように、データ構造５０へインデックス化するために使用されるのはページの仮想アドレスであってもよい。

【００７５】

アクセス頻度追跡構造更新回路４８は、実行されるべき更新を制御するための明示的なソフトウェアストア命令を必要とせずに、アクセス頻度追跡構造の任意の必要な更新を自動的に開始するハードウェア回路論理を備えることができるが、アクセス頻度追跡構造５０は、データ構造内のアドレスを指定するロード命令を発行するソフトウェアによってアクセス可能であってもよく、これは、オペレーティングシステム又は他のソフトウェアが、頻繁にアクセスされた可能性があるページを識別するためにアクセス頻度追跡構造５０を後で解析したい場合に有用であり得る。

【００７６】

図３は、アドレス空間のそれぞれのページへのアクセスの相対頻度を監視する方法を示すフロー図である。工程Ｓ１００において、アクセス頻度追跡回路４０は、メモリアクセ

ス回路１１が所与のページへのアクセスを要求するメモリアクセス要求を受信したか又は処理していることを検出する。メモリアクセス要求が受信又は処理されたことに応答して、工程Ｓ１０２において、偶然依存テスト評価回路４２は、そのメモリアクセス要求に対して評価された偶然依存テストの結果を決定する。偶然依存テストは、偶然に少なくとも部分的に依存し得る不確実な結果を生成する任意のテストとすることができ、その結果、偶然依存テストの結果は、メモリアクセス要求の特性のみに基づいて、又はメモリアクセス要求が許可されるかどうかを管理するために使用される任意の制御パラメータ、又はメモリアクセス要求に応答してアクセスされるメモリシステム内の位置を制御する任意の制御パラメータに基づいて決定論的に導出されることができない。例えば、偶然依存テストは、乱数若しくは擬似乱数、パルス列からの任意にサンプリングされたビット、又はメモリアクセス要求自体と相関していないシステム内でサンプリングされた他の内部状態、或いはメモリアクセス要求が受信されたときに特定の値を有する或る特定の１／Ｎ確率を有し得るメモリアクセス要求の数をカウントするために設けられたカウンタに依存し得る。偶然依存テストが実行される方法にかかわらず、テストは、同じ特性を有する全く同じメモリアクセス要求が何度も発行された場合であっても、偶然依存テストが、同じメモリアクセス要求のそれらの繰り返されたインスタンスに対して異なる結果を有し得るようなものであり得る。これは、アクセス頻度追跡構造５０が、所与のページへのメモリアクセスのあらゆる発生に応答して更新されるのではなく、時には偶然依存テストが成功する場合があることを意味し、したがってアクセス頻度追跡構造が更新され、場合によっては偶然依存テストが失敗し、更新が実行されず、個々のメモリアクセス要求の結果は、本質的に任意であるが、ソフトウェア構成可能な確率レジスタ４４によって制御される構成可能な確率を有する。

【００７７】

したがって、工程Ｓ１０４において、偶然依存テスト評価回路４２又はアクセス頻度追跡構造更新回路４８は、偶然依存テストの結果が第１の結果であるかどうかを判定し、第１の結果である場合、工程Ｓ１０６において、アクセス頻度追跡構造更新回路４８は、アクセス頻度追跡構造５０内のメモリアクセス要求によってアクセスされた所与のページに対応するアクセス頻度追跡インジケータ５２を更新する。偶然依存テストの結果が第１の結果でなかった場合、工程Ｓ１０８において、所与のページに対応するアクセス頻度追跡インジケータ５２のこの更新は省略される。

【００７８】

したがって、アクセス頻度追跡構造５０が、対応するページがアクセスされるたびに更新されないように、第１の結果を提供する確率が１未満である偶然依存テスト４２に基づいて、所与のメモリアクセス要求に対してアクセス頻度追跡構造５０が更新されるかどうかを制御することによって、これは、アクセス頻度追跡構造５０の目的が比較的広い範囲にわたってメモリアクセスをカウントすることである場合でも、アクセス頻度追跡インジケータ５２に必要な任意のカウンタのサイズが、追跡されることを目的とするアクセスの数と同じ範囲でスケーリングする必要がないことを意味する。なぜなら、インクリメントの可能性が低いことは、大きい絶対範囲を測定しながら、より小さいカウンタを採用できることを意味し、したがって、多くの状態ビットを節約し、したがって、アクセス頻度追跡構造を実装する面積、電力、及び性能コストを削減するからである。

【００７９】

図２に示されるように、偶然依存テストは、偶然依存テスト評価回路に提供される或る特定の入力情報５４に依存し得る。この情報は、実装される偶然依存テストの性質に応じて、或る範囲の形式を有することができる。一般に、偶然依存テストは、メモリアクセス要求自体の性質と特に相関しない方法で実質的にランダムに変化し得る入力情報を使用し得る。ソフトウェアの観点から、特定のメモリアクセス要求が偶然依存テストをパスするかどうかは、特定のメモリアクセス要求に対する結果が不確実であるように、サイコロ振りのイベントとして見ることができるが、一般に、所与のページがより頻繁にアクセスされるほど、そのページが獲得するサイコロの振りが多くなり、したがって、そのページが

、他のより頻繁にアクセスされないページと区別するように設定されたそのアクセス頻度追跡インジケータ５２を有する可能性が高くなる。それにもかかわらず、最初にアクセスされたページがたまたまサイコロの目に当たり、以前にアクセスされたことがなくても偶然依存テストが第１の結果を提供する一方で、より頻繁にアクセスされたページはサイコロの目で負け続け、そのアクセス頻度追跡インジケータが更新されることがないという誤検出が時折存在する可能性がある。それにもかかわらず、平均して、より頻繁にアクセスされるページは、この機構を使用して識別される可能性がより高く、したがって、これは、メモリシステム内に或る特定のデータを配置するためのいくつかの配置決定が、より頻繁にアクセスされるページを支持するように最適化されることを可能にすることによって、性能のために依然として有益であり得る。

【００８０】

図４～図８は、偶然依存テストを実装するためのいくつかの異なる例を示す。図４に示されるように、１つの例では、グローバルカウンタ６０は、アクセス頻度追跡回路４０によって維持されてもよく、グローバルカウンタは、全てのメモリアクセス要求の間で共有される。メモリアクセス要求が処理されると、グローバルカウンタ６０はインクリメントされるか、さもなければ次の状態に進められる。カウンタの１つの実施態様は、各メモリアクセス要求に応答して１を加算又は減算することによってインクリメントされるバイナリ値とすることができるが、他の手法では、カウンタを、必ずしもインクリメント又はデクリメントされる必要がないように状態の或る特定のシーケンス間で遷移するシフトレジスタとして表すことができる（例えば、シフトレジスタを１ビットの場所だけ右又は左にシフトすることができる）ことが理解されよう。グローバルカウンタ６０の現在の値は、比較器６２によって所定の値又は閾値と比較され、比較に応じて、信号がアクセス頻度追跡構造更新回路４８に発行されて、対応するアクセス頻度追跡インジケータ５２が、現在のメモリアクセス要求によってアクセスされる対応するページに対して設定又は更新されるかどうかを制御する。したがって、この手法では、偶然依存テストが第１の結果をもたらすかどうかは、グローバルカウンタが閾値に達したか、又は所定の値に一致するかどうかに基づき、これは個々のメモリアクセス要求に対して事実上ランダムである場合があり、その理由は、カウンタがその閾値を経過又は到達したときにどの特定のメモリアクセス要求がたまたま処理されるかは事実上任意である場合があり、他のメモリアクセス要求に対するそのメモリアクセス要求の処理のタイミングに依存でき、それは、順不同実行がサポートされるかどうか、及び動作を再順序付けすることができる任意の動作キューのサイズ等のマイクロアーキテクチャ実装の詳細、並びにトリガされる特定の割り込みのタイミング等の実行ランごとに異なり得るイベントに依存でき、したがって、カウンタ値は、カウンタが第１の結果を決定するために必要な条件を満たすかどうかが偶然の問題として見られ得るように、所与のメモリアクセス要求の特性に相関しないと見られ得るからである。カウンタ６０は、所定の条件が満たされる状態に達したと判定された後、その初期状態にリセットされて、メモリアクセス要求をカウントする別の期間を再開する。

【００８１】

図５は、グローバルカウンタ６０が、各々がメモリアクセス要求の或る特定のサブセットに対応するいくつかのローカルカウンタ６４で置き換えられることを除いて、図４と同様の手法を示す。第１のサブセット内のメモリアクセス要求に遭遇すると、第１のローカルカウンタ６４－１の状態がインクリメント又は進められ、同様に、第２のサブセット内のメモリアクセス要求に遭遇すると、第２のローカルカウンタ６４－２の状態がインクリメント又は進められる。現在のメモリアクセス要求に対する偶然依存テストを評価するために使用するローカルカウンタは、そのメモリアクセス要求がどのサブセットに属するかに基づいて選択され、比較器６２は、選択されたカウンタ６４の状態を図４と同様の所定の値又は閾値と比較する。カウンタが所定の値又は閾値に達すると、アクセス頻度追跡構造更新回路４８は、アクセス頻度追跡構造５０内の関連する追跡インジケータ５２の更新をトリガする。また、選択された関連カウンタ６４は、別の周期のカウントを再開するために、その初期状態にリセットされてもよい。ここでも、メモリアクセス要求を異なるサ

ブセットに分類する異なる方法が存在してもよく、例えば、サブセットは、メモリアクセス要求のタイプ（例えば、ロード対ストア）、メモリアクセス要求によって対象とされるアドレス空間の特定の領域、及び／又はメモリアクセス要求に関連付けられた実行モード若しくは状態若しくは特権レベル（又はこれらの要因の任意の組み合わせ）によって区別されてもよい。

【００８２】

図４及び図５等のカウンタを使用する例では、ソフトウェア構成可能な確率レジスタ４４は、初期状態（更新イベントが発生した直後にカウンタがリセットされる）と次の更新イベントが検出される状態（カウンタが所定の閾値又は条件を満たすとみなされるとき）との間でカウンタ６０、６４を進めるのに必要な遷移の数を制御する値を記憶することができる。例えば、ソフトウェア構成可能な確率レジスタ４４内の値は、偶然依存テスト評価回路４２に、更新イベントが発生した直後にカウンタ６０、６４にリセットされる値を調整させるか、又はカウンタが比較される値又は閾値を調整させ、同様の結果を提供することができる。

【００８３】

図６は、偶然依存テストを実装する別の例を示す。この例では、偶然依存テストのための入力５４は、偶然依存テストが評価されている関連するメモリアクセス要求のときにサンプリングされたシリアルパルス列又は内部状態６８の項目内からサンプリングされたいくつかのビットを含み得る。シリアルパルス列は、経時的なそれぞれのインスタンスにおいて特定の信号又は信号のグループからサンプリングされた値から導出されるビットのシーケンスであり得る。例えば、或る特定の信号の値は、いくつかのクロックサイクルにわたってデジタル的にサンプリングされてもよく、サンプリングされた値が、所定の値に一致する１及び０の或る特定のパターンを提供する場合、更新条件は、満たされたとみなされてもよく（第１の結果は、偶然依存テストにおいて発生する）、一方、１及び０の他のパターンは、発生しない第１の結果に対応してもよい。或いは、クロックサイクルの経過、キャッシュ１０におけるキャッシュミス、アドレス障害又は他のエラーの数等のイベントをカウントするためのイベントカウンタ等、他の機能目的を有する内部状態の項目を使用して内部状態６８を提供することができ、この内部状態６８から或る特定の数のビット５４を抽出し、所定の値と比較して、第１の結果が生じたかどうかを判定することができる。内部状態のどの特定の項目がサンプリングされるか、又はどの特定のアナログ若しくはデジタル信号がシリアルパルス列としてサンプリングされるかの選択は、特定のプロセッサ実装に依存し得る。場合によっては、内部状態の複数の項目を組み合わせて（例えば、排他的論理和（ＸＯＲ）演算に基づいて）、特定のセットの値に対するバイアスがより少ないサンプリングされたビット５４を有することができるパルス列又はデータ値６８を提供することができる。したがって、図６の例では、メモリアクセスを追跡するための専用カウンタ６０、６４を設ける必要はないが、その代わりに、処理システム２内の内部状態又は既存のアナログ若しくはデジタル信号のいくつかの項目を使用するか、又は他の項目と組み合わせて、偶然依存テストを提供することができる。

【００８４】

図７は、乱数又は擬似乱数７０が取得され、所定の値と比較され、乱数又は擬似乱数が所定の値と一致するかどうかに応じて、アクセス頻度追跡構造５０を更新するかどうかが判定される別の例を示す。これは、処理システム２が、暗号化等の他の目的のために乱数発生器又は擬似乱数発生器を既に有している可能性があるという事実を利用することができ、したがって、既に利用可能なランダム性のソースを使用して、或る特定の数のビットを有する乱数又は擬似乱数を提供することができ、その結果、Ｎビットの乱数又は擬似乱数は、第１の結果を表す特定の値を有する１／Ｎの可能性をもたらすことができ、それが所定の値に一致しないとき、第１の結果以外の結果を提供する。乱数又は擬似乱数は、偶然依存テストが評価される各メモリアクセス要求に対して再生成され得る。

【００８５】

図６及び図７の両方の例において、ソフトウェア構成可能な確率レジスタ４４は、現在

アクセスされているページのアクセス頻度追跡インジケータ５２が更新される確率を調整するために、図６の所定の値の間で比較されるサンプル内のビット数、又は図７の所定の値と比較される乱数／擬似乱数７０内のビット数を制御することができる。

【００８６】

図８は、偶然依存テストを実装するための別の手法を示し、この手法は、図４及び図５と同様に、偶然依存テストが所与のメモリアクセス要求に対して第１の結果を提供するかどうかを判定するためのテスト評価カウンタとして機能するローカルカウンタ６４を使用する。しかしながら、図８の例では、ＴＬＢ１４は、ＴＬＢエントリ有効性を示す有効ビット８４及びアドレス空間の所与のページに関連付けられた関連するＴＬＢエントリ内にキャッシュされた任意のページテーブルデータ８２に加えて、ＴＬＢエントリが、その特定のページを対象とするメモリアクセス要求に対応するローカルカウンタとして機能するテスト評価カウンタ６４も記憶するように拡張された各ＴＬＢエントリ８０を有する。したがって、この手法では、ページテーブルデータ８２が有効なＴＬＢエントリ８０内にキャッシュされる各ページは、ＭＭＵ１２によってそのページのために維持される対応するローカルカウンタ６４を有し、その結果、所与のメモリアクセス要求が発行されたとき、アクセス要求がＴＬＢ１４内でヒットした場合、そのページテーブルデータ８２が読み出され、アドレス変換又はアクセス許可の評価を制御するために使用され、カウンタ６４が読み出され、比較器６２によって所定の値又は閾値と比較されて、アクセス頻度追跡構造５０を更新するかどうかを判定する。また、ＴＬＢにヒットがあった場合、対応するＴＬＢエントリ８０内のテスト評価カウンタ６４は、その遷移シーケンス内の次の状態に進むように更新される（例えば、インクリメントされる）。ＴＬＢ１４内の異なるページに対して異なるテスト評価カウンタ６４を維持することによって、これは、アクセス頻度追跡構造５０内のアクセス頻度追跡インジケータ５２が、対応するページに対するアクセスの実際の相対頻度とより密接に相関される見込みを増加させることができ、それらのページがアクセスされたときに実行されるそれぞれの偶然依存テストに対する偶然の影響に起因して、より頻繁にアクセスされるページ及びより頻繁にアクセスされるページとして、より頻繁にアクセスされないページが誤って識別される確率を低減する。

【００８７】

図９～図１１は、メモリベースのアクセス頻度追跡構造５０を実装する異なる方法を示す。これらの技法は、ＴＬＢ等のハードウェアキャッシング構造からのアクセス頻度追跡インジケータがライトバック又はフラッシュされるメモリベースのバッキング構造にも使用することができる。

【００８８】

図９の例では、構造は、ベースアドレス９０から始まるメモリアドレスのブロックとして記憶される線形データ構造を提供するアクセス頻度ビットマップとして実装され、各アクセス頻度追跡インジケータ５２は、アクセス頻度を追跡する期間の開始時に全てのフラグがリセットされる第１の値（この例では０）を有し、その追跡インジケータ５２に対応する関連ページを対象とする特定のメモリアクセス要求に対して偶然依存テストが第１の結果を有するときに第２の値（この例では１）に設定される単一ビットフラグである。したがって、第２の値を有するインジケータ５２は、偶然依存テストが第１の結果を有したときに少なくとも１回アクセスされたページを表す。第２の値を有するフラグは、確率が少なくともＰ回アクセスされたページの半分より大きいことを示すものとして見ることができ、ここで、１／Ｐは、偶然依存テストが第１の結果を有する確率である。アクセス頻度ビットマップ５０へのインデックス化は、ベースアドレス９０と、対応するページのアドレスから導出され得るオフセット９２とに基づいて実行され得る。これは線形データ構造であるので、全てのアクセス頻度追跡インジケータ５２は、仮想アドレス空間の連続ブロック（使用されるアドレスマッピングに応じて物理アドレス空間内で連続していてもよいし、連続していなくてもよい）内に記憶され得る。

【００８９】

図１０は、やはりアクセス頻度追跡構造が、ベースアドレスから始まるアドレスに記憶

され、特定の追跡インジケータ５２に対応するページの（仮想）ページアドレスから導出されたオフセット９２に基づいてインデックス化された線形データ構造である第２の例を示す。しかしながら、この例では、各追跡インジケータは、偶然依存テストが第１の結果を提供したときに対応するページがアクセスされた回数をカウントするマルチビットアクセス頻度追跡カウンタである。これらのカウンタは、図４、図５及び図８に関して説明したテスト評価カウンタ６０、６４とは異なり、これらのカウンタは、個々のメモリアクセス要求に対して偶然依存テストが満足されるかどうかを評価するためのものであるが、特定のページのアクセスの相対頻度の指標を提供するものではないことに留意されたい。上述の偶然依存更新方法を使用することによって、カウンタ５２は、依然としてより大きい範囲にわたってカウントをシミュレートしながら、アクセス頻度追跡構造がメモリアクセス要求ごとに更新された場合に必要とされるよりもはるかに小さくすることができる。図９及び図１０の両方の例において、いくつかの実施態様は、アドレス空間の少なくとも一部分内のそれぞれのページに対してアクセス頻度追跡インジケータ５２を提供することができ、その結果、追跡インジケータを設定する粒度はページ単位である。他の手法は、２つ以上の隣接ページのブロック間で追跡インジケータ５２を共有することができる。

【００９０】

図１１は、ハードウェアキャッシング構造からアクセス頻度追跡インジケータをバックアップするためにメモリベースのアクセス頻度追跡構造又はメモリ内のバッキング構造を維持するための別の手法を示し、図９及び図１０に示す線形データ構造の代わりにツリー構造が使用される。アドレス変換データを提供するためにＭＭＵ１２又はＩＯＭＭＵ２２によって使用されるページテーブルとは別にツリー構造を維持することが可能であるが、実際には、アドレス変換データは、ツリー構造として管理されるページテーブルのセットにおいて定義されてもよく、アクセス頻度追跡インジケータ５２を提供する（又はハードウェアにおいて提供されるキャッシング構造からアクセス頻度追跡インジケータ５２をバックアップする）ために使用され得るいくつかの予備ビットを各ページテーブルエントリにおいて有してもよく、したがって、場合によっては、ツリー構造はページテーブル構造自体であってもよい。図１１の右下に示すように、ツリー構造は、ツリーのより高いレベルのノードがツリーのより低いレベルのそれぞれのノードへのいくつかのポインタを含むいくつかのツリーノード１００を含むことができ、ツリー構造を照会するために使用されるアドレスからのビットの一部は、ツリーの次のレベルのノードの位置を取得するために後続する代替ポインタを選択するために使用され、照会アドレスの異なる部分は、ツリーの異なるレベルのこれらのポインタ間で選択するために使用される。最終的に、ツリーの最終リーフエントリに到達する（例えば、この例ではレベル３であるが、ツリーのより早いレベルのツリーノードが符号化されて、ツリー横断を終了し、それ自体がクエリアドレスに対する関連情報ソートを提供することを示すことも可能である）。リーフエントリは、クエリアドレスに対して求められたデータ、例えば、ページテーブルに対するページテーブルエントリ、及び／又はアクセス頻度追跡データ構造５０に対するアクセス頻度追跡インジケータ５２を提供する。

【００９１】

例えば、特定の仮想ページアドレスに対応するアドレス変換データを取得するために、ＭＭＵ１２又はＩＯＭＭＵ２２は、レベル０（Ｌ０）ベースアドレス１２０と、仮想ページアドレスのＬ０インデックスビット１２４から導出されたオフセット１２２とを使用して、仮想ページアドレスに対応するレベル１（Ｌ１）ページテーブル１３２のベースアドレス１３０へのポインタ１２８を提供するレベル０ページテーブル１２６のエントリを選択することができる。レベル０ページテーブル１２６の異なるエントリは、ツリーがレベル０の後のツリーの次のレベルにある異なるレベル１ノードに分岐することを可能にする異なるポインタを提供することができる。同様に、ツリーの各後続レベルにおいて、インデックスビット１３４、１３６、１３８の更なる部分が、それぞれレベル１、２、及び３ページテーブルのベースアドレス１３０、１４０、１４２に対するオフセットを提供するために使用される。ツリーのブランチノード（実際のページテーブルエントリ自体を提供

するリーフノード以外のノード）において、選択されたエントリは、次のレベルへのポインタを提供する。最終的に、関心のあるページテーブルエントリ１５０を提供する（例えば、対応するページに対するアドレス変換マッピング及び／又はアクセス許可を符号化する）リーフエントリが（レベル３において、又はツリー横断が早期に終了する場合はより早期のレベルにおいて）見つけられる。このページテーブルエントリは、アクセス頻度追跡インジケータ５２を提供するように拡張することができ、アクセス頻度追跡インジケータ５２は、図９及び図１０の例のような単一ビットフラグ又はマルチビットカウンタとすることができる。したがって、この例では、アクセス頻度追跡回路４０は、ページテーブルをトラバースするためにＭＭＵ１２又はＳＭＭＵ２２内に設けられた機構を再使用して、所与のメモリアクセスに関する偶然依存テストが第１の結果を提供するときに関連するアクセス頻度追跡インジケータ５２が更新されるべきアドレスを決定することができる。したがって、複数のメモリアクセスが、更新されるべきアクセス頻度追跡インジケータ５２を見つけるために必要とされ得る。他の例では、ページテーブルから独立し得るツリー構造が使用されてもよく、その場合、独立したツリー横断機構がアクセス頻度追跡回路４０内に実装されてもよい。図１１に示すツリーレベルの数は１つの例にすぎず、４レベルである必要はないことが理解されよう。

【００９２】

図１２は、アクセス頻度追跡ハードウェア構造１５８がＰＥ４又はＩＯＭＭＵ２２に設けられた別の例を示しており、このアクセス頻度追跡ハードウェア構造は、所与のメモリアクセスに対する所与のページのアドレスに基づいてルックアップされて、その所与のページに対するアクセス頻度追跡インジケータ５２を取得することができるいくつかのストレージエントリ１５９を含む。アクセス頻度追跡ハードウェア構造１５８は、アクセス頻度追跡のための専用ハードウェア構造とすることができるが、この例では、アクセス頻度キャッシングハードウェア構造１５８はＴＬＢ１４であり、したがって、各エントリ１５９は、対応するページのページテーブルデータ８２も指定する。この場合も、各エントリは、エントリが有効であるかどうかを示す有効ビット８４を有することができる。

【００９３】

アクセス頻度追跡インジケータ５２は、偶然依存テストが第１の結果を与えた対応するページへのアクセスでインクリメントされるアクセス頻度追跡カウンタであってもよい。

【００９４】

したがって、この例では、ターゲットページへのアクセスが要求されたとき、アドレスがハードウェア構造１５８内でヒットし、偶然依存テストが第１の結果を与える場合、追跡構造１５８内の対応するエントリ内のアクセス頻度追跡インジケータ５２がインクリメントされ、偶然依存テストが別の結果を与える場合、このインクリメントは省略される。

【００９５】

アドレスが追跡構造にない場合、ターゲットページのアドレス及び初期値（例えば、０又は１）に設定されたアクセス頻度追跡インジケータ５２に新しいエントリを割り振ることができる。追跡構造１５８がＴＬＢ自体１４である場合、偶然依存テストが第１の結果を与えるかどうかにかかわらず、ＴＬＢミス時に新しいエントリを割り振ることができる（カウンタもインクリメントされるかどうかにかかわらず、ターゲットページのページテーブルデータ８２をＴＬＢにキャッシュするために新しいエントリがいずれにしても必要とされる場合があるので、その場合、新たに割り振られたエントリの初期値は、偶然依存テストが第１の結果を与えなかった場合は０、偶然依存テストが第１の結果を与えた場合は１とすることができる）。追跡構造１５８がＴＬＢ１４以外の専用構造である場合、偶然依存テストが構造内のミスに対して最初の結果を与える場合に新しいエントリを割り振ることができるが、偶然依存テストが別の結果を与える場合には割り振りを必要としない場合がある。

【００９６】

図１２の例では、偶然依存テストは、図４の例のようにグローバルテスト評価カウンタ６０を使用して実施されるが、図１２の例は、例えば図５～図８に示すような偶然依存テ

ストを実施するための他の技法のいずれかを使用することもできる。図８の例が偶然依存テストのために使用される場合、追跡構造の各エントリ１５９は、アクセス頻度追跡インジケータ５２に加えて、対応するページのためのローカルテスト評価カウンタ６４を含むことができ、その結果、両方のタイプのカウンタ６４、５２が提供され得るが、実際には、図４～図７に示されるオプションは、ＴＬＢエントリ８０ごとに２つの別個のカウンタを提供する必要を回避するために、回路面積に関してより安価であり得る。

【００９７】

いくつかの例では、ハードウェア構造１５８（１４）からアクセス頻度追跡インジケータ５２をバックアップする基礎となるメモリベースの構造が存在しない場合があるが、ソフトウェアがハードウェア構造１５８から所与のページのアクセス頻度追跡インジケータ５２の値を読み取る機構が存在する場合があり、ソフトウェアは、ソフトウェアによって選択されたメモリ位置へのそのような追跡インジケータ５２の書き込みを制御することができる。この場合、追跡構造１４からアクセス頻度追跡インジケータ５２をバックアップするハードウェア管理手段は存在しなくてもよい。この手法では、構造１５８のエントリ１５９が（例えば容量競合のために）無効にされるか又は追い出される場合、対応するアクセス頻度追跡インジケータ５２を単に破棄することができる。例えば、ＴＬＢ１４がアクセス頻度追跡ハードウェア構造１５８として機能する場合、常駐ＴＬＢエントリを有さない十分にまれにアクセスされるページは、アクセス頻度を追跡するための対象ではない可能性があると仮定することができる。ソフトウェアの場合、ＴＬＢエントリを割り振られた、より頻繁にアクセスされるページに関するアクセス頻度追跡情報を知ることで十分であり得る。したがって、メモリへのアクセス頻度追跡インジケータのハードウェア管理ライトバックの追加コストは、正当化されない可能性がある。

【００９８】

しかしながら、任意選択的に、いくつかの実施態様では、アクセス頻度追跡回路４０内のハードウェア回路が、メモリ内のバッキングデータ構造１８０へのアクセス頻度追跡インジケータのライトバックを管理してもよい。アクセス頻度追跡回路４０は、ソフトウェアがバッキングデータ構造１８０のベースアドレスを用いて構成することができるレジスタを有することができる。この場合、図１２の点線で示すように、追跡ハードウェア構造１５８からの所与のページのエントリ１５９に対してライトバックイベントが発生すると、アクセス頻度追跡回路４０は、メモリベースのバッキングデータ構造１８０内の所与のページの対応するアクセス頻度追跡インジケータ１８２を更新するために、１つ以上のメモリアクセス要求を発行させることができる。ライトバックイベントは、例えば、所与のページに対するエントリ１５９の追い出し又は無効化；そのエントリ１５９におけるアクセス頻度追跡インジケータ５２のオーバーフロー；ハードウェア構造１５８内のアクセス頻度追跡インジケータ５２の全て又はサブセットのライトバックをトリガするための定期的なフラッシュイベントの発生、及び／又はハードウェア構造１５８の全てのエントリ１５９について、又は命令によって指定されたフィルタ条件（例えば、エントリ１５９が指定されたアドレス又はアドレス範囲に対応する）を満たすエントリ１５９のみについて、アクセス頻度追跡インジケータ５２がメモリにライトバックされることを要求するための「フラッシュ」命令の実行のいずれかであり得る。

【００９９】

ライトバックは、異なる方法で実行することができる。いくつかの例では、アクセス頻度追跡インジケータ１８２は、ライトバックイベント時に、フラグ１８２の前の値が０である場合、それが１に更新されるが、フラグ１８２がライトバックイベント前に既に１であった場合、それは１のままであるように、単一ビットフラグであり得る。この場合、メモリベースの構造１８２は、対応するページへのアクセスの数が比較的大きい可能性がある（例えば、第１の結果を与える偶然依存テストの確率が１／Ｐである場合、Ｐ／２より大きい可能性が高い）ことを示すために、アクセス頻度追跡インジケータ５２のオーバーフローがそれぞれのページについて発生したかどうかを示す。

【０１００】

代替として、メモリベースの構造１８０内の追跡インジケータ１８２は、マルチビットカウンタであり得る。この場合、所与の追跡インジケータ５２のライトバック時に、そのカウンタの値をメモリ内の対応するインジケータ１８２に書き込むことができる。代替的に、ライトバックは、ライトバックカウンタ５２から導出された値をメモリ内のインジケータ１８２の以前の値に加算して、インジケータ１８２の位置に書き込まれる更新された値を生成することを含むことができる。これは、インジケータ１８２がそれらのソースのいずれかによって開始されたページへのアクセスの総数を追跡するように、メモリアクセスの複数のソースが同じバッキング構造１８０を共有することを可能にするのに有用であり得る。メモリ１８２内のインジケータ値に加算されるインクリメントは、カウンタの正確な値でなくてもよい。例えば、インクリメントは、カウンタ５２の値のスケーリングされたバージョンとすることができ、スケーリング係数は、第１の結果を提供する偶然依存テストの確率１／Ｐに依存して可変であり、例えば、スケーリング係数はＰであってもよい。これは、確率の異なる値を有する偶然依存テストを使用する複数のソースが、アクセスの推定数を追跡する共有インジケータ１８２をメモリ内に維持することを可能にし得る。また、図１３に関して以下で説明するように、アクセス数による確率１／Ｐの変動を使用する実施態様では、推定アクセス数の値は、アクセス頻度追跡カウンタ５２の値と非線形にスケーリングすることができ、したがって、この場合、アクセス頻度追跡カウンタ５２の各可能な値は、カウンタ５２のその値によって表される推定アクセス数を表す対応するインクリメント値に関連付けることができ、その結果、カウンタライトバック時に、メモリ内の対応するインジケータ１８２に加算するために、カウンタ５２の現在の値に基づいて適切なインクリメント値を選択することができる。

【０１０１】

図１３に示されるように、いくつかの実施態様では、偶然依存テストにおいて第１の結果が発生する確率は、所与のページへのアクセスの頻度が増加するにつれて時々低減され得る。確率の低減は、例えば、テスト評価カウンタをリセットするためのリセット値を増加させること、第１の結果が発生したかどうかを判定するためにテスト評価カウンタに対して比較される閾値を増加させること、又は第１の結果が発生したかどうかを評価するために使用されるテスト評価カウンタ、内部状態、パルス列、又は乱数／擬似乱数のサンプリングされた部分におけるビット数を増加させることによって達成され得る。

【０１０２】

例えば、ＴＬＢ１４又は他のアクセス頻度追跡ハードウェア構造１５８がそれぞれのページについてアクセス頻度追跡インジケータ５２を提供する図１２に示される手法では、所与のページについて第１の結果を提供する偶然依存テストの確率は、所与のページについてのアクセス頻度追跡インジケータ５２に依存し得る。例えば、図１３に示されるように、所与のページｘに対するアクセス頻度追跡カウンタ５２の現在の値が第１の閾値１６０に達するとき、確率は第１の値Ａから第２の値Ｂに低減されてもよく、次いで、そのページに対するアクセス頻度追跡カウンタがインクリメントし続けて第２の閾値１６２に達する場合、確率は第２の値Ｂから第３の値Ｃに再び低減され得る。

【０１０３】

例えば、テスト評価カウンタ６０がアクセス頻度追跡インジケータと比較されて、偶然依存テストを評価する実施態様では、第１の結果を与えるテストの確率の段階的な低減を実装する１つの方法は、所定の値（０等）と比較されるカウンタ６０のビット数を変化させることであり、比較に使用されるカウンタのビット数は、アクセス頻度追跡インジケータ５２の値が増加するにつれて増加する。例えば、具体例では、１６ビットのテスト評価カウンタ６０が設けられ、比較に使用されるビット数がアクセス頻度追跡インジケータ５２の値とともに変化して、以下のように異なる確率を与えることができる。

【０１０４】

【表１】

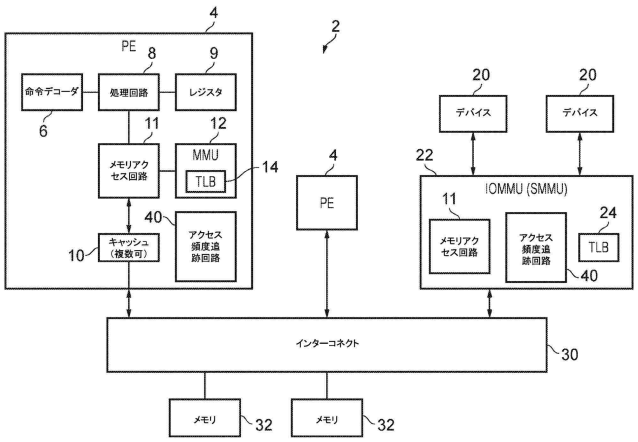
アクセス頻度追跡インジケータ52の値	カウンタから比較されるビット数	第1の結果を与える偶然依存テストの確率
0	8最下位ビット	1／256
1	9最下位ビット	1／512
2	10最下位ビット	1／1024
3	11最下位ビット	1／2056
4	12最下位ビット	1／4098
5	13最下位ビット	1／8192
6	14最下位ビット	1／16384
7	15最下位ビット	1／32768
8以上	全16ビット	1／65536

【０１０５】
これは１つの例にすぎず、他の実施態様は、確率を低減する異なる閾値を選択し得ることが理解されよう。
【０１０６】
一般に、確率を減少させることによって、カウンタ５２の更なるインクリメントを引き起こすために、所与のページへのより多くのアクセス数が平均して必要とされ、その結果、カウンタ値は、対応するページへの実際のアクセス数に対して非線形にスケーリングすることができる。これは、比較的少数のビットのカウンタが、数桁にわたるアクセスの頻度を追跡できるようにするために有用である可能性があり、その結果、カウンタがインクリメントされる可能性がより高いアクセスの閾値数は、均一に分散される必要がない。例えば、確率Ａ、Ｂ、Ｃ、及び確率が適切に切り替えられる閾値点１６０、１６２を設定することによって、ソフトウェアは、カウンタの第１の値が、比較的少数のアクセスが行われたことを示し、カウンタの第２のレベルが、アクセスの数が第１の数の少なくとも１０又は１００倍である可能性が高いことを示すことができ、カウンタの次の値が、カウンタ値自体が線形にインクリメントされる場合であっても、数桁多いアクセスがあった可能性があることを示すことができるように、カウンタを構成することができる。この場合も、これは、或る特定の範囲のアクセス数に関する追跡情報を決定するのに必要なカウンタのサイズを低減するのに役立つ。
【０１０７】
したがって、確率構成レジスタ４４は、以下のいずれかを選択することができる構成データを含むこともできる。
・アクセス頻度追跡カウンタ５２のカウント値の増加に伴って確率を漸減させるか、アクセス頻度追跡カウンタ５２の値にかかわらず一定の確率を維持するか；
・確率が低減されるべきカウンタ５２の値を定義する閾値点１６０、１６２；
・カウンタ値の異なる範囲に対して設定される確率レベルＡ、Ｂ、Ｃ。
確率が切り替えられるべき遷移点の数（図１３は、閾値１６０、１６２における２つの遷移を示すが、ソフトウェアは、異なる数を選択することが可能であり得る）。
【０１０８】
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、定義された動作を実施する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設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、機能を実行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。
【０１０９】
本発明の例示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

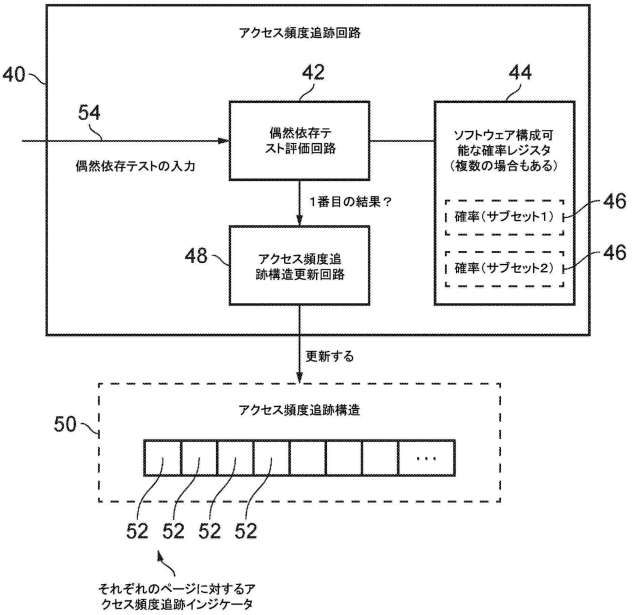
ことなく、実施形態に行われ得ることが理解されよう。

【図面】

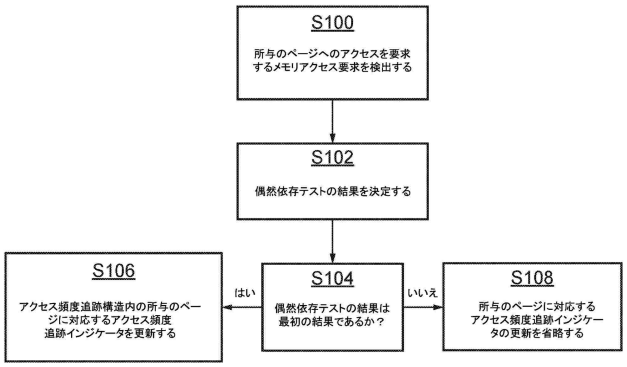
【図１】



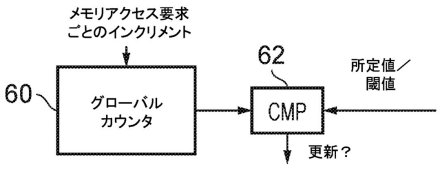
【図２】



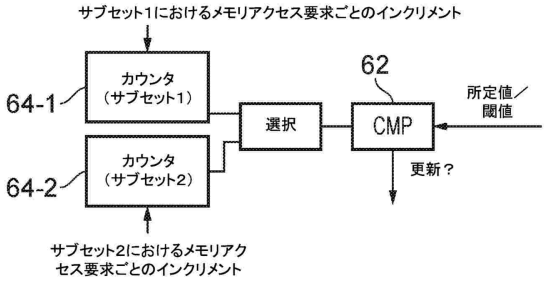
【図３】



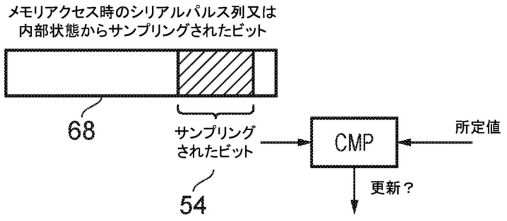
【図４】



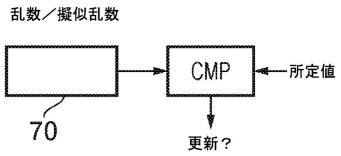
【図５】



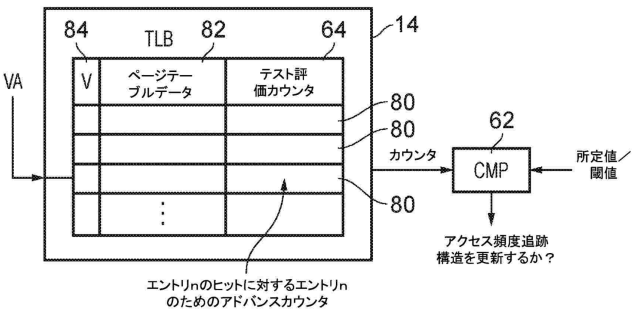
【図６】



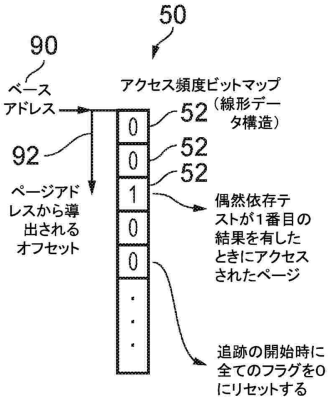
【図７】



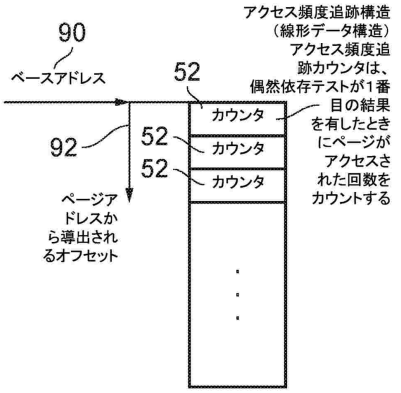
【図８】



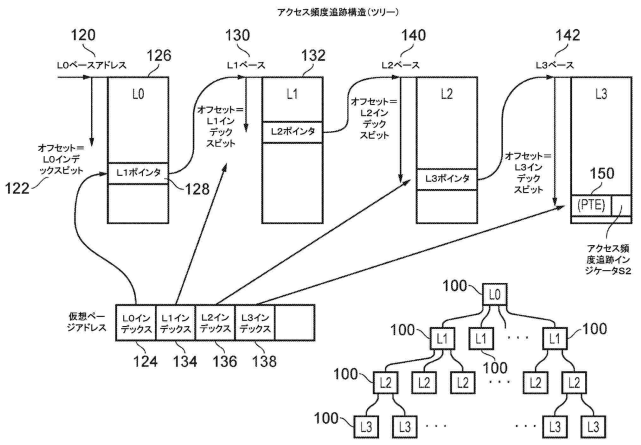
【図９】



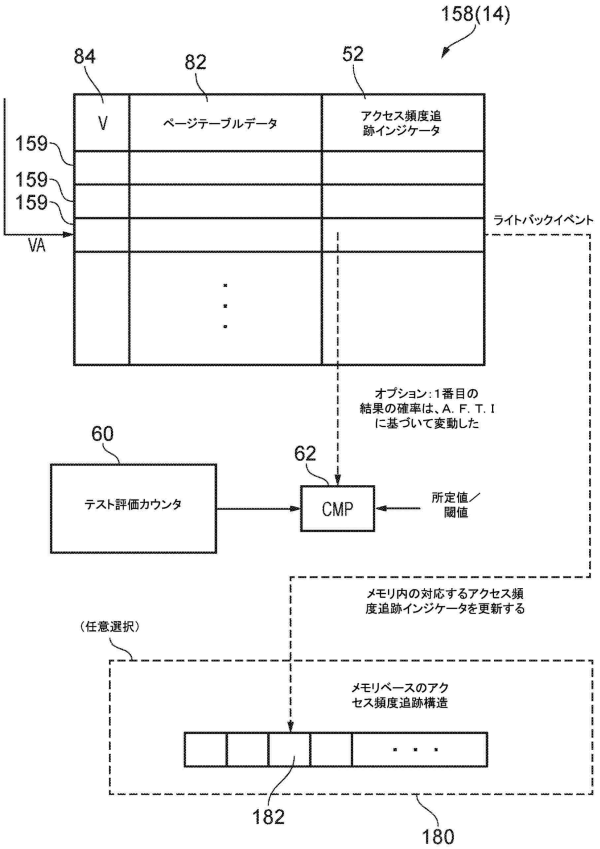
【図１０】



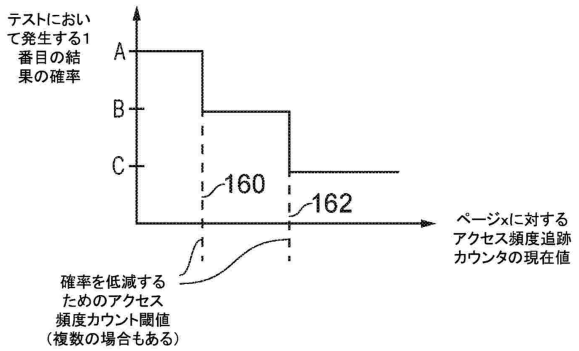
【図１１】



【図１２】



【図１３】



（例えば、テスト評価カウンタのリセット値／閾値を増加させることによって、又は内部状態、パルス列、又は乱数／擬似乱数のサンプリングされた部分のビット数を増加させることによって）

【国際調査報告】

INTERNATIONAL SEARCH REPORT

International application No
PCT/GB2022/051454

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06F12/1009
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

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	YU XIANGYAO ET AL: "Banshee: Bandwidth-Efficient DRAM Caching via Software/Hardware Cooperation", 2017 50TH ANNUAL IEEE/ACM INTERNATIONAL SYMPOSIUM ON MICROARCHITECTURE (MICRO), ACM, 14 October 2017 (2017-10-14), pages 1-14, XP033536660,	1-5, 15-25
A	paragraph [4.2.1] figure 1	6-14
A	US 2018/046378 A1 (COBURN JOEL DYLAN [US] ET AL) 15 February 2018 (2018-02-15) paragraphs [0034], [0039]	1-25

☐ 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
6 September 2022	14/09/2022

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	Braccini, Guido

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

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/GB2022/051454

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2018046378 A1	15-02-2018	US 2018046378 A1	15-02-2018
		US 2019073314 A1	07-03-2019

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

フロントページの続き

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,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,ME,MG,MK,MN,MW,MX,MY,MZ,NA,NG,NI,NO,NZ,OM,PA,PE,PG,PH,PL,PT,QA,RO,RS,RU,RW,SA,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

ー１　９エヌジェイ　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　チャドウィック、アレクサンダー　ドナルド　チャールズ

イギリス国　ケンブリッジシャー　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，アーム　リミテッド宛

Ｆターム（参考）　5B205　JJ13 MM51 RR05 VV03