

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第5752261号
(P5752261)

(45) 発行日 平成27年7月22日(2015. 7. 22)

(24) 登録日 平成27年5月29日(2015. 5. 29)

(51) Int.Cl.

F I

G06F 12/08 (2006.01)

G06F 9/30 (2006.01)

G06F 9/52 (2006.01)

G06F 12/08 519E

G06F 9/30 350A

G06F 9/46 472Z

G06F 12/08 513

請求項の数 15 (全 24 頁)

(21) 出願番号 特願2013-543316 (P2013-543316)
 (86) (22) 出願日 平成23年12月7日(2011. 12. 7)
 (65) 公表番号 特表2013-546088 (P2013-546088A)
 (43) 公表日 平成25年12月26日(2013. 12. 26)
 (86) 国際出願番号 PCT/US2011/063772
 (87) 国際公開番号 W02012/078775
 (87) 国際公開日 平成24年6月14日(2012. 6. 14)
 審査請求日 平成26年12月5日(2014. 12. 5)
 (31) 優先権主張番号 12/961, 819
 (32) 優先日 平成22年12月7日(2010. 12. 7)
 (33) 優先権主張国 米国 (US)

早期審査対象出願

(73) 特許権者 591016172
 アドバンスド・マイクロ・デバイス・
 インコーポレイテッド
 ADVANCED MICRO DEVI
 CES INCORPORATED
 アメリカ合衆国、94088-3453
 カリフォルニア州、サニペール、ピー・
 オウ・ボックス・3453、ワン・エイ・
 エム・ディ・プレイス、メイル・ストップ
 ・68 (番地なし)
 (74) 代理人 100108833
 弁理士 早川 裕司
 (74) 代理人 100111615
 弁理士 佐野 良太

最終頁に続く

(54) 【発明の名称】 記憶されたアトミック手順を使用するプログラム可能なアトミックメモリ

(57) 【特許請求の範囲】

【請求項 1】

マルチプロセッシングコアシステム内のプロセッシングコアを備える装置であって、前記プロセッシングコアは、

指定された命令シーケンスが1つ以上のアトミック性基準の組を満たすことを検証して、1つ以上のキャッシュラインから1つ以上のターゲットレジスタを介してデータにアクセスするように構成された、検証された命令シーケンスを生成することであって、前記1つ以上のアトミック性基準は、前記指定された命令シーケンスにおけるどの命令も、共有メモリから前記1つ以上のターゲットレジスタ以外を介してデータにアクセスするように前記プロセッシングコアに命令しないという基準を含む、ことと、

前記検証された命令シーケンスが、再度検証する必要なくアトミックメモリトランザクションとして実行可能となるように、前記検証された命令シーケンスを記憶することと、

アトミックメモリトランザクションとして前記マルチプロセッシングコアシステムによって実行されるための、記憶された前記検証された命令シーケンスを呼び出すことであって、前記検証された命令シーケンスをアトミックメモリトランザクションとして実行することは、データを、前記1つ以上のキャッシュラインから前記1つ以上のターゲットレジスタに記憶することを含む、ことと、

を行うように構成されている、装置。

【請求項 2】

前記プロセッシングコアは、前記検証された命令シーケンスをアトミックメモリトラン

ザクションとして実行するために、

前記命令シーケンスを実行して、前記 1 つ以上のターゲットレジスタに記憶された前記データを変更することと、

1 つ以上の値を、前記 1 つ以上のターゲットレジスタから前記 1 つ以上のキャッシュラインに記憶することと、を行うように構成されている、請求項 1 に記載の装置。

【請求項 3】

前記プロセッシングコアは、前記アトミックメモリトランザクションを実行するために、

メモリデータを前記 1 つ以上のキャッシュラインから前記 1 つ以上のターゲットレジスタに記憶する前に、前記 1 つ以上のキャッシュラインをロックすることと、

前記値を前記 1 つ以上のターゲットレジスタから前記 1 つ以上のキャッシュラインに記憶した後に、前記 1 つ以上のキャッシュラインのロックを解除することと、を行うように構成されており、

前記プロセッシングコアは、前記 1 つ以上のキャッシュラインをロックすることによって前記 1 つ以上のキャッシュラインがロックされている間、前記 1 つ以上のキャッシュラインに記憶された共有データを、1 つ以上の他のプロセッシングコアに読み取りまたは変更させないようにする、請求項 2 に記載の装置。

【請求項 4】

前記命令シーケンスがアトミックメモリトランザクションとして実行されることを示す特殊用途開始命令が、前記検証された命令シーケンスに先立ってなされ、前記特殊用途開始命令は前記 1 つ以上のキャッシュラインを示す、請求項 2 に記載の装置。

【請求項 5】

前記プロセッシングコアは、マイクロコード検証エージェントを使用して、前記検証することを行うように構成されている、請求項 1 に記載の装置。

【請求項 6】

前記マイクロコード検証エージェントは、

前記命令シーケンスの指標をユーザプログラムから受信し、

前記命令シーケンスの前記指標の受信に応じて、前記検証することと前記記憶することとを行う、ように構成されている、請求項 5 に記載の装置。

【請求項 7】

前記マイクロコード検証エージェントは、前記記憶された前記検証された命令シーケンスの識別子を、前記ユーザプログラムに戻すようにさらに構成されており、

前記アトミックメモリトランザクションを実行することは、前記ユーザプログラムが、前記識別子を使用して前記検証された命令シーケンスを呼び出すことに応じて行われる、請求項 6 に記載の装置。

【請求項 8】

前記マイクロコード検証エージェントは、前記命令シーケンスにおけるどの命令も、メモリ階層にアクセスするように前記プロセッシングコアに命令しないことを検証するようにさらに構成されている、請求項 6 に記載の装置。

【請求項 9】

前記プロセッシングコアは、順方向の進行を保証した前記アトミックメモリトランザクションを実行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 10】

前記 1 つ以上のアトミック性基準の組は、

前記命令シーケンス内のどの命令も、前記命令シーケンス外の命令において実行を継続するように前記プロセッシングコアに命令しないという基準と、

前記命令シーケンス内のどの命令も、汎用レジスタ以外の前記プロセッシングコアのレジスタにアクセスするように前記プロセッシングコアに命令しないという基準と、

前記命令シーケンスの各命令が事前に指定された制限長形式で表現されているという基準と、

10

20

30

40

50

前記命令シーケンスの命令の数が、所与の最大制限値を超えないという基準と、
のうち1つ以上の基準を含む、請求項1に記載の装置。

【請求項11】

マルチプロセッシングコアシステム内のプロセッシングコアが、1つ以上のアトミック性基準の組を命令シーケンスが満たすことを検証して、1つ以上のキャッシュラインから1つ以上のターゲットレジスタを介してデータにアクセスするように構成された、検証された命令シーケンスを生成するステップであって、前記1つ以上のアトミック性基準は、前記命令シーケンスにおけるどの命令も、共有メモリから前記1つ以上のターゲットレジスタ以外を介してデータにアクセスするように前記プロセッシングコアに命令しないという基準を含む、ステップと、

10

前記プロセッシングコアが、前記検証された命令シーケンスが、アトミック性について再度検証する必要なくアトミックメモリトランザクションとして実行可能となるように、前記検証された命令シーケンスを記憶するステップと、

前記記憶するステップの後に、前記検証された命令シーケンスをアトミックメモリトランザクションとして実行するステップであって、前記検証された命令シーケンスをアトミックメモリトランザクションとして実行するステップは、データを、前記1つ以上のキャッシュラインから前記1つ以上のターゲットレジスタに記憶することを含む、ステップと、を含む、方法。

【請求項12】

前記アトミックメモリトランザクションを実行するステップは、

20

前記命令シーケンスを実行して、前記1つ以上のターゲットレジスタに記憶された前記データを変更するステップと、

1つ以上の値を、前記1つ以上のターゲットレジスタから前記1つ以上のキャッシュラインに記憶するステップと、を含む、請求項11に記載の方法。

【請求項13】

前記アトミックメモリトランザクションを実行するステップは、

メモリデータを前記1つ以上のキャッシュラインから前記1つ以上のターゲットレジスタに記憶する前に、前記1つ以上のキャッシュラインをロックするステップと、

前記値を前記1つ以上のターゲットレジスタから前記1つ以上のキャッシュラインに記憶した後に、前記1つ以上のキャッシュラインのロックを解除するステップと、をさらに含み、

30

前記プロセッシングコアは、前記1つ以上のキャッシュラインをロックすることによって前記1つ以上のキャッシュラインがロックされている間、前記1つ以上のキャッシュラインに記憶された共有データを、1つ以上の他のプロセッシングコアに読み取りまたは変更させないようにする、請求項12に記載の方法。

【請求項14】

前記プロセッシングコアは、

前記命令シーケンスの指標をユーザプログラムから受信し、

前記命令シーケンスの前記指標の受信に応じて、前記検証するステップと前記記憶するステップとを行う、

40

ように構成されたマイクロコード検証エージェントを使用して、前記検証するステップを行うように構成されている、請求項11に記載の方法。

【請求項15】

前記1つ以上のアトミック性基準の組は、

前記命令シーケンス内のどの命令も、前記命令シーケンス外の命令において実行を継続するように前記プロセッシングコアに命令しないという基準と、

前記命令シーケンス内のどの命令も、汎用レジスタ以外の前記プロセッシングコアのレジスタにアクセスするように前記プロセッシングコアに命令しないという基準と、

前記命令シーケンスの各命令が事前に指定された制限長形式で表現されているという基準と、

50

前記命令シーケンスの命令の数が、所与の最大制限値を超えないという基準と、
のうち1つ以上の基準を含む、請求項11に記載の方法。

【発明の詳細な説明】

【背景技術】

【0001】

コンピュータアーキテクチャにおける現代の傾向では、単一のシステムおよび/または
プロセッサがメモリを共有し、それぞれが独立して同時実行できる複数のプロセッシング
コアを含む、マルチプロセッシングに向かう動きが見られる。現在、単一チップ上に2、
4若しくは8個のプロセッシングコアを有するチップマルチプロセッサ(CMP)、また
は多数のさらなるプロセッシングコアを有する汎用グラフィクスプロセッシングユニット
(GPGPU)が、比較的一般に見られる。加えて、各チップおよび/またはシステム上
のプロセッシングコアの数は、今後さらに増加する可能性がある。

10

【0002】

現代のプロセッサの高い並列プロセッシング能力を利用するために、ソフトウェアプロ
グラマは、様々な同期設備、例えば、命令セットアーキテクチャ(ISA)に支持される
アトミック命令を利用する。プロセッシングコアは、命令自体が複数のマイクロ命令を含
むが、システム内の他のプロセッシングコアに対して上記命令をアトミック(不可分)に
実行することができる。例えば、x86アーキテクチャにおけるアトミック命令CMPX
CHG(比較および交換)は、所与のメモリ位置のコンテンツを所与の値とアトミックに
比較し、2つの値が同一である場合にのみ、当該メモリ位置のコンテンツを所与の新しい
値に変更するように、プロセッシングコアに命令する汎用アトミック命令である。

20

【0003】

ISAは、制限された数の特殊用途のアトミック命令、例えばアトミックXADD、B
TSなどを提供する場合がある。プログラマの所望の特定機能に対する特殊用途の命令が
存在しない場合には、プログラマは、CMPXCHGなどの汎用命令を使用して、そのよ
うな論理の構成を試みる場合がある。しかしながら、そのような構成は複雑で実装困難で
あり、かつ実行速度が遅い可能性がある。

【発明の概要】

【0004】

マルチプロセッシングコアシステム内のプロセッシングコアは、命令シーケンスを、単
一のアトミックメモリトランザクションとして実行するように構成されている。プロセッ
シングコアは、アトミックシーケンスの命令が、共有メモリにアクセスするようにプロセ
ッシングコアに対して命令しないことを含む、1つ以上のアトミック性基準の組を、命令
シーケンスが満たすことを検証する。命令シーケンスを検証した後、プロセッシングコア
は、命令シーケンスを、単一のアトミックメモリトランザクションとして実行する。この
実行は、共有メモリデータを記憶するソースキャッシュラインをロックすることと、検証
された命令シーケンスを実行することと、アトミックシーケンスの結果をソースキャッシ
ュラインに記憶することと、ソースキャッシュラインのロックを解除することと、を含ん
でよい。いくつかの実施形態では、プロセッシングコアは、順方向の進行を保証したシー
ケンスを実行する。

30

40

【0005】

いくつかの実施形態では、マイクロコード検証エージェントが検証を行い、検証された
シーケンスのコピーを、保護されたメモリ領域に記憶してよい。このような実施形態では
、検証エージェントは、識別子をユーザコードに戻してよく、後にその識別子を使用して
アトミックシーケンスを起動することができる。

【図面の簡単な説明】

【0006】

【図1】いくつかの実施形態に従って、順方向の進行を保証したアトミック命令シーケ
ンスを検証および実行するように構成されたコンピュータシステムを示すブロック図である
。

50

【図 2】いくつかの実施形態に従って、命令シーケンスが、所与のアトミック性基準に適合するマイクロ命令のカスタムシーケンスをアトミックに実行するための方法を示すフロー図である。

【図 3】いくつかの実施形態に従って、命令実行パイプライン内のランタイム検証ユニットを使用して、プロセッシングコアが命令のアトミックシーケンスを検証および実行する方法を示すフロー図である。

【図 4】いくつかの実施形態に従って、実行前検証（例えば、マイクロコード化された検証エージェント）を使用して、プロセッシングコアが命令のアトミックシーケンスを検証および実行する方法を示すフロー図である。

【図 5】いくつかの実施形態に従って、マイクロコード化された検証エージェントを使用して、ランタイム前にアトミックシーケンスを検証および実行するためのさらに詳細な方法を示すフロー図である。

【図 6】いくつかの実施形態に従って、1つ以上のプロセッシングコアおよび/またはデバイスの代わりに、命令のアトミックシーケンスを実行するように構成された特殊用途のプロセッサを含むシステムを示すブロック図である。

【図 7】いくつかの実施形態に従って、アトミックプログラム実行ユニットを使用してアトミックシーケンスを実行するための方法を示すフロー図である。

【図 8】いくつかの実施形態に従って、本明細書に記載されるように、コード検証エージェントを使用して命令のアトミックシーケンスを単一のアトミックトランザクションとして実行するように構成されたコンピュータシステムを示すブロック図である。

【発明を実施するための形態】

【0007】

本明細書は、「一実施形態」に関する言及を含む。「一実施形態では」というフレーズの出現は、必ずしも同一の実施形態を指すとは限らない。特定の機能、構造または特徴は、この開示と一致する任意の適切な方法で組み合わせられてよい。

【0008】

用語について。以降の段落は、この開示（添付の請求項を含む）において見られる用語の定義および/または文脈を提供する。

【0009】

「備える」という用語について。この用語は非制限的である。添付の請求項において使用される場合、この用語は、追加の構造またはステップを除外しない。「1つ以上のプロセッシングユニットを備える装置」を列挙する請求項を考慮されたい。このような請求項は、装置が追加のコンポーネント（例えば、ネットワークインターフェースユニット、グラフィクス回路など）を含むことを除外しない。

【0010】

「～するように構成されている」との記載について。様々なユニット、回路または他のコンポーネントは、1つまたは複数のタスクを行う「ように構成されている」と記載または請求されてもよい。このような文脈において、「～するように構成されている」との表記は、ユニット/回路/コンポーネントが、動作中に1つ以上のタスクを行う構造（例えば、回路）を含むことを示すことによって、構造を暗示するように使用されている。そのようにして、前記ユニット/回路/コンポーネントは、特定のユニット/回路/コンポーネントが現在動作していない（例えば、オンでない）場合であっても、タスクを行うように構成されているとすることができる。「～するように構成されている」との表記とともに使用されるユニット/回路/コンポーネントは、ハードウェア、例えば、回路や、動作を実装するように実行可能なプログラム命令を記憶するメモリなどを含む。ユニット/回路/コンポーネントが1つ以上のタスクを行う「ように構成されている」という表記は、当該ユニット/回路/コンポーネントについて、米国特許法第112条6項を引き合いに出さないことが明示的に意図される。加えて、「～するように構成されている」との表記は、ソフトウェアおよび/またはファームウェア（例えば、FPGAまたはソフトウェアを実行する汎用プロセッサ）により操作され、問題のタスクを行うことができる方法で動

10

20

30

40

50

作する包括的構造（例えば、包括的回路）を含み得る。「～するように構成されている」との表記は、1つ以上のタスクを実装または実行するように適合されたデバイス（例えば、集積回路）を製造するように、製造プロセス（例えば、半導体製造設備）を適合することを含んでもよい。

【0011】

「第1の」、「第2の」などの用語について。本明細書で使用されるこれらの用語は、これらの用語に先行される名詞のラベルとして使用されるものであって、任意の種類の順序（例えば、空間的、時間的、論理的など）を暗示するものではない。例えば、8つのプロセッシング要素またはコアを有するプロセッサにおいて、「第1の」プロセッシング要素および「第2の」プロセッシング要素という用語は、8つのプロセッシング要素のうち任意の2つのプロセッシング要素を指すように使用され得る。つまり、「第1の」および「第2の」プロセッシング要素は、論理的なプロセッシング要素0および1に限定されない。

10

【0012】

「～に基づいて」という用語について。本明細書において使用されるこの用語は、決定に影響する1つ以上の要因を説明するように使用される。この用語は、決定に影響し得る追加の要因を除外しない。つまり、決定は、それらの要因のみに基づき得るし、あるいはそれらの要因の少なくとも一部に基づき得る。「Bに基づいてAを決定する」というフレーズを考慮されたい。BはAの決定に影響する要因であり得るが、このようなフレーズは、Aの決定がCにも基づくことを除外しない。他の例では、AはBのみに基づいて決定されてよい。

20

【0013】

汎用アトミック命令、例えば、CMPXCHGを使用して、単一のメモリ位置（例えば、単一のキャッシュライン）にアトミックにアクセスするための任意論理を構成することができる。本明細書において使用される「アトミックに」という用語は、その中間結果が、プログラム実行の他のスレッドにより観察可能でない実行を指す。したがって、アトミックに実行されるシーケンス（すなわち、アトミックシーケンス）内の命令は、すべての他のスレッドに対して、単一のアトミック動作として実行されたように見える。

【0014】

汎用アトミック命令を使用してアトミック論理を構成するアプローチにはいくつかの短所がある。第一に、これらの汎用命令を使用して様々な機能を実装することは、多くの場合、より複雑なロックフリーアルゴリズムまたはソフトウェアセマフォを必要とする。さらに、競合の存在下でCMPXCHG動作が中断され得るため、これらの動作を利用するアルゴリズムは、順方向の進行を保証することができず、実行の同時スレッド間の問題を解決するソフトウェア競合管理機構を必要とする。このことは、CMPXCHGを利用するユーザプログラムまたはゲストOSが、割り込みまたはページフォールトの可能性なしに、アルゴリズムを通して進行することが保証され得ないことから、性能の損失をもたらす。さらに、CMPXCHG動作は、それ自体が実行するのにいくらか不十分であり、1つの動作のターゲットキャッシュラインに対して少なくとも2つのアクセスを要する。

30

【0015】

プログラマは、制限された一連の固定機能アトミック動作では容易に行えない単一ラインのアトミックトランザクションを必要とする場合が多く、および/またはこれらのアトミック命令を使用して構築される場合には、性能の低下を被る。例えば、いくつかのマルチコアタスク管理技術は、同一のキャッシュライン内に存在し得る2つ以上のデータ値に対するアトミックトランザクションを必要とする。

40

【0016】

近年では、ハードウェアトランザクションメモリシステム（HTM）は、プログラマが、命令のカスタムシーケンスをアトミックに実行するのを可能にするための手段として提案されている。しかしながら、このようなシステムは、複雑なハードウェアサポートを必要とし、依然として、トランザクション試行の中断などのライブロック状況に供される。

50

したがって、このようなシステムは、依然として、競合の存在下で性能の低下を被ることから、デッドロックおよび/またはライブロックを回避するためのコンテンション管理機構を必要とする。

【0017】

本明細書で使用される「デッドロック」という用語は、所与のスレッドの実行が、別のスレッドとの競合に起因して無制限に行き詰まる状況を指す（例えば、2つのスレッドの各々が、他方のスレッドがロックを解除するのを同時に待っている等）。本明細書において使用される「ライブロック」という用語は、スレッドの実行が、別のスレッドとの競合に起因して、同一の命令シーケンスの実行を繰り返し試行し、失敗する（少なくとも1回）という状況を指す（例えば、H T Mシステムが、メモリ競合に起因して、トランザクシ

10

【0018】

様々な実施形態に従って、コンピュータシステムは、順方向の進行が保証された単一のアトミックトランザクションとしてシステムが実行可能なアトミックシーケンスとして、命令のカスタムシーケンスをプログラマが設計することを可能にする設備を提供するように、構成され得る。本明細書に使用される順方向の進行の保証とは、上記において定義されるように、アトミック命令シーケンスの実行がデッドロックまたはライブロックに供されないことを意味する。

20

【0019】

いくつかの実施形態では、システムは、アトミックとして指定されたシーケンスが特定のアトミック性基準に適合するか否かを検証するように構成された検証エージェントを含んでよい。このような実施形態では、アトミック性基準を満たすことは、順方向の進行を保証したシーケンスをアトミックに実行できることを保証する。以下でより詳細に論じられる様々な実施形態では、検証エージェントは、プロセッサ内、メモリサブシステム内またはシステムの他の場所に配置されてよい。

【0020】

図1は、いくつかの実施形態に従って、順方向の進行を保証したアトミック命令シーケンスを検証および実行するように構成されたコンピュータシステムを示すブロック図である。

30

【0021】

例示の実施形態に従って、システム190は、チップマルチプロセッサ(CMP)100を備えている。CMP100は、相互接続160を経由して、様々な他のCMP150および共有メモリ155に接続されている。いくつかの実施形態では、共有メモリ155は、例えば、L3キャッシュおよびメインメモリを備えるメモリ階層に配列されてよい。

【0022】

複数の相互接続されたCMPを有する実施形態を図1に示すが、本明細書に記載されるシステムおよび技術は、それらがシステム内でどのように分配されるかに関係なく、複数のプロセッシングコアを有する他のシステムに一般に適用されてよい。例えば、いくつかの実施形態では、CMP100および/または他のCMP150のいずれかは、通常の単一コアプロセッサと置き換えられてよい。他の実施形態では、CMP100は、複数のコアを備えてよく、システムは、他のCMP（例えば、150）、プロセッサまたはプロセッシングコアを備えていなくてもよい。

40

【0023】

例示の実施形態では、CMP100は、オンチップネットワーク145により接続されており、当該ネットワーク上の通信のために構成された複数のプロセッシングコア（すなわち、105、135）を備えている。オンチップネットワーク145は、異なるネットワークトポロジー（例えば、2D若しくは3Dグリッド、トーラス、線形、リングなど）に従って配列された、様々な型のポイントツーポイントネットワークのいずれかに対応し

50

てよい。他の実施形態では、コア 105 および 135 は、バスおよび / または別の型の相互接続を介して接続されてもよい。

【0024】

いくつかの実施形態では、オンチップ相互接続 (例えば、オンチップネットワーク 145) は、CMP (例えば、CMP 100 上のコア 105 および 135) 上のコアを 1 つ以上の共有データキャッシュ、例えば、データキャッシュ 140 (例えば、L2 キャッシュ) に接続し得る。いくつかの実施形態では、コアは、1 つ以上のプライベートデータキャッシュ、例えば、コア 105 のキャッシュ 115 を含んでもよく、このコアを使用して、共有メモリ 155 からキャッシュされたデータを記憶し得る。

【0025】

キャッシュされたデータのコヒーレンスを維持するために、各コアは、様々なキャッシュコヒーレンス機構、例えば、120 を含んでもよい。キャッシュコヒーレンス機構 120 は、キャッシュコヒーレンスメッセージ (例えば、MESI プロトコルメッセージ) を他のプロセッシングコアと交換し、および / または同期機能 (例えば、キャッシュ 115 内のキャッシュラインを、他のプロセッシングコアから独占的に所有し、拒否し、許容し、および / または回答するキャッシュコヒーレンスプローブとしてマークすること) を実行するための設備を含んでよい。

【0026】

例示の実施形態に従って、コア 105 は、例えば、命令若しくはスタックポインタ、中間結果または他のデータなどのデータを一時的に記憶するのに使用可能な様々なレジスタ 110 を含んでもよい。レジスタ 110 は、様々な特殊用途および / または汎用レジスタを含んでよい。

【0027】

例示の実施形態では、コア 105 は、例えばマイクロコード 132 などのプライベートデータを記憶するための、保護された高速メモリ領域を備えるプライベートメモリ領域 130 を含んでもよい。マイクロコード 132 は、異なるプログラム命令 (例えば、CMPXCHG) および / または任意のマイクロプログラムを実装するために、プロセッシングコア 105 により実行可能なマイクロ命令を含んでよい。いくつかの実施形態では、マイクロコード 132 は、以下に記載されるように、あるアトミック性基準を満たすことが検証エージェント (例えば、134) によって検証されたマイクロプログラムおよび / またはアトミック性マイクロプログラムを含んでよい。

【0028】

図 1 の特定の実施形態では、プライベートメモリ領域 130 は、マイクロコードを含み、マイクロコード化された検証エージェント 134 を実装する。検証エージェント 134 は、命令シーケンスの指標を受信し、命令シーケンスが所与のアトミック性基準を満たすアトミックシーケンスであるか否かを検証し、当該シーケンスのコピーをプライベートメモリ領域 130 に記憶し、および / または記憶したシーケンスのコピーを識別するハンドルを他のプログラムコードに返送するように構成されている。この方法は、図 3 の説明においてさらに詳述される。

【0029】

他の実施形態では、検証エージェント 134 は、必ずしもメモリ領域 130 内にマイクロコード化および記憶される必要はない。例えば、いくつかの実施形態では、検証エージェント 134 は、例えば命令デコード 165 などのように、コア 105 のハードウェアユニットに組み込まれているハードウェア論理によって実装されてよい。さらに他の実施形態では、検証エージェント 134 は、以下に説明されるように、例えば記憶サブシステム内の特殊用途プロセッサなどによって、コア 105 から完全に分離して実装されてよい。

【0030】

図 2 は、マイクロ命令のカスタムシーケンスをアトミックに実行するための方法を示すフロー図である。この命令シーケンスは、いくつかの実施形態に従って、所与のアトミック性基準に適合する。方法 200 は、順方向の進行を保証したアトミック命令シーケンス

10

20

30

40

50

を実行することの一部として、図 1 のプロセッシングコア 1 0 5 などのプロセッシングコアにより実行されてよい。

【 0 0 3 1 】

図 2 に示されるように、方法 2 0 0 は、3 つの段階、具体的にはプロローグ段階 2 1 0 (トランザクションで使用される任意のデータが読み取られ得る段階)、主要な実行段階 2 2 0 (命令が実行される段階) およびエピローグ段階 2 3 0 (任意の結果がメモリに書き戻される段階) を含む。プロローグ段階の間、プロセッシングコアは、2 1 2 にあるように、所与のソースキャッシュラインをロックする。本明細書に使用される、キャッシュラインをロックするプロセッシングコアは、他のプロセッシングコアが、キャッシュラインコンテンツを読み取らないように、またはロックを保持するコアによってロックが解除されるまでキャッシュラインコンテンツを修正することがないように、キャッシュラインに対する排他的読み取り / 書き込み権限を有する。この特性を保証するために必要な特定のステップは、実装間で異なり、用いられる特定のキャッシュコヒーレンス機構およびプロトコルなどの他のシステム詳細に依存する。例えば、いくつかの実施形態では、所与のキャッシュラインをロックするコアは、プローブまたは別のプロセッサから受信する他のキャッシュコヒーレンスメッセージを、当該メッセージがロックされたキャッシュラインに関するものである場合に再利用してよい。いくつかの実施形態では、コアは、アトミックシーケンスの実行後およびキャッシュラインのロック解除後までコアが受信する様々なハードウェア割り込みおよび / またはソフトウェアシグナルのプロセッシングを、遅延させる場合もある。そのようなステップを行うことにより、コアは、割り込みなしにキャッシュラインの内容を読み取り、および / または修正してよい。

【 0 0 3 2 】

2 1 2 にあるように、キャッシュラインをロックした後、プロセッサは、2 1 4 にあるように、ソースキャッシュラインからデータを読み取り得る。2 1 4 におけるソースキャッシュラインの読み取りは、シーケンスの 1 つ以上の命令による後次アクセスのための 1 つ以上のターゲットレジスタに、キャッシュラインからのデータを記憶することを含んでよい。場合によっては、プロセッサは、全体ではないが、ソースキャッシュラインをターゲットレジスタに読み込み得る。例えば、プロセッサは、ソースキャッシュラインのほんの一部分 (例えば、6 4 バイト) を単一のターゲットレジスタに読み込んでもよいし、ソースキャッシュラインの多数の (もしくはすべての) 部分をそれぞれのターゲットレジスタに読み込んでもよい。

【 0 0 3 3 】

図 2 に示されるように、プロローグ段階が完了すると、実行は主要な実行段階 2 2 0 に進行してよく、例えば 2 2 2 において、コアは命令のアトミックシーケンスを単一のアトミックトランザクションとして実行する。これらの様々な命令は、ターゲットレジスタからデータを読み取り、および / またはターゲットレジスタにデータを書き込み得る。

【 0 0 3 4 】

2 2 2 に示されるように、命令のアトミックシーケンスは、コアがシーケンスを単一のアトミックトランザクションとして実行できるようにする特定のアトミック性基準に適合する。いくつかの実施形態では、アトミック性基準は、シーケンスが迅速かつ単純に実行できることを保証するように選択されてよく、実行コアにより順方向の進行が保証され得る。そのような基準は、シーケンスにおけるどの命令もメモリにアクセスしない、またはアトミックシーケンスの外部のコード部分にジャンプしないという条件を含んでよい。いくつかの実施形態では、アトミック性基準は、シーケンスを実行することに関する命令カウント制限または時間制限などの全体シーケンスに関する条件を含んでもよい。アトミック性基準は、以下でさらに詳述される。

【 0 0 3 5 】

アトミックシーケンスにおけるすべての命令が実行されたとき、プロセスはエピローグ段階 2 3 0 に入る。エピローグ段階 2 3 0 では、プロセッシングコアは、実行の結果を、ロックされたキャッシュライン 2 3 2 に書き戻してよい。例えば、コアは、ターゲットレ

ジスタにおけるデータを、ソースキャッシュラインに書き込んでよい。これが行われると、実行コアは、234にあるように、ソースキャッシュラインを解放し得る。

【0036】

命令シーケンスを実行する前に、システムは、まずそのシーケンスが実際に、特定のアトミック性基準に適合するアトミックシーケンスであるかを決定し得る。異なる実施形態では、この決定は、異なる時間および/または異なるコンポーネントにより行われてよい。例えば、いくつかの実施形態では、シーケンスは、例えば命令実行パイプライン内（例えば、デコーダ165内）のハードウェアにより実装される検証エージェントによって、検証が行われてもよい。他の実施形態では、シーケンスは、実行される前に、例えば、命令シーケンスが実行される前に起動されるマイクロコード化された検証エージェントによって、統計的に検証されてよい。さらに他の実施形態では、シーケンスは、他のプロセッシングコアの代わりにアトミック命令シーケンスを実行するように構成された特殊用途のオフチッププロセッサによって、検証および/または実行されてよい。他の可能性も存在するが、これら3つの構成は、さらに以下で詳述される。

10

【0037】

図3は、いくつかの実施形態に従って、プロセッシングコアが、命令実行パイプライン内（例えば、デコーダ内）のランタイム検証ユニットを使用して、命令のアトミックシーケンスを検証および実行する方法を示すフロー図である。いくつかのそのような実施形態では、デコーダ（例えば、図1の165）は、デコード段階の間に、シーケンス内の各命令を検証するように構成された検証ユニットを含んでよい。このような検証エージェントは、シーケンス（および/またはシーケンス内の任意の命令）がアトミック性基準に適合しないことを発見すると、実行を停止し、例外を送出し、および/またはエラーを表示してソースキャッシュラインをロック解除し得る。

20

【0038】

図3は、そのような実施形態において検証エージェントが動作し得る一方法の例を示す。異なる実施形態では、図3の方法300は、デコーダによって、当該デコーダの一部として実装される検証ユニットによって、および/またはプロセッサ若しくはプロセッシングコアの別のハードウェアコンポーネントによって実行されてよい。

【0039】

305に示されるように、方法300は、アトミック実行のために指定されたプログラム命令のシーケンスの開始を検出するシステムに応じて、開始される。例えば、いくつかの実施形態では、シーケンスは、それぞれシーケンスの開始および終了を示す「START」および「STOP」プログラム命令を使用して、プログラムコードにおいて区分され得る。

30

【0040】

コアは、305において、START命令の読み取りに応じて、図2のプロローグに対応する1つ以上のステップを行ってよい。これらのステップは、ソースキャッシュラインを読み取ること（212を参照）と、キャッシュラインをロックすること（214を参照）とを含んでよい。

【0041】

40

いくつかの実施形態では、ソースキャッシュラインおよび/またはターゲットレジスタは、（例えば、1つ以上のフィールドまたはパラメータにおける）START命令によって指標されてよい。例えば、x86互換性アーキテクチャを利用する実施形態では、START命令は、START命令の1つのフィールド（例えば、modrm.r/mフィールド）におけるソースキャッシュラインのメモリアドレスと、別のフィールド（例えば、modrm.regフィールド）におけるターゲットレジスタの指標とを含んでよい。コアは、これらの値の読み取りに応じて、識別されたソースキャッシュラインをロックし（310を参照）、シーケンス内の1つ以上の命令による後次アクセスのために、その指標されたソースキャッシュラインからのデータを、識別されたターゲットレジスタにロードしてよい（315を参照）。

50

【 0 0 4 2 】

いくつかの実施形態では、S T A R T命令を実行することは、ソースキャッシュラインにおけるデータのメモリアドレスを記録すること（320を参照）および/またはアトミックシーケンス内の命令数（325を参照）を記録することをさらに含んでよい。いくつかの実施形態では、アトミック領域における命令の数は、例えばx o p . v v v vフィールドなどのS T A R T命令の別のフィールドに記憶された値によって指標され得る。

【 0 0 4 3 】

ステップ305～325を完了した後（実行のプロローグ段階に対応し得る）、コアは、ステップ330～345により例示されるように、一連のランザクションを実行し得る。この段階の間に、コアはコードシーケンスを通して進行し、シーケンスにおける次の命令がアトミック性基準を満たす場合（330：はい）、当該次の命令を実行する（340を参照）。次の命令がアトミック性基準を満たさない場合（330：いいえ）、システムは、335に見られるように、例外を送出し、シーケンスの実行を停止して、エラーハンドリングコードにジャンプし、および/またはシーケンスがアトミック性基準を満たさないことを示す。

【 0 0 4 4 】

いくつかの実施形態では、アトミック性基準は、迅速かつ単純に実行され、実行コアにより順方向の進行が保証され得る。例えば、いくつかの実施形態では、基準は、シーケンス内のどの命令もメモリにアクセスしないという基準を含んでよい。いくつかの実施形態では、基準は、シーケンス内のどの命令も、汎用レジスタ（G P R）ではないレジスタにアクセス（読み取りおよび/または書き込み）しないという基準を含んでもよい。これらの基準を満たす命令は、それ自体がデータを別の実行スレッドと対立させることができないため、そのような動作のみで構成される命令シーケンスは、デッドロックまたはライブロックに供されない場合がある。したがって、命令シーケンスは、実行されると順方向の進行を保証し得る。様々な実施形態では、アトミック性基準は、様々な他の条件または付加的条件（例えば、ジャンプ命令なし、即時形態の命令なしなど）を含んでよい。アトミックシーケンスが含み得る命令の種類に関連する基準に加えて、強化されたアトミック性基準は、シーケンス内の命令数に対する制限を含んでよい。

【 0 0 4 5 】

アトミック性基準を満たす命令は、共有メモリに直接アクセスすることはできないが、当該命令は、ターゲットレジスタを介してソースキャッシュライン内のデータに間接的にアクセスし得る。したがって、ソースキャッシュライン内のデータに正常にアクセスする命令は、代わりにそのデータをターゲットレジスタから読み取り、および/または他のデータをターゲットレジスタに書き込んでよい。コアがアトミックシーケンスの実行を完了すると（例えば、S T O P命令に遭遇すると）（345：いいえ）、コアは、350に見られるように、ターゲットレジスタに記憶された計算結果を、ソースキャッシュラインに書き戻し、355に見られるように、ソースキャッシュラインを解放する。したがって、コアは、アトミックシーケンス内の命令を単一のアトミック操作として効果的に実行することができる。

【 0 0 4 6 】

いくつかの実施形態では、命令がアトミック性基準を満たすか否かを決定する（ステップ330）デコーダは、アトミック性基準を満たすことが知られている命令を示すオペコードであるか否かを決定するために、命令のオペコードを審査することを含んでよい。例えば、いくつかの実施形態では、x 8 6システム内のデコーダは、命令のオペコードがA D D、A D C、A N D、X O R、O R、S B B、S U B、C M P、C M O V c c、B T、B T S、B T C、B T R、T E S TおよびX C H Gのセットから選択された任意の命令を示す場合に、当該命令がアトミック性基準を満たすことを決定し得る。この例では、列挙された命令は、何もメモリまたは非汎用レジスタにアクセスしないこと、および何もアトミックシーケンス外の地点までプログラム制御を移行するようにプロセッサに命令することができないこと（すなわち、ジャンプ命令なし）というアトミック性基準を満たす。

【 0 0 4 7 】

ステップ 3 3 0 ~ 3 4 5 は、命令の線形実行を示すが、当業者は、複数の命令が 3 4 0 において同時に実行し得る一方で、後次の命令が 3 3 0 においてデコードされる、および / またはアトミック性基準と比較されるように、様々なコアが命令パイプラインを利用し得ることを認識するであろう。加えて、条件の存在は、シーケンス内の異なる命令を複数回実行させ得る場合があるし、全く実行させない場合がある。

【 0 0 4 8 】

いくつかの実施形態では、3 5 5 においてソースキャッシュラインを解放することは、例えば、キャッシュラインに関する後次プローブを許容し、割り込みを再度有効にし、および / または他の実装に固有のステップによって、3 1 0 にて行われるロック手順を取り消すことを含んでよい。

10

【 0 0 4 9 】

いくつかの実施形態では、検証エージェントは、ランタイムではなく実行前に命令シーケンスを検証するように構成されてよい (図 3 を参照) 。例えば、そのような検証エージェントは、マイクロコード (例えば、マイクロコード 1 3 2) に実装されてよい。

【 0 0 5 0 】

図 4 は、いくつかの実施形態に従って、プロセッシングコアが実行前検証 (例えば、マイクロコード化された検証エージェント) を使用して、命令のアトミックシーケンスを検証および実行する方法を示すフロー図である。そのような実施形態では、プライベートメモリ領域 (例えば、図 1 の 1 3 0) は、命令シーケンスが、1 つ以上のアトミック性基準に適合するアトミックシーケンスであるか否かを検証するように構成された検証エージェント (例えば、1 3 4) を実装するプログラム命令を記憶してよい。

20

【 0 0 5 1 】

例示の実施形態に従って、ユーザコードが、プログラム命令のシーケンスを、検証エージェントを用いてアトミックシーケンスとして登録した場合、方法 4 0 0 は、4 0 5 において開始する。いくつかの実施形態では、ユーザコードは、特殊用途の命令 (例えば、A T O M I C D E F) を使用してシーケンスを登録してよい。そのような命令は、命令シーケンスを示すパラメータおよび / またはフィールドを含んでよい。

【 0 0 5 2 】

マイクロコード検証エージェントは、4 0 5 において命令シーケンスを登録する試みに応じて、4 1 0 に見られるように、命令シーケンスがアトミックシーケンスであることを検証しようとする。4 1 0 に示されるように、シーケンスがアトミック性であることを検証することは、シーケンス内の各命令を審査することと、シーケンスが 1 つ以上のアトミック性基準を満たすことを決定することと、を含んでよい。

30

【 0 0 5 3 】

マイクロコード検証エージェントは、ランタイム検証エージェントと同様に、シーケンス内の各命令が検証基準を満たすことをチェックしてよい。これらの基準は、ランタイム検証エージェントにより強化されたもの (例えば、シーケンス内のどの命令も G P R 以外のメモリまたはレジスタにアクセスし得ない) を含んでよい。加えて、マイクロコード検証エージェントは、各命令が迅速かつ容易にデコードされることを保証し得る (例えば、各命令は、4 バイトなどの制限された長さであり、即時形態の命令を含まない) 。マイクロコード検証エージェントは、ハードウェアベースのランタイム検証エージェントと同様に、命令が、アトミック性基準を満たす所与の許容可能な命令群の 1 つを示すことが知られるオペコードを含むことの検出に応じて、当該命令がアトミック性基準を満たすことを決定し得る。

40

【 0 0 5 4 】

シーケンスがアトミックシーケンスである (すなわち、アトミック性基準を満たす) 場合、検証エージェントは、4 1 5 に見られるように、保護されたプライベートのメモリ領域にアトミックシーケンスを記憶してよい。いくつかの実施形態では、コアは、アトミックシーケンスが記憶され得るプライベートメモリ領域 (例えば、1 3 0) を含んでよい。

50

プライベートメモリ領域 130 は、様々な保護領域（例えば、c6 保存領域または c6 保存領域のサブセット）を含んでよく、マイクロコード検証エージェントは、当該保護領域が別のプロセッサにより上書きされる可能性なしに命令シーケンスを記憶し得る。保護されたメモリ領域という用語は、マイクロコードにアクセス可能であるが、ソフトウェアまたは他のプロセッサにはアクセス可能ではないメモリ領域を指し得る。

【0055】

420 では、検証エージェントは、ハンドルを起動ユーザコードに戻す。ハンドルは、記憶されたアトミックシーケンスに固有の識別子を備えてよい。ユーザコードがアトミックシーケンスを実行する必要がある場合、425 に見られるように、このハンドルを使用して記憶されたアトミックシーケンスを起動してよい。例えば、システムは、以下に記載されるように、アトミックシーケンスを起動するための特別命令を認識してよい。

10

【0056】

プロセッシングコアは、識別ハンドルを使用して記憶されたシーケンスを起動するユーザコードに応じて（425 を参照）、430 に見られるように、検証前アトミックシーケンスを実行することは、ソースキャッシュラインをターゲットレジスタに読み込むこと、アトミックシーケンスを実行すること、ターゲットレジスタにおける結果をソースキャッシュラインにライトバックすること、およびソースキャッシュラインをロック解除することを含んでよい。シーケンスは既に検証されているため、430 においてアトミックシーケンスを実行することは、シーケンスを再度検証することを含まなくてよい。

20

【0057】

図5は、いくつかの実施形態に従って、マイクロコード化された検証エージェントを使用して、ランタイム前にアトミックシーケンスを検証および実行するためのより詳細な方法を示すフロー図である。図5に示される方法500は、x86アーキテクチャについて、図4の方法400の特定の実装に対応し得る。しかしながら、異なる実施形態では、特定の命令名、データ通信に使用される命令フィールド、使用される特定のレジスタなどを含む実装は、いくらか異なり得る。この開示は、そのような実施形態をすべて包含することが意図される。

【0058】

例示の実施形態に従って、ユーザコードは、命令シーケンスをマイクロコード検証エージェントによって登録するために、505に見られるように、例えば、ATOMICDEF命令を最初に実行する。例示の実施形態では、ATOMICDEF命令はmodrm.r/mフィールドを含み、modrm.r/mフィールドには、シーケンス内の第1の命令の相対命令ポインタアドレスが記憶されている。加えて、この実施形態におけるATOMICDEF命令は、シーケンス内の命令のカウント（すなわち、数）を記憶するxop.vvvvフィールドを含む。

30

【0059】

ATOMICDEF命令を実行する一部として（またはそれに応じて）、マイクロコード検証エージェントは、（例えば、modrm.r/mフィールド内の）ATOMICDEFにより識別された命令シーケンスがアトミック性基準を満たすという検証を試みる。上述されるように、この検証は、シーケンス内の命令のそれぞれが、アトミック性基準を満たす既知の許容可能な命令群の1つであることを決定することを含んでよい。いくつかの実施形態では、このアトミック性基準は、（a）シーケンス内のどの命令もメモリにアクセスし得ない、（b）シーケンス内のどの命令もGPRでないレジスタにアクセスし得ない、（c）各命令は、特定の制限された長さの形態を使用して特定される、（d）シーケンス内のどの命令も即時形態を使用しない（即時命令形態は、レジスタ識別子としてではなく、番号として特定されるパラメータを含む）という要件、および/またはコアがアトミックシーケンスを順方向の進行の保証した単一のアトミックトランザクションとして実行できるようにする他の要件を含んでよい。

40

【0060】

50

シーケンス内の1つ以上の命令がアトミック性基準を満たさない場合、検証は失敗する(515:いいえ)。いくつかの実施形態では、マイクロコード検証エージェントは、シーケンスの記憶されたバージョンにハンドルを戻すのではなく、520に見られるように、失敗の指標をユーザコードに戻すことによりアトミック性基準を満たすように、シーケンスの失敗をシグナル送信し得る。例えば、検証が失敗した場合、検証エージェントは、記憶されたアトミックシーケンスに対するハンドルよりもむしろ、起動するユーザコードに値0を戻し得る。

【0061】

検証が成功した場合(515:はい)、次にATOMICDEF命令により示されるシーケンスは、アトミック性基準を満たすアトミックシーケンスである。この場合、マイクロコードエージェントは、命令シーケンスを、保護されたメモリ領域、例えば、525におけるc6保存領域にコピーしてもよいし、ATOMICDEF命令が実行された後に、ソフトウェアがデッドロックの可能性を生じる命令をリライトしないことを保証する別の実装特定領域にコピーしてもよい。

【0062】

いくつかの実施形態では、エージェントは、命令のアトミックシーケンスのプロローグ段階を行うために実行可能な1つ以上の命令を用いて、命令シーケンスの保存されたバージョンを始めてよい。例えば、530において、エージェントは、START命令を用いて、保存された命令シーケンスを始める。上述されるように、START命令は、シーケンスの所与のアトミック実行のためのソースキャッシュラインおよびターゲットレジスタを示すために使用可能な個別のフィールドを含んでよい。そのような実施形態では、START命令は、ソースキャッシュラインをロックし、ソースキャッシュラインからターゲットレジスタまでデータをロードするように実行可能であり得る。

【0063】

535では、マイクロコード検証エージェントは、記憶されたアトミックシーケンスを一意的に識別するハンドルを(起動ユーザコードに)戻す。次に、ユーザコードは、固有のハンドルを使用して、前検証されたアトミックシーケンスを起動し、それにより当該アトミックシーケンスをアトミックに実行してよい。例えば、540では、ユーザコードは、ATOMICSTART命令を実行することにより記憶されたアトミックシーケンスを起動する。ATOMICSTART命令は、シーケンス用の固有のハンドルおよびメモリ位置/入力を特定するパラメータおよび/またはフィールドを含む(例えば、ソースキャッシュラインおよびターゲットレジスタ)。

【0064】

プロセッシングコアは、540の起動に応じて、545に見られるように、記憶されたアトミックシーケンスをアトミックに実行する。545においてシーケンスをアトミックに実行することは、図2の方法200と同様のプロローグ段階、主要な段階およびエピローグ段階を実行することを含んでよい。

【0065】

いくつかの実施形態では、上述される基本的なSTART機能性は拡張され、アトミックシーケンスが複数のキャッシュライン上で動作するのを許可し得る。例えば、START2命令は、アトミックシーケンスのプロローグ段階が第2のソースキャッシュラインをロックおよびロードするのを可能にするように定義されてよい。したがって、アトミック命令シーケンスは、複数のキャッシュラインからのデータ上で動作し得る。

【0066】

いくつかの実施形態では、(例えば、STARTおよびSTART2を使用して)複数のキャッシュラインにアクセスするアトミックシーケンスのプロローグは、デッドロック条件を回避する方法で実装され得る。例えば、いくつかの実施形態では、START2は、先行するSTART命令によりロックされたキャッシュラインの直後にあるキャッシュラインのみを読み取り、かつロックするように実行可能であり得る。これを行うことにより、システムは、2つのスレッドが同一の2つのキャッシュラインを反対順序でロックし

10

20

30

40

50

ようとし、それぞれ第1のキャッシュラインのロックに成功したが、他のスレッドが他のキャッシュラインを解放するのを無制限に待機するというデッドロック状況を回避する。

【0067】

いくつかの実施形態では、START2は、先行するSTART命令により特定されるソースキャッシュラインの直後のものであるという必要のない任意のソースキャッシュラインを特定するために使用され得る。しかしながら、デッドロックを回避するために、システムはロックに関する順序を強化し得る。例えば、STARTおよびSTART2を実行するコアが、それらのソースキャッシュラインのそれぞれの物理メモリアドレスにより決定される順序（例えば、昇順または降順）でソースキャッシュラインのそれぞれをロックすることを保証するために、マイクロコードを用いることができる。この順序をすべてのスレッドに課すことにより、システムは、複数のキャッシュラインをロックすることに関連するデッドロック状況を回避することができる。

10

【0068】

いくつかのアトミックシーケンスは、ジャンプ命令の実行を必要とし得る。しかしながら、ジャンプ命令は、プログラムフローをアトミックシーケンスの外側にジャンプさせる場合に問題となり得る。このことは、ランタイム検証エージェントが、アトミック性基準を満たさない任意の命令を、当該命令の実行前に検出するため、（例えば、デコーダにおける）ランタイム検証スキームを伴う実施形態に逆効果をもたらすとは必ずしも限らない。しかしながら、ランタイム前検証エージェント（例えば、マイクロコード化された検証エージェント）は、アトミックシーケンスを事前にチェックするため、ランタイムにおけるシーケンスの外側へのジャンプは、アトミック性基準を満たさない命令が実行されるのを許可し得る。

20

【0069】

いくつかの実施形態では、検証前アトミック命令シーケンスの外側へのジャンプを回避するために、アトミック性基準は、ジャンプ命令を許可しない既定を含んでよい。他の実施形態では、基準は、ジャンプ命令を許可し得るが、当該命令がアトミックシーケンスを超えて実行をジャンプさせないように保証するようチェックする。

【0070】

マイクロコード化された検証エージェントは、本明細書に記載されるように、異なるシステムにおいて、様々な実装に固有のアトミック性基準を許可してもよい。例えば、いくつかの実施形態では、マイクロコード化された検証エージェントは、命令ポインタ相対アドレス指定モードを利用する命令を却下し得る。保存されたアトミックシーケンスを起動することは、命令ポインタを変更し得ることから、IP相対パラメータを含む命令は、予想外のプログラム動作をもたらし得る。

30

【0071】

いくつかの実施形態では、アトミックシーケンス検証および/または実行は、コア以外の全体で行われてよい。例えば、いくつかの実施形態では、システムは、複数のプロセッシングコアおよび/または他のシステムデバイスにより使用するために構成された、別個のオフチップアトミックプログラム実行ユニットを含んでよい。オフチップユニットは、アトミックシーケンスを単一のアトミックメモリトランザクションとして実行することに応じて、アトミックシーケンスの指標を、プロセッシングコアおよび/または別のシステムデバイスから受信するように構成されてよい。そのような実施形態では、アトミック実行ユニットは、指標された命令シーケンスが、アトミック性基準の組の明示性または暗示性を満たすことを検証し得る。例えば、アトミック実行ユニットは、アトミック性基準の組を満たすシーケンスのみを解釈することができるか否かについて、シーケンスを暗示的に検証し得る。そのような実施形態は、以下でさらに詳述される。

40

【0072】

図6は、いくつかの実施形態に従って、1つ以上のプロセッシングコアおよび/またはデバイスの代わりに、命令のアトミックシーケンスを実行するように構成された特殊用途のプロセッサを含むシステムを示すブロック図である。例示の実施形態では、システム6

50

00は、プロセッシングコア610および630を含むチップマルチプロセッサ605を含む。各プロセッシングコアは、オンチップネットワーク640によって、互いに接続されており、かつ、共有キャッシュ635にも接続されている。各コア自体は、1つ以上のプライベートキャッシュ（例えば、615）およびレジスタ（例えば、620）を含む。図1に関して上述されるように、様々な実施形態では、システム600内の各プロセッシングコアは、1つ以上の別個のチップを含む、異なる配置で分配されてよい。例えば、システム600は、650などの1つ以上の他のCMPおよび/またはプロセッサを含んでよい。

【0073】

例示の実施形態に従って、相互接続645は、CMP605を、他のプロセッサ/CMP650および共有メモリサブシステム655に接続する。共有メモリサブシステム655は、共有データ（すなわち、システム内で複数のプロセッシングコアにより共有されるデータ）を記憶する、および/または当該共有データに対するアクセスを提供するための設備を含んでよい。例えば、例示の実施形態では、共有メモリサブシステム655は、システムのプロセッシングコアによる共有データへのアクセスを容易にするために、データおよびメモリコントローラ660を記憶するための1つ以上の共有キャッシュ665を含む。

【0074】

図6の例示の実施形態では、システム600は、アトミックプログラムを実行するための特殊用途プロセッサとして実装され得るアトミックプログラム実行ユニット670を含む。様々な実施形態では、アトミックプログラム実行ユニット670は、共有メモリサブシステム655の一部として実装されてよく、並びに/または共有メモリサブシステム内のメモリコントローラ（例えば、660）、共有キャッシュ（例えば、665）、並びに/若しくは他のコンポーネントおよび/若しくは機能性への高速アクセスを容易にするように、システムの異なる部分に配置されてよい。

【0075】

いくつかの実施形態では、プロセッシングコア（例えば、610）または周辺デバイス（例えば、680）は、特殊用途命令セット（ISA）を利用して、アトミックプログラムをエンコードしてもよく、実行のために当該アトミックプログラム（またはその指標）をアトミック実行ユニット（例えば、670）に送信してもよい。いくつかの実施形態では、ISAの表現力は、ISAを使用してエンコードされた任意の有効プログラムが必ずアトミック性基準の組を満たすように制限されてよい。アトミックプログラムまたはその指標の受信に応じて、アトミック実行ユニットは、単一のアトミックメモリトランザクションとしてアトミックプログラムを解釈および実行してよい。

【0076】

様々な実施形態では、アトミックプログラム実行ユニットは、1つ以上のプロセッシングコア、または例えばデバイス680などの周辺デバイスによって使用されるために構成されてよい。様々な実施形態では、デバイス680は、グラフィクスアダプター、オーディオアダプター、例えばディスクまたは光学ドライブなどのI/Oデバイス、および/または他のデバイスを含んでよい。

【0077】

図7は、いくつかの実施形態に従って、アトミックプログラム実行ユニット（例えば、670）を使用してアトミックシーケンスを実行するための方法を示すフロー図である。例示の実施形態では、方法700は、705に見られるように、特殊用途のアトミックプログラミング言語を用いた命令のアトミックシーケンスを特定するプロセッシングコアを用いて開始する。プロセッシングコアは、命令のアトミックシーケンスを、プロセッシングコアの単一キャッシュライン（すなわち、ソースキャッシュライン）におけるアトミックプログラムとして特定し得る。

【0078】

いくつかの実施形態では、特殊用途のプログラミング言語は、順方向の進行を保証した

10

20

30

40

50

シーケンスをアトミック実行ユニットが実行するのを許可する様々なアトミック性基準を、暗示的に強化するように設計され得る。例えば、いくつかの実施形態では、これらの基準は、アトミックシーケンス内のどの命令も、共有メモリにアクセスするようにおよび/またはアトミックシーケンス外の命令にジャンプするように、アトミック実行ユニットに命令しないという基準を含んでよい。アトミックプログラミング言語の表現性を制限して上記命令を除外することにより、システムは、ステップ700において構成された任意の有効プログラムがこれらのアトミック性基準を満たすことを保証することができる。他のアトミック性基準は、同様に強化されてよい（例えば、汎用レジスタ以外のレジスタにアクセスしないなど）。

【0079】

いくつかの実施形態では、言語は、キャッシュライン内の所定位置（インプレース）で実行され得るスタックベース言語であってよい。インプレース実行は、プログラムが特定されるメモリスペースだけを使用してプログラムを実行する技術を指す。例えば、単一キャッシュラインに記憶されたスタックベースのプログラムは、命令および/またはオペランドをスタックからポップし、それらの命令およびオペランドの結果をスタックに押し戻すことによってインプレースで実行され得、スタックが決して単一キャッシュラインより大きくならないようにする。以下の命令は、1つ以上のアトミック性基準を満たすアトミックプログラムをエンコードするための例示のスタックベースの言語を定義し、アトミック実行ユニットは、アトミックプログラムを単一のアトミックメモリトランザクションとして実行してよい。

LOAD ADDR0 : アドレス0から1ワード（例えば、8バイト）をロードし、結果をプッシュする

LOAD ADDR1 : アドレス1から1ワードをロードし、結果をプッシュする

STORE ADDR0 : スタックのトップをポップし、ワードをアドレス0に記憶する（記憶はENDまで保留）

STORE ADDR1 : スタックのトップをポップし、ワードをアドレス1に記憶する（記憶はENDまで保留）

ADD, SUB : 2つの値をポップし、これらの値を付加/控除して結果をプッシュする
 DUP : スタックのトップのコピーをプッシュする

DUP2 : トップ2つのスタック要素のコピーをプッシュする

SWAP : トップおよび2番目のスタック要素をスワップする

END : プログラムを終了し、記憶を書き出す

BTn : ポップ、ポップされた値のNビット値をプッシュする

PUSHimm : 即時#Nをプッシュする

CMPSIGNED : ポップされた符号付きの値の差異の符号に従って、2つの値をポップし、-1、0、1をプッシュする

CMPUNSIGNED : ポップされた符号なしの値の差異の符号に従って、2つの値をポップし、-1、0、1をプッシュする

DUP2CMP : ポップされた符号付きの値の差異の符号に従って、2つの値をポップし、ポップされた値をスタックにプッシュバックして、-1、0、1をプッシュする

JE, JG, JL : スタックをポップし、順方向のN命令を条件的にジャンプさせる（新しい命令アドレスが、キャッシュラインの外側にある場合はトランザクションを中断する）

J : 順方向のN命令を無条件にジャンプさせる（新しい命令アドレスがキャッシュラインにある場合はトランザクションを中断する）

MOVE, MOVG, MOVL : スタックから3つのアイテムをポップする。スタックのトップは、条件コードとして使用される。条件がTRUEである場合は、次にスタック上の2番目のアイテムをプッシュし、そうでなければ3番目のアイテムをプッシュする

END RETURN: ENDと同様であるが、スタックのトップに値を戻す

cc: 条件付き「TRUEの場合」

E: スタックのトップ(TOS)が0である場合、TRUE

L: スタックの最後が1である場合、TRUE

G: スタックのトップおよび最後の両方が0である場合、TRUE

【0080】

上記の例示的命令セットは、プログラムがエンコードされるキャッシュラインの外側にある共有メモリにアクセスする命令を含まない。さらに、命令セットは、プログラムの外側の位置に実行を移行させ得るジャンプ命令を含まない。したがって、例示の命令セットを使用してエンコードされた任意の有効プログラムは、共有メモリへのアクセスを禁止するアトミック性基準を少なくとも満たし、アトミックシーケンス外の位置にジャンプする。

10

【0081】

上記のスタックベースの言語は、アトミックプログラムを表現するための可能な命令セットの一例に過ぎない。他の変型例も可能であり、この開示の利益を考慮して、当業者に明らかとなり得る。例えば、いくつかの実施形態では、命令セットは、スタックベースの言語ではなく、レジスタ間言語を実装してよい。アトミック性および順方向の進行を保証するアトミック性基準の組を満たすアトミックシーケンスをエンコードするための他の言語およびスキームが可能である。

【0082】

20

705に見られるように、コアがアトミックプログラムを構成すると、コアは、710に見られるように、プログラムを実行するようにアトミック実行エージェントに要求を送信してよい。いくつかの実施形態では、コアは、エンコードされたアトミックプログラムを、実行ユニットに送信してよい。次いで、実行ユニットは、ローカルキャッシュのラインにプログラムを記憶してよい。他の実施形態では、コアは、実行ユニットが使用可能なアドレスまたは他の指標を送信して、アトミックプログラムを検索してよい。

【0083】

いくつかの実施形態では、プロセッシングコアは、メモリマッピング(例えば、MMIOを参照)またはポートマッピング(例えば、PMIOを参照)を経由して、アトミックプログラムをアトミック実行ユニットに通信してよい。例えば、プロセッシングコアは、メモリマッピングを使用して、メモリおよびアトミック実行ユニットの両方をアドレス指定するために同一のアドレスバスを使用してもよい。そのような実施形態では、コアのアドレス指定可能なメモリスペースの領域は、メモリマップされたアトミックユニットに対して一時的または永久的に保存されてよい。次に、コアは、マップされたメモリ領域内でアトミックプログラムを構成し、続いて、そのようなプログラムの実行準備ができた場合、アトミック実行ユニットに通知し得る。そのような指標に応じて、実行ユニットは、715に見られるように、プログラムを単一のアトミックメモリトランザクションとして読み取り、実行し得る。いくつかの実施形態では、メモリマップされた実行ユニットは、割り当てられたアドレス空間に対する変更について、プロセッサのアドレスバスを監視(モニタ)し、したがって、新しいアトミックプログラムが明示的な通知なしに実行準備ができた場合に検出するように構成され得る。

30

40

【0084】

例示の実施形態に示されるように、アトミックプログラムを受信した後、アトミックプログラム実行ユニットは、715に見られるように、プログラムを単一のアトミックトランザクションとして解釈および実行してよい。いくつかの実施形態では、例えばスタックベースのプログラムエンコードを使用するものなどのアトミックプログラムを実行することは、アトミックプログラムが記憶されるローカルキャッシュラインをロックすることと、プログラムをインプレースで実行することと、キャッシュラインをロック解除することと、を含んでよい。したがって、プログラム実行ユニットは、715においてエンコードされたプログラムをアトミックに実行し得る。この例では、アトミックプログラム実行ユ

50

ニットは、プログラムを、単一のロックされたキャッシュライン内ですべて実行するため、ユニットは、デッドロック状況を回避し得る。

【 0 0 8 5 】

様々な実施形態では、プロセッシングコア以外のデバイス（例えば、オーディオ/ビデオカード、他の周辺デバイスなど）は、アトミックプログラム実行ユニットを方法 7 0 0 と同様に利用してよい。例えば、そのようなデバイスは、アトミックプログラムを構成し（7 0 5 を参照）、そのプログラムの実行を要求する（7 1 0 を参照）。

【 0 0 8 6 】

いくつかの実施形態では、いくつかのコード（例えば、オペレーティングシステム）は、保護されたメモリ（7 0 5）において1つ以上のアトミックプログラムを構成および記憶した後、識別ハンドルを使用してアトミック実行ユニット（7 1 0）を起動させて、対応するプログラム（7 1 5）を実行してよい。したがって、オペレーティングシステムは、カーネルによる後の起動のために、1つ以上の信頼できるアトミックプログラムのセットを作成してよい。

【 0 0 8 7 】

図 8 は、本明細書に記載されるように、いくつかの実施形態に従って、コード検証エージェントを使用して、命令のアトミックシーケンスを単一のアトミックトランザクションとして実行するように構成されたコンピュータシステムを示すブロック図である。いくつかの実施形態では、コンピュータシステム 8 0 0 は、図 1 のコンピュータシステム 1 9 0 に対応し得る。

【 0 0 8 8 】

コンピュータシステム 8 0 0 は、パーソナルコンピュータシステム、デスクトップコンピュータ、ラップトップ若しくはノートブックコンピュータ、メインフレームコンピュータシステム、携帯型コンピュータ、ワークステーション、ネットワークコンピュータ、コンシューマーデバイス、アプリケーションサーバー、記憶デバイス、例えばスイッチ、モデム、ルーターなどの周辺デバイス、または一般に任意の種類のコンピューティングデバイスを含むが、これらに限定されない様々な種類のデバイスのいずれかに対応し得る。

【 0 0 8 9 】

コンピュータシステム 8 0 0 は、1つ以上のプロセッサ 8 6 0 を含んでよく、それらのいずれかは、複数の物理コアおよび/または論理コアを含んでよい。プロセッサ 8 6 0 は、例えば検証エージェント 8 7 0 などのように、カスタムアトミックシーケンスが、本明細書に記載されるように或るアトミック性基準を満たすことを検証するための機構を含んでよい。異なる実施形態では、検証エージェント 8 7 0 は、本明細書に記載されるように、ハードウェア実装されるランタイム検証エージェント（例えば、デコーダ内）、またはマイクロコードされたランタイム前検証エージェントに対応し得る。

【 0 0 9 0 】

コンピュータシステム 8 0 0 は、永続的にデータを記憶し得る1つ以上の永続記憶デバイス 8 5 0（例えば、光学記憶、磁気記憶、ハードドライブ、テープドライブ、固体メモリなど）を含んでもよい。

【 0 0 9 1 】

いくつかの実施形態では、システム 8 0 0 は、例えばアトミックプログラム実行ユニット 8 2 4 などの命令のアトミックシーケンスを実行するための特殊用途プロセッサを含んでよい。図 8 は、コア外のアトミックプログラム実行ユニット 8 2 4 および検証エージェント 8 7 0 の両方を示すが、他の実施形態は、これらのコンポーネントの1つのみ、またはその他を含んでよい。

【 0 0 9 2 】

例示の実施形態に従って、コンピュータシステム 8 0 0 は、1つ以上の共有メモリ 8 1 0（例えば、1つ以上のキャッシュ、SRAM、DRAM、RDRAM、EDO RAM、DDR10 RAM、SDRAM、Rambus RAM、EEPROMなど）を含んでよく、共有メモリ 8 1 0 は、複数のプロセッシングコアの間、例えば、プロセッサ 8 6 0

10

20

30

40

50

の1つ以上で共有され得る。1つ以上のプロセッサ860、記憶デバイス(複数可)850、アトミックプログラム実行ユニット824および共有メモリ810は、相互接続840を介して接続され得る。様々な実施形態では、システムは、図8に示されない、より少数または追加のコンポーネント(例えば、ビデオカード、オーディオカード、追加のネットワークインターフェース、周辺デバイス、ネットワークインターフェース、例えば、ATMインターフェース、イーサネット(登録商標)インターフェース、フレームリレーインターフェース、モニタ、キーボード、スピーカーなど)を含んでよい。加えて、図8に示される異なるコンポーネントは、追加のコンポーネントにさらに組み込まれ、または分離されてよい。

【0093】

いくつかの実施形態では、共有メモリ810は、プログラム命令820を記憶してよく、プログラム命令820は、プラットフォームネイティブバイナリ、例えばJava(登録商標)バイトコードなどの任意のインタープリタ型言語、若しくは例えば、C/C++、Java(登録商標)などの任意の他の言語、または任意のそれらの組み合わせでエンコードされてよい。プログラム命令820は、同期構造を必要とする1つ以上のマルチスレッドアプリケーション822を実装するプログラム命令を含んでよく、本明細書に記載されるように、アトミック命令シーケンスを含んでもよい。いくつかの実施形態では、プログラム命令820は、アプリケーション822(例えば、スケジューリング、ソフトウェアアシグナルハンドリングなど)を実行するためのソフトウェアサポートを提供する、オペレーティングシステム824を実装するために実行可能な命令を含んでもよい。

【0094】

例示の実施形態に従って、共有メモリ810は、共有データ830を含んでよく、複数のプロセッサ860および/またはそのプロセッシングコアによりアクセスされ得る。複数のプロセッサ860の一つは、本明細書に記載されるように、ローカルキャッシュ内の共有データ830の様々なコンポーネントをキャッシュし、キャッシュコヒーレンスプロトコルに従ってメッセージを交換することにより、これらのキャッシュ内のデータを調整する。

【0095】

プログラム命令820、例えば、マルチスレッドされたアプリケーション822および/またはオペレーティングシステム824を実装するために使用されるプログラム命令は、コンピュータ可読記憶媒体に記憶されてよい。コンピュータ可読記憶媒体は、マシン(例えば、コンピュータ)により読み取り可能な形態(例えば、ソフトウェア、プロセッシングアプリケーション)で情報を記憶するための任意の機構を含んでよい。コンピュータ可読記憶媒体は、これらに限定されないが、磁気記憶媒体(例えば、フロッピー(登録商標)ディスク)、光学記憶媒体(例えば、CD-ROM)、磁気光学記憶媒体、読み取り専用メモリ(ROM)、ランダムアクセスメモリ(RAM)、消去可能なプログラム可能メモリ(例えば、EPROMおよびEEPROM)、フラッシュメモリ、プログラム命令を記憶するために適した電子媒体または他の種類の媒体を含んでよい。

【0096】

上述されたコンピュータ可読記憶媒体は、いくつかの実施形態において、プログラムにより読み取られた命令を記憶するために、および1つ以上のプロセッサ860を備えるハードウェアを製造するために、直接的または間接的に使用されてよい。例えば、命令は、ハードウェア機能性の動作レベルまたはレジスタ転送レベル(RTL)の説明を例えばVerilogまたはVHDLなどの高レベルの設計言語(HDL)で記述する、1つ以上のデータ構造を説明し得る。この説明は、ネットリストを生成するための記述を合成し得る合成ツールにより読み取られてよい。ネットリストは、プロセッサ500の機能性を表す一連のゲート(例えば、合成ライブラリ内で定義される)を備えてよい。次に、ネットリストは、マスクに適用される幾何学形状を記述するデータセットを生成するように配置およびルーティングされ得る。次に、マスクは、プロセッサ100、605および/または860に対応する半導体回路(複数可)を製造するために、様々な半導体製造ステップ

10

20

30

40

50

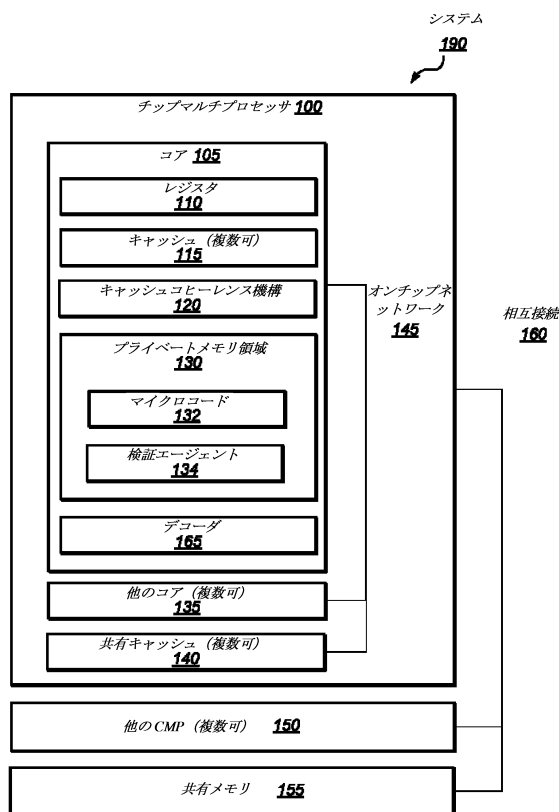
において使用され得る。あるいは、データベースは、必要に応じて、ネットリスト（合成ライブラリの有無にかかわらず）またはデータセットであってよい。

【 0 0 9 7 】

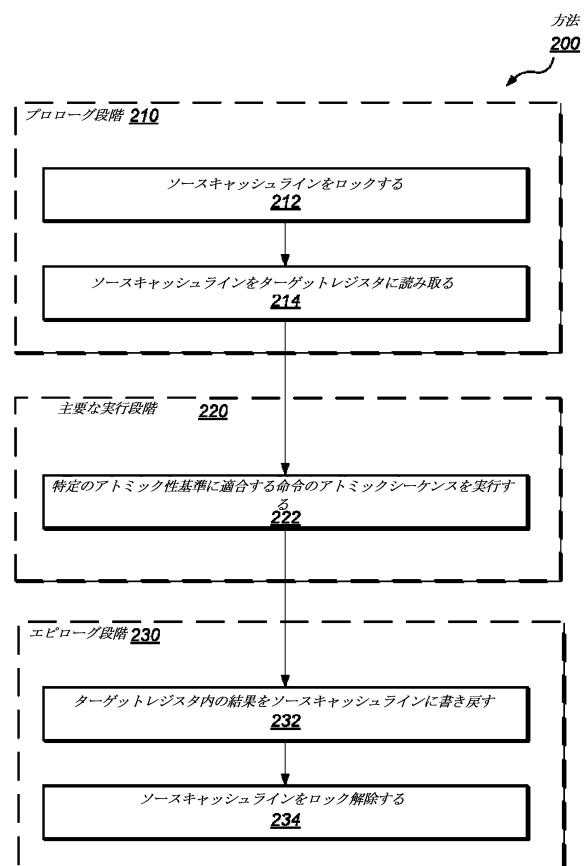
本開示の範囲は、本明細書において対処される問題のいずれかまたはすべてを軽減するか否かにかかわらず、本明細書に開示される任意の特徴もしくは特徴の組み合わせ（明示的または暗示的のいずれか）、またはその任意の一般化を含む。したがって、新しい請求項は、この出願（またはそれに対する優先権を主張する出願）を実行する間に、そのような特徴の任意の組み合わせに対して定式化されてよい。特に、添付の請求項を参照して、従属請求項からの特徴は、独立請求項の特徴と組み合わせられてよく、それぞれの独立請求項からの特徴は、単に添付の請求項において列挙される特定の組み合わせだけでなく、任意の適切な方法で組み合わせられてよい。

10

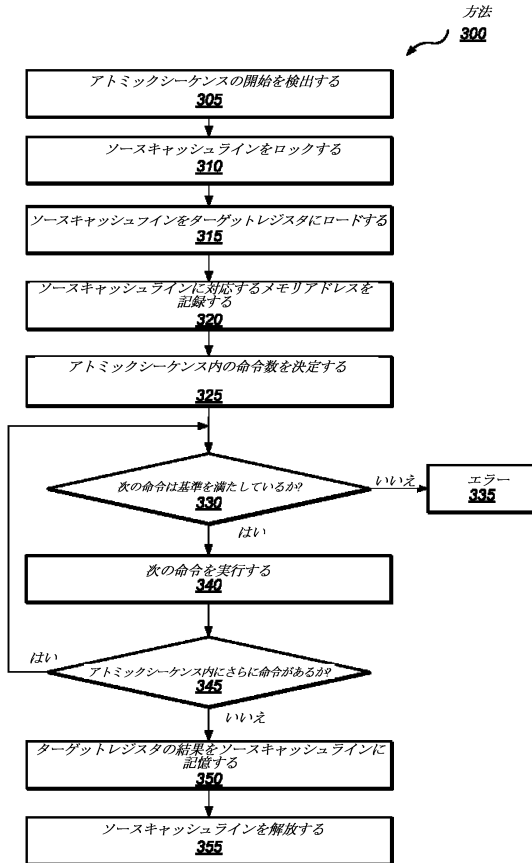
【 図 1 】



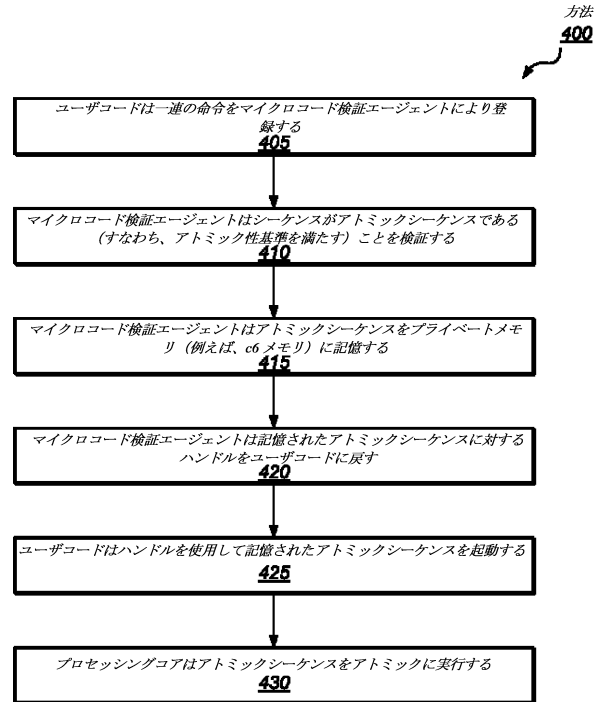
【 図 2 】



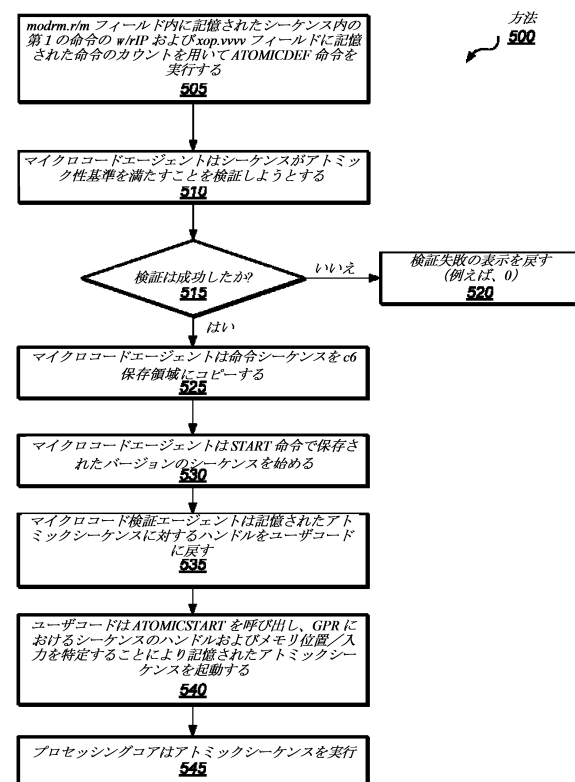
【図 3】



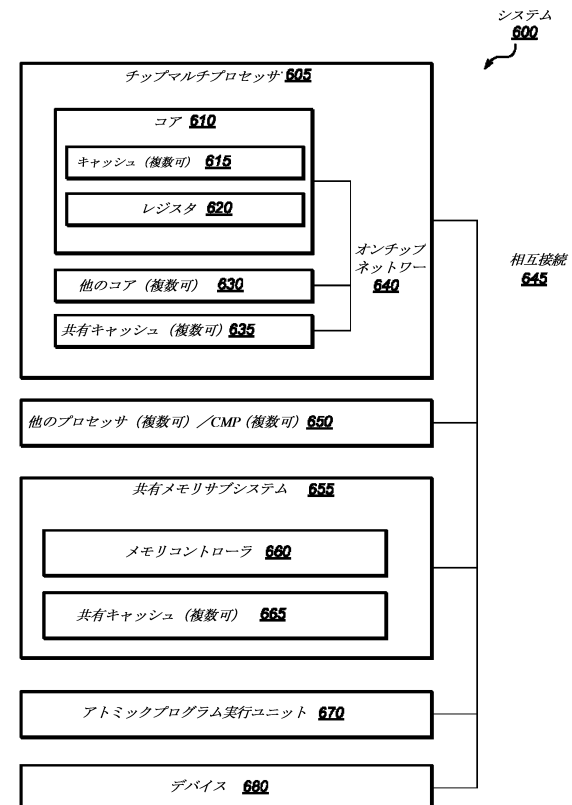
【図 4】



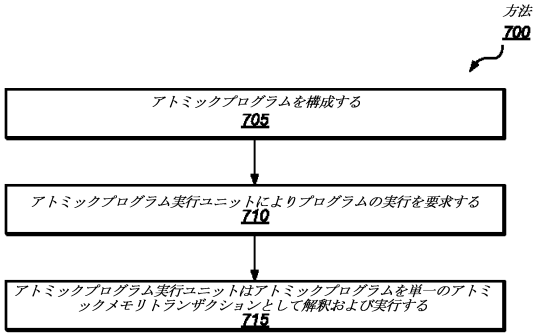
【図 5】



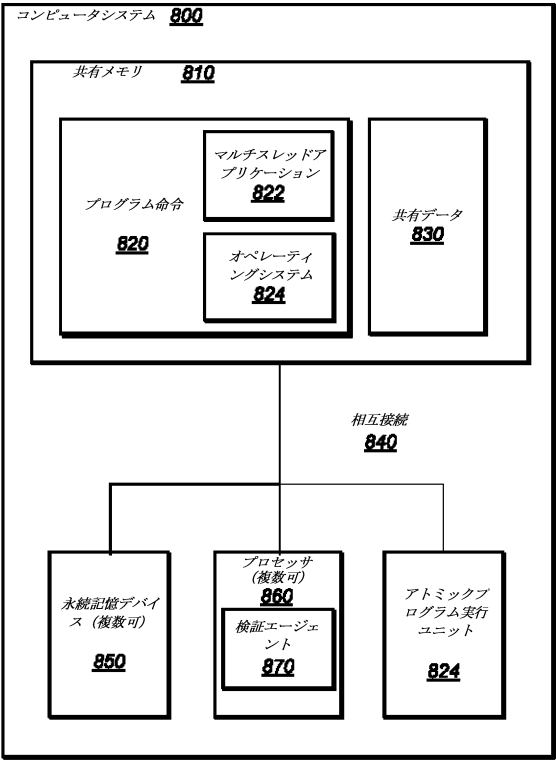
【図 6】



【図 7】



【図 8】



フロントページの続き

(74)代理人 100162156

弁理士 村雨 圭介

(72)発明者 ベンジャミン シー・セレブリン

アメリカ合衆国 94086 カリフォルニア州、サニーベール、マホガニー レーン 771

(72)発明者 デイビッド エー・カプラン

アメリカ合衆国 78749 テキサス州、オースティン、モンテレー オークス ブールバード
4600 アpartment 2137

(72)発明者 アントン シェルノフ

アメリカ合衆国 01460 マサチューセッツ州、リトルトン ピー・オー・ボックス 152
1

審査官 中野 裕二

(56)参考文献 国際公開第2010/014200(WO, A1)

国際公開第2011/106333(WO, A1)

国際公開第2008/155827(WO, A1)

特開平09-146838(JP, A)

米国特許出願公開第2007/0050560(US, A1)

米国特許出願公開第2003/0037223(US, A1)

(58)調査した分野(Int.Cl., DB名)

G06F 12/08 - 12/12

G06F 9/30

G06F 9/52