

(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7674475号
(P7674475)

(45)発行日 令和7年5月9日(2025.5.9)

(24)登録日 令和7年4月28日(2025.4.28)

(51)国際特許分類

F I

H O 4 L 9/32 (2006.01) H O 4 L 9/32 2 0 0 Z

請求項の数 15 (全26頁)

(21)出願番号	特願2023-523289(P2023-523289)	(73)特許権者	591025439
(86)(22)出願日	令和3年6月22日(2021.6.22)		ザイリンクス インコーポレイテッド
(65)公表番号	特表2023-547817(P2023-547817 A)		X I L I N X I N C O R P O R A T E D
(43)公表日	令和5年11月14日(2023.11.14)		アメリカ合衆国 カリフォルニア州 9 5
(86)国際出願番号	PCT/US2021/038528		1 2 4 - 3 4 0 0 サン ホセ ロジック
(87)国際公開番号	WO2022/093335	(74)代理人	ドライブ 2 1 0 0
(87)国際公開日	令和4年5月5日(2022.5.5)		110001195
審査請求日	令和6年6月3日(2024.6.3)		弁理士法人深見特許事務所
(31)優先権主張番号	17/083,195	(72)発明者	ジャバイド, ハリス
(32)優先日	令和2年10月28日(2020.10.28)		アメリカ合衆国、9 5 1 2 4 カリフォ
(33)優先権主張国・地域又は機関	米国(US)		ルニア州、サン・ノゼ、ロジック・ドラ
(31)優先権主張番号	17/084,942	(72)発明者	イブ, 2 1 0 0
(32)優先日	令和2年10月30日(2020.10.30)		ヤン, ジー
	最終頁に続く		アメリカ合衆国、9 5 1 2 4 カリフォ
			ルニア州、サン・ノゼ、ロジック・ドラ
			最終頁に続く

(54)【発明の名称】 ブロックチェーンマシンネットワークアクセラレーションエンジン

(57)【特許請求の範囲】

【請求項1】

コンピューティングシステムであって、
プロセッサと、
ブロックチェーンの台帳を記憶するメモリと、
ハードウェアアクセラレータであって、
前記台帳にコミットされるトランザクションのブロックに対応する複数のパケットを受信することと、
トランザクションの前記ブロック内の異なるコンポーネントのハッシュを生成することと、

前記ハッシュが、以前に計算されたハッシュと一致すると判定すると、前記ハードウェアアクセラレータにおいてトランザクションの前記ブロックを検証するためのタスクを生成することと、を行うように構成された、ハードウェアアクセラレータと、を備え、

前記プロセッサ又は前記ハードウェアアクセラレータのうちの1つが、トランザクションの前記ブロックが有効であると判定すると、トランザクションの前記ブロックを前記台帳にコミットするように構成されている、コンピューティングシステム。

【請求項2】

トランザクションの前記ブロックが、ブロックヘッダ、複数のトランザクション、及びメタデータを含み、前記ブロックヘッダ、前記複数のトランザクションの各々、及び前記メタデータが、前記複数のパケットのうちの別個のパケットで送信される、請求項1に記

載のコンピューティングシステム。

【請求項 3】

前記複数のパケットが、前記ブロック又は前記複数のトランザクションに対応する署名された証明書を含まない、請求項 2 に記載のコンピューティングシステム。

【請求項 4】

前記ハードウェアアクセラレータが、

前記ブロックチェーン内の認められたノードに対応する前記署名された証明書のローカルコピーを記憶するアイデンティティキャッシュと、

前記複数のパケットに含まれる ID を使用して、前記アイデンティティキャッシュから、トランザクションの前記ブロックに対応する署名された証明書を取り出すように構成されたデータ挿入器と、を含む、請求項 3 に記載のコンピューティングシステム。

10

【請求項 5】

前記ハードウェアアクセラレータが、

前記複数のパケット内のデータ、及び取り出された前記署名された証明書を使用して、トランザクションの前記ブロックを再構築するように構成されたデータ抽出器と、

トランザクションの再構築された前記ブロックを使用して、トランザクションの前記ブロックに対応するハッシュを生成するように構成されたハッシュ計算器と、

トランザクションの前記ブロックがシンタックスエラーを含まないことを確実にするために、生成された前記ハッシュが、以前に計算されたハッシュと一致するかどうかをチェックするように構成されたハッシュチェッカと、を含む、請求項 4 に記載のコンピューティングシステム。

20

【請求項 6】

コンピューティングシステムであって、

プロセッサと、

ブロックチェーンの台帳を記憶するメモリと、

ハードウェアアクセラレータであって、

前記台帳にコミットされるトランザクションのブロックを受信することと、

前記ブロックの署名を確認することと、

前記ブロック内の前記トランザクションの各々を検証することと、

前記トランザクションの検証結果を記憶することと、を行うように構成された、ハードウェアアクセラレータと、を備え、

30

前記プロセッサ又は前記ハードウェアアクセラレータのうちの 1 つが、前記トランザクションを前記台帳にコミットするように構成されている、コンピューティングシステム。

【請求項 7】

前記ハードウェアアクセラレータが、

前記ブロックチェーン内の別のノードからトランザクションの前記ブロックを受信するように構成されたネットワークインタフェースであって、トランザクションの前記ブロックが、複数のパケットに含まれる、ネットワークインタフェースと、

前記複数のパケットを解析して、前記トランザクションに関するデータを生成するように構成されたプロトコルプロセッサと、

40

前記プロトコルプロセッサから前記データを受信すること、前記ブロックの前記署名を確認すること、及び前記ブロック内の前記トランザクションの各々を検証することを行うように構成されたブロックプロセッサと、を含む、請求項 6 に記載のコンピューティングシステム。

【請求項 8】

前記ブロックプロセッサが、

前記ブロックの前記署名が、前記ブロックチェーン内のオーダーの既知の署名と一致することを確認するように構成された第 1 の署名アルゴリズムエンジンを含むブロック確認と、

前記ブロック内の前記トランザクションの各々を検証するように構成されたブロック検

50

証と、を含む、請求項 7 に記載のコンピューティングシステム。

【請求項 9】

前記ブロック検証が、

前記トランザクションに対応する署名が、前記ブロックチェーンを使用することを許可されたクライアントの既知の署名と一致することを確認するように構成された第 2 の署名アルゴリズムエンジンを含むトランザクション確認モジュールと、

トランザクション検証システムチェーンコード (V S C C) ブロックであって、

前記トランザクションにおける承認が、前記ブロックチェーン内の既知の承認ノードによって署名されたことを確認するように構成された第 3 の署名アルゴリズムエンジンと、

前記承認が、承認ポリシーを満たすことを確実にするように構成された承認ポリシー評価器と、を含む、トランザクション V S C C ブロックと、

状態データベース内の前記トランザクションに関連付けられたキー値ペアを読み取りかつ書き込むように構成されたトランザクションマルチバージョン同時実行制御 (M V C C) 書き込みブロックと、を含む、請求項 8 に記載のコンピューティングシステム。

【請求項 10】

前記ハードウェアアクセラレータが、システムオンチップ (S o C)、フィールドプログラマブルゲートアレイ (F P G A)、又は特定用途向け集積回路 (A S I C) のうちの少なくとも 1 つを含む、請求項 1 又は 6 に記載のコンピューティングシステム。

【請求項 11】

方法であって、

ハードウェアアクセラレータにおいて、ブロックチェーンの台帳にコミットされるトランザクションのブロックに対応する複数のパケットを受信することと、

前記ハードウェアアクセラレータにおいて、トランザクションの前記ブロック内の異なるコンポーネントのハッシュを計算することと、

前記ハードウェアアクセラレータにおいて、前記ハッシュが、以前に計算されたハッシュと一致すると判定することと、

前記ハードウェアアクセラレータにおいて、トランザクションの前記ブロックを検証するためのタスクを生成することと、を含む、方法。

【請求項 12】

トランザクションの前記ブロックが、ブロックヘッダ、複数のトランザクション、及びメタデータを含み、前記ブロックヘッダ、前記複数のトランザクションの各々、及び前記メタデータが、前記複数のパケットのうちの別個のパケットで送信される、請求項 11 に記載の方法。

【請求項 13】

前記複数のパケットが、前記ブロック又は前記複数のトランザクションに対応する署名された証明書を含まない、請求項 12 に記載の方法。

【請求項 14】

前記ハードウェアアクセラレータのアイデンティティキャッシュに、前記ブロックチェーン内の認められたノードに対応する前記署名された証明書のローカルコピーを記憶することと、

前記複数のパケットに含まれる ID を使用して、前記アイデンティティキャッシュから、トランザクションの前記ブロックに対応する署名された証明書を取り出すことと、を更に含む、請求項 13 に記載の方法。

【請求項 15】

前記複数のパケット内のデータ及び取り出された前記署名された証明書を使用して、トランザクションの前記ブロックを再構築することを更に含み、

前記ハッシュが、トランザクションの再構築された前記ブロックを使用して計算され、

前記ハッシュが、前記以前に計算されたハッシュと一致すると前記判定することが、トランザクションの前記ブロックがシンタックスエラーを含むかどうかを示す、請求項 14 に記載の方法。

【発明の詳細な説明】

【技術分野】

【0001】

本開示の実施例は、概して、ブロックチェーン内のノードのためのハードウェアアクセラレータに関する。

【背景技術】

【0002】

Hyperledger Fabricは、許可されたブロックチェーンのためのオープンソースの企業グレードの実装プラットフォームである。Hyperledger Fabricにおけるトランザクションフローは、実行ー順序付けー検証モデルに従い、トランザクションが最初に実行され、次いでブロックに順序付けられ、最後に検証され、（これまでコミットされたブロックのグローバル状態を保持するための状態データベースとともに）台帳にコミットされる。その結果、ファブリックネットワークは、ピア、オーダーラ、クライアントなどの異なるタイプのノードを含み、各ノードは、メンバーシップサービスプロバイダ（Membership Service Provider、MSP）によって提供されるアイデンティティを有する。

【0003】

許可されたブロックチェーン（Hyperledger Fabric、Quorum、Cordaなどのような）は、の一部となるためにアクセスを必要とするブロックチェーンネットワークである。これらのブロックチェーンは、トランザクションが、ブロックチェーンの台帳に追加される前に検証されることを必要とする。しかしながら、検証プロセスは、複数のトランザクションを検証しなければならないときにボトルネックをしばしば経験する特定のノードによって実行されなければならない。このボトルネックは、ブロックチェーンが新しいトランザクションを迅速にコミットする能力を制限することができる。

【発明の概要】

【0004】

一実施形態は、プロセッサと、ブロックチェーンのレジを記憶するメモリと、ハードウェアアクセラレータと、を含むコンピューティングシステムを説明する。したがって、ハードウェアアクセラレータは、台帳にコミットされるトランザクションのブロックに対応する複数のパケットを受信することと、トランザクションのブロック内の異なるコンポーネントのハッシュを生成することと、ハッシュが以前に計算されたハッシュと一致すると判定すると、ハードウェアアクセラレータにおいてトランザクションのブロックを検証するためのタスクを生成することと、を行うように構成されている。更に、プロセッサ又はハードウェアアクセラレータのうちの1つは、トランザクションのブロックが有効であると判定すると、トランザクションのブロックを台帳にコミットするように構成されている。

【0005】

本明細書で説明される別の実施形態は、プロセッサと、ブロックチェーンのレジを記憶するメモリと、台帳にコミットされるトランザクションのブロックを受信することと、ブロックの署名を確認することと、ブロック内のトランザクションの各々を検証することと、トランザクションの検証結果を記憶することと、を行うように構成されたハードウェアアクセラレータと、を含むコンピューティングシステムである。更に、プロセッサ又はハードウェアアクセラレータのうちの1つは、トランザクションを台帳にコミットするように構成されている。

【0006】

本明細書で説明される別の実施形態は、ハードウェアアクセラレータにおいて、ブロックチェーンの台帳にコミットされるトランザクションのブロックに対応する複数のパケットを受信することと、ハードウェアアクセラレータにおいて、トランザクションのブロック内の異なるコンポーネントのハッシュを計算することと、ハードウェアアクセラレータ

10

20

30

40

50

において、ハッシュが以前に計算されたハッシュと一致すると判定することと、ハードウェアアクセラレータにおいて、トランザクションのブロックを検証するためのタスクを生成することと、を含む方法である。

【0007】

上記の特徴が詳細に理解することができるように、上記で簡単に要約されたより具体的な説明が、例示的な実装形態を参照することによって行われ得、それらの実装形態のうちのいくつかは添付の図面に示される。しかしながら、添付の図面は、典型的な例示的な実装形態のみを例解しており、したがって、その範囲を限定するものと見なされるべきではないことに留意されたい。

【図面の簡単な説明】

【0008】

【図1】一例による、許可されたブロックチェーンに対応するタイミングチャートである。

【図2A】一例による、ハードウェアアクセラレータを有するブロックチェーン内のノードのブロック図である。

【図2B】一例による、ハードウェアアクセラレータを有するブロックチェーン内のノードのブロック図である。

【図3】一例による、ハードウェアアクセラレータ内のプロトコルプロセッサとブロックプロセッサとの間のインターフェースを例解する。

【図4】一例による、パケットを解析して、ハードウェア処理のためにデータを準備するためのフローチャートである。

【図5A】一例による、トランザクションのブロック内の各コンポーネントについて別個のパケットを送信することを例解する。

【図5B】一例による、ブロック内の個々のコンポーネントを送信するためのパケットを例解する。

【図6】一例による、ハードウェアアクセラレータ内のプロトコルプロセッサのブロック図である。

【図7】一例による、プロトコルプロセッサ内のブロック抽出器のブロック図である。

【図8】一例による、トランザクションをブロックチェーン台帳にコミットする前にトランザクションを検証するためのフローチャートである。

【図9】一例による、ブロックプロセッサのブロック図である。

【図10】一例による、ブロック検証のブロック図である。

【図11】一例による、ハードウェアアクセラレータ内のレジスタとCPUとの間の通信を例解する。

【発明を実施するための形態】

【0009】

様々な特徴が、図面を参照して以下に説明される。図面は縮尺通りに描かれている場合もあるし、描かれていない場合もあり、同様の構造又は機能の要素は図面全体を通して同様の参照番号によって表されていることに留意されたい。図面は、特徴の説明を容易にすることのみを意図していることに留意されたい。それらは、明細書の網羅的な説明として、又は特許請求の範囲に対する限定として意図されていない。更に、例解された実施例は、示されたすべての態様又は利点を有する必要はない。特定の実施例に関連して説明される態様又は利点は、必ずしもその実施例に限定されず、そのように例解されていない場合、又はそのように明示的に説明されていない場合であっても、任意の他の実施例において実施することができる。

【0010】

本明細書の実施形態は、ブロックチェーンマシン又はノードのためのハードウェアアクセラレータ（例えば、ネットワークアクセラレーションエンジン又は計算アクセラレーションエンジン）を説明する。ハードウェアアクセラレータは、トランザクションのブロックの別個のコンポーネントを含むパケットを解析して、検証プロセスを実行するためのデータを生成する。すなわち、ネットワークプロトコル（例えば、TCP（Transmi

10

20

30

40

50

ssion Control Protocol)) を使用して伝送されるデータは、通常、パケットが最初にソフトウェアによって処理されることなくハードウェアアクセラレータによって使用されるのには適していない。ソフトウェアを使用することに伴う待ち時間を回避するために、本明細書の実施形態は、データが、ソフトウェアの介入なしにアクセラレータ内の下流コンポーネントによって消費され得るように、パケットを解析し、かつデータを準備するハードウェアアクセラレータ内のプロトコルプロセッサを説明する。次いで、これらの下流コンポーネントは、検証動作を実行して、1つ以上のトランザクションが、許可されたブロックチェーン又は許可のないブロックチェーンの台帳にコミットされる（すなわち、追加される）前に、それらのトランザクションを検証することができる。

【0011】

ブロックチェーンは、複数のピアノードを含み得、その各々は、サーバ又はコンテナ上で動作する標準ソフトウェアを含む。バリデータノードとして知られるいくつかのピアノードは、それらのトランザクションがブロックチェーン台帳にコミットされ得る前に、数十又は数百のトランザクションのブロックを迅速に検証する必要があるため、システム性能の主なボトルネックである。ソフトウェアを使用してトランザクションのブロックを検証する代わりに、ハードウェアアクセラレータは、わずかな時間でトランザクションを検証することができる。次いで、ピアノードソフトウェアは、ハードウェアアクセラレータから検証結果を収集し、その結果を、受信したブロックデータと組み合わせて、ブロックを導出し、次いで、ブロックは、記憶された台帳にコミットされる。実験的な設定において、ハードウェアアクセラレータを有するノードは、ネットワーキングアクセラレーションエンジンに結合されたとき、マルチコアサーバ上で実行するソフトウェアのみのピアと比較して、トランザクションコミットスループットにおいて10倍を超える改善を達成した。

【0012】

図1は、一例による、許可されたブロックチェーン100に対応するタイミングチャートである。図1の許可されたブロックチェーン100のタイミングチャートは、特に、Hyperledger Fabricに関するが、本明細書の実施形態は、任意のタイプの許可されたブロックチェーンに適用することができる。更に、本明細書の実施形態は、トランザクションが台帳にコミットされる前にトランザクションに対して検証プロセスを実行する非許可ブロックチェーンにも適用し得る。したがって、Hyperledger Fabricは、以下に説明されるハードウェアアクセラレータから利益を得ることができる好適なブロックチェーンネットワークの単なる一例として提供される。

【0013】

ブロックチェーン100におけるトランザクションフローは、実行—順序付け—検証モデルに従い、トランザクションが最初に実行され、次いでブロックに順序付けされ、最後に検証され、（これまでコミットされたブロックのグローバル状態を保持するための状態データベースとともに）台帳にコミットされる。その結果、許可されたブロックチェーン100は、ピア、オーダーラ、クライアントなどの異なるタイプのノードを含み、各ノードは、MSPによって提供されるアイデンティティを有する。この識別は、証明書の形態で提供することができる。

【0014】

クライアントは、ブロックチェーン100上でコミットされるトランザクションを提出する任意のエンティティであり得る。例えば、ブロックチェーン100が送金を追跡するために金融機関によって使用される場合、クライアントは、資金を第1の口座から（同じ金融機関又は異なる機関における）第2の口座に移動させるためのトランザクションを提出し得る。ステップ1において、クライアントは、ブロックチェーンにコミットされるトランザクションを提出する。具体的には、トランザクションは、複数の承認ノード（又はピア）上で受信される。承認ノードは、トランザクションを実行／承認すること、及びブロックを台帳に対して検証／コミットすることの両方を行う。各承認ノードは、それ自体の状態データベースに対してトランザクションを実行して、トランザクションの読み取り

10

20

30

40

50

一書き込みセットを計算する（図1においてEとしてマークされる）。読み取りセットは、アクセスされたキー及びそれらのバージョン番号であり、書き込みセットは、それらの新しい値で更新されるキーである。

【0015】

承認プロセスが成功した場合（すなわち、エラーがない場合）、ステップ2において、承認ノードは、それらの承認をトランザクションに追加し、トランザクションをクライアントに返す。クライアントが十分な数の承認を収集した後、ステップ3で、クライアントは、順序付けサービスに、トランザクションを検証プロセスに提出するように求める。一実施形態では、順序付けサービスは、コンセンサス機構を使用して、トランザクションの全順序付けを確立するオーダー（例えば、コンピューティングノード）を含む。Raft及びApache Kafka／Zookeeperベースのコンセンサス機構など、複数のプラグブルコンセンサス機構が利用可能である。

10

【0016】

ステップ4において、順序付けサービスは、ブロックへの包含のためにトランザクションが受け入れられた後にクライアントに応答する（ステップ4）。次いで、順序付けサービスは、順序付けされたトランザクションからトランザクションのブロック105を作成する。一実施形態では、順序付けサービスは、ユーザ構成タイムアウトが満了したときか、又はブロックサイズに対するユーザ構成制限に達したときに、順序付けられたトランザクションからブロック105を作成する。

【0017】

20

ブロック105が作成されると、順序付けサービスは、ステップ5において、例えばゴシッププロトコルを介して、それをすべての承認及び非承認ノードにブロードキャストする。各ノードは、ブロック105内のすべてのトランザクションを検証し、次いで、ブロックを台帳及び状態データベース（Vとしてマークされる）にコミットする。最後に、ノードのうちの1つが、トランザクションがコミットされたことか、又はトランザクションが台帳において無効又は有効としてマークされたかどうかの通知をクライアントに送信する（ステップ6）。

【0018】

図1は、検証フェーズの検証ワークフロー110を右側により詳細に示している。ワークフロー110は、ブロック105を受信するすべてのノード上で実行される4つのステップを示す。ゴシッププロトコルを介して順序付けサービス（又はリードノード）からトランザクションのブロック105を受信すると、ステップ1において、ノードは、ブロックの構文構造をチェックし、その署名を確認し、次いで、以下でより詳細に説明する様々な動作のパイプラインを介してそれを送信する。ステップ2では、ブロック内の各トランザクションが構文的にチェックされ、その署名が確認される。次いで、検証システムチェーンコード（validation system chaincode、VSCC）が、承認が検証され、関連付けられたチェーンコードの承認ポリシーが評価される各トランザクション上で実行される。トランザクションは、その承認ポリシーが満たされない場合、無効としてマークされる。

30

【0019】

ステップ3において、マルチバージョン同時実行制御（multi-version concurrency control、MVCC）チェックが実行される。このチェックは、有効なトランザクション間に読み取り一書き込み競合がないことを確実にする。換言すれば、それは、1つのトランザクションのみが意図されたときに2つのトランザクションがコミットされる二重支払い問題を回避する。各トランザクションの読み取りセットは、状態データベース（図1において「state db」として例解される）にアクセスすることによって再び計算され、承認フェーズからの読み取りセットと比較される。これらの読み取りセットが異なる場合、他の何らかのトランザクション（このブロック105又は以前のブロックのいずれかにおける）が既に同じキーを変更しており、したがって、このトランザクションは無効としてマークされる。

40

【0020】

50

最後のステップ4において、ブロックは、ノードにおいて、記憶された台帳にコミットされる。一実施形態では、ブロック全体が最初に、そのトランザクションの有効／無効フラグとともに台帳に書き込まれる。次いで、有効なトランザクションの書き込みセットが状態データベースにコミットされる。

【0021】

図2A及び図2Bは、一例による、ハードウェアアクセラレータ210を有するブロックチェーン内のノード200のブロック図である。一実施形態では、ノード200は、トランザクションをブロックチェーンにコミットするときに検証プロセスを実行する任意のコンピューティングシステムである。例えば、ノード200は、図1に示すように、承認又は非承認ノード又はピアであり得る。一実施形態では、ノード200は、サーバ又は他のコンピューティングシステムである。

10

【0022】

図2Aにおいて、ノード200Aは、CPU205、ハードウェアアクセラレータ210、及びメモリ235を含む。CPU205は、それぞれが任意の数の処理コアを含むことができる任意の数のプロセッサを表す。メモリ235は、揮発性メモリ、不揮発性メモリ（例えば、ハードディスクドライブ）、及びこれらの組み合わせを表す。示されるように、メモリ235は、ブロックチェーンのコミットされたトランザクションをリストする台帳240を記憶する。

【0023】

ハードウェアアクセラレータ210は、図1に例解される検証ワークフロー110を実行するための様々な回路要素を含む。一実施形態では、ハードウェアアクセラレータ210は、集積回路である。別の実施形態では、ハードウェアアクセラレータ210は、1つ以上の集積回路が実装される基板（例えば、PCIEカードなどのプリント回路基板（printed circuit board、PCB））である。一実施形態において、集積回路は、プログラマブルロジックを含む、フィールドプログラマブルゲートアレイ（例えば、field program mable gate array、FPGA）又はシステムオンチップ（system on a chip、SoC）である。この例では、アクセラレータ210内の様々な回路ブロックは、プログラマブルロジックで実装される。しかしながら、別の実施形態では、集積回路は、アクセラレータ210の回路ブロックが、強化された回路においてのみ実装される、特定用途向け集積回路（application specific integrated circuit、ASIC）であり得る。プログラマブルロジックを有するFPGA及びSoCを使用することは、検証プロセスが変更された場合に再プログラムされる柔軟性をアクセラレータ210に与える一方、ASICを使用することは、スペースを節約し得る。

20

30

【0024】

アクセラレータ210は、トランザクションに関するデータを含むイーサネット（登録商標）パケットを受信するためのネットワークインタフェース215と、データを再フォーマットするためのプロトコルプロセッサ220と、検証ワークフローを実行するためのブロックプロセッサ225と、検証の結果を記憶するレジスタマップ230（reg_map）（例えば、メモリレジスタ）と、を含む。プロトコルプロセッサ220、ブロックプロセッサ225、及びレジスタマップ230は、以下でより詳細に説明される。一般に、これらのハードウェアブロックは、受信したトランザクションのブロックを検証するために協働する。すなわち、ネットワークインタフェース215は、トランザクションのブロックに対応するデータを含む複数のパケットを受信する。このデータは処理に適さないフォーマットであり得るため、プロトコルプロセッサ220は、ブロックプロセッサ225が消費のためにデータを再フォーマットし、出力することができる。ブロックプロセッサ225は、検証ワークフローにおけるほとんどのステップを実行するが、これらのステップのいくつかは、プロトコルプロセッサ220及びレジスタマップ230によって実行され得る。更に、台帳240が、メモリ235（アクセラレータ210によって直接アクセス可能ではない場合がある）に記憶されるため、ノード200Aは、検証されたトランザクションを台帳240にコミットするために、CPU205に依存し得る。すなわち、

40

50

アクセラレータ 210 は、CPU 205 が評価することができる検証結果をレジスタマップ 230 に記憶し、次いで、トランザクションを台帳にコミットすることができる。すなわち、一実施形態では、すべてのトランザクションがレジにコミットされるが、検証フラグは、どのトランザクションが有効であり、どのトランザクションが無効であったかに関する情報を記憶する。しかしながら、状態データベース（以下に説明する）に対しては、正常に有効にされたトランザクションのみがコミットされる。検証の大部分はハードウェアアクセラレータ 210 において実行されるが、トランザクションを台帳 240 にコミットすることは、CPU 205 上で実行されるソフトウェアによって実行され得る。

【0025】

図 2 B は、ネットワークインターフェースカード（network interface card、NIC）245 の追加を除いて、図 2 A のノード 200 A と同じであるノード 200 B を例解する。一実施形態では、NIC 245 は、どのネットワークトラフィックがアクセラレータ 210 を通って流れ、どのネットワークトラフィックが NIC 245 を通って流れるかを判定する能力をノード 200 B に提供する。一実施形態では、ブロックチェーンに関連するすべてのトラフィックは、アクセラレータ 210 を介して送信され得、ブロックチェーンに関連しない受信されたネットワークトラフィックは、NIC 245 によって処理される。対照的に、ノード 200 A では、ノード 200 A によって受信されたすべてのネットワークトラフィック（ブロックチェーントラフィックであろうと非ブロックチェーントラフィックであろうと）は、アクセラレータ 210 において受信され得る。例えば、プロトコルプロセッサ 220 は、検証に関連するネットワークトラフィックをブロックプロセッサ 225 に転送するが、他のすべてのトラフィックを CPU 205 に転送し得る。

【0026】

別の実施形態において、アクセラレータ 210 は、アクセラレータ 210 におけるトランザクションの検証に関連するネットワークトラフィックのみを受信する一方、他のすべてのトラフィック（承認要求などの他のタイプのブロックチェーントラフィックであるか、非ブロックチェーントラフィックであるかにかかわらず）は、NIC 245 によって受信され、処理される。

【0027】

図 2 A 又は図 2 B のいずれにも示されていない更に別の実施形態では、アクセラレータ 210 は、CPU 205 の支援なしにすべてのブロックチェーン動作を実行し得る。その実施形態では、NIC 245 及び CPU 205 は、ブロックチェーンタスクを実行するために使用されないが、CPU 205 は、アクセラレータ 210 を構成又は制御するために使用され得る。このシナリオでは、全てのネットワークトラフィックは、ブロックチェーン関連パケットを処理するが他のパケットを CPU 205 へ／から転送するアクセラレータ 210 を通過する。

【0028】

図 3 は、一例による、ハードウェアアクセラレータ内のプロトコルプロセッサとブロックプロセッサとの間のインターフェース 300 を例解する。この例では、インターフェース 300 は、ブロックデータ、トランザクションデータ（図 3 では tx データとラベル付けされている）、承認データ、読み取りセットデータ、及び書き込みセットデータをブロックプロセッサ 225 に伝送するために使用される。すなわち、プロトコルプロセッサ 220 は、トランザクションのブロックを検証するために使用される情報を含むイーサネット（登録商標）パケットを、ブロックチェーン内の他のノードから（例えば、順序付けサービス内のオーダーから）受信する。次いで、プロトコルプロセッサ 220 は、ブロックプロセッサ 225 によって消費することができるように、データを再フォーマット／解析する。

【0029】

図 4 は、一例による、パケットを解析して、ハードウェア処理のためにデータを準備するための方法 400 のフローチャートである。説明を容易にするために、方法 400 におけるブロックは、以下の図と並行して説明される。

10

20

30

40

50

【0030】

ブロック405において、アクセラレータ内のネットワークインタフェースは、トランザクションのブロックの別個のコンポーネントを含むパケットを受信する。ブロックは通常、単一のパケットで送信するには大きすぎるため（例えば、ブロックは1メガバイトを超える可能性がある）、ソフトウェアアプリケーションは、ネットワークプロトコル又はソフトウェアドライバに依存して、ブロックを複数のパケットにチャンク化する。しかしながら、ネットワークプロトコル／ドライバは、典型的には、データがブロック内でどのように構造化されるかについての知識を有しておらず、したがって、データは、典型的には、アドホック方式で伝送される。これは、ハードウェアアクセラレータが、受信されたパケットを解析し、トランザクションのブロックを再構築することを、不可能ではないにしても困難にする。しかしながら、本明細書の実施形態は、ハードウェアアクセラレータが、トランザクションのブロック内の異なるコンポーネントを再構築するために、パケットを解析することができるように、トランザクションのブロックを伝送するための技法を説明する。

10

【0031】

図5Aは、一例による、トランザクションのブロック505内の各コンポーネントについて別個のパケットを伝送することを例解する。具体的には、図5Aは、送信者515（例えば、ブロックチェーン内のオーダー又は他のピアノード）がハードウェアアクセラレータ210にパケット520を伝送し、したがって、ブロックメッセージ505が再構築され得るようにこれらのパケットが正確に解析され得る、通信システム500を例解する。そうするために、送信者515内のソフトウェアは、ブロックメッセージ505を、ここではヘッダ、トランザクション（TX）1～5、及びメタデータとして示される、その別個のブロックコンポーネント510に分割する。すなわち、ブロックメッセージ505内のデータを別個のパケットにどのように分割するかを判定するために、ネットワークプロトコル又はドライバに依存するのではなく、（ブロック505の構造を認識している）送信者515上のソフトウェアが、ブロックをそのコンポーネント510に分割することができる。

20

【0032】

図5Aは、ネットワーク、例えば、インターネットなどのプライベートネットワーク又はパブリックネットワークを介して、パケット520A～520Gをハードウェアアクセラレータ210に伝送する送信者515を例解する。示されるように、各パケット520は、コンポーネント510のうちの1つのみを含み、すなわち、パケット520Aは、ブロック505のヘッダを含み、パケット520B～520Fは各々、トランザクションのうちの1つを含み、パケット520Gは、ブロック505のメタデータを含む。ソフトウェアがブロック505をそのコンポーネント510に分割しなければ、ネットワークプロトコル又はドライバは、ヘッダ及びTX1の一部を含む第1のパケット、TX1の残りのデータ及びTX2の一部を含む第2のパケットなどを送信したことがある。ブロック505内の異なるコンポーネント510のこの予測不能な分岐は、ブロック505が再構築され得るようにデータを解析するように、ハードウェアアクセラレータ210を設計することを困難にし得る。

30

40

【0033】

また、図5Aには示されていないが、送信者515は、コンポーネント510に関連付けられた署名された証明書を削除し得る。これらの証明書を使用して、ブロック505及びブロック内のトランザクションを検証し得る。しかしながら、これらの証明書（例えば、x509証明書）は大きく、これは、それらをアクセラレータ210に送信することが、かなりの量の帯域幅を必要とすることを意味する。一実施形態において、送信者515は、パケット520を形成し、次いでアクセラレータ210に送信される前に、ブロック505から証明書を除去する。代わりに、送信者515は、証明書をID（例えば、8ビットコード）で置き換えることができる。一実施形態では、送信者515は、証明書を別々に一度にアクセラレータ210に伝送し、アクセラレータ210は、次いで、後で取り

50

出すために証明書をキャッシュに記憶することができる。このようにして、証明書がブロック505内の複数のコンポーネント510に関連付けられている場合、証明書は繰り返して、ではなく1回送信される。

【0034】

図5Bは、一例による、ブロック内の個々のコンポーネントを送信するためのパケット520を示す。520は、L2及びIP/UDPデータ、トランスポートヘッダ525、ブロックマシンプロトコルヘッダ530、及びブロックメッセージペイロード535を含む。トランスポートヘッダ525は、シーケンス番号(Seq)、肯定応答番号(Ack)、及び制御データ(Ctrl)を含むことができる。シーケンス番号は、パケットがブロック505を表すパケットのシーケンス内のどこにあるかを示す。肯定応答番号は、送信者515に返送するためのAckパケットを生成するときに、アクセラレータ210によって使用される。制御データは、アプリケーションレベルプロトコル障害及び受信機バッファステータスなどの情報を含むことができる。

【0035】

ブロックマシンプロトコルヘッダ530は、メッセージタイプ(MsgType)及び注釈を含む。メッセージタイプは、ペイロード535内のデータがブロックヘッダであるか、トランザクションであるか、メタデータであるかを示す。注釈は、重要なデータの位置を指摘する。例えば、注釈は、パケット520内の関連データ(例えば、ペイロード535内の特定のデータを見つけることができる場所)を指すポインタ、及びキャッシュ内又は異なるパケット(しかし同じブロック)内のデータを指すロケータを含み得る。一実施形態において、注釈内のロケータは、パケット520のIDをマークするために使用される。ペイロード535は、パケット内で送信される特定のコンポーネント510に対応するデータ、すなわち、ブロック505からのメタデータを含むことができる。

【0036】

各パケット520はブロック505のコンポーネント510のうちの1つを含むため、これは、アクセラレータ210がブロックのすべてのパケット520を受信する前にデータを処理し始めることができるという利点を提供する。すなわち、ソフトウェアがすべてのパケットを受信し、次いで検証のためにブロック505を再構築するのを待つ代わりに、アクセラレータ210は、トランザクションが入ってきたときにトランザクションを処理することができる。例えば、プロトコルプロセッサは、TX2を含むパケット520Cがアクセラレータ210において受信される前に、TX1に対応するパケット520Bを解析することができる。したがって、アクセラレータ210は、トランザクションが再構築され、検証され得る前にすべてのパケットが受信されなければならないソフトウェアソリューションよりもはるかに早くトランザクションを処理し始めることができる。

【0037】

図6は、一例による、ハードウェアアクセラレータ内のプロトコルプロセッサ220のブロック図である。図示のように、データパケット520は、レベル4~7(L4~L7)パケットヘッダ(例えば、ブロックマシンプロトコルヘッダ530)を解析して、メッセージタイプ及び注釈を抽出するパケットフィルタ610において受信される。メッセージタイプは、ブロック開始、ブロックメタデータ、トランザクションなどを含み得る。メッセージペイロードの注釈は、ブロックヘッダの位置、ブロック署名の場所、ブロックID、トランザクションデータ開始、チャンネル名、トランザクションID、チェーンコード名、作成者認証局(certificate authority, CA)ID、トランザクションアクション、トランザクションCA ID、契約名、契約入力、承認者アクション、読み取り/書き込みセット、承認者リスト、トランザクション署名、オーダーCA ID、オーダー署名、証明書ID、証明書公開キーデータ、情報以来有効な状態、情報を通じて有効な状態などを含むことができる。パケットフィルタ610はまた、トランスポートヘッダ内のシーケンス番号を識別し、この番号を応答生成器605に転送し、応答生成器210は、応答パケット(例えば、肯定応答パケット)を送信者(例えば、オーダー)に返送する。

【0038】

パケットフィルタ 610 は、メッセージタイプ、注釈、及びパケットペイロードを、注釈に基づいてメッセージペイロードを処理し、ブロックプロセッサ 225 のための関連データを抽出するブロック抽出器 615 に転送する。ブロック抽出器 615 はまた、応答生成器 605 にメッセージ有効信号を提供し、それにより、生成器 605 は、ブロック及びそのトランザクションが有効であるか無効であるかを送信者に通知することができる。ブロックプロセッサ 225 の詳細は、方法 400 の残りの部分及び以下の図において説明される。

【0039】

方法 400 に戻ると、ブロック 410 において、プロトコルプロセッサ内のブロック抽出器は、パケット内の ID を使用して、パケットに対応する署名された証明書を識別する。上述したように、送信者は、ブロックを複数のパケットでハードウェアアクセラレータに伝送する前に、ブロックから証明書を取り除き得る。これは、証明書が大きく、めったに変更されないために行われ得る。したがって、証明書を参照するパケット内のはるかに小さい ID 又はキーで証明書を置き換えることは、かなりの帯域幅を節約し、トランザクションのブロック内の各コンポーネントが単一のパケットで送信され得ることを確実にし得る。

【0040】

しかしながら、パケットが受信されると、ブロック抽出器は、ブロック（及びブロック内のトランザクション）を検証し、ブロック及びトランザクションのシンタックスが正しいことを確実にするために、証明書を必要とする場合がある。したがって、ブロック抽出器はコンポーネントを再構築することができ、これは、ブロック抽出器が証明書を識別し、取り出すことを意味する。

【0041】

図 7 は、一実施例による、ブロック抽出器 615 のブロック図である。示されるように、ブロック抽出器 615 は、パケットペイロード及び注釈を受信して、そのペイロードに対応する証明書（又は複数の証明書）を識別するデータ挿入器 705 を含む。例えば、ペイロードがブロックメタデータである場合、ペイロードは、ブロックを準備したオーダーの証明書に対応する ID を含み得る。ペイロードがトランザクションを含む場合、ペイロードは、トランザクションを提出したクライアントに対応する ID、及びトランザクションを承認した承認ノードに対応する 1 つ以上の ID を含み得る。したがって、データ挿入器 705 は、ペイロード内の複数の ID を識別し得る。

【0042】

ID を使用して、データ挿入器 705 は、アイデンティティキャッシュ 710 内でルックアップを実行して、ID に対応する署名された証明書を取り出す。すなわち、オーダーが証明書をアクセラレータに送信するとき、アクセラレータは、それらの証明書（及びそれらの対応する ID）をアイデンティティキャッシュ 710 に記憶し、その結果、トランザクションのブロックを検証するときに、これらの証明書を取り出すことができる。

【0043】

方法 400 に戻ると、ブロック 415 において、ブロック抽出器 615 は、ブロックの別個のコンポーネントを再構築する。すなわち、各ペイロードが受信されると、ブロック抽出器 615 は、対応する証明書を取り出すことができる。これは図 7 に示されており、データ挿入器 705 は、取り出された証明書（又は複数の証明書）をペイロード内の他のデータとともにデータ抽出器 715 に伝送する。

【0044】

データ抽出器 715 は、異なる時間にトランザクションのブロック内のコンポーネントを再構築し得る。例えば、時間 1 の間、データ抽出器 715 は、第 1 の受信されたパケットを使用してブロックのヘッダを再構築し、時間 2 において、データ抽出器 715 は、第 2 の受信されたパケットを使用してブロック内の第 1 のトランザクションを再構築し、以下同様である。したがって、ブロック抽出器 615 は、ブロック内の異なるコンポーネントが抽出器 615 内の異なるステージ（例えば、異なる回路モジュール）で並列に実行さ

10

20

30

40

50

れ得るようにパイプライン化され得る。

【0045】

ブロック420において、ハッシュ計算器720は、ブロック内のセパレータコンポーネントのハッシュを計算する。一実施形態では、ハッシュ計算器720は、ブロック全体、ブロックのあらゆるトランザクション、及び各トランザクションのあらゆる承認に対するハッシュを生成する。ハッシュ計算器720は、これらのハッシュを異なる時間に生成し得る。例えば、ハッシュ計算器720は、トランザクションに対応するパケットを受信したときに、特定のトランザクションに対するハッシュ（及びそのトランザクションに関連付けられたすべての承認に対するハッシュ）を生成することができる。しかしながら、ハッシュ計算器720は、ブロックについてのすべてのパケットを受信した後、ブロック全体に対するハッシュを計算するのを待つ場合がある。

10

【0046】

ブロック425において、ハッシュチェッカ725は、ハッシュ計算器720によって計算されたハッシュがパケット内のハッシュと一致するかどうかを判定する。すなわち、送信者によって伝送されたパケットは、ハッシュ計算器720によって生成されたハッシュと比較することができる以前に計算されたハッシュ（又は少なくともハッシュへのポインタ）を含み得る。例えば、送信者は、ブロック、ブロック内の各トランザクション、及びトランザクションにおける各承認に対するハッシュを計算し、それらのハッシュをアクセラレータ210に伝送し得る。これらのハッシュがハッシュチェッカ725によって生成されたローカルハッシュと一致する場合、これは、メッセージが有効であり、ブロック及びトランザクションのための適切なシンタックスが従われたことを意味する。ハッシュが一致しない場合、方法はブロック445に進み、ハードウェアアクセラレータは、受信されたデータがシンタックスエラーを有することを示す。一実施形態では、ハードウェアアクセラレータは、検証プロセスが失敗したことを示す応答メッセージを送信者に送信する。

20

【0047】

ハッシュが、受信されたハッシュと一致すると仮定すると、方法400は、ブロック430に進み、ハードウェアアクセラレータは、受信されたデータが正常に解析されたことを示す。一実施形態では、図6の応答生成器605は、トランザクションのブロックが、プロトコルプロセッサ220によって正常に受信され、処理されたことを示す応答又は肯定応答（acknowledge、ACK）メッセージを送信者に伝送する。

30

【0048】

ブロック435において、タスクジェネレータ730は、検証プロセスを完了するために、ブロックプロセッサに対するタスクを生成する。一実施形態では、タスクジェネレータ730は、ブロックタスク、トランザクションタスク、及び承認者タスクを生成する。ブロックタスクは、ブロックID、ブロック署名（例えば、ブロックを生成したオーダーの証明書）などを含み得る。トランザクションタスクは、トランザクションID、トランザクション署名（例えば、トランザクションを生成したクライアントの証明書）、トランザクション読み取り／書き込みセットなどを含み得る。一実施形態では、タスクジェネレータ730は、ブロック内の各トランザクションに対するトランザクションタスクを生成する。タスクジェネレータ730はまた、トランザクションにおける各承認に対する承認者タスクを生成することができる。すなわち、各トランザクションはいくつかの承認を受け取ることができるため、タスクジェネレータ730は、それらの承認の各々に対するタスクを生成することができる。

40

【0049】

ブロック440において、タスクジェネレータ730は、タスク及び対応するデータをハードウェアアクセラレータ内のブロックプロセッサに転送する。次いで、ブロックプロセッサは、ブロック、トランザクション、及び承認を検証することができる。一般に、プロトコルプロセッサ（及び図5～図7に例解されるその回路コンポーネント）は、受信されたパケットを解析して、ブロックのデータを、ソフトウェアの助けなしにブロックプロ

50

セッサによって要約又は処理され得るフォーマットに再フォーマットする。したがって、ブロックチェーンにおける検証プロセスは、ハードウェアによって完全に実行され、それによって、このプロセスに必要な時間及び計算リソースを低減することができる。

【0050】

図8は、一例による、トランザクションをブロックチェーン台帳にコミットする前にトランザクションを検証するための方法800のフローチャートである。説明を容易にするために、方法800における異なるステージは、以下の図9～図11と併せて説明される。更に、方法800におけるステージは、図1に例解される検証ワークフロー110のステップ1～4に相関する。

【0051】

方法800は、ハードウェアアクセラレータが、例えば、順序付けサービスから、台帳にコミットされ得る前に検証される必要があるトランザクションのブロックをすでに受信していると仮定する。更に、方法800は、プロトコルプロセッサがブロックシンタックスチェック（例えば、図1の「blkシンタックスチェック」）及びトランザクションシンタックスチェック（例えば、図1の「txシンタックスチェック」）を実行して、検証を実行するために必要なすべてのデータが受信されたことを確認したと仮定する。

【0052】

ステージ805において、ハードウェアアクセラレータは、トランザクションのブロック上の署名を確認する。より具体的には、ブロックプロセッサは、図3に示される情報をプロトコルプロセッサから受信し、ブロック確認を実行することができる。図9に示すように、ブロックプロセッサ225は、2つのハードウェアサブモジュール：ブロック確認905及びブロック検証910を含む。ブロック確認905は、ブロック上のオーダー署名を確認するための楕円曲線デジタル署名アルゴリズム（elliptic curve digital signature algorithm、ECDSA）エンジン920を含む。言い換えれば、ECDSAエンジン920は、ブロック内の受信された署名を、オーダーのための既知の署名（例えば、証明書）と比較して、それらが一致することを確実にする。このようにして、ブロック確認905は、トランザクションのブロックが認められたノードから来たことを確かめることができる。ECDSAエンジン920が示されているが、任意の好適な署名アルゴリズムエンジンを使用することができる。

【0053】

オーダー署名を含むことに加えて、ブロック確認905で受信されたブロックデータはまた、ブロック番号、ブロック内のトランザクションの数などを含むこともできる。

【0054】

方法800に戻ると、ステージ810において、ハードウェアアクセラレータは、ブロック内の複数のトランザクションを検証する。図9において、この機能は、ブロック検証910によって実行される。すなわち、ブロック確認905は、ブロックが既知のオーダーによって受信されたことを確実にする一方、ブロック検証910は、ブロック内の個々のトランザクションを検証する。これを行うために、ブロック検証910は、プロトコルプロセッサ（図9には図示せず）から、トランザクションデータ、承認データ、読み取りセットデータ、及び書き込みセットデータを受信する。ブロック検証910は、次いで、ブロック番号、有効／無効トランザクションフラグ、待ち時間などの検証結果を含むブロック検証データを出力する。

【0055】

一実施形態では、ブロック確認905及びブロック検証910は、ブロックレベルでパイプライン化される。すなわち、ブロックプロセッサ225は、ブロック確認905においてトランザクションの第1のブロックを処理することができる（パイプラインステージ1）一方、ブロック検証910は、トランザクションの第2のブロックを処理する（パイプラインステージ2）。言い換えれば、ブロック確認905は、ECDSAエンジン920を使用して、ブロック検証910が第2のブロック内の個々のトランザクションを検証するのと同時に、許可されたオーダーが第1のブロックを伝送したことを確実にするこ

10

20

30

40

50

とができる。

【0056】

ブロックプロセッサ225はまた、ブロック確認905及びブロック検証910から受信した信号を監視することによって、ブロックレベル及びトランザクションレベルの統計値を収集するブロックモニタ915も含む。例えば、ブロックモニタ915は、トランザクションのブロックを検証するのに必要な時間若しくは待ち時間、又はブロックプロセッサ225のスループット（例えば、単位時間当たりに処理されるブロックの数）を判定し得る。

【0057】

更に、図8は、ステージ810がステージ815～825に細分することができることを例解する。一実施形態では、方法800のステージ815～825は、図9のブロック検証910のより詳細な図を例解する図10のパイプラインステージ2a、2b、及び2cに対応する。

【0058】

ステージ815において、ブロック検証910における第1のパイプラインステージ2aは、各トランザクションの署名を確認する。図10に示すように、ステージ2aは、各々がECDSAエンジン1035を含む複数のトランザクション確認ブロック1005を含む。これらのエンジン1035は、トランザクションのクライアント（又は作成者）の署名を確認する。すなわち、ECDSAエンジン1035は、受信された署名を、txデータ及び作成者の公開キーペアに基づいて計算された署名と比較することによって、トランザクションが既知のクライアントによって署名されたことを確実にする。

【0059】

複数のトランザクション確認ブロック1005が存在するため、ステージ2aは、複数のトランザクションに対するクライアント署名を並行して検証することができる。ブロック検証910が含むトランザクション確認ブロック1005の数を判定することは、設計上の選択である。追加のトランザクション確認ブロック1005を有することは、ステージ2aが、より多くのトランザクションを並列に処理することができるが、アクセラレータ内の追加の空間及び電力を使用するという犠牲を払うことを意味する。

【0060】

方法800に戻ると、ステージ820において、ブロック検証910は、承認ポリシーを使用して各トランザクションの承認を確認する。図10に示すように、パイプラインステージ2bは、複数のトランザクションVSCCブロック1010を含み、トランザクションVSCCブロック1010は、各々が複数のECDSAエンジン1015（又は任意の他のタイプの署名確認エンジン）及び承認ポリシー評価器1020を含む。トランザクションVSCCブロック1010は各々、特定のトランザクションの承認を確認する。そうするために、トランザクションVSCCブロック1010は、承認者ID及び確認データを含む承認データを受信する。クライアントトランザクションは、（図1に示されるように）複数の承認を受信し得るため、各ECDSAエンジン1015は、承認のうちの1つを並行して評価することができる。すなわち、トランザクションAが2つの承認を受信した場合、ECDSAエンジン1015Aは、第1の承認が、認められた承認ノードによって署名されたことを確認することができ、同時に、ECDSAエンジン1015Bは、第2の承認が、認められた承認ノードによって署名されたことを確認する。ここでも、各ブロック1010内のECDSAエンジン1015の数は、設計上の選択である。

【0061】

一実施形態では、承認ポリシー評価器1020は、チェーンコードごとに承認ポリシーを維持することができる。評価器1020は、トランザクションが、適切な承認を受信したことを確かめる。すなわち、認められた承認ノードによって承認が与えられたと仮定すると、評価器1020は、トランザクションが、適切な承認ノードから承認を受信したことを確かめる。例えば、金銭が2つの銀行間で移転されるべきであることをトランザクションが示す場合、評価器1020は、トランザクションがそれらの銀行の両方によって運

10

20

30

40

50

用される承認ノードから承認を受信したことを確実にすることをチェックし得る。トランザクションが、1つの銀行のみに対する承認ノードによって、又はトランザクションによって影響を受けない異なる銀行によって承認された場合、評価器1020は、トランザクションを無効にし得る。

【0062】

ステージ2bは複数のトランザクションVSCCブロック1010を含むため、ブロック検証910は、複数のトランザクションに対する承認を並行して確認することができる。すなわち、ステージ2aが、トランザクションが、認められたクライアントによって発信されたことを確認した後、ステージ2bは、トランザクションの承認が、有効であり、1つ以上の承認ポリシーを満たすことを確認することができる。ブロック検証910内のトランザクションVSCCブロック1010の数は、設計上の選択である。

【0063】

方法800に戻ると、ステージ825において、ブロック検証910は、バージョンチェックを実行し、状態データベース1030への書き込みキーをコミットする。これは、状態データベース1030及びレジスタマップ（図10には図示せず）に通信可能に結合されているトランザクションMVCC書き込みブロック1025を含むステージ2cによって実行される。更に、レジスタマップは、任意選択で、ノード内のCPU（図示せず）に結合されている。一実施形態では、トランザクションMVCC書き込みブロック1025は、状態データベース1030から読み取りキーを調べて、バージョンチェックを実行し、確かめられた場合、有効なトランザクションの更新された書き込みキーをデータベース1030にコミットする。そうするために、トランザクションMVCC書き込みブロック1025は、プロトコルプロセッサから読み取りセットデータ（各要素が読み取りキーバージョンペアを含む）及び書き込みセットデータ（各要素が書き込みキー値ペアを含む）を受信する。一実施形態では、状態データベース1030は、現在書き込まれている（又は更新されている）キーの読み取りを許可しないための内部ロック機構を含む。

【0064】

図10のステージ2a～2cはパイプライン化することができる。すなわち、図9は、ブロック確認905とブロック検証910との間のブロックレベルパイプラインを例解するが、図10は、特定のブロック内のトランザクションをステージ2a～2c内で並列に処理することができるトランザクションレベルパイプラインを例解する。すなわち、ステージ2aは、ブロック内のトランザクションの第1のセットを処理することができる一方、ステージ2bは、同じブロック内のトランザクションの第2のセットを処理し、ステージ2cは、同じブロック内のトランザクションの第3のセットを処理する。しかしながら、一実施形態では、トランザクション間の依存性のために、ブロック検証910は、同じブロックからのトランザクションのみを処理し得る。すなわち、ブロック検証910内の1つのステージは、第1のブロックからのトランザクションを処理することができない場合があるが、異なるステージは、第2のブロックからのトランザクションを処理する。

【0065】

方法800に戻ると、ステージ830において、ブロック検証910は、検証プロセスを実行した結果をレジスタ（例えば、レジスタマップ230）に記憶する。図11に示すように、レジスタマップは、ブロック番号、有効／無効トランザクションフラグ、待ち時間（ブロックモニタによって測定される）などを含み得るブロック検証910からブロック検証データを受信する。検証結果は、レジスタマップに書き込まれ、CPU205は、AXI-E-Lite又はPCIeインターフェースを使用して、検証結果にアクセスすることができる。一実施形態では、現在記憶されている検証結果がCPU205によって読み出されるまで、新しい検証結果をレジスタマップ230に書き込むことができない。更に、図11は、上述したように、ハードウェアアクセラレータ内のレジスタマップ230がCPU205によってアクセス可能であるシステム1100を例解するが、アクセラレータは、CPU205を使用しなくてもよく、代わりに、CPU205上で実行されるソフトウェアの支援なしに検証プロセスを完了することができる。

【0066】

ステージ835において、CPU（又はハードウェアアクセラレータ）は、トランザクションが有効であるか無効であるかをクライアントに通知する。すなわち、CPUは、検証結果を評価して、トランザクションのブロック内の各個々のトランザクションが検証されたかどうかを判定することができる。次いで、クライアントは、無効なトランザクションを再提出することを選択し得る。

【0067】

ステージ840において、トランザクションは、台帳にコミットされる。一実施形態では、有効トランザクションと無効トランザクションの両方が台帳にコミットされ、コミットされたトランザクションが有効であるか否かを示すための有効／無効フラグを含むことができる。

【0068】

前述では、本開示において提示される実施形態が参照される。しかしながら、本開示の範囲は、特定の記載された実施形態に限定されない。代わりに、説明される特徴及び要素の任意の組み合わせは、異なる実施形態に関連するか否かにかかわらず、企図される実施形態を実装及び実践するために企図される。更に、本明細書に開示される実施形態は、他の可能な解決策又は従来技術に勝る利点を達成し得るが、特定の利点が所与の実施形態によって達成されるか否かは、本開示の範囲を限定するものではない。したがって、前述の態様、特徴、実施形態、及び利点は、単に例示的なものであり、特許請求の範囲に明示的に記載されている場合を除き、添付の特許請求の範囲の要素又は限定とは見なされない。

【0069】

当業者によって理解されるように、本明細書に開示される実施形態は、システム、方法、又はコンピュータプログラム製品として具現化され得る。したがって、態様は、完全にハードウェアの実施形態、完全にソフトウェアの実施形態（ファームウェア、常駐ソフトウェア、マイクロコードなどを含む）、又は本明細書ではすべて一般に「回路」、「モジュール」、若しくは「システム」と呼ばれ得るソフトウェア態様とハードウェア態様とを組み合わせた実施形態の形態をとり得る。更に、態様は、コンピュータ可読プログラムコードが具現化された1つ以上のコンピュータ可読媒体において具現化されたコンピュータプログラム製品の形態をとり得る。

【0070】

1つ以上のコンピュータ可読媒体の任意の組合せを利用し得る。コンピュータ可読媒体は、コンピュータ可読信号媒体又はコンピュータ可読記憶媒体であり得る。コンピュータ可読記憶媒体は、例えば、電子、磁気、光学、電磁気、赤外線、若しくは半導体のシステム、装置、若しくはデバイス、又は前述の任意の好適な組み合わせであり得るが、これらに限定されない。コンピュータ可読記憶媒体のより具体的な例（非網羅的なリスト）は、1つ以上のワイヤを有する電気接続、ポータブルコンピュータディスクレット、ハードディスク、ランダムアクセスメモリ（random access memory、RAM）、読み取り専用メモリ（read-only memory、ROM）、消去可能プログラマブル読み取り専用メモリ（erasable programmable read-only memory、EPROM又はフラッシュメモリ）、光ファイバ、ポータブルコンパクトディスク読み取り専用メモリ（portable compact disc read-only memory、CD-ROM）、光記憶デバイス、磁気記憶デバイス、又は前述の任意の好適な組み合わせを含む。本明細書の文脈では、コンピュータ可読記憶媒体は、命令実行システム、装置、又はデバイスによって、又はそれに関連して使用するためのプログラムを含むか、又は記憶することができる任意の有形媒体である。

【0071】

コンピュータ可読信号媒体は、例えば、ベースバンドにおいて、又は搬送波の一部として、コンピュータ可読プログラムコードが具現化された伝搬データ信号を含み得る。そのような伝搬信号は、電磁気、光学、又はそれらの任意の好適な組み合わせを含むが、これらに限定されない、種々の形態のうちのいずれかをとり得る。コンピュータ可読信号媒体は、コンピュータ可読記憶媒体ではなく、命令実行システム、装置、又はデバイスによっ

10

20

30

40

50

て、又はそれに関連して使用するためのプログラムを通信、伝搬、又は移送することができる任意のコンピュータ可読媒体であり得る。

【0072】

コンピュータ可読媒体上に具現化されたプログラムコードは、ワイヤレス、ワイヤライン、光ファイバケーブル、RFなど、又は前述の任意の好適な組合せを含むが、それらに限定されない、任意の適切な媒体を使用して伝送され得る。

【0073】

本開示の態様の動作を実行するためのコンピュータプログラムコードは、例えば、Java（登録商標）、Smalltalk、C++などのオブジェクト指向プログラミング言語、及び「C」プログラミング言語又は同様のプログラミング言語などの従来の手続き型プログラミング言語を含む、1つ以上のプログラミング言語の任意の組合せで書き込まれ得る。プログラムコードは、ユーザのコンピュータ上で完全に、ユーザのコンピュータ上で部分的に、スタンドアロンソフトウェアパッケージとして、ユーザのコンピュータ上で部分的に、リモートコンピュータ上で部分的に、又はリモートコンピュータ若しくはサーバ上で完全に実行し得る。後者のシナリオでは、リモートコンピュータは、ローカルエリアネットワーク（local area network、LAN）若しくは広域ネットワーク（wide area network、WAN）を含む任意のタイプのネットワークを介してユーザのコンピュータに接続され得るか、又は外部コンピュータ（例えば、インターネットサービスプロバイダを使用するインターネットを介して）に接続が行われ得る。

【0074】

本開示の態様は、本開示に提示された実施形態による方法、装置（システム）、及びコンピュータプログラム製品のフローチャート例解図及び／又はブロック図を参照して以下に記載されている。フローチャート例解図及び／又はブロック図の各ブロック、並びにフローチャート例解図及び／又はブロック図におけるブロックの組み合わせは、コンピュータプログラム命令によって実装することができることが理解されよう。これらのコンピュータプログラム命令は、コンピュータ又は他のプログラム可能なデータ処理装置のプロセッサを介して実行される命令が、フローチャート及び／又はブロック図のブロックで指定された機能／行為を実施するための手段を作成するような機械をもたらすように、汎用コンピュータ、専用コンピュータ、又は他のプログラム可能なデータ処理装置のプロセッサに提供され得る。

【0075】

これらのコンピュータプログラム命令はまた、コンピュータ可読記憶媒体に記憶された命令が、フローチャート及び／又はブロック図のブロックで指定された機能／行為の態様を実装する命令を含む製造物品を生成するように、コンピュータ、プログラマブルデータ処理装置、及び／又は他のデバイスに、特定の方法で機能するように指示することができる、コンピュータ可読記憶媒体に記憶され得る。

【0076】

コンピュータプログラム命令はまた、コンピュータ、他のプログラマブルデータ処理装置、又は他のデバイスにロードされて、一連の動作ステップを、コンピュータ、他のプログラマブル装置、又は他のデバイス上で実行させて、コンピュータ実装プロセスを生成し得、そのため、コンピュータ、又は他のプログラマブル装置上で実行される命令は、フローチャート及び／又はブロック図のブロックに指定される機能／行為を実装するためのプロセスを提供する。

【0077】

図中のフローチャート及びブロック図は、本発明の様々な実施例によるシステム、方法、及びコンピュータプログラム製品の可能な実装形態のアーキテクチャ、機能、及び動作を例解する。これに関して、フローチャート又はブロック図の各ブロックは、指定された論理機能を実装するための1つ以上の実行可能命令を含む、命令のモジュール、セグメント、又は部分を表し得る。いくつかの代替的な実装形態では、ブロックに記載されている機能は、図に記載された順序から外れて発生し得る。例えば、連続して示される2つのブ

ロックは、実際には実質的に同時に実行され得るか、又はブロックは、関与する機能に応じて、逆の順序で実行され得る場合がある。ブロック図及び／又はフローチャート例解図の各ブロック、並びにブロック図及び／又はフローチャート例解図におけるブロックの組み合わせは、指定された機能若しくは行為を実行するか、又は専用ハードウェアとコンピュータ命令との組み合わせを行う、専用ハードウェアベースのシステムによって実装することができることに留意されたい。

【0078】

開示の技術は、特許請求の範囲に記載されたものに加えて、以下の非限定的な実施例によって表現され得る。

【0079】

実施例1．ブロックチェーン内のノードからトランザクションのブロックを含む複数のパケットを受信するように構成されたネットワークインタフェースと、複数のパケットを解析して、トランザクションに関するデータを生成するように構成されたプロトコルプロセッサと、ブロックの署名を確認することと、ブロック内のトランザクションの各々を検証することと、トランザクションの検証結果を記憶することと、を行うように構成されたブロックプロセッサと、を備える、ブロックチェーンのための検証プロセスを加速させるための集積回路。

【0080】

実施例2．集積回路が、システムオンチップ（SoC）、フィールドプログラマブルゲートアレイ（FPGA）、又は特定用途向け集積回路（ASIC）のうちの少なくとも1つを含む、実施例1に記載の集積回路。

【0081】

実施例3．SoC又はFPGAが、プログラマブルロジックを含む。実施例2に記載の集積回路。

【0082】

実施例4．ASICが、硬化回路のみを含む、実施例2に記載の集積回路。

実施例5．ブロックプロセッサが、ブロックの署名がブロックチェーン内のオーダーの既知の署名と一致することを確認するように構成された第1の署名アルゴリズムエンジンを含むブロック確認と、ブロック内のトランザクションの各々を検証するように構成されたブロック検証と、を含む、実施例1に記載の集積回路。

【0083】

実施例6．ブロック検証が、トランザクションに対応する署名がブロックチェーンを使用することを許可されたクライアントの既知の署名と一致することを確認するように構成された第2の署名アルゴリズムエンジンを含むトランザクション確認モジュールと、トランザクションにおける承認が、ブロックチェーン内の既知の承認ノードによって署名されたことを確認するように構成された第3の署名アルゴリズムエンジンと、承認が承認ポリシーを満たすことを確実にするように構成された承認ポリシー評価器と、状態データベース内のトランザクションに関連付けられたキー値ペアを読み取りかつ書き込むように構成されたトランザクションマルチバージョン同時実行制御（MVCC）書き込みブロックと、を含む、トランザクション検証システムチェーンコード（VSCC）ブロックと、を含む、実施例5に記載の集積回路。

【0084】

実施例7．ブロック検証が、トランザクションの第1のセットの署名を確認するために並列に動作するように構成された複数のトランザクション確認モジュールと、トランザクションの第2のセットの承認を確認するために並列に動作するように構成された複数のトランザクションVSCCブロックと、を含み、複数のトランザクションVSCCブロックが第2のセットのトランザクションを処理すると同時に、複数のトランザクション確認モジュールが第1のセットのトランザクションを処理するように、複数のトランザクション確認モジュールが、パイプラインの第1のステージを形成し、複数のトランザクションVSCCブロックが、パイプラインの第2のステージを形成する、実施例6に記載の集積

10

20

30

40

50

回路。

【0085】

実施例8．複数のトランザクション確認モジュールの各々が、トランザクションのうちの1つにおける承認のセットを並列に確認するための複数のアルゴリズム署名エンジンを含む、実施例7に記載の集積回路。

【0086】

実施例9．ブロック検証がブロックチェーンのトランザクションの第2のブロックを処理すると同時に、ブロック確認がブロックチェーンのトランザクションの第1のブロックを処理することができるように、ブロック確認がパイプラインにおける第1のステージを形成し、ブロック検証がパイプラインの第2のステージを形成する、実施例2に記載の集積回路。

10

【0087】

実施例10．ハードウェアアクセラレータにおいて、ブロックチェーンの台帳にコミットされるトランザクションのブロックを受信することと、ハードウェアアクセラレータにおいて、ブロックの署名を確認することと、ハードウェアアクセラレータにおいて、ブロック内のトランザクションの各々を検証することと、ハードウェアアクセラレータにおいて、トランザクションの検証結果を記憶することと、トランザクションを台帳にコミットし、検証結果に基づいてトランザクションが有効であるか否かを示すことと、を含む、方法。

【0088】

20

実施例11．トランザクションのブロックを受信することが、トランザクションのブロックを含む複数のパケットを受信することを含み、方法が、ハードウェアアクセラレータにおいて、複数のパケットを解析して、トランザクションに関するデータを生成することを更に含む、実施例10に記載の方法。

【0089】

実施例12．ブロック内のトランザクションの各々を検証することが、ブロックの署名がブロックチェーン内のオーダーの既知の署名と一致することを確認することを更に含む、実施例10に記載の方法。

【0090】

実施例13．ブロック内のトランザクションの各々を検証することが、トランザクションにおける承認がブロックチェーン内の既知の承認ノードによって署名されたことを確認することと、トランザクションにおける承認が承認ポリシーを満たすことを確実にすることと、を更に含む、実施例10に記載の方法。

30

【0091】

実施例14．ブロックチェーンのための検証プロセスを加速させるための集積回路であって、ブロックチェーンの台帳にコミットされるトランザクションのブロックに対応する複数のパケットを受信するように構成されたデータ挿入器と、トランザクションのブロック内の異なるコンポーネントのハッシュを生成するように構成されたハッシュ計算器と、ハッシュが、以前に計算されたハッシュと一致すると判定するように構成されたハッシュチェッカと、トランザクションのブロックを検証するためのタスクを生成するように構成されたタスクジェネレータと、を含む、集積回路。

40

【0092】

実施例15．集積回路が、システムオンチップ（SoC）、フィールドプログラマブルゲートアレイ（FPGA）、又は特定用途向け集積回路（ASIC）のうちの少なくとも1つを含む、実施例14に記載の集積回路。

【0093】

実施例16．SoC又はFPGAが、プログラマブルロジックを含む、実施例15に記載の集積回路。

【0094】

実施例17．ASICが、硬化回路のみを含む、実施例15に記載の集積回路。

50

実施例 18. トランザクションのブロックが、ブロックヘッダ、複数のトランザクション、及びメタデータを含み、ブロックヘッダ、複数のトランザクションの各々、及びメタデータの各々が、複数のパケットのうちの別個のパケットで送信され、複数のパケットが、ブロック又は複数のトランザクションに対応する署名された証明書を含まない、実施例 14 に記載の集積回路。

【0095】

実施例 19. ブロックチェーン内の認められたノードに対応する署名された証明書のローカルコピーを記憶するアイデンティティキャッシュを更に備え、データ挿入器が、複数のパケットに含まれる ID を使用して、トランザクションのブロックに対応する署名された証明書をアイデンティティキャッシュから取り出すように構成されている、実施例 18 に記載の集積回路。

10

【0096】

実施例 20. 複数のパケット内のデータ及び取り出された署名された証明書を使用して、トランザクションのブロックを再構築するように構成されたデータ抽出器を更に備え、ハッシュ計算器が、トランザクションの再構築されたブロックを使用して、ハッシュを生成するように構成されており、ハッシュチェッカが、トランザクションのブロックがシンタックスエラーを含まないことを確実にするために、生成されたハッシュが、以前に計算されたハッシュと一致するかどうかをチェックするように構成されている、実施例 19 に記載の集積回路。

【0097】

20

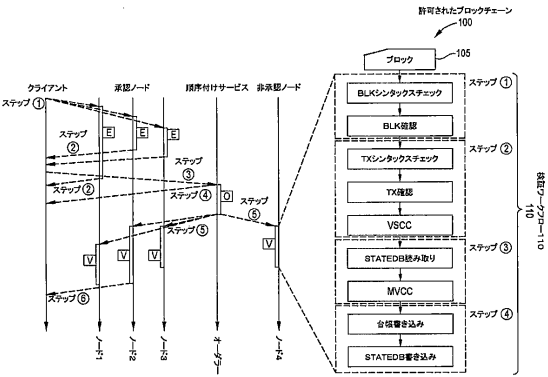
前述は特定の実施例を対象とするが、他の実施例及び更なる実施例が、その基本的な範囲から逸脱することなく考案され得、その範囲は、以下の特許請求の範囲によって決定される。

30

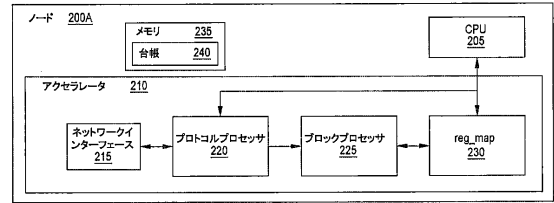
40

50

【図面】
【図 1】

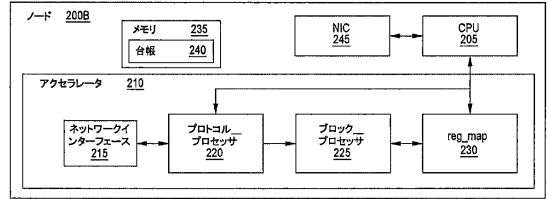


【図 2 A】

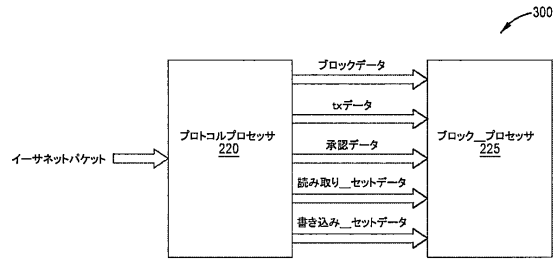


10

【図 2 B】



【図 3】



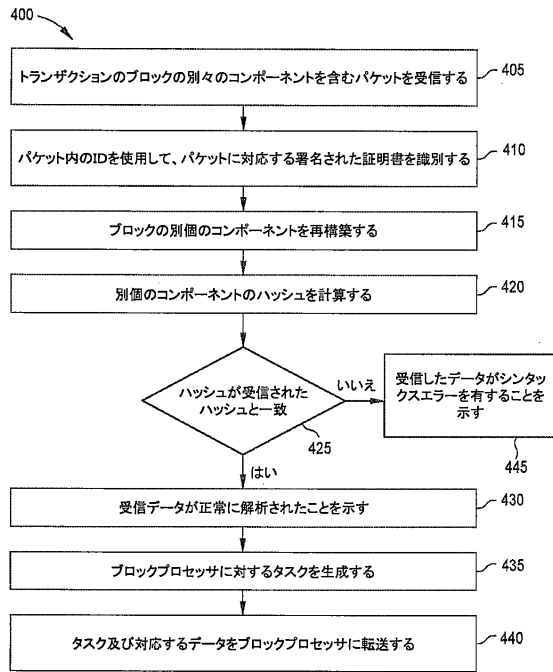
20

30

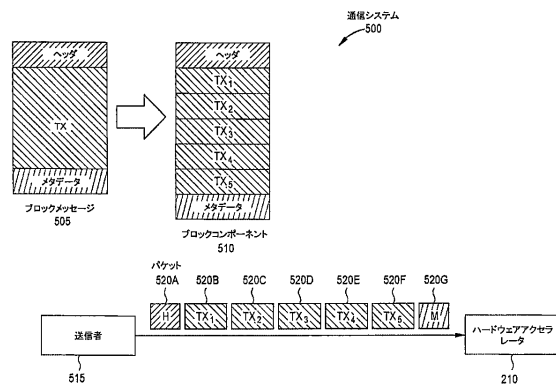
40

50

【図 4】



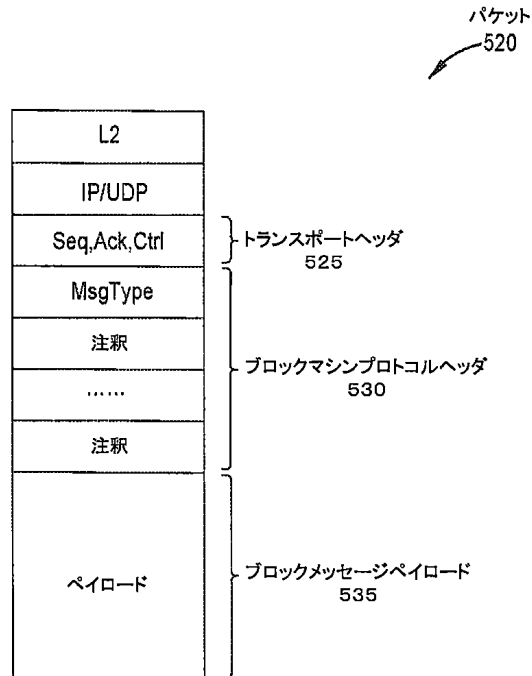
【図 5 A】



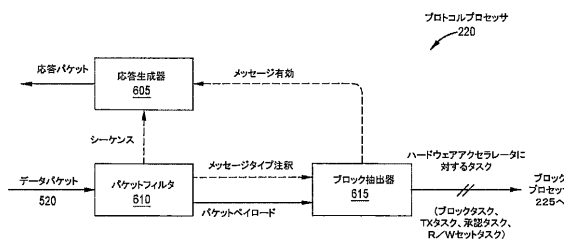
10

20

【図 5 B】



【図 6】

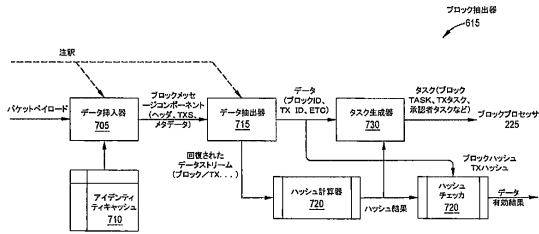


30

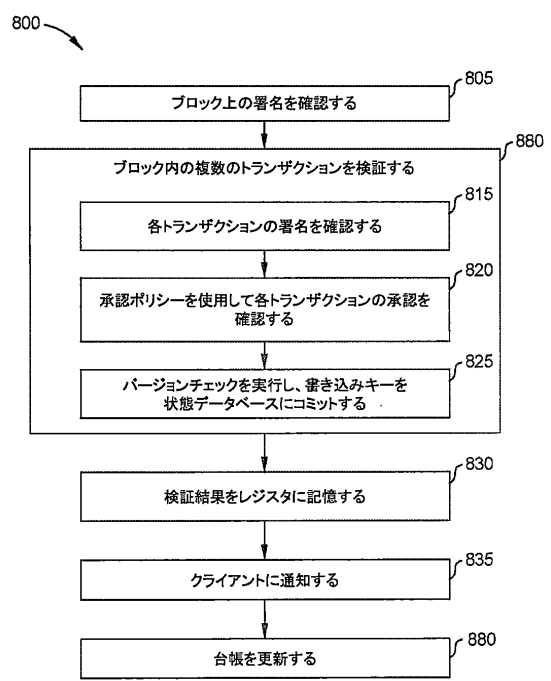
40

50

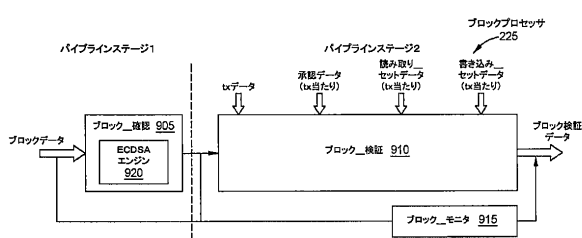
【図 7】



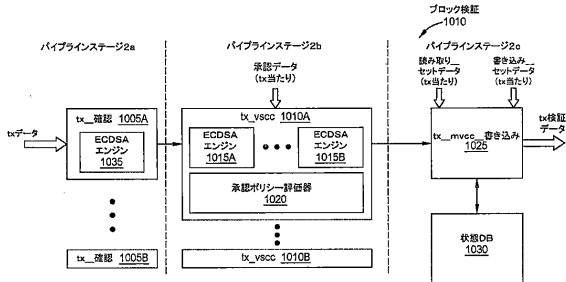
【図 8】



【図 9】



【図 10】



10

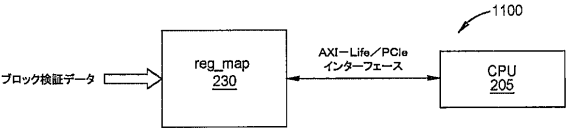
20

30

40

50

【図 11】



10

20

30

40

50

フロントページの続き

(33)優先権主張国・地域又は機関

米国(US)

イブ、2100

(72)発明者 モハン, サンダラジャラオ

アメリカ合衆国、95124 カリフォルニア州、サン・ノゼ、ロジック・ドライブ、2100

(72)発明者 ブレブナー, ゴードン・ジョン

アメリカ合衆国、95124 カリフォルニア州、サン・ノゼ、ロジック・ドライブ、2100

審査官 平井 誠

(56)参考文献 米国特許出願公開第2019/0026146 (US, A1)

米国特許出願公開第2020/0092360 (US, A1)

(58)調査した分野 (Int.Cl., DB名)

G09C 1/00-5/00

H04L 9/00-40