

(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7123146号
(P7123146)

(45)発行日 令和4年8月22日(2022.8.22)

(24)登録日 令和4年8月12日(2022.8.12)

(51)国際特許分類 F I
H 0 4 L 45/02 (2022.01) H 0 4 L 45/02

請求項の数 20 (全15頁)

(21)出願番号	特願2020-534180(P2020-534180)	(73)特許権者	591016172
(86)(22)出願日	平成30年9月19日(2018.9.19)		アドバンスト・マイクロ・デバイス
(65)公表番号	特表2021-508963(P2021-508963 A)		・インコーポレイテッド
(43)公表日	令和3年3月11日(2021.3.11)		ADVANCED MICRO DEVI
(86)国際出願番号	PCT/US2018/051782		CES INCORPORATED
(87)国際公開番号	WO2019/125561		アメリカ合衆国 9 5 0 5 4 カリフォル
(87)国際公開日	令和1年6月27日(2019.6.27)		ニア州、 サンタ クララ、 オーガスティ
審査請求日	令和3年9月17日(2021.9.17)	(74)代理人	ンドライブ 2 4 8 5
(31)優先権主張番号	15/850,616		100108833
(32)優先日	平成29年12月21日(2017.12.21)		弁理士 早川 裕司
(33)優先権主張国・地域又は機関	米国(US)	(74)代理人	100111615
早期審査対象出願			弁理士 佐野 良太
		(74)代理人	100162156
			弁理士 村雨 圭介
		(72)発明者	ヴィドヒャナサン カリヤナスンダラム
			最終頁に続く

(54)【発明の名称】 相互接続トポロジの自己識別

(57)【特許請求の範囲】

【請求項1】

複数のコンポーネントを備える通信ファブリックと、
前記通信ファブリックに結合されたプロセッサであって、
ファブリックブロック・インスタンスカウトレジスタから値を読み取ることであって、
前記値は、前記通信ファブリック内のコンポーネントの数を指定する、前記読み取るこ
とと、

各コンポーネントのレジスタを読み取ることによって、前記通信ファブリックの各コン
ポーネントに問い合わせを行うことであって、各レジスタは、対応するコンポーネントの
ブロックタイプの指標を少なくとも含み、処理ユニットであるブロックタイプについては
、プロセッサコアの数と、プロセッサコア当たりのメモリチャネルの数と、を含む、こと
と、

前記通信ファブリックの各コンポーネントの問い合わせに基づいて、ネットワークグラ
フを作成することと、

前記ネットワークグラフに基づいて、ルーティングテーブルを生成することと、
を行うように構成される、前記プロセッサと、
を備える、システム。

【請求項2】

前記プロセッサは、前記ルーティングテーブルを前記通信ファブリックの前記複数のコ
ンポーネントにプログラムするようにさらに構成される、請求項1に記載のシステム。

【請求項 3】

前記通信ファブリックの各コンポーネントは、それぞれのルーティングテーブルに基づいて、受信パケットをルーティングするように構成される、請求項 2 に記載のシステム。

【請求項 4】

各レジスタは、コマンド及びデータポートのタイプ及び数、データバス幅、ネイバーインスタンス識別子 (ID) 及び/またはファブリック ID のうちの 1 つまたは複数を備える、請求項 1 に記載のシステム。

【請求項 5】

前記ファブリックブロック・インスタンスカウントレジスタは、前記システム内のファブリックコンポーネントの合計数の指標を格納し、

前記ファブリックブロック・インスタンスカウントレジスタは、固定アドレスに存在する、請求項 1 に記載のシステム。

【請求項 6】

前記ネットワークグラフは、コンポーネントの数及びブロックタイプの指標のうち 1 つ又は複数に基づいて作成される、請求項 1 に記載のシステム。

【請求項 7】

前記ブロックタイプの指標は、対応するブロックが、コヒーレントマスタ、非コヒーレントマスタ、コヒーレントスレーブ、非コヒーレントスレーブまたはクロスバーのうち何れかであることの指標である、請求項 1 に記載のシステム。

【請求項 8】

プロセッサによって、ファブリックブロック・インスタンスカウントレジスタから値を読み取ることによって、前記値は、通信ファブリック内のコンポーネントの数を指定する、前記読み取ることと、

前記プロセッサによって、各コンポーネントのレジスタを読み取ることによって、前記通信ファブリックの各コンポーネントに問い合わせを行うことによって、各レジスタは、対応するコンポーネントのブロックタイプの指標を少なくとも含み、処理ユニットであるブロックタイプについては、プロセッサコアの数と、プロセッサコア当たりのメモリチャネルの数と、を含む、ことと、

前記プロセッサによって、前記通信ファブリックの各コンポーネントの問い合わせに基づいて、ネットワークグラフを作成することと、

前記プロセッサによって、前記ネットワークグラフに基づいて、ルーティングテーブルを生成することと、
を備える、方法。

【請求項 9】

前記ルーティングテーブルを前記通信ファブリックの複数のコンポーネントにプログラムすることをさらに備える、請求項 8 に記載の方法。

【請求項 10】

前記プロセッサによって、ブロックタイプの指標、処理ユニット、プロセッサコアの数、コア当たりのメモリチャネルの数のうち 1 つ又は複数に基づいて、前記ネットワークを作成することをさらに備える、請求項 9 に記載の方法。

【請求項 11】

各レジスタは、コマンド及びデータポートのタイプ及び数、データバス幅、ネイバーインスタンス識別子 (ID) 及び/またはファブリック ID のうちの 1 つまたは複数を備える、請求項 8 に記載の方法。

【請求項 12】

前記ファブリックブロック・インスタンスカウントレジスタは、前記通信ファブリック内のファブリックコンポーネントの合計数の指標を格納し、前記ファブリックブロック・インスタンスカウントレジスタは固定アドレスに存在する、請求項 8 に記載の方法。

【請求項 13】

セキュリティプロセッサが、前記ネットワークグラフを作成することと、前記ネットワ

10

20

30

40

50

ークグラフに基づいてルーティングテーブルを生成することとをさらに備える、請求項 8 に記載の方法。

【請求項 14】

前記ブロックタイプの指標は、対応するブロックが、コヒーレントマスタ、非コヒーレントマスタ、コヒーレントスレーブ、非コヒーレントスレーブまたはクロスバーのうち何れかであることの指標である、請求項 8 に記載の方法。

【請求項 15】

装置であって、

ファブリックブロック・インスタンスカウントレジスタから値を読み取ることであって、前記値は、通信ファブリック内のコンポーネントの数を指定する、前記読み取ることと、各コンポーネントのレジスタを読み取ることによって、前記通信ファブリックの各コンポーネントに問い合わせを行うことであって、各レジスタは、対応するコンポーネントのブロックタイプの指標を少なくとも含み、処理ユニットであるブロックタイプについては、プロセッサコアの数と、プロセッサコア当たりのメモリチャネルの数と、を含む、ことと、

前記通信ファブリックの各コンポーネントの問い合わせに基づいて、ネットワークグラフを作成することと、

前記ネットワークグラフに基づいて、ルーティングテーブルを生成することと、を行うように構成される、前記装置。

【請求項 16】

前記装置は、前記ルーティングテーブルを前記通信ファブリックの複数のコンポーネントにプログラムするようにさらに構成される、請求項 15 に記載の装置。

【請求項 17】

前記ネットワークグラフは、コンポーネントの数及びブロックタイプの指標のうち 1 つ又は複数に基づいて作成される、請求項 16 に記載の装置。

【請求項 18】

各レジスタは、コマンド及びデータポートのタイプ及び数、データバス幅、ネイバーインスタンス識別子 (ID) 及び / またはファブリック ID のうちの 1 つまたは複数を備える、請求項 15 に記載の装置。

【請求項 19】

前記ファブリックブロック・インスタンスカウントレジスタは、前記通信ファブリック内のファブリックコンポーネントの合計数の指標を格納し、前記ファブリックブロック・インスタンスカウントレジスタは固定アドレスに存在する、請求項 15 に記載の装置。

【請求項 20】

前記ブロックタイプの指標は、対応するブロックが、コヒーレントマスタ、非コヒーレントマスタ、コヒーレントスレーブ、非コヒーレントスレーブまたはクロスバーのうち何れかであることの指標である、請求項 15 に記載の装置。

【発明の詳細な説明】

【背景技術】

【0001】

コンピューティングシステムでは、単一の集積回路 (すなわち、チップ) またはマルチチップモジュール上に多数の異なるタイプのコンポーネントを統合することが増えている。システムの複雑性及び消費電力は、異なるタイプのコンポーネントの数に伴い増加する。多くの場合、これらのコンポーネントは、スイッチ、ルータ、通信バス、ブリッジ、バッファ、コントローラ、コヒーレントデバイス及びその他のリンクを介して接続されている。これらの相互接続コンポーネントの組み合わせは、本明細書において「通信ファブリック」または略して「ファブリック」と呼ばれる。

【0002】

一般的に、ファブリックは、チップまたはマルチチップモジュール上の複数のコンポーネント間でメッセージをルーティングすることにより、通信を容易にする。ファブリック

10

20

30

40

50

を介して通信されるメッセージの例には、メモリアクセス要求、ステータス更新、データ転送、コヒーレンシー・プローブ、コヒーレンシー・プローブ応答などがある。コンピューティングシステムの複雑性が増すと、システムコンポーネント間で結合及び通信する相互接続ファブリックの複雑性も増す。ラベル及びルートを静的に割り当てることは、通常、実行不可能であり、ファブリックトポロジの規則性について仮定することも不可能である。したがって、不規則なファブリックトポロジを管理するための改善された技術が求められている。

【0003】

添付図面と共に以下の説明を参照することによって、本明細書に記載される方法及びメカニズムの利点はより良好に理解され得る。

【図面の簡単な説明】

【0004】

【図1】コンピューティングシステムの一実施形態のブロック図である。

【図2】ファブリックの一実施形態のブロック図である。

【図3】コンピューティングシステムの別の実施形態のブロック図である。

【図4】ファブリックに結合されたセキュリティプロセッサの一実施形態のブロック図である。

【図5】ネットワークグラフの一実施形態のブロック図である。

【図6】ファブリックコンポーネントの一実施形態のブロック図である。

【図7】ファブリックトポロジの自動発見プロセスを実行するための方法の一実施形態を示す一般化されたフロー図である。

【発明を実施するための形態】

【0005】

以下の説明では、本明細書に提示する方法及びメカニズムの完全な理解を提供するために、多くの具体的な詳細が記載されている。しかしながら、当業者は、それらの特定の詳細がなくとも、様々な実施形態を実施可能であることを認識するべきである。いくつかの例では、本明細書に記載される手法を不明瞭にしないために、周知の構造、コンポーネント、信号、コンピュータプログラム命令及び技術を詳細には示さない。例示を簡潔及び明確にするために、図面に示される要素は、必ずしも縮尺通りに描写されるわけではないことが認識されるだろう。例えば、要素のいくつかの寸法は、他の要素に対して拡張されてよい。

【0006】

本明細書では、相互接続ファブリックトポロジの自己識別発見プロセスを実装するための様々なシステム、装置、方法及びコンピュータ可読媒体を開示する。一実施形態では、システムは、少なくとも1つまたは複数の処理ユニット、1つまたは複数の入出力（I/O）インターフェース、不明トポロジの相互接続ファブリック、及び1つまたは複数のメモリデバイスを含む。一実施形態では、システムは、任意のファブリックトポロジで動作することができる自律的なブートストラップ発見プロセスを実施する。異なるファブリックシステムのプログラミングを簡素化するために、ファブリックが複数のノード（ダイまたはソケット）にまたがる場合、任意のネットワークオンチップ（NoC）の発見及びプログラミングに対する一般的な解決法が実施される。一実施形態では、情報レジスタのセット及びアルゴリズムを実装してネットワークブロックを横断し、ブロックタイプ及び機能ならびにネットワークコンポーネント間の接続性を発見する。

【0007】

一実施形態では、ファブリックの発見は、セキュリティプロセッサによって実行される。別の実施形態では、ファブリックの発見は、システム管理プロセッサによって実行される。他の実施形態では、ファブリックの発見は、他のタイプのコンポーネントによって実行することができる。一実施形態では、ファブリックの発見は、ファブリックを発見する際に取得された情報を使用してシステムハードウェアにリセット条件を作成する場合には、設計フローの一部として実行される。一実施形態では、ファブリックの発見は、セキュ

10

20

30

40

50

リティブロセッサがファブリックブロック・インスタンスカウントレジスタを読み取ることによって開始する。ファブリックブロック・インスタンスカウントレジスタは、システム内のファブリックブロックの合計数の表示を記憶し、ファブリックブロック・インスタンスカウントレジスタは、システム内の固定アドレスに存在する。ファブリックブロック・インスタンスカウントレジスタを使用すると、システムのサイズまたはトポロジに関係なく、ファームウェアは一般的な開始点を有することができる。

【0008】

ファブリックブロックの合計数がファブリックブロック・インスタンスカウントレジスタから取得されると、セキュリティプロセッサは各ブロックを通過してブロックインスタンス情報レジスタを読み取る。これらのレジスタには、ブロックタイプ（例えば、コヒーレントマスタ、非コヒーレントマスタ、クロスバー、コヒーレントスレーブ、非コヒーレントスレーブ）、コマンド及びデータポートのタイプ及び数、データバス幅、クロック速度、ネイバーインスタンス識別子（ID）、ファブリックID及びその他の属性などの情報が含まれる。この情報が各ブロックについて取得されると、以降の処理で使用するデータ構造としてネットワークグラフが構築される。ネットワークグラフに基づいてルーティングアルゴリズムを実行し、NoC及びマルチノードシステムに必要なルーティングを決定することができる。ルート、ブロックタイプ及び機能が発見されると、ファームウェアはファブリックを初期化するために、マスタ、スレーブ及びスイッチの適切なプログラミングを開始する。

【0009】

ここで図1を参照すると、コンピューティングシステム100の一実施形態のブロック図が示されている。一実施形態では、コンピューティングシステム100は、少なくとも処理ユニット110A~B、ファブリック115、入出力（I/O）インターフェース120、メモリコントローラ130（複数可）、ディスプレイコントローラ135、他のデバイス140（複数可）及びセキュリティプロセッサ145を含む。他の実施形態では、コンピューティングシステム100は、他のコンポーネントを含むことができ、及び/またはコンピューティングシステム100は異なる方式で配置することができる。処理ユニット110A~Bは、任意の数及びタイプの処理ユニットを表す。例えば、一実施形態では、処理ユニット110Aは中央処理ユニット（CPU）であり、処理ユニット110Bはグラフィック処理ユニット（GPU）である。別の実施形態では、処理ユニット110A~Bは、他の数またはタイプの処理ユニット（例えば、デジタルシグナルプロセッサ（DSP）、フィールドプログラマブルゲートアレイ（FPGA）、特定用途向け集積回路（ASIC））を含むことができる。

【0010】

ファブリック115は、システム100のコンポーネント間で通信するために利用される、様々なタイプのプロトコルのいずれかによる任意の通信相互接続を表す。ファブリック115は、処理ユニット110A~B、I/Oインターフェース120、メモリコントローラ130（複数可）、ディスプレイコントローラ135及び他のデバイス（複数可）140を互いに接続するデータバス、スイッチ、ルータ及び他のロジックを提供する。ファブリック115は、要求、応答及びデータトラフィックを処理するだけでなく、トラフィックをプローブしてコヒーレンスを容易にする。ファブリック115は、割り込み要求ルーティング及びシステム100の様々なコンポーネントへの構成アクセスパスも処理する。加えて、ファブリック115は、構成要求、応答及び構成データトラフィックを処理する。ファブリック115は、共有バス構成、クロスバー構成及びブリッジを備えた階層バスを含む、バスベースとすることができる。ファブリック115は、パケットベースにすることもでき、ブリッジ、クロスバー、ポイントツーポイントまたは他の相互接続で階層化することができる。ファブリック115の観点から、システム100の他のコンポーネントは「クライアント」と呼ばれることがある。ファブリック115は、様々なクライアントによって生成された要求を処理し、その要求を他のクライアントに渡すように構成される。

10

20

30

40

50

【 0 0 1 1 】

一実施形態では、セキュリティプロセッサ 1 4 5 は、ファブリック 1 1 5 のコンポーネントのトポロジの自律的な発見を開始するように構成される。これは、システム 1 0 0 が通常の構成を有しておらず、システム 1 0 0 の異なる実装内に様々な異なるタイプの機能が含まれる可能性がある状況において役立つことがある。例えば、システム 1 0 0 の 1 つの実装では、コアあたり 1 6 のプロセッサコア及び 2 つのメモリチャネルを含むことがあり、システム 1 0 0 の第 2 の実装では、コアあたり 8 つのプロセッサコア及び 1 つのメモリチャネルを含むことがある。システム 1 0 0 の他の実装は、他の数のコア、メモリチャネル、メモリコントローラ、メモリデバイスなどを含むことがある。したがって、システムの起動前にファブリック 1 1 5 の構成及び初期化を行う機能は制限される。したがって、これらの状況では、セキュリティプロセッサ 1 4 5 は、システム 1 0 0 の最初の起動中にシステム 1 0 0 の所与の実装の固有トポロジを発見するように構成される。

10

【 0 0 1 2 】

一実施形態では、セキュリティプロセッサ 1 4 5 は、コンポーネントに関連する様々な属性を取得するために、ファブリック 1 1 5 の各コンポーネントに問い合わせを行うように構成される。一実施形態では、ファブリック 1 1 5 の各コンポーネントは、それぞれのコンポーネントの属性を指定する値を格納するための 1 つまたは複数のレジスタを含む。次いで、ファブリック 1 1 5 のコンポーネントに問い合わせを行った後、セキュリティプロセッサ 1 4 5 は、様々なコンポーネントから取得された属性に基づいて、ネットワークグラフを作成するように構成される。次に、セキュリティプロセッサ 1 4 5 は、決定されたルート、ブロックタイプ及び機能に基づいて、ファブリックを初期化するために、マスタ、スレーブ及びスイッチをプログラムするように構成される。例えば、一実施形態では、ネットワークグラフを作成した後、セキュリティプロセッサ 1 4 5 は、ネットワークグラフからルーティングテーブルを生成するように構成される。この実施形態では、セキュリティプロセッサ 1 4 5 は、ルーティングテーブルをファブリック 1 1 5 の様々なコンポーネントにプログラムする。次いで、ファブリック 1 1 5 のコンポーネントは、プログラムされたルーティングテーブルを利用して、受信パケットをルーティングする方法を決定する。

20

【 0 0 1 3 】

一実施形態では、セキュリティプロセッサ 1 4 5 はさらに、システム 1 0 0 の構成及びセキュリティを管理するように構成される。セキュリティプロセッサ 1 4 5 は、システム 1 0 0 にセキュリティ保護を提供する認証及び検証機能を行うための命令を実行するように構成される。さらに、セキュリティプロセッサ 1 4 5 は、システム 1 0 0 の他の部分へのアクセスを不可能にする 1 つまたは複数の一意の暗号化 / 復号キーを格納する。したがって、セキュリティプロセッサ 1 4 5 は、システム 1 0 0 にハードウェアベースの信頼の基点を提供し、システム 1 0 0 が安全な環境で起動することを可能にする。一実施形態では、セキュリティプロセッサ 1 4 5 は、システム 1 0 0 の起動プロセスを管理して、システム 1 0 0 が認証された起動コードで起動することを確実にする。セキュリティプロセッサ 1 4 5 はさらに、システム 1 0 0 の起動プロセスに関連する他の様々な機能を管理する。次いで、セキュリティプロセッサ 1 4 5 は、処理ユニット 1 1 0 A ~ N を解放して、ブートコードを実行し、システム 1 0 0 のオペレーティングシステムを起動する。

30

40

【 0 0 1 4 】

メモリコントローラ 1 3 0 (複数可) は、任意の数及びタイプのメモリデバイス (複数可) に結合することができる任意の数及びタイプのメモリコントローラを表す。例えば、メモリコントローラ 1 3 0 (複数可) に結合されるメモリデバイス (複数可) のタイプは、ダイナミックランダムアクセスメモリ (DRAM)、スタティックランダムアクセスメモリ (SRAM)、NANDフラッシュメモリ、NORフラッシュメモリ、強誘電体ランダムアクセスメモリ (FeRAM) またはその他を含むことができる。メモリコントローラ 1 3 0 (複数可) は、ファブリック 1 1 5 を介して、処理ユニット 1 1 0 A ~ B、I/O インターフェース 1 2 0、ディスプレイコントローラ 1 3 5 及び他のデバイス 1 4 0 (

50

複数可)にアクセス可能である。I/Oインターフェース120は、任意の数及びタイプのI/Oインターフェース(例えば、周辺コンポーネント相互接続(PCI)バス、PCI-Extended(PCI-X)、PCI Express)バス、ギガビットイーサネット(GBE)バス、ユニバーサルシリアルバス(USB))を表す。様々なタイプの周辺デバイスは、I/Oインターフェース120に結合することができる。係る周辺デバイスは、ディスプレイ、キーボード、マウス、プリンタ、スキャナ、ジョイスティックまたは他のタイプのゲームコントローラ、メディア記録デバイス、外部記憶装置、ネットワークインターフェースカードなどを含む(しかしながら、それらに限定されない)。他のデバイス140(複数可)は、任意の数及びタイプのデバイス(例えば、マルチメディアデバイス、ビデオコーデック)を表す。

10

【0015】

様々な実施形態では、コンピューティングシステム100は、コンピュータ、ラップトップ、モバイルデバイス、サーバまたは様々な他のタイプのコンピューティングシステムもしくはデバイスのいずれかであることができる。コンピューティングシステム100のコンポーネントの数は、実施形態によって変更することがあることに留意されたい。各々のコンポーネントは、図1に示される数よりも多いまたは少ない場合がある。さらに、コンピューティングシステム100は、図1に示されていない他のコンポーネントを含むことがあることに留意されたい。加えて、他の実施形態では、コンピューティングシステム100は、図1に示された以外の方式で構造化することができる。

【0016】

20

ここで図2に注目すると、ファブリック200の一実施形態のブロック図が示される。一実施形態では、ファブリック200は、(図1の)コンピューティングシステム100のファブリック115として実装することができる。一実施形態では、ファブリック200は、不規則なトポロジに配置された複数の異なるコンポーネントを含む。いくつかの実施形態では、ファブリック200は複数の領域に分割され、領域ごとに任意の数のコンポーネントがあり、各領域には任意のタイプのトポロジがある。一実施形態では、ファブリック200の各領域は、異なるパワーゲーティングドメインに関連付けられている。

【0017】

図2の左側では、ファブリック200は、クロスバー210に結合されたマスタ205ならびにスレーブ215及び220を含む。本明細書で使用される場合、「マスタ」は、要求を生成するコンポーネントとして定義され、「スレーブ」は、要求を提供するコンポーネントとして定義される。クロスバー210はクロスバー225に結合され、クロスバー225はマスタ230A~Bに結合される。クロスバー225はクロスバー245に結合され、クロスバー245はクロスバー250及び260ならびにマスタ255Aに結合される。クロスバー250はマスタ255B~Cに結合され、クロスバー260はマスタ265A~B及びスレーブ265Cに結合される。図2の右端では、ファブリック200は、スレーブ275及びマスタ280に結合されたクロスバー270を含む。一実施形態では、スレーブ275は、メモリコントローラに結合される。

30

【0018】

一実施形態では、ファブリック200の各コンポーネントは、それぞれのコンポーネントに関する情報を指定する属性を格納するためのレジスタを含む。例えば、一実施形態では、レジスタは、コンポーネントのタイプ、コマンド及びデータポートのタイプ及び数、データバス幅、クロック速度、ネイバーインスタンスID、ファブリックIDなどの情報、ならびに他の情報を格納する。一実施形態では、セキュリティプロセッサ(例えば、図1のセキュリティプロセッサ145)は、対応するレジスタに格納された情報を取得するためにファブリック200の各コンポーネントに問い合わせを行うように構成される。次いで、セキュリティプロセッサはこの情報を利用して、ファブリック200のネットワークグラフ及びネットワークグラフに基づいたルーティングテーブルを作成することができる。

40

【0019】

50

ファブリック 200 のコンポーネントのレイアウト、数及びタイプは、1つの実施形態を示すにすぎないことに留意されたい。他の実施形態では、ファブリック 200 は、異なった方式で配置することができる、及び/または他の数及びタイプのコンポーネントを含むことができる。さらに、ファブリック 200 の一部である様々なクロスバー 210、225、245、250、260 及び 270 は、スイッチまたはルータと呼ぶこともできることに留意されたい。クロスバーには、スイッチ接続と、クロスバーが受信した要求をバッファリングするバッファリングを含めることができる。クロスバーには、要求を送信元から宛先に送信するためのフロー制御メカニズムを実施するためのクレジット制御ロジックを含めることもできる。各クロスバーによって、クロスバーに結合された任意の送信元から宛先のいずれかへの接続を要求することができる。各クロスバーには、他のクロスバーへの1つまたは複数の接続も含まれ、それぞれのクロスバーに直接的には結合されていない他の宛先への要求を送信することができる。

10

【0020】

ここで図3を参照すると、コンピューティングシステム300の別の実施形態のブロック図が示される。システム300は、少なくともセキュリティプロセッサ305、ファブリックブロック・インスタンスカウントレジスタ310、ファブリック315、処理ユニット335A~N、I/Oインターフェース340、メモリコントローラ345(複数可)及び図を不明瞭しないために示されていない任意の数の他のコンポーネントを含む。一実施形態では、システム300の電源を最初に投入すると、システム300にはファブリック315のトポロジまたは構造の知識が全くなく、システム300は、起動プロセスの一部としてこのトポロジを発見するように構成される。その結果、ファブリックトポロジの発見は、同じ基本コンポーネントを使用して、様々なネットワークトポロジ及びリソースを備えた異なるタイプのシステムを構築する柔軟性を提供する。

20

【0021】

一実施形態では、セキュリティプロセッサ305は、ファブリックブロック・インスタンスカウントレジスタ310に問い合わせを行い、ファブリック315のコンポーネント320A~Nの合計数の表示を取得するように構成される。次いで、セキュリティプロセッサ305は、ファブリック315のコンポーネント320A~Nの自動発見プロセスを開始する。セキュリティプロセッサ305は、ソフトウェア、ハードウェア及び/またはファームウェアの任意の適切な組み合わせを使用して実装することができる。一実施形態では、セキュリティプロセッサ305は、システム300の様々なセキュリティ機能を実行するように構成された専用のマイクロプロセッサである。例えば、この実施形態では、セキュリティプロセッサ305は、システム300にハードウェアベースの信頼の基点を提供するように構成される。

30

【0022】

コンポーネント320A~Nは、ファブリック315内の任意のタイプのトポロジに配置された任意の数及びタイプのコンポーネントを表す。例えば、コンポーネント320A~Nは、クロスバー、スイッチ、ルータ、非コヒーレントマスタ、コヒーレントマスタ、非コヒーレントスレーブ、コヒーレントスレーブなどを含むことができる。コンポーネント320A~Nは、本明細書では「ブロック」と呼ばれることもあることに留意されたい。一実施形態では、各コンポーネント320A~Nは、それぞれのコンポーネントの様々なメタデータを格納する対応するレジスタ325A~Nを有する。各レジスタ325A~Nは、任意の数の属性を格納するための任意の数のレジスタまたは他のタイプの格納場所を表す。例えば、各レジスタ325A~Nは、コンポーネントのタイプ、コマンド及びデータポートのタイプ及び数、データバス幅、クロック速度、ネイバーインスタンスID、ファブリックIDなどを指定することができる。

40

【0023】

一実施形態では、セキュリティプロセッサ305は、ファブリック315のコンポーネント320A~Nを一度に1つずつ横断するように構成される。一実施形態では、セキュリティプロセッサ305は、主要なデータパスを介して、ファブリック315のコンポー

50

ネット 3 2 0 A ~ N に結合される。別の実施形態では、セキュリティプロセッサ 3 0 5 は、パケットがファブリック 3 1 5 を通過するときに横断する主要なデータパス接続とは異なるサイドバンド接続を介してファブリック 3 1 5 のコンポーネント 3 2 0 A ~ N に結合される。自動発見プロセスを開始するために、セキュリティプロセッサ 3 0 5 は、セキュリティプロセッサ 3 0 5 に最も近いコンポーネント 3 2 0 A のレジスタ 3 2 5 A に問い合わせることによって開始する。次いで、セキュリティプロセッサ 3 0 5 は、コンポーネント 3 2 0 A のネイバーに問い合わせることにより継続する。一実施形態では、各コンポーネント 3 2 0 A ~ N には、ファブリック 3 1 5 内で一意の ID が割り当てられる。

【 0 0 2 4 】

セキュリティプロセッサ 3 0 5 は、ファブリック 3 1 5 の末端に到達するまで、ネイバーのネイバーなどを經由してファブリック 3 1 5 を横断し続ける。セキュリティプロセッサ 3 0 5 がコンポーネント 3 2 0 A ~ N の全てのレジスタ 3 2 5 A ~ N からメタデータを取得した後、セキュリティプロセッサ 3 0 5 は、発見されたコンポーネントのネットワークグラフを構築するように構成される。次いで、セキュリティプロセッサ 3 0 5 は、ネットワークグラフを利用して、各コンポーネント 3 2 0 A ~ N のルーティングテーブルを生成する。ルーティングテーブルを生成した後、セキュリティプロセッサ 3 0 5 は、受信パケットをルーティングする方法を決定するときに使用するために、ルーティングテーブルを各コンポーネント 3 2 0 A ~ N に伝達する。

【 0 0 2 5 】

ここで図 4 を参照すると、ファブリック 4 1 5 に結合されたセキュリティプロセッサ 4 0 5 の一実施形態のブロック図が示されている。セキュリティプロセッサ 4 0 5 は、図 3 に関する説明において既に記載したように、ファブリック発見プロセス中にファブリック 4 1 5 のコンポーネント 4 2 0 A ~ N のレジスタ 4 2 5 A ~ N からコンポーネント属性 4 0 8 を取得するように構成される。コンポーネント属性 4 0 8 を取得した後、セキュリティプロセッサ 4 0 5 は、ネットワークグラフ 4 1 0 を作成して、コンポーネント及びファブリック 4 1 5 のコンポーネント相互接続のトポロジを表すように構成される。

【 0 0 2 6 】

一実施形態では、セキュリティプロセッサ 4 0 5 は、ネットワークグラフ 4 1 0 を利用して、ルーティングテーブル 4 3 0 A ~ N を生成する。例えば、セキュリティプロセッサ 4 0 5 は、ネットワークグラフ 4 1 0 の所与のノードで開始することによってネットワークグラフ 4 1 0 を横断し、ここで所与のノードは所与のコンポーネントを表す。次いで、セキュリティプロセッサ 4 0 5 は、所与のノードから発生する接続を追跡しながらリーフノードに到達するまで、所与のノードからネットワークグラフ 4 1 0 を横断する。セキュリティプロセッサ 4 0 5 は、所与のノードからリーフノードへの各横断パスを利用して、所与のノードのルーティングテーブルを構築する。ルーティングテーブルを構築した後、セキュリティプロセッサ 4 0 5 は、このルーティングテーブルを用いて所与のコンポーネントをプログラムする。次いで、セキュリティプロセッサ 4 0 5 は、ネットワークグラフ 4 1 0 の他のノードに対して同様のプロセスを実行することができる。

【 0 0 2 7 】

例えば、セキュリティプロセッサ 4 0 5 は、ネットワークグラフ 4 1 0 を利用して、コンポーネント 4 2 0 A のルーティングテーブル 4 3 0 A を生成し、次いで、セキュリティプロセッサ 4 0 5 は、ルーティングテーブル 4 3 0 A をコンポーネント 4 2 0 A にプログラムする。ファブリック 4 1 5 を含むホストシステムの動作中、コンポーネント 4 2 0 A は、ルーティングテーブル 4 3 0 A を利用して、着信パケットをルーティングする方法を決定する。同様に、セキュリティプロセッサ 4 0 5 は、ルーティングテーブル 4 3 0 B ~ N をコンポーネント 4 2 0 B ~ N にそれぞれプログラムし、次いでコンポーネント 4 2 0 B ~ N は、それらの対応するルーティングテーブル 4 3 0 B ~ N を利用して、ホストシステムの実際の動作中に受信パケットをルーティングする方法を決定する。ファブリック 4 1 5 にはルーティングテーブルが事前にプログラムされているわけではなく、ホストシステムの初期化中に実施される自己発見プロセス中にルーティングテーブル 4 3 0 A ~ N が

10

20

30

40

50

生成されることを理解すべきである。

【 0 0 2 8 】

ここで図 5 に注目すると、ネットワークグラフ 5 0 0 の一実施形態のブロック図が示されている。ネットワークグラフ 5 0 0 は、自動発見プロセス中に検出されたファブリックのコンポーネントを表すノードを含む。一実施形態では、ファブリックのコンポーネントは、ネットワークグラフ 5 0 0 の相互作用ノードとしてモデル化される。セキュリティプロセッサ（例えば、図 3 のセキュリティプロセッサ 3 0 5 ）は、ファブリックの様々なコンポーネントの検出及び問い合わせを行い、これらのコンポーネントに関連する属性を取得するように構成される。一実施形態では、セキュリティプロセッサは、ファブリックコンポーネントの属性をテーブルのエントリに格納し、ここで各コンポーネントには個別のエントリがあり、各属性には個別の列がある。次いで、セキュリティプロセッサは、テーブルのエントリ内の取得された属性に基づいて、ネットワークグラフ 5 0 0 を作成するように構成される。

10

【 0 0 2 9 】

ネットワークグラフ 5 0 0 は、所与の相互接続ファブリックのコンポーネントを表すことを意図している。ネットワークグラフ 5 0 0 の各ノードは、発見プロセス中に発見されたファブリックのコンポーネントを表す。例えば、図 5 に示すように、ネットワークグラフは、コンポーネント 5 1 0、5 1 5 及び 5 2 0 に結合されたコンポーネント 5 0 5、コンポーネント 5 0 5、5 1 0、5 1 5 及び 5 3 5 に結合されたコンポーネント 5 2 0、コンポーネント 5 1 5 に結合されたコンポーネント 5 3 0、コンポーネント 5 1 5 に結合されたコンポーネント 5 2 5、コンポーネント 5 2 0 に結合されたコンポーネント 5 3 5 ならびにコンポーネント 5 1 0 に結合されたコンポーネント 5 4 0 を含む。ネットワークグラフ 5 0 0 を作成した後、セキュリティプロセッサはネットワークグラフ 5 0 0 を利用して、ファブリックの各コンポーネントのルーティングテーブルを生成する。次いで、セキュリティプロセッサはルーティングテーブルをコンポーネントにプログラムし、受信パケットのルーティング方法を決定するときにコンポーネントがルーティングテーブルを利用できるようにする。ネットワークグラフ 5 0 0 は、ネットワークグラフの単に 1 つの表現であることに留意されたい。他の実施形態では、セキュリティプロセッサによって他のタイプのネットワークグラフを作成することができる。

20

【 0 0 3 0 】

ここで図 6 に注目すると、ファブリックコンポーネント 6 0 0 の一実施形態のブロック図が示されている。一実施形態では、ファブリックコンポーネント 6 0 0 は、入力バッファ 6 0 5、クロスバー 6 1 5、出力バッファ 6 2 0、制御ユニット 6 3 0 及び 1 つまたは複数のルーティングテーブル 6 3 5 を含む。入力バッファ 6 0 5 は、1 つまたは複数の入力ポート 6 1 0 A ~ N に結合される。各入力ポート 6 1 0 A ~ N は、ファブリックの内部または外部の対応するコンポーネントに結合され、入力バッファ 6 0 5 は、入力ポート 6 1 0 A ~ N を介して受信されたメッセージデータをバッファするように構成される。同様に、出力バッファ 6 2 0 は、1 つまたは複数の出力ポート 6 2 5 A ~ N に接続されている。各出力ポート 6 2 5 A ~ N は、ファブリックの内部または外部の対応するコンポーネントに結合され、出力バッファ 6 2 0 は、対応するリンクのクロスバー 6 1 5 から受信されたメッセージデータをバッファするように構成される。一実施形態では、（図 3 の）コンポーネント 3 2 0 A ~ N のそれぞれは、ファブリックコンポーネント 6 0 0 のロジックを含む。ファブリックコンポーネント 6 0 0 におけるロジック構成は、1 つの特定の実施形態を表すことを理解すべきである。他の実施形態では、他の適切なロジック構成を利用することができる、及び/または他のロジックをファブリックコンポーネント 6 0 0 に含めることができる。

30

40

【 0 0 3 1 】

一実施形態では、クロスバー 6 1 5 は、制御ユニット 6 3 0 によって提供される制御シグナリングに基づいて、入力ポート 6 1 0 A ~ N から出力ポート 6 2 5 A ~ N を通過するパケットを切り替えるマルチプレクサを含む。一実施形態では、制御ユニット 6 3 0 は、

50

ソフトウェアによって指定された構成パラメータ及び1つまたは複数のルーティングテーブル635で表されるルーティング情報を利用して、クロスバー615を制御し、入力ポート610A~Nから出力ポート625A~Nへの入力メッセージデータの特定のルーティングを行う。制御ユニット630は、着信メッセージヘッダを検査し、ルーティングテーブル635のルックアップを実行して次のホップを決定し、クロスバー615を制御して、データを適切な出力ポート625A~Nに転送する。制御ユニット630はさらに、仮想チャンネルを管理し、ソフトウェアによって提供される構成データごとにアービトレーション及びフィルタリングを実装し、別様には、1つまたは複数のルーティングプロトコルのコンポーネントを実装する。

【0032】

様々な実施形態では、ルーティングテーブル635は、ファブリックコンポーネント600を通過するパケットのルーティング情報を提供する。一実施形態では、ルーティングテーブル635は、複数のテーブルエントリとして実装され、各エントリは、送信元コンポーネントと宛先コンポーネントとの間の指定されたルーティングパスに従って、対応する宛先、ネクストホップ及びポートフィールドに関連付けられる。他の実施形態では、ルーティングテーブル635のエントリはさらに、代替ルート、パスの長さ、リンクエラーステータスなど、制御ユニット630によって使用される他のフィールドを含むことができる。ネクストホップフィールドは、マルチホップルートの次のコンポーネントのアドレスとすることができ、あるいはネクストホップが最終的な宛先である場合は宛先と同じにすることができる。様々な実施形態では、テーブル635のエントリは、自律的な発見プロセス中の実行時に決定される。

【0033】

したがって、テーブル635は、様々なルートを実装するために、またはダイモしくはネットワークトポロジの異なる数または配置の再構成を可能にするために、書き込み可能か、別様にはプログラム可能である。様々な実施形態では、テーブル635は、RAMまたはフラッシュメモリなどにレジスタファイルなどの多様な構成可能ストレージ要素のいずれかを使用して実装することができる。これらの構成可能要素及び書き込み可能なテーブルエントリは、オペレーティングシステム、ハイパーバイザ、セキュリティプロセッサ、基本的な入出力システム(BIOS)、ファームウェアまたはそれらの組み合わせを含む、多数の要素によって管理することができる。一例として、システムの起動中に、セキュリティプロセッサ(例えば、図3のセキュリティプロセッサ305)は、ファブリック全体のトポロジ、コンポーネントの数及びシステム相互接続を発見した後、ルーティングテーブル635をプログラムすることができる。一実施形態では、物理システムの異なるバージョンと実装との間で変化する様々な数のコンポーネント及び接続によりファブリックのトポロジは事前には分からない。一実施形態では、セキュリティプロセッサは、クロスバー及び相互接続トポロジを検査して、実装されたファブリックトポロジを発見するように構成される。セキュリティプロセッサはさらに、コンポーネントを再構成して、ネットワーク及びメモリの帯域幅保証などのサービス品質ポリシーを実施することができる。

【0034】

ここで図7を参照すると、ファブリックトポロジの自動発見プロセスを実行するための方法700の一実施形態が示されている。説明の目的上、本実施形態のステップは順番に示される。しかしながら、記載される方法の様々な実施形態では、記載される要素のうちの1つまたは複数とは異なる順序で実行されるか、または全体が省略されることに留意されたい。他の追加の要素も所望により実行されることがある。本明細書に記載される様々なシステムまたは装置のいずれも、方法700を実施するように構成されている。

【0035】

プロセッサ(例えば、図3のセキュリティプロセッサ305)は、ファブリックブロック・インスタンスカウントレジスタ(例えば、ファブリックブロック・インスタンスカウントレジスタ310)に問い合わせを行い、相互接続ファブリックのコンポーネントの数

10

20

30

40

50

を決定する（ブロック705）。次に、プロセッサは、複数のコンポーネントの各コンポーネントに問い合わせを行い、各コンポーネントに関連する様々な属性を対応するレジスタから取得する（ブロック710）。様々な実施形態では、属性は、ブロックタイプ（例えば、コヒーレントマスタ、非コヒーレントマスタ、クロスバー、コヒーレントスレーブ、非コヒーレントスレーブ）、コマンド及びデータポートのタイプ及び数、データバス幅、クロック速度、ネイバーインスタンス識別子（ID）、ファブリックID及びその他の属性を含むことができる。

【0036】

次いで、プロセッサは、取得された属性を利用して、ファブリックコンポーネントのネットワークグラフを作成する（ブロック715）。次に、プロセッサはネットワークグラフからルーティングテーブルを生成する（ブロック720）。次いで、プロセッサはルーティングテーブルをファブリックコンポーネントにプログラムする（ブロック725）。ルーティングテーブルでプログラムされた後、ファブリックコンポーネントはルーティングテーブルを利用して、着信パケットをルーティングする方法を決定する（ブロック730）。ブロック730の後に、方法700は終了する。

【0037】

様々な実施形態では、ソフトウェアアプリケーションのプログラム命令は、本明細書に記載される方法及び/またはメカニズムを実装するために使用される。例えば、汎用または専用プロセッサによって実行可能なプログラム命令が企図される。様々な実施形態では、係るプログラム命令は、高レベルプログラミング言語として表すことができる。他の実施形態では、プログラム命令は、高レベルプログラミング言語から、バイナリ、中間または他の形式にコンパイルすることができる。代替的には、ハードウェアの挙動または設計を記述したプログラム命令を書き込むことができる。係るプログラム命令は、Cなどの高レベルプログラミング言語によって表すことができる。代替的には、Verilogなどのハードウェア設計言語（HDL）を使用することができる。様々な実施形態では、プログラム命令は、様々な非一時的コンピュータ可読記憶媒体のいずれかに記憶される。記憶媒体は、プログラム実行のためにプログラム命令をコンピューティングシステムに提供するために使用される間にコンピューティングシステムによってアクセス可能である。一般的に、係るコンピューティングシステムは、プログラム命令を実行するように構成された少なくとも1つまたは複数のメモリ及び1つまたは複数のプロセッサを含む。

【0038】

上記の実施形態は実装の非限定的な例にすぎないことが強調されるべきである。上記の開示を十分に理解すれば、当業者には多数の変形形態及び修正形態が明らかになるであろう。以下の特許請求の範囲は、全ての係る変形形態及び修正形態を包含すると解釈されることが意図される。

10

20

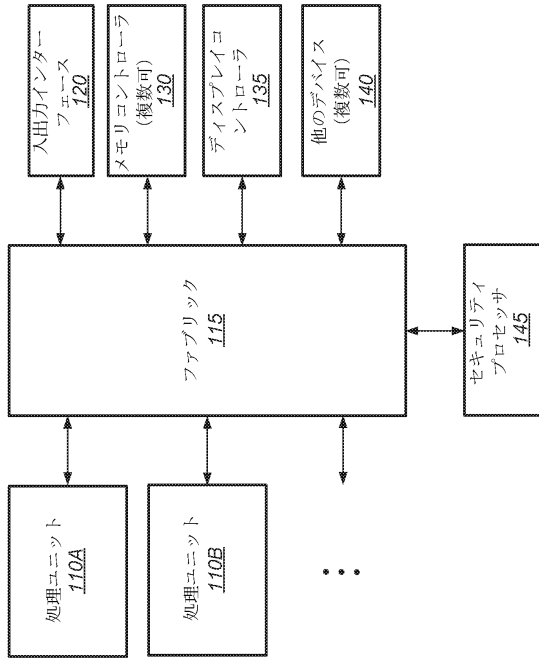
30

40

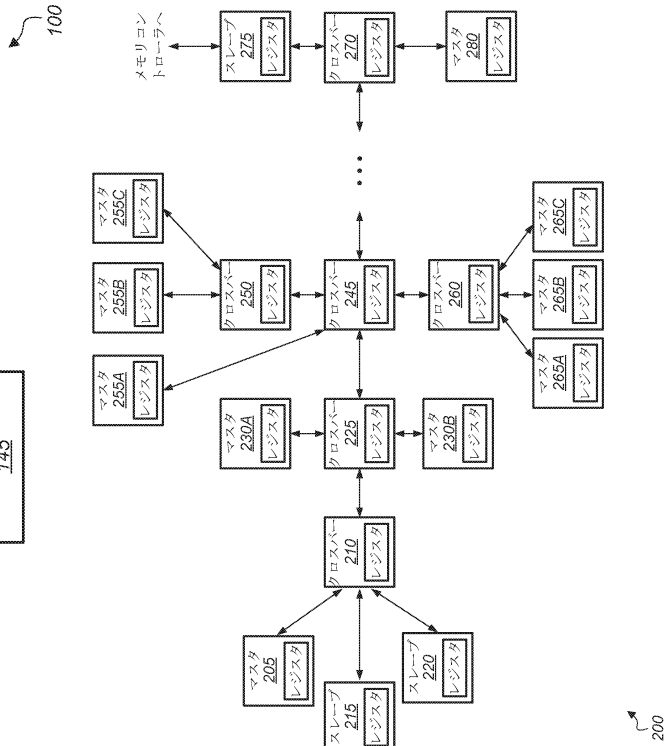
50

【図面】

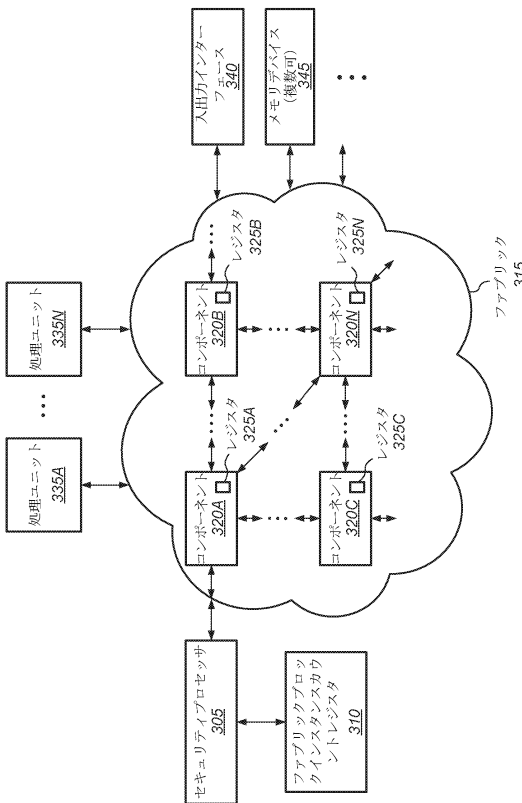
【図 1】



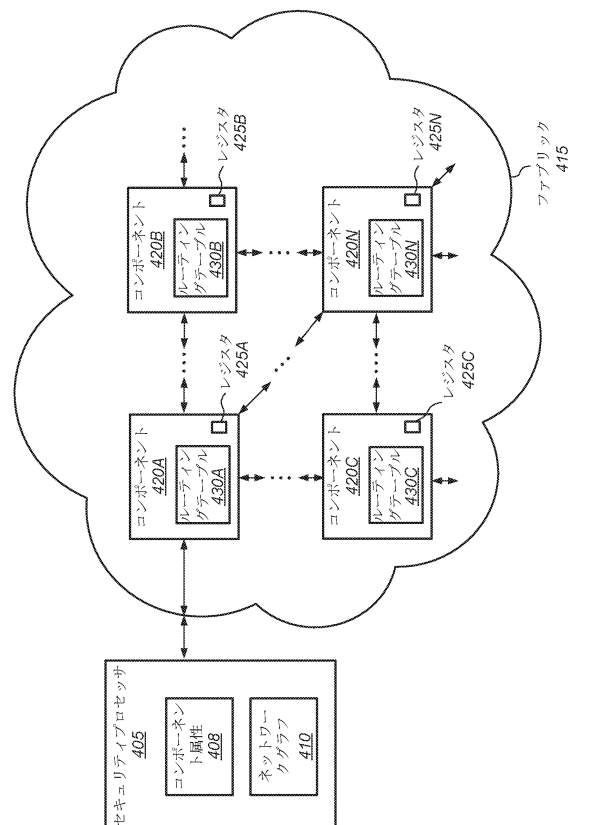
【図 2】



【図 3】



【図 4】



10

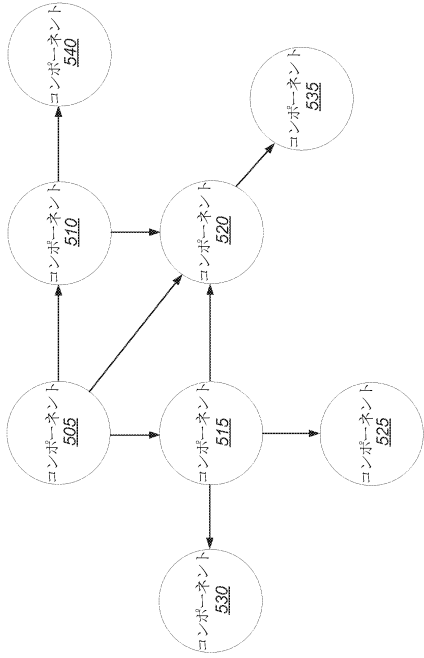
20

30

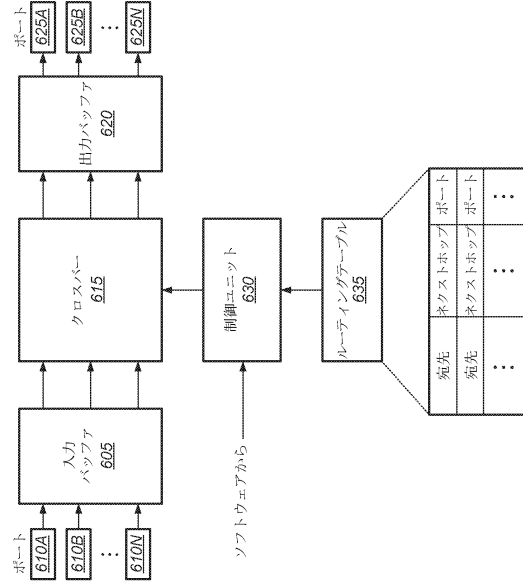
40

50

【図 5】



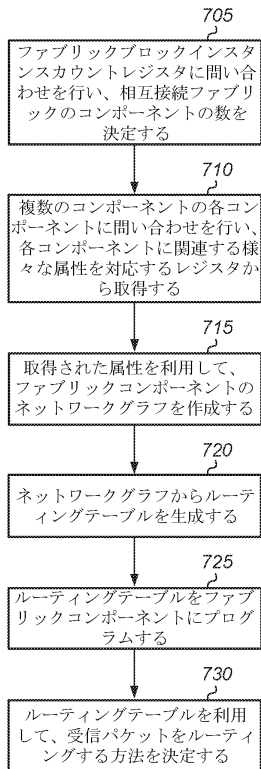
【図 6】



10

20

【図 7】



30

40

50

フロントページの続き

- アメリカ合衆国 9 5 1 3 8 カリフォルニア州、サンノゼ、ロングヒル コート 3 5 1
(72)発明者 エリック クリストファー モートン
アメリカ合衆国 7 8 7 3 2 テキサス州、オースティン、マジェスティック オークス ドライブ
1 2 9 0 1
(72)発明者 アラン ドッドソン スミス
アメリカ合衆国 7 8 7 0 3 テキサス州、オースティン、メリデン レーン 1 0 0 3
(72)発明者 ジョー ジー . クルツ
アメリカ合衆国 0 1 5 3 6 マサチューセッツ州、ノース グラフトン、ストーニーブルック ロ
ード 1 3
審査官 羽岡 さやか
(56)参考文献 米国特許出願公開第 2 0 0 9 / 0 0 1 6 3 5 5 (U S , A 1)
米国特許出願公開第 2 0 1 6 / 0 1 2 7 1 9 1 (U S , A 1)
米国特許出願公開第 2 0 1 6 / 0 9 4 4 3 5 (U S , A 1)
国際公開第 0 0 / 3 1 9 2 5 (W O , A 1)
米国特許出願公開第 2 0 0 6 / 0 1 6 5 1 1 5 (U S , A 1)
(58)調査した分野 (Int.Cl. , D B 名)
H 0 4 L 4 5 / 0 0 - 4 5 / 8 5 1