

(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号  
特許第7160682号  
(P7160682)

(45)発行日 令和4年10月25日(2022.10.25)

(24)登録日 令和4年10月17日(2022.10.17)

(51)国際特許分類 F I  
G 0 6 F 12/0815(2016.01) G 0 6 F 12/0815  
G 0 6 F 12/0817(2016.01) G 0 6 F 12/0817

請求項の数 15 (全20頁)

(21)出願番号	特願2018-555617(P2018-555617)	(73)特許権者	591016172
(86)(22)出願日	平成29年5月2日(2017.5.2)		アドバンスト・マイクロ・デバイス
(65)公表番号	特表2019-517687(P2019-517687 A)		・インコーポレイテッド
(43)公表日	令和1年6月24日(2019.6.24)		ADVANCED MICRO DEVI
(86)国際出願番号	PCT/US2017/030586		CES INCORPORATED
(87)国際公開番号	WO2017/209883		アメリカ合衆国 9 5 0 5 4 カリフォル
(87)国際公開日	平成29年12月7日(2017.12.7)		ニア州、 サンタ クララ、 オーガスティ
審査請求日	令和2年5月1日(2020.5.1)	(74)代理人	ン ドライブ 2 4 8 5
(31)優先権主張番号	15/169,118		100108833
(32)優先日	平成28年5月31日(2016.5.31)		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
			弁理士 佐野 良太
		(74)代理人	100162156
			弁理士 村雨 圭介
		(72)発明者	マイケル ダブリュ. ボイヤー
			最終頁に続く

(54)【発明の名称】 メモリにおける処理のためのキャッシュコヒーレンス

(57)【特許請求の範囲】

【請求項1】

第1キャッシュコヒーレンシブプロトコルを実行するプロセッサインメモリを含むシステムであって、

前記プロセッサインメモリは、

メモリダイと、

前記メモリダイに積層されたロジック集積回路ダイと、を備え、

前記ロジック集積回路ダイは、

前記メモリダイに存在するデータブロックのコピーを記憶する、対応するプロセッサの少なくとも1つのキャッシュと、

前記データブロックのプロセッサインメモリコヒーレンス状態を記憶するように構成されたプロセッサインメモリディレクトリであって、前記プロセッサインメモリコヒーレンス状態は、前記少なくとも1つのキャッシュ内の前記データブロックのコピーの状態を示す、プロセッサインメモリディレクトリと、

前記プロセッサインメモリディレクトリの内容を用いて、前記データブロックのデータコヒーレンスを維持するように構成されたプロセッサインメモリ内コヒーレンスコントローラと、

前記データブロックのブリッジコヒーレンス状態を記憶するように構成されたシャドウディレクトリストレージであって、前記ブリッジコヒーレンス状態は、前記プロセッサインメモリコヒーレンス状態とは別のものであり、前記プロセッサインメモリの外部の全て

のコンポーネントを単一の追加の潜在的な共有デバイスとして表し、前記プロセッサインメモリの外部に記憶された前記データブロックの外部コピーの状態を示す、シャドウディレクトリストレージと、

前記プロセッサインメモリに接続されたホストプロセッサのホスト内コヒーレンス制御ロジックからコヒーレンスメッセージを受信し、前記コヒーレンスメッセージを前記ホストプロセッサの第2キャッシュコヒーレンシプロトコルから前記第1キャッシュコヒーレンシプロトコルに変換し、変換されたコヒーレンスメッセージをブリッジプロトコル制御ロジックに提供するように構成されたブリッジプロトコル変換ロジックであって、前記第2キャッシュコヒーレンシプロトコルは、前記第1キャッシュコヒーレンシプロトコルと異なっている、ブリッジプロトコル変換ロジックと、

10

前記変換されたコヒーレンスメッセージと、前記シャドウディレクトリストレージの内容とを用いて、前記データブロックのデータコヒーレンスを維持するように構成されたブリッジプロトコル制御ロジックと、を備える、

システム。

【請求項2】

前記第2キャッシュコヒーレンシプロトコルは、前記第1キャッシュコヒーレンシプロトコルのブロックサイズと異なるブロックサイズを使用する、

請求項1のシステム。

【請求項3】

前記少なくとも1つのキャッシュは、前記プロセッサインメモリ内コヒーレンスコントローラによって維持される前記プロセッサインメモリのコヒーレンスドメイン内に存在する、

20

請求項1のシステム。

【請求項4】

前記少なくとも1つのキャッシュは、ライトスルーキャッシュであり、前記データブロックの前記ブリッジコヒーレンス状態は、no-host-copy状態、host-read状態、又は、host-written状態であり、

前記no-host-copy状態は、ブリッジコヒーレンスドメイン内のホストプロセッサが、前記データブロックの何れの部分もホストキャッシュに記憶していないことを示しており、前記host-read状態は、前記ブリッジコヒーレンスドメイン内の少なくとも1つのプロセッサが、前記データブロックの一部を読み出し、ホストキャッシュに前記一部の未修正のコピーを有する可能性があることを示しており、前記host-written状態は、前記ブリッジコヒーレンスドメイン内の少なくとも1つのプロセッサが、前記データブロックの排他的所有権を有しており、ホストキャッシュに前記データブロックを書き込んでいる可能性があることを示している、

30

請求項1、2又は3のシステム。

【請求項5】

前記少なくとも1つのキャッシュは、ライトバックキャッシュであり、前記データブロックの前記ブリッジコヒーレンス状態は、no-host-copy状態、host-read状態、host-written状態、又は、processor-in-memory-modified状態であり、

40

前記no-host-copy状態は、ブリッジコヒーレンスドメイン内のホストプロセッサが、前記データブロックの何れの部分もホストキャッシュに記憶していないことを示しており、前記host-read状態は、前記ブリッジコヒーレンスドメイン内の少なくとも1つのプロセッサが、前記データブロックの一部を読み出し、ホストキャッシュに前記一部の未修正のコピーを有する可能性があることを示しており、前記host-written状態は、前記ブリッジコヒーレンスドメイン内の少なくとも1つのプロセッサが、前記データブロックの排他的所有権を有しており、ホストキャッシュに前記データブロックを書き込んでいる可能性があることを示しており、前記processor-in-memory-modified状態は、前記データブロックの何れかの部分が、前

50

記プロセッサインメモリによって前記少なくとも1つのキャッシュにおいて変更されたかどうかを示している、

請求項 1、2 又は 3 のシステム。

【請求項 6】

前記プロセッサインメモリは、

前記プロセッサインメモリのコヒーレンスドメインの外部にあるメモリサイドキャッシュであって、前記ブリッジプロトコル制御ロジックは、前記データブロックへのメモリアクセスに応じて、前記メモリサイドキャッシュに記憶された前記データブロックの任意のコピーの状態をチェックするように構成されている、メモリサイドキャッシュをさらに備える、

10

請求項 1、2 又は 3 のシステム。

【請求項 7】

前記ホストプロセッサは、前記プロセッサインメモリから受信したコヒーレンスメッセージを前記第 1 キャッシュコヒーレンシプロトコルから前記第 2 キャッシュコヒーレンシプロトコルに変換するように構成されたホストブリッジ変換ロジックを含み、

前記システムは、

前記ホストプロセッサのコヒーレンスドメインの外側、且つ、前記プロセッサインメモリのコヒーレンスドメインの外側のメモリサイドキャッシュであって、前記シャドウディレクトリストレージは、前記メモリサイドキャッシュに記憶された前記データブロックの任意のコピーの状態を保存する、メモリサイドキャッシュと、

20

ホストコピーデータブロックのホストコヒーレンス状態を記憶するように構成されたホストディレクトリであって、ホストコピーは前記ホストプロセッサ内に記憶される、ホストディレクトリと、

前記ホストディレクトリを用いて、前記データブロックのホストコピーと前記プロセッサインメモリに存在するデータブロックとのデータコヒーレンスを維持するように構成された前記ホスト内コヒーレンス制御ロジックと、をさらに備える、

請求項 1、2 又は 3 のシステム。

【請求項 8】

第 1 のキャッシュコヒーレンシプロトコルを実行するプロセッサインメモリのメモリに存在するデータブロックのブリッジコヒーレンス状態をシャドウディレクトリに記憶するステップであって、前記ブリッジコヒーレンス状態は、前記データブロックのプロセッサインメモリコヒーレンス状態とは別のものであり、前記プロセッサインメモリの外部の全てのコンポーネントを単一の追加の潜在的な共有デバイスとして表し、前記プロセッサインメモリの外部に記憶された前記データブロックの外部コピーの状態を示す、ステップと、

30

前記データブロックのコピーを、前記プロセッサインメモリの対応するプロセッサの少なくとも1つのキャッシュに記憶するステップと、

前記データブロックのコピーのプロセッサインメモリコヒーレンス状態をプロセッサインメモリディレクトリに記憶するステップであって、前記プロセッサインメモリコヒーレンス状態は、前記少なくとも1つのキャッシュに存在する前記データブロックのコピーの状態を示す、ステップと、

40

前記プロセッサインメモリディレクトリを使用して、前記少なくとも1つのキャッシュに存在する前記データブロックのデータコヒーレンスを維持するステップと、

前記プロセッサインメモリに接続されたホストプロセッサのホスト内コヒーレンス制御ロジックから受信したコヒーレンスメッセージを、前記ホストプロセッサの第 2 キャッシュコヒーレンシプロトコルから前記第 1 のキャッシュコヒーレンシプロトコルに変換するステップであって、前記第 2 キャッシュコヒーレンシプロトコルは、前記第 1 キャッシュコヒーレンシプロトコルと異なっている、ステップと、

変換されたコヒーレンスメッセージと、前記シャドウディレクトリの内容とを用いて、前記データブロックのデータコヒーレンスを維持するステップと、を含む、

方法。

50

## 【請求項 9】

前記第 2 キャッシュコヒーレンシプロトコルは、前記第 1 キャッシュコヒーレンシプロトコルのブロックサイズと異なるブロックサイズを使用する、  
請求項 8 の方法。

## 【請求項 10】

前記プロセッサインメモリの前記メモリのデータコヒーレンスを維持するステップは、前記データブロックのアクセスに応じて、前記データブロックのブリッジコヒーレンス状態を更新するステップを含む、  
請求項 8 又は 9 の方法。

## 【請求項 11】

前記プロセッサインメモリの前記メモリのデータコヒーレンスを維持するステップは、前記少なくとも 1 つのキャッシュと、前記プロセッサインメモリの前記メモリとのコヒーレンスを維持するステップであって、前記少なくとも 1 つのキャッシュは、前記プロセッサインメモリのコヒーレンスドメイン内に存在する、ステップと、を含む、  
請求項 8 又は 9 の方法。

## 【請求項 12】

前記プロセッサインメモリの前記メモリのアクセスに応じて、前記プロセッサインメモリのメモリサイドキャッシュをチェックするステップと、

前記プロセッサインメモリのコヒーレンスドメインのキャッシュから前記プロセッサインメモリの前記メモリサイドキャッシュにライトスルーキャッシュポリシーを実施するステップであって、前記シャドウディレクトリの前記データブロックの前記ブリッジコヒーレンス状態は、no-host-copy 状態、host-read 状態、又は、host-written 状態である、ステップと、をさらに含み、

前記no-host-copy 状態は、ブリッジコヒーレンスドメイン内のホストプロセッサが、前記データブロックの何れの部分もホストキャッシュに記憶していないことを示しており、前記host-read 状態は、前記ブリッジコヒーレンスドメイン内の少なくとも 1 つのプロセッサが、前記データブロックの一部を読み出し、ホストキャッシュに前記一部の未修正のコピーを有する可能性があることを示しており、前記host-written 状態は、前記ブリッジコヒーレンスドメイン内の少なくとも 1 つのプロセッサが、前記データブロックの排他的所有権を有しており、ホストキャッシュに前記データブロックを書き込んでいる可能性があることを示している、

請求項 8 又は 9 の方法。

## 【請求項 13】

前記プロセッサインメモリのコヒーレンスドメインのキャッシュから前記プロセッサインメモリのメモリサイドキャッシュにライトバックキャッシュポリシーを実施するステップであって、前記シャドウディレクトリの前記データブロックの前記ブリッジコヒーレンス状態は、no-host-copy 状態、host-read 状態、host-written 状態、又は、processor-in-memory-modified 状態である、ステップをさらに含み、

前記no-host-copy 状態は、ブリッジコヒーレンスドメイン内のホストプロセッサが、前記データブロックの何れの部分もホストキャッシュに記憶していないことを示しており、前記host-read 状態は、前記ブリッジコヒーレンスドメイン内の少なくとも 1 つのプロセッサが、前記データブロックの一部を読み出し、ホストキャッシュに前記一部の未修正のコピーを有する可能性があることを示しており、前記host-written 状態は、前記ブリッジコヒーレンスドメイン内の少なくとも 1 つのプロセッサが、前記データブロックの排他的所有権を有しており、ホストキャッシュに前記データブロックを書き込んでいる可能性があることを示しており、前記processor-in-memory-modified 状態は、前記データブロックの何れかの部分が、前記プロセッサインメモリによって前記少なくとも 1 つのキャッシュにおいて変更されたかどうかを示している、

10

20

30

40

50

請求項 8 又は 9 の方法。

【請求項 1 4】

前記データブロックのデータコヒーレンスを維持するステップは、前記データブロックへのメモリアクセスに応じてコヒーレンスメッセージを生成するステップを含む、

請求項 8 又は 9 の方法。

【請求項 1 5】

前記ホストプロセッサのホストディレクト内の前記データブロックのホストコピーのホストコヒーレンス状態を記憶するステップと、

前記ホストコヒーレンス状態を用いて、前記ホストプロセッサに存在するホストコピーデータブロックと前記プロセッサインメモリに存在する前記データブロックとのデータコヒーレンスを維持するステップと、をさらに含む、

10

請求項 8 又は 9 の方法。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、処理システムに関し、より詳細には、処理システムにおけるキャッシュコヒーレンスに関する。

【背景技術】

【0002】

一般に、プロセッサインメモリ（すなわち、プロセッシングインメモリ又はニアメモリ処理）技術は、処理要素をメモリ（例えば、ランダムアクセスメモリ）に組み込むことによって、データ集約型アプリケーションのエネルギー効率及び性能を向上させる。異種システムアーキテクチャフレームワーク内にプロセッシングインメモリを組み込み、従来のプログラミングモデルをこれらのシステムプロセッサに拡張するには、キャッシュコヒーレンスが必要となる。高度なプロセッサインメモリ設計では、独自のキャッシュ階層を有する複雑な処理エンジンが組み込まれている。しかしながら、システムは、適切な動作のために、これらのキャッシュとホストプロセッサ（すなわち、メインプロセッサ）のキャッシュとの間のコヒーレンスを維持すべきである。シンプルでキャッシュレスなプロセッサインメモリの設計であっても、ホストキャッシュと、プロセッサインメモリで処理されるデータとの間のコヒーレンスを確保する必要がある。

20

30

【0003】

プロセッサインメモリを含むシステムにおいてデータコヒーレンスを提供するための従来技術では、ハードウェアサポートを含まない場合があり、ホストとプロセッサインメモリとの間のキャッシュコヒーレンスを調整するソフトウェアを必要とすることがある。このようなソリューションは、複雑なプログラミングが必要になる。他の技術では、既存のキャッシュコヒーレンスソリューションを拡張して、任意のプロセッサインメモリ設計を含む。システム内の全てのプロセッサ間の統合を必要とする従来技術のキャッシュコヒーレンスソリューションをプロセッサインメモリに拡張することは、実用的でない場合がある。これは、このソリューションが、システムがどのくらい多くのメモリデバイス（すなわち、プロセッサインメモリデバイス）を含むかを知るためのホスト上のキャッシュコヒーレンス実装を必要とし、システム設計を、設計の相互運用性及びスケラビリティを制限する特定のバージョン又はタイプのプロセッサに制限するためである。また、キャッシュコヒーレンスプロトコル設計は複雑であるため、かかる設計は、確認及び実証が難しく、結果として、既存の実績のあるホストキャッシュコヒーレンスプロトコルに対する望ましくない変更が生じる。既存のコヒーレンスプロトコルの変更を必要としない他の提案されたプロセッサインメモリキャッシュコヒーレンススキームでは、例えばリード共有（すなわち、ホスト及びプロセッサインメモリの両方におけるリードオンリーデータのキャッシュ）等の一般的に使用されるシナリオが許可されていない。

40

【0004】

プロセッサインメモリアーキテクチャに対する他のキャッシュコヒーレンス策は、プロ

50

セッサインメモリデバイスのキャッシュを考慮せず、プロセッサインメモリデバイスがデータを変更する際にホストキャッシュがコヒーレントであることを確保するように、ホスト側のキャッシュコヒーレンス技術を拡張することに依存する。このような技術では、プロセッサインメモリデバイスが、ホストと一致するキャッシュラインサイズ及びプロトコルを使用することを必要とする。さらに、これらの策は、メモリ（例えば、ダイナミックランダムアクセスメモリ（DRAM））内の各キャッシュラインと共に、1つ以上の専用のキャッシュコヒーレンス状態ビットの記憶を必要とし得るが、専門のメモリモジュールを必要とし、容易に利用可能なメモリコンポーネントが使用できなくなる場合がある。したがって、プロセッサインメモリデバイスを組み込んだシステムのキャッシュコヒーレンスを実現するための改良された技術が望まれている。

10

【発明の概要】

【課題を解決するための手段】

【0005】

本発明の少なくとも1つの実施形態では、システムは、プロセッサインメモリを含む。プロセッサインメモリは、プロセッサインメモリのメモリに存在するデータブロックのブリッジコヒーレンス状態を記憶するように構成されたシャドウディレクトリストレージを含む。ブリッジコヒーレンス状態は、プロセッサインメモリの外部に記憶されたデータブロックの任意のコピーの状態を示している。プロセッサインメモリは、プロセッサインメモリの第1キャッシュコヒーレンスプロトコルと、ホストプロセッサの第2キャッシュコヒーレンスプロトコルとの間でコヒーレンスメッセージを変換するように構成されたブリッジプロトコル変換ロジックを含む。第2キャッシュコヒーレンスプロトコルは、第1キャッシュコヒーレンスプロトコルと異なっている。プロセッサインメモリは、コヒーレンスメッセージ及びシャドウディレクトリストレージの内容を用いて、プロセッサインメモリのメモリのデータコヒーレンスを維持するように構成されたブリッジプロトコル制御ロジックを含む。プロセッサインメモリは、集積回路ダイのスタックを含んでもよい。プロセッサインメモリのメモリは、集積回路ダイのスタックのうち複数のメモリ集積回路ダイに含まれてもよい。ブリッジプロトコル制御ロジック及びブリッジプロトコル変換ロジックは、集積回路ダイのスタックのうちロジック集積回路ダイに含まれてもよい。第2キャッシュコヒーレンスプロトコルは、第1キャッシュコヒーレンスプロトコルのブロックサイズとは異なるブロックサイズを使用してもよい。プロセッサインメモリは、対応するプロセッサの少なくとも1つのキャッシュを含んでもよい。少なくとも1つのキャッシュは、プロセッサインメモリのコヒーレンスドメイン内にあってもよい。少なくとも1つのキャッシュは、ライトスルーキャッシュであってもよく、データブロックのブリッジコヒーレンス状態は、no-host-copy状態、host-read状態及びhost-written状態のうち何れか1つであってもよい。少なくとも1つのキャッシュは、ライトバックキャッシュであってもよく、データブロックのブリッジコヒーレンス状態は、no-host-copy状態、host-read状態、host-written状態及びprocessor-in-memory-modified状態のうち何れか1つであってもよい。プロセッサインメモリは、プロセッサインメモリのコヒーレンスドメインの外部にあるメモリサイドキャッシュを含んでもよい。ブリッジプロトコル制御ロジックは、データブロックへのメモリアクセスに応じて、メモリサイドキャッシュに記憶されたデータブロックの任意のコピーの状態をチェックするように構成されてもよい。プロセッサインメモリは、対応するプロセッサの少なくとも1つのキャッシュ内に記憶されたデータブロックのプロセッサインメモリコヒーレンス状態を記憶するように構成されたプロセッサインメモリディレクトリを含んでもよい。プロセッサインメモリは、プロセッサインメモリディレクトリの内容を用いて、少なくとも1つのキャッシュに記憶されたデータブロックのデータコヒーレンスを維持するように構成されたプロセッサインメモリ内コヒーレンスコントローラを含んでもよい。システムは、プロセッサインメモリに接続されたホストプロセッサを含んでもよい。ホストプロセッサは、プロセッサインメモリの第1キャッシュコヒーレンスプロトコルと、ホストプロセッサの第2キャッシュコヒーレン

20

30

40

50

スプロトコルとの間でコヒーレンスメッセージを変換するように構成されたホストブリッジ変換ロジックを含んでもよい。ホストプロセッサは、ホストのコヒーレンスドメインの外側であって、プロセッサインメモリのコヒーレンスドメインの外側にあるメモリサイドキャッシュをさらに含んでもよい。シャドウディレクトリストレージは、メモリサイドキャッシュに記憶されたデータブロックの任意のコピーの状態を保存してもよい。ホストプロセッサは、ホストコピーデータブロックのホストコヒーレンス状態を記憶するように構成されたホストディレクトリであって、ホストプロセッサ内に記憶されたホストディレクトリと、ホストディレクトリを用いて、ホストプロセッサに存在するホストコピーデータブロックのデータコヒーレンスを維持するように構成されたホスト内コヒーレンス制御ロジックと、をさらに含んでもよい。

10

## 【0006】

本発明の少なくとも1つの実施形態では、方法は、プロセッサインメモリのメモリに存在するデータブロックのブリッジコヒーレンス状態をシャドウディレクトリに記憶するステップを含む。ブリッジコヒーレンス状態は、プロセッサインメモリの外部に記憶されたデータブロックの任意のコピーの状態を示している。方法は、プロセッサインメモリの第1キャッシュコヒーレンスプロトコルと、ホストプロセッサの第2キャッシュコヒーレンスプロトコルとの間でコヒーレンスメッセージを変換するステップを含む。第2キャッシュコヒーレンスプロトコルは、第1キャッシュコヒーレンスプロトコルと異なっている。方法は、コヒーレンスメッセージ及びシャドウディレクトリストレージの内容を用いて、プロセッサインメモリのメモリのデータコヒーレンスを維持するステップを含む。第2キャッシュコヒーレンスプロトコルは、第1キャッシュコヒーレンスプロトコルのブロックサイズとは異なるブロックサイズを使用してもよい。プロセッサインメモリのメモリのデータコヒーレンスを維持するステップは、データブロックへのアクセスに応じて、データブロックのブリッジコヒーレンス状態を更新するステップを含んでもよい。プロセッサインメモリのメモリのデータコヒーレンスを維持するステップは、プロセッサインメモリのプロセッサの少なくとも1つのキャッシュと、プロセッサインメモリのメモリとのコヒーレンスを維持するステップを含んでもよい。少なくとも1つのキャッシュは、プロセッサインメモリのコヒーレンスドメイン内に存在してもよい。方法は、プロセッサインメモリのメモリのアクセスに応じて、プロセッサインメモリのメモリサイドキャッシュをチェックするステップを含んでもよい。方法は、プロセッサインメモリのコヒーレンスドメインのキャッシュからプロセッサインメモリのメモリサイドキャッシュにライトスルーキャッシュポリシーを実施するステップを含んでもよい。シャドウディレクトリのデータブロックのブリッジコヒーレンス状態は、`no-host-copy`状態、`host-read`状態及び`host-written`状態のうち何れか1つであってもよい。方法は、プロセッサインメモリのコヒーレンスドメインのキャッシュからプロセッサインメモリのメモリサイドキャッシュにライトバックキャッシュポリシーを実施するステップを含んでもよい。シャドウディレクトリのデータブロックのブリッジコヒーレンス状態は、`no-host-copy`状態、`host-read`状態、`host-written`状態及び`processor-in-memory-modified`状態のうち何れか1つであってもよい。プロセッサインメモリのメモリのデータコヒーレンスを維持するステップは、データブロックへのメモリアクセスに応じてコヒーレンスメッセージを生成するステップを含んでもよい。方法は、対応するプロセッサの少なくとも1つのキャッシュ内に記憶されたデータブロックのプロセッサインメモリコヒーレンス状態を記憶するステップを含んでもよい。方法は、プロセッサインメモリディレクトリの内容を用いて、少なくとも1つのキャッシュに存在するデータブロックのデータコヒーレンスを維持するステップを含んでもよい。方法は、ホストプロセッサ内に記憶されたホストコヒーレンス状態であって、ホストコピーデータブロックのホストコヒーレンス状態を記憶するステップを含んでもよい。方法は、ホストディレクトリを用いて、ホストプロセッサに存在するホストコピーデータブロックのデータコヒーレンスを維持するステップを含んでもよい。

20

30

40

## 【0007】

50

本発明の少なくとも1つの実施形態では、装置はホストプロセッサを含む。ホストプロセッサは、プロセッサインメモリの第1キャッシュコヒーレンスプロトコルと、ホストプロセッサの第2キャッシュコヒーレンスプロトコルとの間でコヒーレンスメッセージを変換するように構成されたホストブリッジ変換ロジックを含む。第2キャッシュコヒーレンスプロトコルは、第1キャッシュコヒーレンスプロトコルと異なっている。ホストプロセッサは、プロセッサインメモリに存在するデータブロックのホストコヒーレンス状態であって、ホストプロセッサ内に記憶されたホストコヒーレンス状態であって、ホストコピーに対応するホストコヒーレンス状態を記憶するように構成されたホストディレクトリを含んでもよい。ホストプロセッサは、ホストディレクトリを用いて、ホストプロセッサに存在するデータブロックのデータコヒーレンスを維持するように構成されたホスト内コヒーレンス制御ロジックを含んでもよい。第2キャッシュコヒーレンスプロトコルは、第1キャッシュコヒーレンスプロトコルのブロックサイズとは異なるブロックサイズを使用してもよい。

10

【0008】

本発明は、添付の図面を参照することによってより良く理解され、その多くの目的、特徴及び利点が当業者には明らかであろう。

【図面の簡単な説明】

【0009】

【図1】プロセッサインメモリデバイスを含む例示的な処理システムの機能ブロック図である。

20

【図2】例示的なプロセッサインメモリの機能ブロック図である。

【図3】本発明の少なくとも1つの実施形態と一致する、ホストプロセッサとプロセッサインメモリとの間の例示的なキャッシュコヒーレントリンクの機能ブロック図である。

【図4】本発明の少なくとも1つの実施形態と一致する、複数のホストと様々なプロセッサインメモリデバイスとの間の例示的なキャッシュコヒーレントリンクの機能ブロック図である。

【図5】例示的なプロセッサインメモリキャッシュ構成を含む例示的なプロセッサインメモリの機能ブロック図である。

【図6】本発明の少なくとも1つの実施形態と一致する、ホストプロセッサと、第1キャッシュプロトコルを有するプロセッサインメモリとの間の例示的なキャッシュコヒーレンスブリッジプロトコルの状態図である。

30

【図7】本発明の少なくとも1つの実施形態と一致する、ホストプロセッサと、第2キャッシュプロトコルを有するプロセッサインメモリとの間の例示的なキャッシュコヒーレンスブリッジプロトコルの状態図である。

【発明を実施するための形態】

【0010】

異なる図面における同じ参照記号の使用は、類似又は同一の項目を示している。

【0011】

キャッシュコヒーレンスブリッジプロトコルは、ホストプロセッサのキャッシュコヒーレンスプロトコルと、プロセッサインメモリのキャッシュコヒーレンスプロトコルとの間のインタフェースを提供することによって、ホストプロセッサ及びプロセッサインメモリのコヒーレンスメカニズムを分離する。キャッシュコヒーレンスブリッジプロトコルでは、既存のホストプロセッサキャッシュコヒーレンスプロトコルに対する変更を制限する必要がある。キャッシュコヒーレンスブリッジプロトコルは、ホストプロセッサと、異なるベンダによって設計されたプロセッサインメモリデバイスとの間の相互運用性を容易にするために使用され、ホストプロセッサ及びプロセッサインメモリデバイスの両方は、各プロセッサ内のコンピューティングユニット間のコヒーレンス技術を実施してもよい。キャッシュコヒーレンスブリッジプロトコルは、ホストプロセッサ及び/又はプロセッサインメモリのキャッシュコヒーレンスプロトコルによって使用される粒度とは異なるキャッシュコヒーレンス許可の粒度をサポートしてもよい。キャッシュコヒーレンスブリッジプロ

40

50

トコルは、データを含むプロセッサインメモリの外部のシステムにキャッシュされた当該データのコピーの集約ビューを示す状態情報を維持するシャドウディレクトリを使用している。対照的に、従来のコヒーレンスプロトコルは、プロセッサ内のデータのローカルコピーの状態を追跡する。

#### 【0012】

プロセッサインメモリは、1つ以上のメモリダイ内の処理ロジックとして、又は、1つ以上のメモリダイと共にスタックされた別個のロジックダイとして実装されてもよい。図1を参照すると、システム100は、ホスト110と、インターポーザ112上の4つのプロセッサインメモリデバイスと、を含む。プロセッサインメモリ102は、処理ロジック104を含み、処理ロジック104は、複数のメモリダイ106と共にスタックされた別個のダイに含まれてもよい。処理ロジック104は、少なくとも1つの加速処理ユニット（すなわち、中央処理ユニット及びグラフィックス処理ユニットを含む高度処理ユニット）、中央処理ユニット、グラフィックス処理ユニット、又は、他のプロセッサを含むことができ、コプロセッサ又は固定機能処理（fixed-function processing）ハードウェアを含んでもよい。

10

#### 【0013】

図2は、例示的な処理ロジック104を示す図である。処理ロジック104は、グラフィックス処理ユニット（例えば、コンピューティングユニットクラスタ206）を含む高度処理ユニットを含み、グラフィックス処理ユニットは、複数のグラフィックス処理ユニットコンピューティングユニット214を含む。高度処理ユニットは、複数の中央処理ユニットコア216を含む中央処理ユニットクラスタ208を含んでもよい。高度処理ユニットは、複数のハードウェアアクセラレータ218を含む固定機能ハードウェア210を含んでもよい。各プロセッサインメモリ102内のコンポーネント（例えば、処理ロジック104の各グラフィックス処理ユニットコンピューティングユニット214及び各中央処理ユニットコア216）は、コンポーネント間のハードウェアキャッシュコヒーレンスを実施してもよい。

20

#### 【0014】

図1に戻ると、ホスト110は、複数の処理コアを含むことができ、これらの処理コア間でハードウェアキャッシュコヒーレンスを実施してもよい。ホスト110は、マルチソケット実装インタフェース（例えば、コヒーレントHyperTransport）を介して、複数のホストデバイス間のキャッシュコヒーレンスをサポートしてもよい。従来のキャッシュコヒーレンスポリシーの何れかを拡張してプロセッサインメモリを含むことは、ホスト及びプロセッサインメモリが同じキャッシュコヒーレンスプロトコルを実装することを必要とする。この要件は、異なるベンダが異なる既存のコヒーレンスプロトコルを有しているため、ホストと、異なるベンダからのプロセッサインメモリ設計との間の相互運用性を妨げることにより、プロセッサインメモリ設計のビジネスチャンスを制限する。さらに、それぞれプロセッサインメモリを含み得る多数のメモリを有する高性能システムをサポートするためには、ホスト及びプロセッサインメモリのキャッシュコヒーレンスプロトコルを、コヒーレントノードの可変数（すなわち、プロセッサインメモリデバイスの無いシステムより大きい数であって、従来のシステムより大きい数）にスケラブルであるように、従来技術のコヒーレンス状態を追跡するオーバーヘッドを大きく増加させる。

30

40

#### 【0015】

キャッシュコヒーレンスブリッジプロトコルは、既存のキャッシュコヒーレンスプロトコル（例えば、ホストキャッシュコヒーレンスプロトコル及びプロセッサインメモリキャッシュコヒーレンスプロトコル）間のインタフェースを提供する。キャッシュコヒーレンスブリッジプロトコルは、任意の数のプロセッサインメモリデバイスのシステム内の任意の特定のキャッシュラインに対して単一の潜在的な共有デバイスを追加することを含むホスト側サポートを使用しており、結果として、オーバーヘッドがごく僅かしか増加せず、システム内のプロセッサインメモリデバイスの数に関係なく、必要とされる追加状態が制限される。キャッシュコヒーレンスブリッジプロトコルは、単一の追加の潜在的な共有部

50

(sharer)によって、ホスト側の全てのコンポーネントのプロセッサインメモリ側表現を含み、結果として、プロセッサインメモリのオーバーヘッドがごく僅かしか増加しないだけでなく、システム内のホストの性質及び数に関係なく、必要とされる追加状態の数を制限する。

**【0016】**

キャッシュコヒーレンスブリッジプロトコルは、ブリッジ変換ユニットと、プロセッサインメモリに実装されたブリッジプロトコルコントローラと、を含む。キャッシュコヒーレンスブリッジプロトコルは、プロセッサインメモリに記憶されたシャドウディレクトリを含む。シャドウディレクトリは、キャッシュコヒーレンスを追跡する状態情報を外部の視点(すなわち、プロセッサインメモリ上のキャッシュされたコピーの状態を追跡する代わりに)で記憶する。シャドウディレクトリは、データ要素が存在するプロセッサインメモリの外部にキャッシュされた当該データ要素の全てのコピーの集約ビューを追跡するブリッジコヒーレンス状態を記憶する。シャドウディレクトリは、ホストディレクトリに含まれ得る情報のサブセットを、プロセッサインメモリの視点から記憶する。少なくとも1つの実施形態では、キャッシュコヒーレンスブリッジプロトコルは、従来のキャッシュコヒーレンスプロトコルによって既に生成されたキャッシュコヒーレンスメッセージのサブセットを使用することによって、ホスト上及びプロセッサインメモリ内に実装される可能性のある、以前に開発され検証されたキャッシュコヒーレンスプロトコルに適合するための変更を制限する。

**【0017】**

図3を参照すると、キャッシュコヒーレンスブリッジプロトコルの少なくとも1つの実施形態では、ホスト内コヒーレンスコントローラ302は、プロセッサインメモリ102がホストコヒーレンスプロトコルにおける単なる別のノードであるかのように、プロセッサインメモリ102に対するキャッシュコヒーレンスメッセージを生成する。ブリッジ変換ユニット304は、これらのキャッシュコヒーレンスメッセージを、ブリッジプロトコルコヒーレンスメッセージに変換してもよい。プロセッサインメモリ102において、ブリッジ変換ユニット316は、受信したキャッシュコヒーレンスメッセージを、ホストコヒーレンスプロトコル又は中間ブリッジコヒーレンスプロトコルから、ブリッジプロトコル制御ロジック322が使用可能な形式に変換してもよい。反対方向では、ブリッジプロトコル制御ロジック322は、シャドウディレクトリ324の内容に基づいて、ホストのホストコヒーレンスプロトコル又は中間ブリッジコヒーレンスプロトコルに一致したキャッシュコヒーレンスメッセージを生成し、プロセッサインメモリ内コヒーレンスコントローラ320からのメッセージは、ディレクトリ326の内容、及び/又は、プロセッサインメモリ102によって受信若しくは生成されたメモリ要求に基づいてもよい。

**【0018】**

ブリッジ変換ユニット316は、ブリッジプロトコル制御ロジック322から受信したキャッシュコヒーレンスメッセージを、ブリッジプロトコルメッセージ又はホストコヒーレンスコントローラプロトコルメッセージに変換する。ホスト110のブリッジ変換ユニット304は、キャッシュコヒーレンスメッセージを、ブリッジプロトコルからホストプロトコルに変換してもよい。個別のロジックとして示されているが、ブリッジ変換ユニット304は、メモリインタフェースコントローラ308と一体となっている部分として実装されてもよく、ブリッジ変換ユニット316は、メモリインタフェースコントローラ314と一体となっている部分として実装されてもよい。キャッシュコヒーレンスブリッジプロトコルの少なくとも1つの実施形態では、中間キャッシュコヒーレンスメッセージが使用されず、ブリッジ変換ユニット304、316は、システムのリセット時又は他のユーザの構成に応じて構成され得る所定のプロトコルフォーマットに基づいて、キャッシュコヒーレンスメッセージを、プロセッサインメモリキャッシュコヒーレンスプロトコルメッセージ又はホストキャッシュコヒーレンスプロトコルメッセージに直接変換する。

**【0019】**

ブリッジプロトコル制御ロジック322は、プロセッサインメモリ内コヒーレンスメッ

10

20

30

40

50

ページを監視し、場合によっては、プロセッサインメモリ102の1つ以上のメモリチャネルの、ホスト110によるメモリアクセス（例えば、キャッシュミス）に介入する。ブリッジプロトコル制御ロジック322は、プロセッサインメモリ102上に実装されるので、ブリッジプロトコル制御ロジック322は、追加のオフデバイストラフィックを生成することなく、プロセッサインメモリ102のメモリチャネル内で発生する全てのアクセス（プロセッサインメモリ102内で生成されるアクセス、及び、ホスト110又は他のデバイスによって外部で生成されるアクセス）を可視化する。ブリッジプロトコル制御ロジック322の機能を以下にさらに説明する。

#### 【0020】

図4を参照すると、キャッシュコヒーレンスブリッジプロトコルを含む例示的なシステムは、複数のプロセッサインメモリデバイスと、ホスト間コヒーレントリンク411によって接続されたホスト410、412と、を含む。プロセッサインメモリ402、406の各々は、キャッシュコヒーレントブリッジプロトコルリンク403及びキャッシュコヒーレントブリッジプロトコルリンク405によって、ホスト410に接続されている。プロセッサインメモリ404、408の各々は、キャッシュコヒーレントブリッジプロトコルリンク407及びキャッシュコヒーレントブリッジプロトコルリンク409によって、ホスト412に接続されている。本明細書で言及するように、特定のメモリアドレスに関して、ホームプロセッサインメモリは、特定のアドレスのデータを記憶するプロセッサインメモリである。本明細書で言及するように、ホームノードは、特定のアドレスについてメモリ内のホームプロセッサインメモリに接続されたホストである。ホストコヒーレンスプロトコルディレクトリは、コヒーレンスディレクトリ目的のホームノードが、データのインメモリコピーを保持するプロセッサインメモリに接続されたノードであるように分散されてもよい。本明細書で言及するように、データブロックは、ホストとプロセッサインメモリとの間のコヒーレンス追跡の粒度である。データブロックは、ホスト内及びホスト間のコヒーレンス追跡の粒度の整数倍である。キャッシュコヒーレンスブリッジプロトコル、ホスト410、ホスト412、プロセッサインメモリ402、プロセッサインメモリ404、プロセッサインメモリ406及びプロセッサインメモリ408を含むシステムの少なくとも1つの実施形態では、ライトアロケートキャッシュコヒーレンスポリシーを実施する（すなわち、ホスト又はプロセッサインメモリは、キャッシュラインの任意の部分を書き込む前に、キャッシュラインをメモリから対応するキャッシュに読み出す）。

#### 【0021】

図3を参照して上述したように、キャッシュコヒーレンスブリッジプロトコルは、ホスト側サポートを使用し、これは、単一の共有デバイスをホストディレクトリ328に追加することを含んでもよい。ホストディレクトリ328内の単一の共有デバイスを使用して、ホストコヒーレンスプロトコルの観点から、ホームプロセッサインメモリのキャッシュコヒーレンス状態を追跡してもよい。図4を参照すると、例えば、プロセッサインメモリ402がホームプロセッサインメモリであり、ホームノードホスト410のホストコヒーレンスディレクトリで追跡されるキャッシュラインの場合、ホスト410のホストコヒーレンスディレクトリは、プロセッサインメモリ402内のキャッシュラインのキャッシュコヒーレンス状態を追跡する。各プロセッサインメモリのシャドウディレクトリは、システムの他の場所に存在し得る任意のデータブロックのコピーの集約ビューを維持する。少なくとも1つの実施形態では、ホームプロセッサインメモリがプロセッサインメモリ402であるキャッシュブロックの場合、プロセッサインメモリ402のシャドウディレクトリは、ホスト410又はホスト412のキャッシュに存在し得るキャッシュラインの任意のコピーの集約ビューを維持する。

#### 【0022】

一般に、プロセッサインメモリキャッシュは、ライトスルーキャッシュ又はライトバックキャッシュである。プロセッサインメモリキャッシュがライトスルーキャッシュである場合、ホストによるメモリアクセスに応じてプロセッサインメモリキャッシュをチェックする必要はない。プロセッサインメモリキャッシュがライトバックキャッシュである場合

10

20

30

40

50

、ホストメモリアクセスを処理する (servicing) ことは、最新のデータについてプロセッサインメモリキャッシュをチェックすることを含む。例えば、図 5 を参照すると、レベル 1 のキャッシュは、メモリ相互接続部 204 のプロセッサ側のコヒーレンスドメインのライトスルーキャッシュであり、レベル 2 のキャッシュは、メモリ相互接続部 204 のプロセッサ側のコヒーレンスドメインの外部のメモリへの経路のメモリサイドキャッシュである。ホストによる全てのメモリアクセスは、プロセッサインメモリキャッシュコヒーレンスドメインの外部のメモリサイドキャッシュ (例えば、レベル 2 のキャッシュ) をチェックする。他の実施形態では、処理ロジック 104 がキャッシュを含まないので、プロセッサインメモリ内コヒーレンスを実施しない。

#### 【0023】

図 3 を参照すると、各プロセッサインメモリ 102 は、プロセッサインメモリ 102 のメモリに記憶された各データブロックのブリッジコヒーレンス状態を追跡するブリッジプロトコル制御ロジック 322 を含む。そのブリッジコヒーレンス状態は、シャドウディレクトリ 324 に記憶される。ブリッジプロトコル制御ロジック 322 は、シャドウディレクトリ 324 の状態遷移を制御し、対応するインジケータを記憶する。図 3 及び図 6 を参照すると、少なくとも 1 つの実施形態では、プロセッサインメモリの各データブロックについて、シャドウディレクトリ 324 は、no-host-copy (I) 状態 602、host-read (S) 状態 604、及び、host-written (M) 状態 606 のブリッジコヒーレンス状態のインジケータを記憶する。状態図は、ライトスルーレベル 1 のキャッシュと、プロセッサインメモリキャッシュコヒーレンスドメインの外側のメモリサイド (例えば、レベル 2) キャッシュと、を有するプロセッサインメモリ 102 に対応する。ブリッジプロトコル制御ロジック 322 は、(例えば、プロセッサインメモリ 102 又はホスト 110 からの) メモリへの各アクセスに応じて、メモリサイドキャッシュをチェックする。no-host-copy (I) 状態 602 は、ブリッジキャッシュコヒーレンスドメインの何れのホストも、データブロックの任意の部分をキャッシュメモリに記憶していないことを示している。このブリッジコヒーレンス状態は、従来の MSI、MESI、MOESI キャッシュコヒーレンスプロトコルの Invalid 状態に類似しており、M は変更 (modified) を表し、O は所有 (owned) を表し、E は排他 (exclusive) を表し、S は共有 (shared) を表し、I は無効 (invalid) を表す。host-read (S) 状態 604 は、ブリッジコヒーレンスドメインの 1 つ以上のホストがデータブロックの一部を読み出し、それらのキャッシュに未修正のコピーを有し得るが、それを書き込んでいないことを示している。このブリッジコヒーレンス状態は、MSI、MESI、MOESI キャッシュコヒーレンスプロトコルにおける共有状態に類似している。host-written (M) 状態 606 は、ブリッジコヒーレンスドメインの 1 つ以上のホストがデータブロックの排他的所有権を有し、その一部を書き込んでいる可能性があることを示している (例えば、ホスト側のコアは、MSI キャッシュコヒーレンスプロトコルの変更状態のラインのコピー、MESI キャッシュコヒーレンスプロトコルの変更状態若しくは排他状態のラインのコピー、又は、MOESI キャッシュコヒーレンスプロトコルの変更状態、所有状態若しくは排他状態のラインのコピーを有する)。

#### 【0024】

プロセッサインメモリ 102 によるローカルデータ (すなわち、プロセッサインメモリ 102 がホームプロセッサインメモリであるメモリアドレスに関連付けられたデータ) の読み出し、及び、プロセッサインメモリ 102 によるローカルデータの書き込みの各々は、P.r 及び P.w で示され、ブリッジプロトコル制御ロジック 322 は、シャドウディレクトリ 324 の対応するエントリの関連する状態遷移に回答する。ホスト 110 によるプロセッサインメモリのローカルデータの読み出し、及び、ホスト 110 によるプロセッサインメモリのローカルデータの書き込みの各々は、H.r 及び H.w で示され、ブリッジプロトコル制御ロジック 322 は、シャドウディレクトリ 324 の対応するエントリの関連する状態遷移で回答する。遷移の中間状態及びアクションは、明瞭にするために、状態図に示されていない。

10

20

30

40

50

## 【 0 0 2 5 】

ローカルデータブロックへのプロセッサインメモリの読み出し ( P . r ) アクセスに応じて、データブロックがシャドウディレクトリ 3 2 4 に `no - host - copy ( I )` 状態 6 0 2 又は `host - read ( S )` 状態 6 0 4 を有する場合、ブリッジプロトコル制御ロジック 3 2 2 は、プロセッサインメモリの外部でコヒーレンスメッセージを生成せず、現在の状態である `no - host - copy ( I )` 状態 6 0 2 又は `host - read ( S )` 状態 6 0 4 に留まる。データブロックがシャドウディレクトリに `host - written ( M )` 状態 6 0 6 を有する場合、ブリッジプロトコル制御ロジック 3 2 2 は、プロセッサインメモリに取り付けられたホームノードへの読み出しコヒーレンス要求を生成し、コヒーレンス状態を `host - read ( S )` 状態 6 0 4 に遷移させる ( 7 0 4 )。そのホストはホームノードでもあるので、その要求が他の読み出し要求であるかのように、ホストのコヒーレンスプロトコルと統合される。プロセッサインメモリは、読み出しを進める前に、コヒーレンス要求が完了し、コヒーレンスの確認応答を受信し、更新データが使用可能になるまで待機する必要がある。

10

## 【 0 0 2 6 】

ローカルデータへのプロセッサインメモリの書き込み ( P . w ) アクセスに応じて、データブロックがシャドウディレクトリ 3 2 4 に `no - host - copy ( I )` 状態 6 0 2 を有する場合、ブリッジプロトコル制御ロジック 3 2 2 は、プロセッサインメモリの外側でコヒーレンスメッセージを生成せず、現在の状態である `no - host - copy ( I )` 状態 6 0 2 に留まる。データブロックがシャドウディレクトリ 3 2 4 に `host - read ( S )` 状態 6 0 4 を有する場合、ブリッジプロトコル制御ロジック 3 2 2 は、ホストに対する無効化要求を生成し、完了信号を待った後に、書き込み及び `no - host - copy ( I )` 状態 6 0 2 への遷移 ( 7 0 2 ) を進める。データブロックがシャドウディレクトリ 3 2 4 に `host - written ( M )` 状態 6 0 6 を有する場合、ブリッジプロトコル制御ロジック 3 2 2 は、無効化要求を生成し、変更されたデータを待ってから、 `no - host - copy ( I )` 状態 6 0 2 への状態遷移 ( 7 0 7 ) を開始する。

20

## 【 0 0 2 7 】

ホスト又はホストを介してメモリにアクセスする他のコンポーネントで発生し得るホスト側からの読み出し ( H . r ) アクセスに応じて、データブロックがシャドウディレクトリ 3 2 4 に `no - host - copy ( I )` 状態 6 0 2 又は `host - written ( M )` 状態 6 0 6 を有する場合、ブリッジプロトコル制御ロジック 3 2 2 は、状態を `host - read ( S )` 状態 6 0 4 に遷移させる ( 7 0 2 又は 7 0 4 )。 `host - written ( M )` 状態 6 0 6 にある間のホスト側からの追加の読み出し要求は、 ( `host - written ( M )` 状態 6 0 6 に導かれた ) ホストキャッシュの変更されたコピーがホスト側キャッシュコヒーレンスプロトコルによって既に降格されているのを示唆していることに留意されたい。データブロックがシャドウディレクトリ 3 2 4 に `host - read ( S )` 状態 6 0 4 を有する場合、ブリッジプロトコル制御ロジック 3 2 2 は、状態を変化させない。

30

## 【 0 0 2 8 】

ホスト側からの所有権読み出し ( H . w ) アクセスに応じて、データブロックがシャドウディレクトリに `no - host - copy ( I )` 状態 6 0 2 又は `host - read ( S )` 状態 6 0 4 を有する場合、ブリッジプロトコル制御ロジック 3 2 2 は、状態を `host - written ( M )` 状態 6 0 6 に遷移させる ( 7 0 6 又は 7 1 1 )。また、ブリッジプロトコル制御ロジック 3 2 2 は、プロセッサインメモリキャッシュコヒーレンスドメインの何れかのキャッシュ (例えば、レベル 1 キャッシュ) のデータブロックの何れかのコピーを無効にし、要求されたデータで所有権読み出しアクセスに応答する。データブロックがシャドウディレクトリ 3 2 4 に `host - written ( M )` 状態 6 0 6 を有する場合、ブリッジプロトコル制御ロジック 3 2 2 は、状態を変化させず、要求されたデータで読み出し要求に応答する。

40

## 【 0 0 2 9 】

50

ホームノードは、追加のホストコヒーレンス動作を実行することができる。例えば、M S I、M E S I及びM O E S IキャッシュコヒーレンスプロトコルにおけるI又はSからE、M又はOへの全ての状態遷移（または、他のプロトコルにおける同等の遷移）に対して、ホームノードは、ホームプロセッサインメモリ（例えば、プロセッサインメモリ102）へのtransition-to-M要求を生成しなければならない。これは、割り当てられていないラインの全てがデフォルトでプロセッサインメモリ上のhost-read(S)状態604であると仮定するようにホストプロトコルを変更することによって、トリガされてもよい。transition-to-M要求は、メモリインタフェースプロトコルがそうすることをサポートする場合、書き込みに先行する読み出し要求（割り当てのための）と関連していてもよい。少なくとも1つの実施形態では、ホームノードは、最後にキャッシュされたコピーがホスト側のキャッシュコヒーレンスドメイン全体から追い出されたことに応じて、プロセッサインメモリへのtransition-to-I要求を生成する。この要求は、ホスト側からの最後の追い出し時にダーティデータからフラッシュすることと併せて生成することができる。この要求の生成は、後述するように、メモリ側ラストレベルキャッシュを有するホストに影響を与える可能性がある。他の実施形態においては、ホストは、このような要求を生成しない。

10

#### 【0030】

図5に戻ると、少なくとも1つの実施形態では、プロセッサロジック104は、プロセッサ側（例えば、GPU CU 214、CPUコア216及びハードウェアアクセラレータ218のレベル1キャッシュ）のライトバックキャッシュを含む。結果として、各データブロックのブリッジコヒーレンス状態は、追加の状態であるprocessor-in-memory-modified(P)状態を用いて、データブロックの何れかの部分がプロセッサインメモリ102によってプロセッサ側キャッシュにおいて変更されたか否かも示す。図7は、シャドウディレクトリ324内のブリッジコヒーレンス状態情報への更新に関するブリッジプロトコル制御ロジック322の変更された状態図である。データブロックがシャドウディレクトリ324にprocessor-in-memory-modified(P)状態608を有する場合、ブリッジプロトコル制御ロジック322は、プロセッサインメモリの外部からの読み出し又は書き込みアクセスに応じて、更新されたデータをプロセッサインメモリキャッシュから提供し、（外部要求が排他又は書き込み許可用である場合に）必要に応じて、ローカルコピーを無効化し、データブロックのブリッジコヒーレンス状態を、遷移710又は遷移717を介して適切に変更する。少なくとも1つの実施形態では、プロセッサインメモリのプロセッササイドキャッシュ内の特定のデータブロックの変更されたコピーがキャッシュから追い出されると(P.e)、特定のデータブロックの状態は、シャドウディレクトリ324のno-host-copy(I)状態602(715)に遷移する。

20

30

#### 【0031】

少なくとも1つの実施形態では、ホームノードは、特定のデータブロックの最後にキャッシュされたコピーがホストサイドキャッシュコヒーレンスドメイン全体から追い出されたことに応じて、transition-to-I要求をホームプロセッサインメモリに生成する。特定のデータブロックがシャドウディレクトリ324にhost-read(S)状態604又はhost-written(M)状態606を有する場合、状態が、シャドウディレクトリ324のno-host-copy(I)状態602に遷移する。この要求は、最後の追い出しの際にダーティデータをフラッシュすることと併せて生成することができる。この要求生成は、メモリ側ラストレベルキャッシュを有するホストに影響を与え得る。このような要求は、キャッシュコヒーレンスブリッジプロトコルではオプションであり（図7には示されていない）、他の実施形態では、ホームノードは、かかる要求を生成しない。

40

#### 【0032】

上述したキャッシュコヒーレンスブリッジプロトコルは、コヒーレンス追跡の粒度とは無関係である。キャッシュコヒーレンスブリッジプロトコルは、キャッシュライン粒度を

50

有してもよいし、より粗い粒度（例えば、複数のキャッシュラインを含む領域のコヒーレンス）を有してもよい。したがって、より粗い粒度を有するキャッシュコヒーレンスブリッジプロトコルの実装は、より少ない状態記憶、及び、より小さいシャドウディレクトリしか必要としない。本明細書に記載されたキャッシュコヒーレンスブリッジプロトコルは、可変粒度でも実施可能である。例えば、シャドウディレクトリの一部を使用して、より粗い粒度を有するデータブロックを追跡してもよく、シャドウディレクトリの他の部分を使用して、より細かい粒度を有するデータブロックを追跡してもよい。

#### 【 0 0 3 3 】

キャッシュコヒーレンスブリッジプロトコルの少なくとも1つの実施形態では、ホストコヒーレンスプロトコルのコヒーレンス粒度は、キャッシュコヒーレンスブリッジプロトコルのコヒーレンス粒度及びプロセッサインメモリのコヒーレンス粒度と異なっている。例えば、個々のキャッシュラインよりも粗い粒度（例えば、4KB）でコヒーレンス許可を追跡する領域コヒーレンスは、コヒーレンストラフィック（例えば、ホストとプロセッサインメモリとの間のコヒーレンストラフィック）を大幅に低減する可能性がある。したがって、ブリッジコヒーレンスプロトコルは、既存のホスト側コヒーレンスプロトコルが従来のキャッシュライン粒度（例えば、64B）で動作し続ける間に、粗い粒度（例えば、4KB）で動作することができる。さらに、ホストコヒーレンスプロトコル及びプロセッサインメモリキャッシュコヒーレンスプロトコルの異なるコヒーレンス粒度のキャッシュコヒーレンスブリッジプロトコルのサポートは、異なるベンダからのコンポーネント（例えば、異なるキャッシュラインサイズを有する製品）間の相互運用を容易にする。効率を向上させるために、システム内のより粗いコヒーレンス粒度は、そのシステム内のより細かいコヒーレンス粒度の倍数であり、このことは、従来のシステムが2の累乗数のバイトのコヒーレンス粒度を使用することから、典型的である。

#### 【 0 0 3 4 】

少なくとも1つの実施形態では、ブリッジ変換ユニットは、キャッシュコヒーレンス要求を、粗い粒度のコヒーレンス許可要求（例えば、プロセッサインメモリから4KBの領域要求）からより細かい粒度のコヒーレンス許可要求（例えば、ホストへの64Bのキャッシュライン）に変換する。この変換は、粗い要求のアドレス範囲の全体に及び（例えば、プロセッサインメモリからの単一の4KBの要求に応じて、ホストへの64個の64Bの要求）、受信側の対応するコヒーレンスコントローラに要求を発行する。次に、受信ブリッジ変換ユニットは、要求するブリッジ変換ユニットに完了を通知する前に、そのセット内の全ての要求に対する応答を収集する。

#### 【 0 0 3 5 】

少なくとも1つの実施形態では、送信ブリッジ変換ユニット又は受信ブリッジ変換ユニットは、細かい粒度の許可要求（例えば、ホストからの64Bの要求）から、粗い許可（例えば、プロセッサインメモリの4KBの領域）を追跡するレスポンド（responder）に変換する。送信ブリッジ変換ユニット又は受信ブリッジ変換ユニットは、その要求を、元の要求のアドレス範囲を含む粗い粒度の領域に変換する。リクエスタ（例えば、ホスト）は、要求した領域よりも大きい領域のコヒーレンス許可を受信したことに気付かず、同じ領域内で追加の要求を引き続き発行する可能性があることに留意されたい。非効率を低減するために、送信ブリッジ変換ユニットは、最近受信した領域許可のキャッシュを維持することができる。同じリクエスタからの後続の細かい粒度の要求がこの領域許可キャッシュのエントリと一致する場合、送信ブリッジ変換ユニットは、他のエンティティへの要求を開始することなく完了を通知することができる。

#### 【 0 0 3 6 】

上述したように、細かい粒度でコヒーレンス降格に関するメッセージを生成するシステムでは、性能を強化するためにのみ降格メッセージが用いられることから、送信ブリッジ変換ユニットは、降格メッセージを破棄する場合がある。他の実施形態では、細かい粒度で、送信ブリッジ変換ユニットは、粗い粒度でコヒーレンスを追跡する受信者に降格を伝える前に、連続する粗い粒度の領域が降格されるまで降格メッセージを収集する。送信ブ

10

20

30

40

50

リッジ変換ユニットは、細かい粒度の降格を伝えてもよく、受信ブリッジ変換ユニットが、それらを粗い粒度の領域に集約するのを予期してもよい。粗い粒度の領域の他の部分への降格要求を生成する送信ブリッジ変換ユニットは、粗い粒度の領域全体の降格を強制し、その領域の降格を伝えてもよい。少なくとも1つの実施形態では、送信ブリッジ変換ユニットは、先行するメモリ要求に基づくヒューリスティクスに従って上記の1つ以上のオプションを実行してもよい。

**【0037】**

図3に戻ると、シャドウディレクトリ324は、各データブロックのブリッジコヒーレンス状態をプロセッサインメモリ内に記憶するために、個別の記憶構造（例えば、個別のダイのプロセッサインメモリ102におけるSRAM、又は、プロセッサインメモリ102と共にスタックされた分散されたSRAM）を用いて実装されてもよい。この記憶構造は、複数のメモリチャンネルに亘って物理的に分散されてもよいし、プロセッサインメモリの各メモリチャンネルは、この構造に対する帯域幅制限を緩和し得る個別のテーブルを維持してもよい。

**【0038】**

少なくとも1つの実施形態では、シャドウディレクトリ324は、例えばDRAMが十分な追加のビットを含むようにカスタマイズされ、非標準的なアドレッシングのサポートを必要としないような設計において、メモリと統合され、データブロック毎のブリッジコヒーレンス状態を、データブロックを記憶するメモリの同じ行のビットのサブセットに記憶する。少なくとも1つの実施形態では、シャドウディレクトリ324は、データブロックとは別に、ブリッジコヒーレンス状態の記憶専用の各メモリチャンネルのメモリ空間のサブセット（例えば、メモリ空間の最上位のアドレス範囲）に配置される。アドレッシングは、特定のメモリアドレスのキャッシュコヒーレンス状態が、DRAMページの競合を低減するために、特定のメモリアドレスのデータを含むチャンネル以外のチャンネルに常駐するように設計されてもよい。アドレス空間が、プロセッサインメモリデバイス間、及び、比較的細かい粒度（すなわち、単一のメモリチャンネルの容量よりはるかに小さい粒度）でプロセッサインメモリ内のチャンネル間で分散される場合には、そのマッピング技術は、特別なアドレッシングのサポートが殆どなくてもよい。DRAM及び記載されたメモリ空間の代替のサブセットを用いて実装されたシャドウディレクトリは、最近アクセスされたシャドウディレクトリエントリを記憶するためのSRAMキャッシュによって拡張され、シャドウディレクトリ状態にアクセスして変更するために必要なDRAMアクセスを低減することができる。

**【0039】**

プロセッサインメモリの観点からは、ホスト側コヒーレンスドメインに含まれていないホストのメモリサイドラストレベルキャッシュが、ホストの一部として見られる。キャッシュコヒーレンスブリッジプロトコルは、上述したシャドウディレクトリ追跡スキームにおいて、デフォルトでホストにキャッシュされたデータブロックの状態を保存する。メモリサイドキャッシュが変更されたデータを含む唯一のものである場合（すなわち、ホストディレクトリによって追跡された他のキャッシュに当該データのコピーが存在しない場合）、ホストキャッシュからの変更されたデータに対する要求を含むプロセッサインメモリからのキャッシュコヒーレンス要求に応じて、キャッシュコヒーレンスブリッジプロトコルは、メモリサイドラストレベルキャッシュを、他のホストキャッシュであるかのように変更してもよい。ホストの観点からは、メモリサイドキャッシュは、キャッシュコヒーレンスブリッジプロトコルからの特別な処理を必要としない。しかしながら、上述したように、ディレクトリ追跡されたホストキャッシュからの追い出しがプロセッサインメモリに通知される場合、これらのメッセージは、ホストメモリサイドキャッシュのデータを適切に考慮する必要がある。メモリサイドキャッシュの状態がこれらのメッセージと一致しない場合、これらのメッセージは、降格をプロセッサインメモリに伝えない。オプションとして、キャッシュコヒーレンスブリッジプロトコルは、メモリサイドラストレベルキャッシュからのデータブロック追い出しに応じて、このようなメッセージを生成する。

10

20

30

40

50

【 0 0 4 0 】

このように、キャッシュコヒーレンスブリッジプロトコルについて説明した。本明細書で説明した構造は、プロセッサ上で実行されるソフトウェア（ファームウェアを含む）を用いて、又は、ソフトウェアとハードウェアとの組み合わせによって実装されてもよい。ソフトウェアは、本明細書で説明したように、少なくとも1つの有形の（すなわち、非一時的な）コンピュータ可読媒体内でコード化されてもよい。本明細書で言及されるように、有形のコンピュータ可読媒体は、ディスク、テープ若しくは他の時期記憶媒体、光学記憶媒体、又は、電子記憶媒体を含む。本明細書に記載された発明の説明は例示的なものであり、以下の特許請求の範囲に記載された本発明の範囲を限定するものではない。例えば、本発明は、プロセッサインメモリデバイスがキャッシュを含む実施形態で説明されているが、当業者であれば、本明細書の教示が、キャッシュを含まない単純なプロセッサインメモリデバイスを利用し得るが、ホストキャッシュがプロセッサインメモリアクセスとコヒーレントであることを必要としているのを理解するであろう。本明細書に開示された実施形態の変形及び変更は、添付の特許請求の範囲に記載された本発明の範囲から逸脱することなく、本明細書に記載された発明に基づいてなされてもよい。

10

【 図 面 】

【 図 1 】

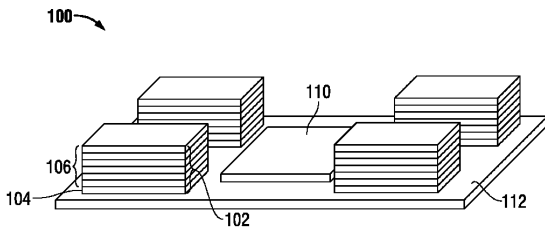
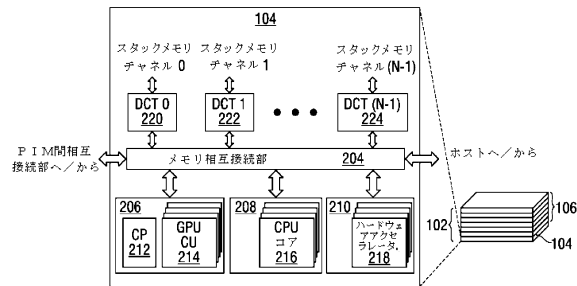


FIG. 1

【 図 2 】



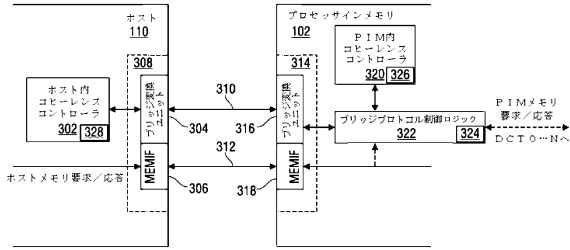
20

30

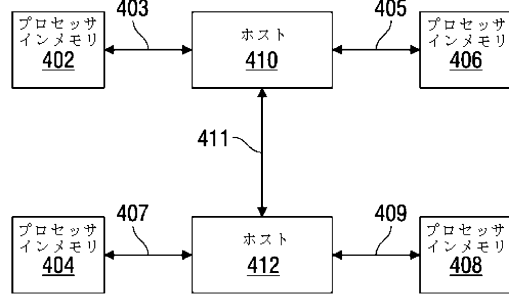
40

50

【図3】

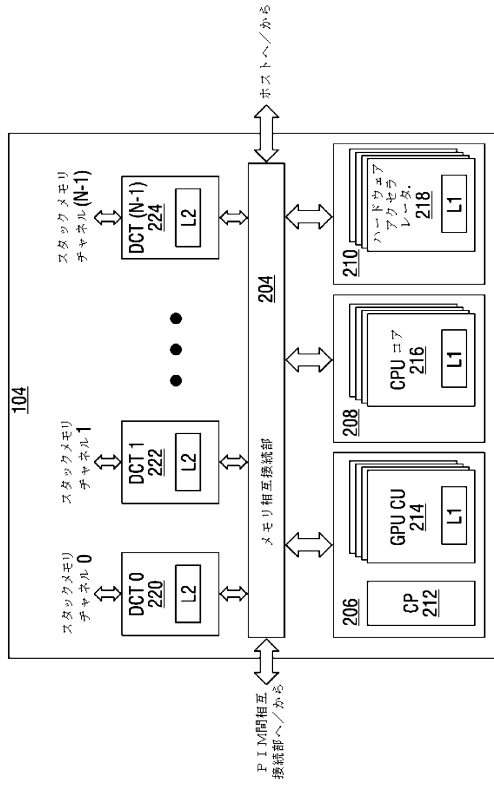


【図4】



10

【図5】



【図6】

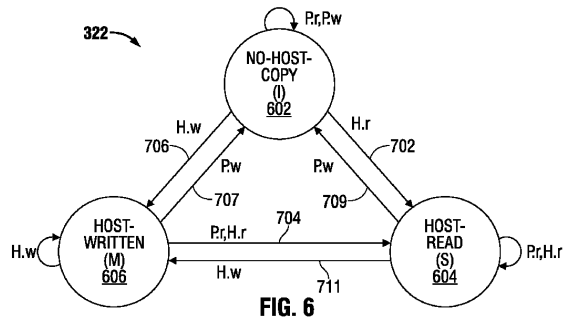


FIG. 6

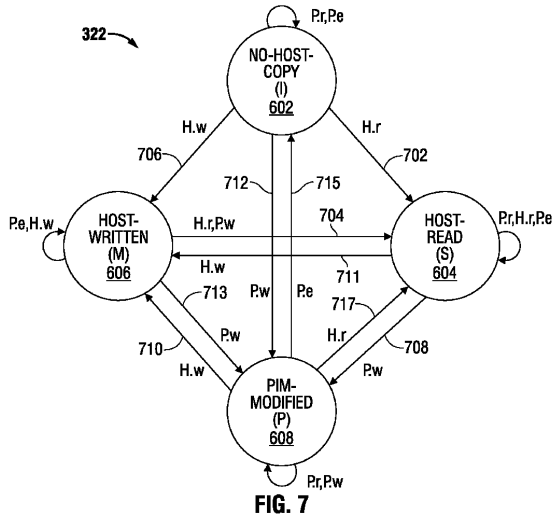
20

30

40

50

【 7 】



10

20

30

40

50

## フロントページの続き

アメリカ合衆国 98007 ワシントン州、ベルビュー、2002 156番 アベニュー ノース  
イースト、スイート 300、エイ・エム・ディ・ユー・エス ベルビュー

(72)発明者 ヌワン ジャヤセーナ

アメリカ合衆国 94088 カリフォルニア州、サニーベール、ワン・エイ・エム・ディ・プレイ  
ス、エイ・エム・ディ・ユー・エス サニーベール

審査官 北村 学

(56)参考文献 米国特許出願公開第2007/0022254(US, A1)

米国特許出願公開第2001/0034816(US, A1)

米国特許出願公開第2014/0149682(US, A1)

米国特許出願公開第2015/0324290(US, A1)

米国特許出願公開第2014/0181417(US, A1)

国際公開第2015/171914(WO, A1)

特開平10-187633(JP, A)

特開平08-016474(JP, A)

特表2015-503160(JP, A)

特開2000-250882(JP, A)

(58)調査した分野 (Int.Cl., DB名)

IPC G06F 12/08 - 12/128