

(19) 日本国特許庁 (JP)

(12) 公 開 特 許 公 報 (A)

(11) 特許出願公開番号
特開2010-27048
(P2010-27048A)

(43) 公開日 平成22年2月4日 (2010. 2. 4)

(51) Int.Cl.	F I	テーマコード (参考)
G 0 6 F 12/08 (2006.01)	G O 6 F 12/08 5 3 1 E	5 B 0 0 5
G 0 6 F 13/42 (2006.01)	G O 6 F 12/08 5 1 5 P	5 B 0 7 7
	G O 6 F 12/08 5 3 1 B	
	G O 6 F 12/08 5 1 9 C	
	G O 6 F 12/08 5 5 1 C	

審査請求 有 請求項の数 28 O L 外国語出願 (全 29 頁) 最終頁に続く

(21) 出願番号 特願2009-159801 (P2009-159801)	(71) 出願人 593096712 インテル コーポレーション アメリカ合衆国 95052 カリフォル ニア州 サンタ クララ ミッション カ レッジ ブールバード 2200
(22) 出願日 平成21年7月6日 (2009. 7. 6)	(74) 代理人 100070150 弁理士 伊東 忠彦
(31) 優先権主張番号 12/168, 606	(74) 代理人 100091214 弁理士 大貫 進介
(32) 優先日 平成20年7月7日 (2008. 7. 7)	(74) 代理人 100107766 弁理士 伊東 忠重
(33) 優先権主張国 米国 (US)	(72) 発明者 ロバート エイチ. ビアーズ アメリカ合衆国 97006 オレゴン州 ビーヴァートン ノースウェスト パン ドン コート 17245 最終頁に続く

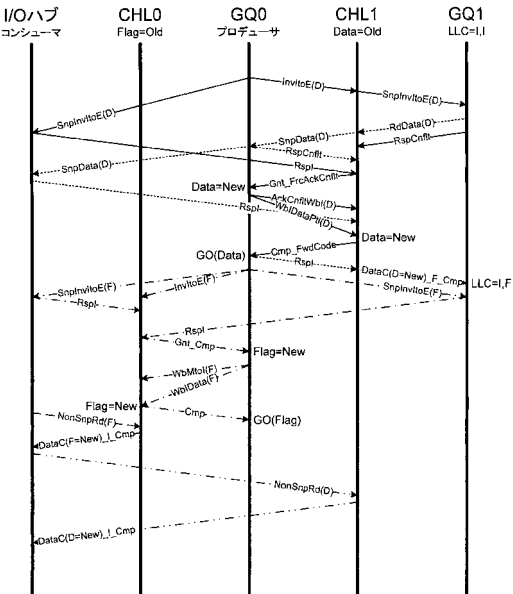
(54) 【発明の名称】 部分書込と非スヌープアクセスとの間のメモリ順序付け要件の充足

(57) 【要約】

【課題】部分的且つ非コヒーレントなメモリアクセスの観点からキャッシュ・コヒーレンス・リンクに基づく相互接続でメモリ順序付けを保つ方法及び装置を提供する。

【解決手段】例えば部分書込のような部分的なメモリアクセスに関連するコンフリクトが検出される場合に、ライトバックフェーズがコンフリクトフェーズで挿入され、ホームエージェントへ部分的データをライトバックする。コンフリクトフェーズでライトバックフェーズを挿入するメッセージの例には、コンフリクトを認めて、コンフリクトフェーズの開始時にライトバックマーカを、コンフリクトフェーズ前にライトバックマーカメッセージを、コンフリクトフェーズ内でライトバックマーカメッセージを、コンフリクトフェーズ後にライトバックマーカメッセージ及びポスタブル・メッセージを提供するAcknowledge Conflict Write-backメッセージがある。

【選択図】 図 5 a



【特許請求の範囲】**【請求項 1】**

部分的な書込を実行して部分的データを更新する第1のロジックと、

前記部分的データに関連するコンフリクトメッセージの受信に応答してコンフリクトフェーズでライトバックフェーズを開始し、該ライトバックフェーズは、前記部分的データを該部分的データに関連するホームエージェントへ供給する第2のロジックと、

前記第2のロジックへ結合され、前記ライトバックフェーズの間前記ホームエージェントへ前記部分的データを供給する第3のロジックと
を有する装置。

【請求項 2】

10

前記第1のロジック及び前記第2のロジックは、相互接続アーキテクチャ階層化スタックの物理レイヤに関連するプロトコルロジックであり、

前記第3のロジックは、前記相互接続アーキテクチャ階層化スタックの物理レイヤに関連する物理ロジックを有する、請求項1記載の装置。

【請求項 3】

コンフリクトフェーズでライトバックフェーズを開始する前記第2のロジックは、確認応答コンフリクト及びライトバックメッセージを生成する前記第2のロジックを有する、請求項1記載の装置。

【請求項 4】

20

前記部分的データに関連するコンフリクトメッセージの受信に応答して確認応答コンフリクト及びライトバックメッセージを生成する前記第2のロジックは、前記ホームエージェントからのフォース確認応答コンフリクトメッセージの受信に応答して前記確認応答コンフリクト及びライトバックメッセージを生成する前記第2のロジックを有する、請求項3記載の装置。

【請求項 5】

前記確認応答コンフリクト及びライトバックメッセージは、Acknowledge Conflict Write-back Invalid (AckCnfltWbI) メッセージを有し、

前記ホームエージェントからの前記フォース確認応答コンフリクトメッセージは、Grant_Force Acknowledge Conflict (Gnt_FrcAckCnflt) メッセージを有する、請求項4記載の装置。

30

【請求項 6】

前記部分的データに関連するコンフリクトメッセージの受信に応答してコンフリクトフェーズでライトバックフェーズを開始する前記第2のロジックは、前記部分的データに関連するコンフリクトメッセージの受信に応答してライトバックマーカーメッセージ及びライトバックデータメッセージを生成する前記第2のロジックを有する、請求項1記載の装置。

【請求項 7】

前記ライトバックマーカーメッセージは、Write-back Modified to Invalid (WbMtoI) マーカーメッセージを有し、

ライトバックデータメッセージは、Write-Back Invalid Data Partial (WbIDataPartial) メッセージを有し、

40

前記コンフリクトメッセージは、Grant_Force Acknowledge Conflict (Gnt_FrcAckCnflt) メッセージを有する、請求項6記載の装置。

【請求項 8】

前記コンフリクトメッセージは、Grant_Force Acknowledge Conflict (Gnt_FrcAckCnflt) メッセージと、Acknowledge Conflict (AckCnflt) メッセージと、Complete_Forward Code (Cmp_FwdCode) メッセージとを有するメッセージのグループから選択される、請求項7記載の装置。

【請求項 9】

前記部分的データに関連するコンフリクトメッセージの受信に応答してコンフリクトフ

50

エーズでライトバックフェーズを開始する前記第2のロジックは、前記部分的データに関連するコンフリクトメッセージの受信に 응답してポスタブル・ライトバックメッセージを生成する前記第2のロジックを有する、請求項1記載の装置。

【請求項10】

前記ポスタブル・ライトバックメッセージは、Response Invalid Write-back Postable (RspIWbPost) メッセージと、Write-back Invalid Data Partial Postable (WbIDPP) メッセージと、Write-back Postable (WbPost) メッセージとを有するグループから選択されるメッセージを有し、

前記コンフリクトメッセージは、Grant_Force Acknowledge Conflict (Gnt_FrcAckCnflt) メッセージと、Acknowledge Conflict (AckCnflt) メッセージと、Complete_Forward Code (Cmp_FwdCode) メッセージとを有するメッセージのグループから選択される、請求項9記載の装置。

10

【請求項11】

キャッシュラインを有するキャッシュメモリと、

前記キャッシュメモリへ結合され、前記キャッシュラインへの部分的データの部分書込を実行する第1のロジックと、

前記部分的データに関連するホームノードからのコンフリクトメッセージを受信する第2のロジックと、

前記第2のロジックへ結合され、前記コンフリクトメッセージの確認応答と、前記部分的データが、前記第1のロジックが当該部分的データの前記部分書込を実行し且つ前記第2のロジックが前記コンフリクトメッセージを受信したことに応答して、データメッセージにおいて送信されるべきことを前記ホームノードへ示す結合メッセージを生成する第3のロジックと、

20

前記第3のロジックへ結合され、前記結合メッセージを前記ホームノードへ送信する第4のロジックとを有する装置。

【請求項12】

前記結合データは、Acknowledge Conflict Write-back (AckCnfltWb) メッセージを有する、請求項11記載の装置。

【請求項13】

前記コンフリクトメッセージは、Force Acknowledge Conflict (FrcAckCnflt) メッセージを有する、請求項12記載の装置。

30

【請求項14】

前記第3のロジックは、次のトランザクションを開始する前に前記AckCnfltWbメッセージ及び前記データメッセージに関連する前記ホームノードからの完了メッセージを待つ、請求項12記載の装置。

【請求項15】

前記完了メッセージは、Complete_Forward Code (Cmp_FwdCode) メッセージを有する、請求項14記載の装置。

【請求項16】

前記第2のロジック及び前記第4のロジックは、相互接続階層化スタックの物理レイヤロジックに関連し、

前記第3のロジックは、前記相互接続階層化スタックのプロトコルレイヤロジックに関連する、請求項11記載の装置。

40

【請求項17】

部分的データに関連するホームノードからのコンフリクトメッセージの受信と、当該ピアノードが前記部分的データに対して部分書込を実行することに応答してコンフリクトフェーズでライトバックフェーズを開始するライトバックメッセージを生成する第1のロジックを有するピアノードを有し、

前記ホームノードは、ポイント・ツー・ポイントリンクを介して前記ピアノードへ結合

50

され、

前記ホームノードは、前記ライトバックフェーズの間の前記ピアノードからの前記部分的データの受信に応答して前記コンフリクトフェーズで前記ライトバックフェーズに係る完了メッセージを生成する第2のロジックを有する、システム。

【請求項18】

前記ライトバックメッセージは、Acknowledge Conflict Write-backメッセージと、Write-back to Invalidメッセージと、Response Write-back Postableメッセージと、Write-back Data Partial Postableメッセージと、Write-back Postableメッセージとを有するグループから選択される、請求項17記載のシステム。

【請求項19】

前記第1のロジックは、相互接続アーキテクチャ・コヒーレンス・プロトコルを実施するプロトコルロジックを有し、

前記第2のロジックは、前記相互接続アーキテクチャ・コヒーレンス・プロトコルを実施するプロトコルロジックを有する、請求項17記載のシステム。

【請求項20】

前記ピアノードに含まれる前記第1のロジックは、次の書込トランザクションを開始する前に前記ホームノードから前記ライトバックフェーズに係る前記完了メッセージを受信するまで待機する、請求項17記載のシステム。

【請求項21】

前記次の書込トランザクションは、前記部分的データが取り出される準備ができていることを第2のノードに示すべく設定されるようフラグを更新することを有し、

前記第2のノードは、前記フラグの非スヌープ読出を実行し、

前記フラグが設定されることに応答して、前記第2のノードは、前記ホームノードからの前記部分的データの第2の非スヌープ読出を実行する、請求項20記載のシステム。

【請求項22】

部分的データをピアキャッシングエージェントで新たな部分的データへと更新する段階と、

前記部分的データに関連して前記ピアキャッシングエージェントによりコンフリクトメッセージを受信する段階と、

前記新たな部分的データが、前記ピアキャッシングエージェントにより前記コンフリクトメッセージを受信することに応答して、ホームエージェントへライトバックされることを示すライトバックメッセージを生成する段階と、

前記ホームエージェントが前記ピアキャッシングエージェントから新たな部分的データを受信した後、前記ピアキャッシングエージェントにより前記ホームノードから完了メッセージを受信する段階と

を有する方法。

【請求項23】

前記部分的データが変形状態で前記ピアキャッシングエージェントに保持されていることに応答して、前記部分的データを前記新たな部分的データへと更新する前に前記部分的データを前記ホームエージェントへライトバックする段階を更に有し、

前記部分的データを前記新たな部分的データへと更新する段階は、前記コンフリクトメッセージを受信する前に起こる、請求項22記載の方法。

【請求項24】

前記ピアキャッシングエージェントで前記部分的データを前記新たな部分的データへと更新する段階は、前記コンフリクトメッセージを受信した後に起こる、請求項22記載の方法。

【請求項25】

前記ホームエージェントでの応答コンフリクトメッセージの受信に応答して前記ホームエージェントにより前記コンフリクトメッセージを生成する段階と、

前記ホームエージェントにより前記ピアキャッシングエージェントから前記新たな部分

10

20

30

40

50

的データを受信する段階と、

前記ホームエージェントに関連するメモリへ前記新たな部分的データをコミットする段階と、

前記ホームエージェントにより前記新たな部分的データの受信と、前記ホームエージェントに関連するメモリへの前記新たな部分的データのコミットとに応答して、前記ホームエージェントにより前記完了メッセージを生成する段階と

を更に有する請求項 2 2 記載の方法。

【請求項 2 6】

前記ホームエージェントから前記完了メッセージを受信した後に前記ピアキャッシングエージェントによりフラグへ書き込み、該フラグを設定する段階と、

第 2 のピアキャッシングエージェントにより前記フラグの非コヒーレント読出を実行する段階と、

前記フラグが前記第 2 のピアキャッシングエージェントによる当該フラグの前記非コヒーレント読出で設定されることに応答して、前記第 2 のピアキャッシングエージェントにより前記ホームエージェントから前記新たな部分的データの非コヒーレント読出を実行する段階と

を更に有する請求項 2 5 記載の方法。

【請求項 2 7】

前記ライトバックメッセージは、Acknowledge Conflict Write-backメッセージと、Write-back to Invalidメッセージと、Response Write-back Postableメッセージと、Write-back Data Partial Postableメッセージと、Write-back Postableメッセージとを有するグループから選択される、請求項 1 7 記載のシステム。

【請求項 2 8】

前記コンフリクトメッセージはフォース確認応答コンフリクトメッセージを有し、

前記完了メッセージは完了メッセージ又は完了フォワードコードメッセージである、請求項 2 7 記載のシステム。

【発明の詳細な説明】

【技術分野】

【0 0 0 1】

本発明は、高速ポイント・ツー・ポイント相互接続リンクに関し、より具体的には、コヒーレント相互接続アーキテクチャを実施するコヒーレンス・プロトコルに関する。

【背景技術】

【0 0 0 2】

電子システムが複数のキャッシュメモリを有する場合に、システムに保持されているデータの妥当性、すなわちコヒーレンスは、データ演算のインテグリティを確かにするよう保たれている。コヒーレンスは、しばしば、キャッシュ・コヒーレンス・プロトコルに従ってデータを扱うことによって達成される。キャッシュ及び/又はプロセッサの数が増えるにつれて、コヒーレンスを保つ複雑性は増す。これにより、潜在的に、システムの複数のコンポーネントの間でのコンフリクトの数は増大することとなる。

【0 0 0 3】

例えば、複数のコンポーネント（例えば、キャッシュメモリ、プロセッサ）が同じデータブロックを要求する場合に、これらの複数のコンポーネントの間でのコンフリクトは、データの妥当性を保つように解消される。以前のキャッシュ・コヒーレンス・プロトコルは、通常、コンフリクト解決に関与する単一コンポーネントを有する。しかし、システムの複雑性が増すにつれ、コンフリクト解決のための単一コンポーネントへの依存はシステム全体の性能を低下させうる。

【0 0 0 4】

一般に、キャッシュ・コヒーレンスを提供する 2 つの基本的なスキームがある。それは、スヌーピング（現在、しばしば、対称型マルチプロセッシング（SMP）と呼ばれる。）及びディレクトリ（しばしば、分散共有メモリ（DSM）と呼ばれる。）である。これ

10

20

30

40

50

ら 2 つのスキームの間の基本的な違いは、メタ情報、すなわち、どこにキャッシュラインのコピーが保持されているかに関する情報への配置及びアクセスを中心に展開する。

【 0 0 0 5 】

ディレクトリに基づくスキームに関して、キャッシュ無しデータが記憶されている固定位置に加えて、どこにキャッシュコピーが存在するかを示す固定ロケーション、すなわちディレクトリがある。新しい方法でキャッシュラインにアクセスするために、ノードは、ディレクトリを有するノード（通常、キャッシュ無しデータを有する同じノードである。）と通信すべきであり、主記憶コピーが有効である場合に応答ノードがデータを供給することを可能にする。ディレクトリシステムにおけるこのようなノードは、ホームノードと呼ばれる。

10

【 0 0 0 6 】

ディレクトリは、二通りに分配され得る。第 1 に、主記憶データ（キャッシュ無しリボジトリ）は、しばしば、ノード間で分配され、一方、ディレクトリも同じように分配される。第 2 に、ラインがキャッシュされるかどうか及びラインがキャッシュされる場合にはどこに単一コピーが存在するかについて、少ない情報としてホームノードに保持しながら、メタ情報自体が分配され得る。例えば、S C I はこのスキームを使用し、キャッシュコピーを有する各ノードは、キャッシュコピーを有する他のノードへのリンクを保ち、集合的に完全なディレクトリを保持する。

【 0 0 0 7 】

スヌーピングキャッシュに関して、メタ情報は、キャッシュラインの有効なコピーの夫々が、キャッシュラインにアクセスして然るべく応答する要求を認識すべきユニットによって保持されるように、キャッシュコピー自体を有して分配される。通常、データがキャッシュされない状態で、すなわち、その元のロケーションに、保持されているリボジトリがある。しばしば、この元のロケーションに関与するエージェント又はユニットは、ホームノードと呼ばれる。キャッシュラインへのアクセスを要求する場合に、要求ノードは、しばしば、必要とされるパーミッションと共に、要求されるキャッシュラインのアドレスを送信し、要求されるキャッシュラインのコピーを保持するノードは然るべく応答する。

20

【 発明の概要 】

【 発明が解決しようとする課題 】

【 0 0 0 8 】

しかし、現在のコヒーレンス及びメッセージングプロトコルの幾つかで、キャッシュラインの部分読出及び非スヌープメモリアクセス、すなわち、コヒーレントなアーキテクチャでの非コヒーレントアクセスは、或る環境下で、潜在的に、要求側へ無効なデータを転送する事態を生じさせる。しばしば、かかる環境は、プロトコル内のメモリ順序付け規則の違反に起因する。結果として、不正な動作が、無効なデータを受け取ったエージェントによって行われる。これは、潜在的に、システム内の不安定性につながる。

30

【 課題を解決するための手段 】

【 0 0 0 9 】

本発明の一実施形態で、

部分的な書込を実行して部分的データを更新する第 1 のロジックと、

40

前記部分的データに関連するコンフリクトメッセージの受信に応答してコンフリクトフェーズでライトバックフェーズを開始し、該ライトバックフェーズは、前記部分的データを該部分的データに関連するホームエージェントへ供給する第 2 のロジックと、

前記第 2 のロジックへ結合され、前記ライトバックフェーズの間前記ホームエージェントへ前記部分的データを供給する第 3 のロジックと
を有する装置が提供される。

【 0 0 1 0 】

また、本発明の一実施形態で、

キャッシュラインを有するキャッシュメモリと、

前記キャッシュメモリへ結合され、前記キャッシュラインへの部分的データの部分書込

50

を実行する第 1 のロジックと、

前記部分的データに関連するホームノードからのコンフリクトメッセージを受信する第 2 のロジックと、

前記第 2 のロジックへ結合され、前記コンフリクトメッセージの確認応答と、前記部分的データが、前記第 1 のロジックが当該部分的データの前記部分書込を実行し且つ前記第 2 のロジックが前記コンフリクトメッセージを受信したことに応答して、データメッセージにおいて送信されるべきこととを前記ホームノードへ示す結合メッセージを生成する第 3 のロジックと、

前記第 3 のロジックへ結合され、前記結合メッセージを前記ホームノードへ送信する第 4 のロジックと

10

を有する装置が提供される。

【0011】

また、本発明の一実施形態で、

部分的データに関連するホームノードからのコンフリクトメッセージの受信と、当該ピアノードが前記部分的データに対して部分書込を実行することとに併答してコンフリクトフェーズでライトバックフェーズを開始するライトバックメッセージを生成する第 1 のロジックを有するピアノードを有し、

前記ホームノードは、ポイント・ツー・ポイントリンクを介して前記ピアノードへ結合され、

前記ホームノードは、前記ライトバックフェーズの間の前記ピアノードからの前記部分的データの受信に併答して前記コンフリクトフェーズで前記ライトバックフェーズに係る完了メッセージを生成する第 2 のロジックを有する、システムが提供される。

20

【0012】

また、本発明の一実施形態で、

部分的データをピアキャッシングエージェントで新たな部分的データへと更新する段階と、

前記部分的データに関連して前記ピアキャッシングエージェントによりコンフリクトメッセージを受信する段階と、

前記新たな部分的データが、前記ピアキャッシングエージェントにより前記コンフリクトメッセージを受信することとに併答して、ホームエージェントへライトバックされることを示すライトバックメッセージを生成する段階と、

30

前記ホームエージェントが前記ピアキャッシングエージェントから新たな部分的データを受信した後、前記ピアキャッシングエージェントにより前記ホームノードから完了メッセージを受信する段階と

を有する方法が提供される。

【発明の効果】

【0013】

本発明の実施形態により、部分的且つ非コヒーレントなメモリアクセスの観点からキャッシュ・コヒーレンス・リンクに基づく相互接続においてメモリ順序付けを保つことが可能となる。

40

【図面の簡単な説明】

【0014】

【図 1】プロトコルアーキテクチャの実施例を表す。

【図 2】ポイント・ツー・ポイント相互接続を用いてチップセットへ結合される複数のプロセッサを有するシステムの実施例を表す。

【図 3】階層化入出力 (I/O) スタックを用いる双方向相互接続アーキテクチャに係るブロック図の実施例を表す。

【図 4 a】潜在的にメモリ順序付け違背を生じさせる複数のノード間のメッセージングプロトコルに係るプロトコルダイアグラムの実施例を表す。

【図 4 b】潜在的にメモリ順序付け違背を生じさせる複数のノード間のメッセージングプ

50

ロトコルに係るプロトコルダイアグラムの他の実施例を表す。

【図 5 a】適切なメモリ順序付け要求に従うようコンフリクトフェーズでメッセージを有するメッセージングプロトコルに係るプロトコルダイアグラムの実施例を表す。

【図 5 b】適切なメモリ順序付け要求に従うようコンフリクトフェーズでメッセージを有するメッセージングプロトコルに係るプロトコルダイアグラムの他の実施例を表す。

【図 6 a】適切なメモリ順序付け要求に従うようコンフリクトフェーズの前に挿入されるデータライトバックフェーズを有するメッセージングプロトコルに係るプロトコルダイアグラムの実施例を表す。

【図 6 b】適切なメモリ順序付け要求に従うようコンフリクトフェーズの前に挿入されるデータライトバックフェーズを有するメッセージングプロトコルに係るプロトコルダイアグラムの他の実施例を表す。

【図 6 c】適切なメモリ順序付け要求に従うようコンフリクトフェーズでメッセージを有するメッセージングプロトコルに係るプロトコルダイアグラムの他の実施例を表す。

【図 7 a】適切なメモリ順序付け要求に従うようコンフリクトフェーズの後にメッセージを有するメッセージングプロトコルに係るプロトコルダイアグラムの実施例を表す。

【図 7 b】適切なメモリ順序付け要求に従うようコンフリクトフェーズの後にメッセージを有するメッセージングプロトコルに係るプロトコルダイアグラムの他の実施例を表す。

【図 7 c】適切なメモリ順序付け要求に従うようコンフリクトフェーズの後にメッセージを有するメッセージングプロトコルに係るプロトコルダイアグラムの実施例を表す。

【発明を実施するための形態】

【0015】

以下、添付の図面を参照して、一例として本発明の実施形態をあらわす。なお、本発明はこの実施形態に限定されない。

【0016】

以下の記載で、特定の相互接続アーキテクチャ、特定のメッセージ、特定の擬似コード、特定の相互接続アーキテクチャレイヤ等の例のような多数の特定の詳細が、本発明の徹底的な理解を与えるために挙げられている。なお、当業者には明らかなように、これらの特定の詳細が本発明を実施するために用いられる必要はない。他の例で、特定の相互接続レイヤの構造上の詳細や、特定の非適合メッセージ及びプロトコルや、プロセッサ及びキャッシュの特定の動作上の詳細等、よく知られているコンポーネント又は方法は、本発明を不必要に不明りょうとすることを避けるために、詳細には記載されていない。

【0017】

ここで記載される方法及び装置は、相互接続アーキテクチャにおいて部分アクセスと非スヌープメモリアクセスとの間で適切なメモリ順序付けを提供するためのものである。特に、メモリ順序付け要件の順守は、主として、実例となるキャッシュ - コヒーレントリンクに基づく相互接続アーキテクチャに関して論じられている。なお、適切なメモリ順序付け要件を提供する方法及び装置はそのように限定されず、それらは如何なる相互接続アーキテクチャにより実施されてもよい。

【0018】

[実例となる相互接続アーキテクチャの実施例]

図 1 は、高レベル簡易プロトコルアーキテクチャの実施例を表す。一実施例で、プロトコルアーキテクチャは、衝突する要求の間と同様に、通常のトランザクションの間、アーキテクチャ内のキャッシュにおいてデータ間のデータ整合性を保つべきである。更に、プロトコルアーキテクチャは、一実施例で、また、それぞれの要求が満足され引込まれるように、夫々の要求についてフォワード・プログレス (forward progress) を提供する。

【0019】

プロトコルアーキテクチャは、ノード又はエージェントを幾つ有してもよい。一実施例で、ノードは、内部キャッシュメモリ、外部キャッシュメモリ、及び / 又は外部メモリと関連するプロセッサを有する。なお、ノード又はエージェントは、例えば、プロセッサ、

10

20

30

40

50

メモリコントローラハブ、及びI/Oハブ、汎用コントローラハブ、I/Oデバイス、I/Oデバイスの一群、あるいは、メモリにアクセス可能ないずれかの他のデバイス/コンポーネント等、相互接続アーキテクチャにおける如何なる装置又はそのような装置の一群であってもよい。他の実施例で、ノードは他の電子システムと相互接続されている電子システム（例えば、コンピュータシステム、モバイルデバイス）である。また、他のタイプのノード構成が使用されてもよい。

【0020】

表されるように、アーキテクチャ100は、キャッシングノード/エージェント105及び106と、ホームエージェント110乃至112とを有する。一実施例で、ホームエージェント110乃至112は、コヒーレントなメモリ空間の最終リポジトリであるメモリロケーションを保護すべきである。言い換えると、エージェント110は、コヒーレントなメモリ空間のデータについて複数の元の物理的なメモリロケーションに關与するホームエージェントである。ホームエージェント110乃至112は、キャッシングエージェント105及び106からのキャッシュ状態トランザクションの追跡と、キャッシングエージェント105及び106の間のコンフリクトの管理と、メモリへのインターフェースと、データ及び/又はオーナーシップの提供と、いずれかの他のホームエージェントに關連するタスクとを含むタスクの如何なる組合せも実行することができる。

10

【0021】

一実施例で、キャッシングエージェント105及び106は、キャッシュメモリ及びキャッシングI/Oプロキシエンティティを含むプロセッサ等、キャッシュメモリと關連するエージェントを有する。キャッシングエージェント105及び106は、コヒーレントメモリ空間への読出及び書込要求の実行、コヒーレントメモリ空間からのデータのキャッシュコピーの保持、及び他のピアキャッシングエージェントへのキャッシュコピーの供給等のタスク及びいずれかの他のタスクの如何なる組合せも実行することができる。キャッシングエージェント又はノードは、また、他のキャッシングエージェントに關してピアエージェント/ノードと呼ばれることがある。図1に具体的に現されていないが、プロトコルアーキテクチャは、入出力(I/O)デバイスに代わってトランザクションに参加し又はそれを經驗するI/Oハブ等の非キャッシングエージェントを更に有してよい。

20

【0022】

一実施例で、キャッシングエージェント105及び106並びにホームエージェント110乃至112は、ネットワーク構造101を介するメッセージの交換を通して、データ整合性を保持しフォワード・プログレスを提供する。一実施例で、構造101は、ポイント・ツー・ポイント相互接続ネットワークを通して1つのエージェント/ノードから他のエージェント/ノードへのメッセージの伝送を助ける。しばしば、図1は、基盤ネットワークのキャッシュ・コヒーレンス・プロトコルの抽象的な見方を表すと言われる。

30

【0023】

一実施例で、相互接続アーキテクチャ100は、下記の同時係属出願で記載されるように、新規の特徴の如何なる組合せも組み入れてよい。例えば、米国特許第6,922,756号(発明の名称「forward State for Use in Cache Coherency in a Multi-Node System」)は、フォワードキャッシュ・コヒーレンス状態(Fステート)の利用について記載する。更に、様々な状況でのコヒーレンス・プロトコルの例は、2004年4月27日に出願された米国特許出願第10/833,963号(事件整理番号P15925)(発明の名称「A Two-Hop Cache Coherency Protocol」)、2004年4月27日に出願された米国特許出願第10/833,965号(事件整理番号P18890)(発明の名称「A Messaging Protocol」)、2004年4月27日に出願された米国特許出願第10/833,977号(事件整理番号P18891)(発明の名称「A Cache Coherence Protocol」)、2006年1月11日に出願された米国特許出願第11/330,977号(事件整理番号P22376)(発明の名称「A Two-Hop Source Snop Based Cache Coherence Protocol」)、及び2006年1月11日に出願された米国特許出願第11/331,301号(事件整理番号P23106)(発明の名称「A

40

50

Two-Hop Source Snoop Based Messaging Protocol」)で論じられている。

【0024】

他の例は、2002年12月19日に出願された米国特許出願第10/325,427号(事件整理番号P13923)(発明の名称「Speculative Distributed Conflict Resolution for a Cache Coherency Protocol」)、2002年12月19日に出願された米国特許出願第10/326,234号(事件整理番号P13984)(発明の名称「Hierarchical Directories for Cache Coherency in a Multiprocessor System」)、2002年12月19日に出願された米国特許出願第10/324,711号(事件整理番号P13985)(発明の名称「Hierarchical Virtual Model of a Cache Hierarchy in a Multiprocessor System」)、及び2002年12月19日に出願された米国特許出願第10/326,232号(事件整理番号P13986)(発明の名称「Non-Speculative Distributed Conflict Resolution for a Cache Coherency Protocol」)を含む。

【0025】

前述の同時係属出願に記載される特徴は、本願明細書等に記載される実施例に組み込まれ得る点に留意すべきである。なお、本願明細書等に記載される実施例はそのように限定されず、それらは、潜在的に前述の特徴のいずれも有さないだけでなく、付加的な特徴を有してよい。

【0026】

図2は、ポイント・ツー・ポイント相互接続アーキテクチャを利用するチップセットへ結合されている複数のプロセッサを有するシステムの実施例を表す。図2のシステムは、また、複数のプロセッサを有してよい。それらのプロセッサのうち2つのプロセッサ205及び210しか明りょうさのために示されない。表されるように、プロセッサ205及び210は、夫々、2つの処理要素(PE)206及び207又は211及び212を有する。なお、プロセッサ205及び210には処理要素が幾つ含まれてもよい。

【0027】

処理要素は、スレッドユニット、処理ユニット、コンテキスト、論理プロセッサ、ハードウェアスレッド、コア、及び/又はいずれかの他の要素を言い、例えば実行状態又は構造上の状態等のプロセッサの状態を保つことができる。言い換えると、処理要素は、一実施例で、例えば、ソフトウェアスレッド、オペレーティングシステム、アプリケーション、又は他のコード等のコードと独立して関わるることができる如何なるハードウェアも言う。一例として、通常、物理プロセッサは集積回路を表し、これは潜在的にコア又はハードウェアスレッド等の他の処理要素を幾つでも有する。

【0028】

コアは、しばしば、独立した構造上の状態を保つことができる集積回路に配置されるロジックを言う。夫々の独立して保たれる構造上の状態は、少なくとも幾つかの専用の実行リソースと関連する。コアとは対照的に、物理スレッドと呼ばれることもあるハードウェアスレッドは、通常、独立した構造上の状態を保つことができる集積回路に配置されるあらゆるロジックを言う。独立して保たれる構造上の状態は、実行リソースへのアクセスを共有する。そのため、明らかなように、例えば単スレッド・アプリケーションの複数の複製等の複数のソフトウェアスレッドは、一実施例で、コア又はハードウェアスレッド等の前述の処理要素のいずれかの組合せを含む複数の処理要素で並行して実行され得る。

【0029】

また、プロセッサ205及び210の中にはリソース208及び213が表されている。リソース208及び213は、通常、レジスタ、ユニット、ロジック、ファームウェア、メモリ、及び他のリソースを有し、コードを実行し又は他のデバイスとインターフェース接続する。上述されるように、リソースの幾つかは、部分的に又は完全に処理要素のために設けられてよく、一方、残りのリソースは、処理要素の間で共有される。例えば、命令ポインタ及びリネームロジック等のより小さいリソースは、物理スレッドについて複製されてよい。リオーダー/リタイアメント・ユニットにおけるリオーダーバッファ、命令

ルックアサイド・トランスレーション・バッファ (ILTB)、負荷/記憶バッファ、及びキュー等の幾つかのリソースは、パーティショニングを通して共有されてよい。汎用内部レジスタ、ページテーブル・ベース・レジスタ、低レベルデータキャッシュ、データTLB、実行ユニット、及びアウト・オブ・オーダー・ユニット等の他のリソースは、潜在的に完全にスレッドの間で共有される。対照的に、コアは、第2レベルキャッシュ(L2)等のより高いレベルのキャッシュの少なくとも一部を共有しながら、専用の実行リソースを有してよい。

【0030】

一実施例で、リソース208及び213はプロセッサ・パイプラインを有する。プロセッサ・パイプラインは、パイプライン段を幾つでも有してよい。パイプライン段の一般的な例には、命令ポイント段、フェッチ段、デコード段、ドライブ段、及び割当て段、リネーム段、キュー段、リオーダー段、スケジュール段、ディスパッチ段、実行段、メモリアクセス段、並びにレジスタアクセス段が含まれる。ここに列挙される段は、あらゆる既知のパイプライン段がプロセッサ100に含まれうる場合に、プロセッサ・パイプライン段の例となる非包括的な羅列を有する点に留意すべきである。

10

【0031】

プロセッサ205及び210は、夫々、メモリコントローラ又はローカルメモリコントローラハブ(MCH)を更に有してよく、然るべくメモリ209又は214とインターフェース接続する。メモリ209及び214は、例えば、ランダムアクセスメモリ(RAM)、キャッシュメモリ、フラッシュメモリ、又は他のメモリデバイス等のあらゆるメモリデバイスを有する。一実施例で、メモリ214は上位キャッシュメモリを有し、一方、リソース213は下位キャッシュメモリを有する。他の実施例で、メモリ209は、プロセッサ205に付随する動的ランダムアクセスメモリ(DRAM)を有する。プロセッサ205は、DRAM209からのデータをキャッシュに格納するようキャッシュメモリを有する。これは実例となる実施形態である点に留意すべきであり、メモリ209及び214は、潜在的に如何なるスタイルメモリデバイスも有する。

20

【0032】

一実施例で、メモリ209及び214が、プロセッサ205及び210に含まれているか又は(表されるように)その外部にあるキャッシュメモリを有する場合に、プロセッサ205及び210は、ピアキャッシングノードはもちろんホームノードであることができる。例えば、トランザクションがメモリ209内のメモリロケーションを参照する場合に、メモリ209に関連するエージェント、すなわちプロセッサ205は、トランザクション及びメモリロケーションに関してホームエージェントであると判断される。同様に、トランザクションが他のメモリロケーション(例えば、メモリ214でのロケーション)を参照する場合に、プロセッサ205はピアキャッシングエージェントであると判断される。

30

【0033】

明らかなように、ポイント・ツー・ポイントリンク220乃至222は、ポイント・ツー・ポイント形式でコンポーネントを連結させる。物理リンク220乃至222は、一実施例で、夫々、図3を参照して以下で論じられる物理レイヤと関連する物理リンク等の、双方向の差動シグナリング相互接続を有する。結果として、プロセッサ205及び210並びにチップセット230は、互いに直接に通信することができる。

40

【0034】

チップセット230は、しばしば、入出力(I/O)ハブへ結合されているメモリコントローラハブ等の、複数の集積回路への総称指示である。なお、一実施例で、エージェントが夫々メモリとインターフェース接続するメモリコントローラハブのバージョンを有する場合に、チップセット230はI/Oハブ又は他のコントローラハブを言う。一実施例で、チップセット230は、上述されるように、トランザクションに参加し又はそれを経験する非キャッシングエージェントである。なお、チップセット230はそのように限定されず、他の実施例では、チップセット230は、キャッシュメモリを有するキャッシン

50

グエージェント及び／又はデータの元のメモリロケーション・リポジトリを有するメモリを有するホームエージェントである。

【 0 0 3 5 】

表されるように、チップセット 2 3 0 は、また、例えば、周辺コンポーネント相互接続 (P C I) 又は P C I エクスプレス (P C I - E) デバイス 2 6 1、集積デバイスエレクトロニクス (I D E) 又はアドバンスド・トランスファー・アタッチメント (A T A) デバイス 2 6 2、ユニバーサル・シリアル・バス (U S B) デバイス 2 6 3、ローカルエリアネットワーク (L A N) 又は無線 L A N (W L A N) デバイス 2 6 4、オーディオデバイス 2 6 5、及び他の I / O デバイス 2 6 6 等の複数の相互接続及び I / O デバイスとインターフェース接続すべきである。チップセット 2 3 0 は、また、ここで記載されるように I / O デバイスを結合する他の相互接続アーキテクチャを有してよい。

10

【 0 0 3 6 】

図 3 を参照すると、階層化相互接続スタックを利用する双方向相互接続アーキテクチャに係るブロック図の実施例が表されている。例えば物理レイヤ 3 0 2 等の図 3 のレイヤへの言及には、例えば物理レイヤ 3 0 2 a 及び物理レイヤ 3 0 2 b 等の、異なるエージェントで実施され得る一般的なレイヤの議論が含まれる。表されるように、相互接続スタックは 5 つのレイヤに分けられる。かかるレイヤのうち 1 又はそれ以上は、潜在的に、設計実施に基づいて任意である。例えば、ルーティングレイヤ 3 0 4 は、一実施例で、リンクレイヤ 3 0 3 の機能性に組み込まれる。従って、ルーティングレイヤは、一実施例では、別個のレイヤでない。

20

【 0 0 3 7 】

一実施例で、物理レイヤ 3 0 2 は、物理媒体での情報の電気伝送に関与する。例えば、物理的なポイント・ツー・ポイントリンクがリンクレイヤエンティティ 3 0 3 a 及び 3 0 3 b の間で利用される。実例として、物理リンクは、双方向差動シグナリング対 3 5 1 及び 3 5 2 を有する差動シグナリングスキームを有する。この場合に、物理レイヤは、潜在的に論理的に電気サブブロック及び論理サブブロックに分けられ、これにより、物理レイヤは、情報の電気伝送から残りのスタックを分離し且つリンクレイヤ 3 0 3 と通信することができる。

【 0 0 3 8 】

一実施例で、リンクレイヤ 3 0 3 は、スタックの上位レイヤから物理レイヤ 3 0 2 を抽出し、例えば、接続されるエージェント／エンティティの間の信頼できるデータ伝送及びフロー制御並びに多重仮想チャネル又はメッセージクラスへの物理チャネル／インターフェースの仮想化等のリンク関連サービスを提供する。ここで、仮想チャネルは、スタックの上位層による使用のための多重仮想ネットワークと考えられてよい。例えば、プロトコルレイヤ 3 0 6 は、プロトコルメッセージをメッセージクラスに、ひいては、1 又はそれ以上の仮想チャネルへマッピングするよう、潜在的に、リンクレイヤ 3 0 3 によって提供されるアブストラクションに依存する。

30

【 0 0 3 9 】

ルーティングレイヤ 3 0 4 は、一実施例で、ソースからあて先へパケットを送るための柔軟な方法を提供する。上述されるように、極めて簡単なトポロジで、ルーティングレイヤ 3 0 4 は明示的でなく、むしろリンクレイヤ 3 0 3 の機能性に組み込まれうる。例えば、ルーティングレイヤ 3 0 4 は、パケットを送るべく < ポート、仮想ネットワーク > 対を特定するためにリンクレイヤ 3 0 3 のアブストラクションに依存してよい。ここで、ルーティングテーブル情報は、パケットにルーティング情報を提供するように保持される。

40

【 0 0 4 0 】

一実施例で、トランスポートレイヤ 3 0 5 は、エンド・ツー・エンドの信頼できる伝送サービスを提供する。ルーティングレイヤ 3 0 4 と同じく、トランスポートレイヤ 3 0 5 も設計実施に基づいて任意である。一例として、トランスポートレイヤ 3 0 5 は、プロトコルレイヤ 3 0 6 に信頼できる伝送サポートを提供するようルーティングレイヤ 3 0 4 サービスに依存する。相互接続アーキテクチャ内で、一実施例で、一部のコンポーネントは

50

トランスポートレイヤ 305 を有する。結果として、この一部のコンポーネントは、トランスポートレイヤ 305 に関するパケットのサブフィールドを定義し、一方、他のコンポーネントは、潜在的にかかるサブフィールドを定義しない。

【0041】

プロトコルレイヤ 306 は、一実施例で、例えばキャッシュ・コヒーレンス、順序付け、ピア・ツー・ピア通信、割り込み転送等の、ノード/エージェント間の上位通信プロトコルを実施することができる。言い換えると、プロトコルレイヤ 306 は、ホームノード、ピアノード、キャッシングノード、及び非キャッシングノード等のノード又はエージェントについて、然るべく、許容されるメッセージ、要求、応答、フェーズ、コヒーレンス状態等を定義する。ホームノードメッセージ、スヌープメッセージ、応答メッセージ等のメッセージの例については以下で論じる。

10

【0042】

レイヤ及びそれに関連するロジックの議論は如何なる方法で結合されてもよい点に留意すべきである。例えば、プロトコルロジックは物理レイヤ、すなわち、送信又は受信ロジックへ結合されてもよい。ここで、図 3 から分かるように、一実施例で、プロトコルロジックは、物理レイヤロジックへ直接には結合されずに、むしろ他のレイヤロジックを介して結合されてよい。更に、相互接続スタックは、一実施例で、適切なキャッシュ・コヒーレンス動作を開始するよう、例えばキャッシュ制御又はキャッシュメモリロジック等の内部コンポーネントロジックへ結合される。

20

【0043】

[M E S I F プロトコルの実施例の概説]

一実施例で、基本 M E S I F (Modified Exclusive Shared Invalid Forward) プロトコルは、単一のシリアライジング・パスの潜在的な制限を伴わずにスヌープ・プロトコルと同様のプロトコルを提供する。スヌーピング・キャッシュ・プロトコルのように、M E S I F は、コヒーレンスを保つために、データのキャッシュコピーを有するノードに依存する。同期集中ブロードバンドよりむしろポイント・ツー・ポイントリンクの使用は、タイムワープの問題、すなわち、事象が異なるノードの側からすれば異なる順序で起こるように見えるという事実を持ち込む。一例として、M E S I F プロトコルは、タイムワープに起因する潜在的エラーの認識を通してタイムワープを扱い、そのプロトコル又はソフトウェアソリューションを提供する。

30

【0044】

ホームノードは、しばしば、データのキャッシュ無しコピーと関連する。結果として、ホームノードは、そのホームノードと関連するデータに関するトランザクションに関与する。しかし、ホームノードは、トランザクションと関連する“クリティカルパス”に含まれる必要はなく、むしろ、ホームノードは、コンフリクト及びタイムワープ問題を解決するようトランザクションに介入する。かかるスキームの同時ブロードキャスト性のために、一実施例で、M E S I F は、スヌーピング・プロトコルと関連する低レイテンシーを達成し、一方で、ある場合には、最低限のレイテンシー、すなわち、単一ラウンドトリップ要求 - 応答で、データのキャッシュ可能なコピーを取得する。

【0045】

40

一実施例で、M E S I F プロトコルに関する基本的なトランザクションは、ホームノードはもちろん全てのピアノードへ最初の要求を送信することを伴う。コピーがステート E、F 又は M コヒーレンス状態でキャッシュされる場合に、それは応答に含まれる。次いで、第 2 のメッセージがホームノードへ送信され、要求が満足されたことをホームノードに知らせる。要求されるラインがキャッシュされない場合、又は S ステートのコピーしか存在しない場合は、ホームノードへ送信される第 2 の要求は、前の要求を確認するために使用される。この前の要求をホームノードは既に自身のメモリからフェッチしてよい。いずれの場合にも、ホームノードは、同期及びコンフリクトの解消のために第 2 の要求（及び潜在的に、たとえそれらが時々結合され得るとしても、第 1 の要求）に応答する。ホームノードは 1 又はそれ以上のキャッシュを有してよい点に留意すべきであり、従って、

50

それは、いずれの他のノードとも同じように最初の要求に応答することができる。

【 0 0 4 6 】

一実施例で、コンフリクトは分散的に扱われる。タイムワープ問題は、個々の要求が任意に長い時間遅延しうるので、コンフリクトを検出することを困難にする。しかし、コンフリクトは、夫々のノードが要求後にコンフリクトについて監視する場合に検出される。複数のノードが潜在的にコンフリクトを検出してよいが、一例として、それらのノードのうちの少なくとも1つがコンフリクトを検出する。結果として、ノードからの応答は、一実施例で、潜在的にコンフリクト情報を有する。

【 0 0 4 7 】

一実施例で、応答からデータのコピーを受け取ったノードは、受け取った直後に内部データを使用することを許可されるが、ノードが確認を受け取るまでは、他のシステムにとって可視的な、すなわち、グローバルに可視的なデータを使用するという作用を引き起こさない。確認は、また、要求ノードがそのコピーを他のノードへ転送すべき命令を含み、ことによると、そのノードを自身のキャッシュから退かせる。

【 0 0 4 8 】

最後に、ノードがキャッシュデータを供給することによって他のノードからの要求に応答する場合に、ノードは、一実施例で、ノードがデータを転送したと認めるホームノードからの応答をノードが受け取るまで、同じキャッシュラインについて自身が受け取る他の要求を留保し、このようにして、全てのノードが（場合により書込可能な）キャッシュラインの伝送の同じ順序を守ることを確かにする。

【 0 0 4 9 】

ホームノードは、上述されるように、キャッシュ無しデータのリポジトリであるが、ホームノードは、また、プロセッサ及びキャッシュを有してよい。ここで、ホームノードのプロセッサがキャッシュを逃す場合に、ホームノードは他の全ての（ピア）ノードへ要求を送信し、ホームノードは、ホームノードに到着したいずれの他の要求と同じく内部でその要求を扱う。これは、ホームノードが自身（ホームノード）へメッセージを明示的に送信する特別の場合である点に留意すべきである。更に、外部からの要求が、局所的にキャッシュされるデータについて到着する場合に、ホームノードは適切に応答する。

【 0 0 5 0 】

開示されているメッセージプロトコルは、コヒーレンス（キャッシュ及びホーム）エージェント、非キャッシングエージェント、及び他のエージェント（メモリコントローラ、プロセッサ等）の間の許容されているメッセージの組を定義する。コヒーレンス・プロトコルは、考えられるコヒーレントを表すアルゴリズムでワード及び文法としてメッセージを使用する。このアルゴリズムは、分別よく要求を順序付け、コンフリクトを解消し、キャッシングエージェント間の相互作用を表す。M E S I F プロトコルについて上述されたが、M E S I F キャッシュ・コヒーレンス・プロトコルは利用される必要がない。例えば、既知のM E S I F プロトコルの利用をもたらしフォワード状態は利用されなくてよい。更に、上記にはM E S I F プロトコルにかかる実施例の典型的な概説が含まれる点に留意すべきである。結果として、上述される様々なコンポーネントは別々の実施例で相違する。メッセージング及び/又はコヒーレンス・プロトコルで利用される潜在的なメッセージの非包括的な例となる羅列は以下に含まれる。

【 0 0 5 1 】

[スヌープメッセージ]

一実施例で、スヌープメッセージは、例えばピアキャッシングエージェント等のキャッシングエージェントへ向けられる。通常、スヌープメッセージはホームノード識別子（ID）を有さず、これより、それらはホームノードには送られずにピアキャッシングエージェントに送られる。

【 0 0 5 2 】

Snoop Data (SnpData) : これは、E、F又はS状態でデータを得るスヌープである。

10

20

30

40

50

【 0 0 5 3 】

Snoop Invalidation Own (SnpInvOwn) : これは、E 又は M ステートでデータを得るスヌープである。一実施例で、ピアエージェントは、データを自身のローカルメモリに E 又は M ステートで保持することに応答してデータを提供することができる。

【 0 0 5 4 】

Snoop Invalidation Invalid to Exclusive (SnpInvItoE) : これは、ピアキャッシングエージェントを無効にし、ホームノードへのあらゆる M ステートデータをフラッシュ (flush) するスヌープである。

【 0 0 5 5 】

Snoop Invalidation X to Invalid (SnpInvXtoI) : これは、ピアキャッシングエージェントを無効にし、ホームノードへのあらゆる M ステートデータをフラッシュするスヌープである。

10

【 0 0 5 6 】

[ホームメッセージ]

下記のメッセージはホームノードと関連する。一例として、下記の幾つかはホームノードへの要求メッセージである。実際には、スヌープメッセージ及びホームメッセージのうち幾つかの間の類似性に留意すべきである。なお、これらの類似するメッセージは、しばしば、ホームノードへのメッセージがホームノードへ送られるべきホームノード ID を有し、一方、送信されるピアノードメッセージはホームノードへの参照を有さなくてもよい点で相違する。

20

【 0 0 5 7 】

他のホームメッセージは、ホーム・ライトバック・マーカメッセージを含む。ここで、データは、WbData メッセージを介して別個に送信されてよく、マーカメッセージに対して如何なる順序で受信されてもよい。以下に含まれるホームメッセージの他の例は、キャッシングエージェントからホームエージェントへのホームスヌープ応答メッセージ、及びコンフリクトフェーズの開始を知らせるコンフリクトメッセージのホーム確認応答である。

【 0 0 5 8 】

Read Data (RdData) ; これは、排他的 (Exclusive) (E) 状態、更には、潜在的に、実施例に依存してフォワード (Forward) (F) 状態又は共有 (Shared) (S) 状態でのデータの要求である。

30

【 0 0 5 9 】

Read Invalidate Own (RdInvOwn) : これは、M 又は E ステートでのキャッシュラインの要求である。

【 0 0 6 0 】

Invalidate Invalid to Exclusive (InvItoE) : これは、データを有さないキャッシュラインの要求である。

【 0 0 6 1 】

Non-Snoop Read (NonSnpRd) : これは、メモリからのコヒーレントでない読出を実行する要求である。

40

【 0 0 6 2 】

Non-Snoop Write (NonSnpW) : これは、メモリへのコヒーレントでない書込を実行する要求である。

【 0 0 6 3 】

WriteBack Modified to Invalid、Shared、又は Exclusive (WbMtOI、WbMtOS、及び WbMtOE) : これは、M ステートにあるキャッシュラインをメモリにライトバックして、なるべくキャッシュラインの状態を無効状態、共有状態、又は排他状態へ遷移させる要求である。一実施例で、かかるメッセージは、データが送信されるべきことを示すマーカメッセージであり、一方、データは、実際には、別々のメッセージ (例えば、Wb*Data*メッセージ) を介して伝送される。

50

【 0 0 6 4 】

Response Invalid又はShared (RspI又はRspS) : ピアエージェントは、然るべく無効状態で又は共有状態でキャッシュを持ち続ける。

【 0 0 6 5 】

Response Forward Invalid又はShared (RspFwdI又はRspFwdS) : ピアはデータを要求側へ送信しており、ラインは然るべく無効状態又は共有状態のままである。

【 0 0 6 6 】

Response Forward Invalid又はShared WriteBack (RspFwdIwb又はRspFwdSWb) : ピアはデータを要求側へ及びWbIDataをホームノードへ送信しており、ラインは然るべく無効状態又は共有状態のままである。

10

【 0 0 6 7 】

Response Invalid又はShared WriteBack (RspIwb又はRspSWb) : ピアはインフライト (in-flight) のWbIDataメッセージを有するデータをホームへ退かせており、如何なるメッセージも要求側へ送信していない。

【 0 0 6 8 】

Response*Writeback Post (Rsp*WbPost) : ピアは、インフライトのWb*Dataメッセージを有するデータをホームへ退かせており、如何なるメッセージも要求側へ送信していない。更に、一実施例で、Rsp*WbPostメッセージに関連するAckCnfltメッセージの送信側は、Cmp_Fwd*メッセージの受信に回答して割当てを解除されない。一例として、ホームエージェントは、メモリへの部分的データのコミットに回答して完全なメッセージを送信する。

20

【 0 0 6 9 】

WbPost : ピアは、インフライトのWb*Dataメッセージを有するデータをホームへ退かせている。更に、一実施例で、WbPostメッセージの送信側は、Cmp_Fwd*メッセージの受信に回答して割当てを解除されない。ここで、送信側は、ホームノードからの完了を待つ。一例として、ホームエージェントは、メモリへのデータのコミットに回答して完全なメッセージを送信する。

【 0 0 7 0 】

Response Conflict (RspCnflt) : ピアは無効状態でラインを持ち続け、ピアはコンフリクトする対抗する要求を有する。

30

【 0 0 7 1 】

Acknowledge Conflict (AckCnflt) : データ完了 (completion) / グラント (grant) 及び完了 / フォース (force) 確認応答コンフリクトの確認応答受信。

【 0 0 7 2 】

Acknowledge Conflict Write-back Invalid (AckCnfltWbI) : 一実施例で、それは、上述されるように、AckCnfltを伝える。更に、一実施例で、このメッセージは、更に、部分的データのライトバックを伝える。部分的データは、後述されるように、別個のWbIDataに含まれてよい。更に、データに関連するキャッシュラインは無効状態へと遷移させられる。

40

【 0 0 7 3 】

[無データ応答メッセージ]

一実施例で、無データ応答メッセージは、データを送信することなくラインのオーナーシップを認めるグラントメッセージを有する。無データ応答メッセージは、また、完了を知らせる完了メッセージを有し、潜在的に、コンフリクト状態でオーナーからデータを取り出す。

【 0 0 7 4 】

Grant_Completion (GntCmp) : データによらずに排他的オーナーシップを付与。

【 0 0 7 5 】

Grant Force Acknowledge Conflict (Gnt_FrcAckCnflt) : データ及びフォース確認応答コンフリクトによらずに排他的オーナーシップを付与。

50

【 0 0 7 6 】

Completion (CMP) : 全てのスヌープ応答が収集され、コンフリクトは検出されない。

【 0 0 7 7 】

Force Acknowledge Conflict (FrcAckCnflt) : 確認応答コンフリクトについて、スヌープ応答が収集される。

【 0 0 7 8 】

Complete_Foward Code (Cmp_FwdCode) : 要求を完了し、フォワード状態又は共有状態でのラインを要求側へ転送し、且つ、ローカルコピーを無効にするか又はそれを共有状態のままとする。

【 0 0 7 9 】

Complete Forward Invalidate Own (Cmp_FwdInvOwn) : 要求を完了し、排他的状態又は変形 (Modified) (M) 状態でのラインを要求側へ転送し、ローカルコピーを無効化する。

10

【 0 0 8 0 】

[データ応答メッセージ]

DataC_* : 対応する M E S I F 状態でのデータ。

【 0 0 8 1 】

DataC_*_Cmp : 完了を伴う対応する E S I F 状態でのデータ。

【 0 0 8 2 】

DataC_*_FrcAckCnflt : FrcAckCnfltを伴う対応する E S I F 状態でのデータ。

20

【 0 0 8 3 】

Wb*Data : データをライトバックし、対応する E、S 又は I ステートへ遷移。

【 0 0 8 4 】

WbIDataPtl 又は WbEDataPtl : 部分的データのライトバック、対応する無効状態又は排他的状態への遷移。

【 0 0 8 5 】

Wb*DataPtlPost : 部分的データのライトバック、対応するコヒーレンス状態への遷移。更に、一実施例で、Wb*DataPtlPostメッセージに関連するAckCnfltメッセージの送信側は、Cmp_Fwd*メッセージの受信に応答して割当てを解除されない。一例として、ホームエージェントは、メモリへの部分的データのコミットに応答して完全なメッセージを送信する。

30

【 0 0 8 6 】

Non-Snoop Write Data (NonSnprWrData) : これは、メモリへの非コヒーレントな書込である。

【 0 0 8 7 】

Non-Snoop Write Data Partial (NonSnprWrDataPtl) : これは、部分的データの非コヒーレントな書込である。

【 0 0 8 8 】

図 4 a を参照すると、潜在的にメモリ順序付け違背を引き起こす複数のノードの間でのメッセージングのためのプロトコルダイアグラムの実施例が表されている。表されるように、プロトコルダイアグラムは、5つのノード、すなわち、多数のトランザクションに参加する入出力ハブ (IOH) (コンシューマ)、“フラグ (Flag)” のためのホームノード 0 (CHL0)、プロデューサノード (GQ0)、“データ (Data)” のためのホームノード 1 (CHL1)、及び他のキャッシングエージェント (GQ1) を表す。留意すべきは、表されるエージェントの数及びタイプは単に実例であり、相互接続アーキテクチャは極めて柔軟性があり且つ如何なる数又はスタイルのエージェントを有してもよい点である。

40

【 0 0 8 9 】

ここで、プロデューサ GQ0 は、データに関連するキャッシュラインに対する要求を開始する。InvltoE(D)メッセージはホームノード 1 (CHL1) へ送信され、SnprInvltoE(D)

50

)メッセージはピアエージェント(コンシューマGQ1)へ送信/転送される。なお、このトランザクションの間、GQ1は、データの読出要求を有する別のトランザクションを開始する。留意すべきは、GQ1及びGQ0の両エージェントは、この場合に、コンフリクトを検出する点である。このコンフリクトは、ピアエージェントGQ0及びGQ1からホームエージェントCHL1へのRspCnfltメッセージによって反映される。ホームノードCHL1が対抗するメッセージを受信した後、それは、オーナーシップを付与し且つコンフリクトの確認応答を余儀なくさせるようGnt_FrcAckCnfltをGQ0へ送信する。

【0090】

この時点で、プロデューサとしてのGQ0は、データへの部分書込、すなわち、Data = Newを実行し、Gnt_FrcAckCnfltメッセージの受信を確認するAckCnfltメッセージを送信する。一実施例で、AckCnfltメッセージは、また、トランザクションのコンフリクトフェーズの開始を示し、一方、AckCnfltメッセージに付随する完了は、コンフリクトフェーズの終了を表す。次いで、ホームノードCHL1は、データをホームノードCHL1に押し返すようCmp_FwdCodeメッセージを送信する。しかし、部分的データ、すなわち、RspiWb(D)メッセージ及びWbIDataPtI(D)メッセージのライトバックは、例えばチャネル又はルーティング遅延のような幾つかの理由のために遅延する。

【0091】

一方、データを更新し、ホームノードCHL1へ部分的データ応答を供給し、ホームノードCHL1からのCmp_FwdCodeによってデータトランザクションへの部分書込から割当てを解除されると、GQ0は、フラグへの書込、すなわち、CHL0へのInvIttoE(F)メッセージ並びにピアエージェントIOH及びGQ1への対応するスヌープを開始する。ここで、GQ0は、フラグについてのキャッシュラインのオーナーシップ、すなわち、ホームノードCHL0からのGnt_Cmpを取得し、フラグをNewへと更新する。更に、フラグの新たな値は、ホームノードCHL0へライトバックされる(すなわち、WbMtoI(F)及びWbIData(F))。しばしば、データ及びフラグに係るこのようなシナリオはI/O動作で利用される。この場合に、1のエージェントはデータを更新し、次いで、データが更新されており且ついつでも回復(retrieval)する状態になっていることを他のエージェントに知らせるようフラグを設定する。

【0092】

従って、IOHは、いつデータが取り出される準備ができるかを決定するよう、周期的に又は一定時間後にフラグをチェックしながらフラグに対してスピニング(spinning)を行っている。結果として、IOHは、フラグの非スヌープな又は非コヒーレントな読出、すなわち、NonSnprd(F)を実行する。ここで、ホームノードCHL0のメモリからの読出は、データがいつでも取り出されることを示すFlag = New値を取り出す。もはやフラグに対してスピニングを行っていないIOHは、データのNonSnprd要求を開始する。なお、NonSnprdが受信される場合に、データは、GQ0からのData = Newのライトバックが遅延したために、依然としてOld値を保持する。結果として、コンシューマIOHは、無効な古いデータを取り出すが、データが有効であると信じる。これにより、潜在的に、不適法な動作又は実行が引き起こされる。

【0093】

留意すべきは、フラグ及びデータの例は、単に実例であって、部分的なメモリアクセス及び非コヒーレントなメモリアクセスの無効なメモリ順序付けが潜在的に無効な/不適法な動作をもたらすところの潜在的な環境を表す。結果として、ここで記載される実施例は、如何なる部分メモリアクセスシナリオとともに利用されてもよい。

【0094】

同様に、図4bは、潜在的にメモリ順序付け違背を引き起こす複数のノードの間でのメッセージングのためのプロトコルダイアグラムの他の実施例が表されている。ここで、GQ0は、上述されるように、ラインを要求する代わりに、データに係るキャッシュラインを変形状態に保つ。結果として、データへの書込の前に、GQ0は、変形されたコピーをWbMtoE(D)及びWbEData(D)によりホームノードCHL1へ返し、排他的状態へと遷移する

。次いで、G Q 0 は、自身のローカルメモリでデータへの部分書込を実行する。なお、上記のように、G Q 1 は、データに対する読出要求を発する。留意すべきは、上記とは異なって、G Q 1 でなく、G Q 0 のみがコンフリクトを検出する。上述されるように、複数のノード/エージェントがコンフリクトを検出してよいが、プロトコルは、一実施例で、少なくとも1つのノードに依存してコンフリクトを検出する。

【0095】

結果として、G Q 0 は、解決のために、ホームノードC H L 1 へRspCnfltメッセージを供給する。コンフリクトフェーズへの同様の遷移が、同一の応答、すなわち、ホームノードC H L 1 からのGnt_FrcAckCnflt、G Q 1 からのAckCnflt、C H L 1 からのCmp_FwdCode、及び、次いで、前述同様に遅延するメッセージRspIwb(d)及びWbIDataPtl(D)によるデータのライトバックを有して起こる。次いで、フラグはNewへ更新され、I O H は、フラグのメモリからの非コヒーレントな読出を実行する。フラグがNewへ更新されたことによりデータがいつでも回復する状態となっていると確信すると、I O H はデータの非スヌープ読出を要求する。上記のように、I O H は、G Q 0 の部分書込、データの非コヒーレントな読出、及び新しい(New)データのライトバックの遅延に応答して、更新された新しい(New)データの代わりに、無効な古い(Old)データを受信する。

【0096】

従って、一実施例で、単一又は複数のメッセージを有しうるライトバックフェーズは、例えば部分書込のような部分メモリアクセスに関連するコンフリクトフェーズで挿入されて、部分的データをホームノードへ返す。第1の例として、コンフリクトを認めてライトバックマーカを供給するAckCnfltWbIメッセージは、コンフリクトフェーズで生成される。他の例として、ライトバックメッセージを用いるライトバックフェーズは、AckCnfltメッセージがホームノードへ供給される前に挿入される。更なる他の例として、ライトバックメッセージを用いるライトバックフェーズは、コンフリクトフェーズ内に、すなわち、AckCnfltメッセージの後に含まれる。最後の具体例として、ライトバックフェーズは、Rsp*Post、WbDataPostメッセージ、又はWbPostメッセージを用いるコンフリクトフェーズ内又はその後に含まれ、これにより、AckCnfltメッセージの送信側は、コンフリクトフェーズの完了メッセージを受信した後に割当てを解除されない。

【0097】

ライトバックフェーズがコンフリクトフェーズで開始されるこのような例は単に実例であり、コンフリクトの際に部分的データをホームノードへ押し返す如何なる既知のメッセージ又は方法も利用されてよい。なお、議論を進めるよう、かかる具体例について、以下、図5 a乃至7 cを参照して論じられる。

【0098】

図5 aを参照すると、コンフリクトフェーズの開始時にコンフリクト確認応答及びデータライトバックメッセージを用いるプロトコルダイアグラムの実施例が表されている。図5 aのシナリオは、G Q 0 がデータに関連するキャッシュラインの排他的オーナーシップを要求し、G Q 1 がデータの読出要求を発し、コンフリクトがG Q 0 及びG Q 1 によって検出され、データに関連するホームノードC H L 1 がGnt_FrcAckCnfltメッセージをG Q 0 へ送信する点で、図4 aのシナリオに類似する。しかし、一実施例で、プロデューサが単にAckCnfltメッセージだけ供給してホームノードC H L 1 からのGnt_FrcAckCnfltの受信を確認することに代えて、結合AckCnfltWbI(D)メッセージがG Q 0 によって生成されて送信される。ここで、AckCnfltWbIメッセージは2つの目的、すなわち、Gnt_FrcAckCnfltの確認応答と、データが例えばWbIDataPtl(D)のようなデータメッセージで供給されるホームノードC H L 1 へのマーカとを果たす。

【0099】

言い換えると、プロデューサG Q 0 がデータへの部分書込を実行し且つホームノードC H L 1 からコンフリクトメッセージを受信することに応答して、G Q 0 は、ホームノードへの部分的データのライトバックを実行する、すなわち、コンフリクトフェーズでライトバックフェーズを実行する。これまで、図4 aで、ライトバックはコンフリクトフェーズ

が完了した後でその後に実行された。ここで、G Q 0 は、Cmp_FwdCodeを用いてコンフリクトフェーズの終わりまでに割当てを解除され、結果として、次のトランザクションは、いつ部分的なデータがホームノードC H L 1に達するかに関わらず開始することができる。このことは、I O Hによる無効なデータの取得を生じさせる。

【0100】

対照的に、図5 aから明らかなように、Cmp_FwdCodeは、部分的データがホームノードC H L 1に返されるか又はそのメモリにコミットされるまで送信されない。結果として、G Q 0は割当てを解除されず、フラグを更新する次のトランザクションはデータがC H L 1で更新され且つCmp_FwdCodeメッセージがG Q 0で受信された後まで始まらない。これにより、結果として、フラグが更新される前にデータが更新されることが確かとなる。従って、I O Hがフラグ、次いでデータの非コヒーレントな読出を実行する場合に、C H L 1にあるデータはG Q 0からの最新のコピーであることが保証される。

10

【0101】

同様に、図5 bを参照すると、コンフリクトフェーズの開始時にコンフリクト確認応答及びデータライトバックメッセージを用いるプロトコルダイアグラムの他の実施例が表されている。ここで、プロトコルフローは、G Q 0がデータの変形されたO l dコピーを保持し、そのO l dコピーをホームノードC H L 1へライトバックし、データを新しい値へ更新し、G Q 1がデータの読出要求を発し、コンフリクトが検出され、コンフリクトメッセージのフォース確認応答がホームノードC H L 1によって生成されてプロデューサG Q 0へ送信される点で、図4 aのプロトコルフローと類似する。

20

【0102】

ここで、図5 aを参照して論じられたように、一実施例で、G Q 0はデータへの部分書込を実行して、コンフリクトメッセージはG Q 0によってホームノードC H L 1から受信されるので、AckCnfltWblメッセージがホームノードC H L 1へ供給される。AckCnfltWblメッセージは、コンフリクトが確認され且つデータメッセージ(すなわち、WblDataPtIメッセージ)がデータをホームノードC H L 1へ供給すべきことをホームノードに知らせる。潜在的なタイムワープに係る前述の議論から留意すべきは、WblDataPtIは潜在的にAckCnfltWblの前に受信され得る点である。しかし、プロトコルの動作を通して、C H L 1は、プロトコルフローと一致する方法でいずれも行う。結果として、フラグ・トランザクションは、G Q 0がホームノードC H L 1から完了信号を受信するまで開始されない。完了信号は、部分データの受信後しか送信されない。従って、上述されるように、I O Hは、フラグがNewへ更新されるまでフラグに対してスピニングを行う。フラグの更新は、目下、データがC H L 1で更新された後であることを確かにされ、結果として、適切なメモリ順序付けと、有効なデータがI O Hへ供給されることとがもたらされる。

30

【0103】

次に、図6 aを参照すると、コンフリクトフェーズでライトバックフェーズを用いるプロトコルダイアグラムの他の実施例が表されている。上述されるように、図6 aのプロトコルは、少なくとも最初は、図4 aのプロトコルに類似する。対照的に、ホームノードC H L 1からのGnt_FrcAckCnfltメッセージの受信に応答して、プロデューサエージェントG Q 0はデータへの部分書込を実行し、WbMtoIデータマーカメッセージ及びWblDataPtIデータメッセージを用いてホームノードC H L 1へ部分的データをライトバックする。G Q 0がライトバックの完了メッセージを受信した後、すなわち、部分的なデータがC H L 1でコミットされ且つ完了メッセージがC H L 1によって送信された後、G Q 0はAckCnfltメッセージをホームノードC H L 1へ送信し、ホームノードC H L 1はCmp_FwdCodeメッセージをG Q 0へ送信する。

40

【0104】

上述されるように、フラグ書込トランザクションは、Cmp_FwdCodeメッセージがG Q 0によって受信された後まで開始されず、Cmp_FwdCodeメッセージは、新しいデータがC H L 1へ返された後まで送信されない。結果として、I O Hは、フラグがNewへ更新されるまでデータを読み出さず、一方、フラグは、データがC H L 1でコミットされるまで更

50

新されない。言い換えると、フラグが更新される前にグローバルに観測されるデータのメモリ順序付けが、実施される。

【0105】

同様に、図6bは、コンフリクトフェーズでライトバックフェーズを用いるプロトコルダイアグラムの他の実施例を表す。ここで、プロトコルフローは、やはり、図4bのプロトコルフローに類似する。しかし、上記の図6aごとく、データへの部分書込を実行し且つホームノードCHL1からFrcAckCnfltメッセージを受信することに応答して、プロデューサエージェントGQ0は新しい(New)データのライトバックを実行する。GQ0からのAckCnfltから始まり、ホームノードCHL1からのCmp_FwdCodeで終わるコンフリクトフェーズは、新しい(New)データの受信に応答してCHL1からの完了がGQ0によって受信されるまで始まらない。結果として、フラグ値の更新の前にホームノードCHL1でデータを更新するメモリ順序付けが、実施される。これにより、結果として、正確な/有効な新しい(New)データがコンシューマIOHへ供給される。

10

【0106】

図6cを参照すると、データのライトバックを有するライトバックフェーズを用いるプロトコルダイアグラムの他の実施例が表されている。ここで、ライトバックフェーズは、図6aのライトバックフェーズに類似する。しかし、ライトバックフェーズは後で実行される。更に、ここで、RsplWbメッセージ及びWbIDataPtlメッセージを用いる代わりに、上記の図4aのごとく、WbMtoiメッセージ及びWbIDataPtlメッセージがデータのライトバックのために用いられる。言い換えると、ライトバックフェーズが、応答フェーズの代わりにコンフリクトフェーズで用いられる。これは、図4aのようにCHL1からGQ0への完了を伴わないデータの単なる転送に代えて、GQ0がフラグ・トランザクションに移る前にホームノードCHL1からの完了メッセージを要求する。

20

【0107】

図7aを参照すると、データのライトバックのためにポスタブル書込メッセージを用いるプロトコルダイアグラムの実施例が表されている。ここで、フローは、図6cのフローに極めて類似する。しかし、留意すべきは、RsplWbPost(D)マーカメッセージがWbMtoiマーカメッセージの代わりに用いられる点である。従って、一実施例で、RsplWbPostによって開始されるフェーズは応答フェーズと呼ばれる。しかし、完了信号がホームノードCHL1からGQ0で受信されるまで適切に応答していたGQ0の割当てを解除しないRsplWbPostの機能性に留意すべきである。明らかなように、動作において、RsplWbPostは、GQ0が、GQ0がフラグ・トランザクションに移ることができる前にホームノードCHL1からの完了を待つ点で、GQ0がフラグ・トランザクションに移る前に完了を待たない図4aのRsplWbよりも、図6cのWbMtoiに類似する。言い換えると、RsplWbPostは、RsplWbPostを送信したノードGQ0が割当てを解除される前に完了を待つ“ポスタブル書込(postable write)”を示す。従って、一実施例で、RsplWbPostの利用は、ライトバックフェーズと呼ばれる。

30

【0108】

同様に、図7bは、データのライトバックのためにポスタブル書込メッセージを用いるプロトコルダイアグラムの他の実施例を表す。ここで、留意すべきは、“ポスタブル書込”は、マーカメッセージRsplWbの代わりに、データメッセージWbIDataPtlPostによって示され得る点である。更に、図7cに表されるように、別のメッセージWbPostは、RsplWb及びWbIDataPtlが一実施例でライトバックフェーズの部分であることを示すために用いられてよい。この場合に、GQ0は、メモリへの部分的データのコミットに応答してホームノードCHL1からの完了が受信されるまで割当てを解除されない。

40

【0109】

以上から明らかなように、一実施例で、コンフリクトフェーズでライトバックフェーズを実行し、挿入し、そのためのメッセージを生成、又は開始することは、コンフリクトフェーズに対して異なる時間的關係を有する。一実施例で、コンフリクトフェーズでのライトバックフェーズは、例えば、ライトバックマーカメッセージと結合されている確認応

50

答コンフリクトメッセージ、すなわち、AckCnfltWbl(D)のように、コンフリクトフェーズの開始時に含める。他の実施例で、コンフリクトフェーズでのライトバックフェーズは、例えば、Gnt_FrcAckCnfltの後であって且つAckCnfltメッセージの前、すなわち、図6 a及び6 bに関する議論のように、コンフリクトフェーズの前に含める。更なる他の実施例で、コンフリクトフェーズでのライトバックフェーズは、例えば、AckCnflt及びCmp_FwdCodeメッセージの後、すなわち、図6 c乃至7 cに関する議論のように、コンフリクトフェーズの後に含める。

【0110】

更に、ライトバックフェーズは、一実施例で、例えばAckCnfltWbl、WbMtoi、及びWblDataPtlのようなライトバックメッセージを参照する。他の実施例で、ライトバックフェーズは、また、例えば、RspIWbPost、WblDataPtlPost、及びWbPostのような、ライトバック機能性をエミュレートする応答関連メッセージを有する。一例として、ライトバックフェーズの機能性をエミュレートすることは、ライトバックの完了メッセージが受信されるまで送信エージェントの割当てを解除しないこと又は送信エージェントが待機することを有する。

10

【0111】

従って、以上から明らかなように、メモリ順序づけは、部分メモリアクセス及び非スヌープ/非コヒーレントアクセスが行われる場合に、コンフリクトが検出される場合でさえ実施され得る。従って、データが読み出されることを示すようフラグを設定する第2のエージェントからのデータに第1のエージェントが付随している場合等の環境下で、適切なメモリ順序付けの実施は、一貫性のない無効なデータの代わりに、有効な/新しいデータが供給されることを確かにする。

20

【0112】

ここで使用されるモジュールは、如何なるハードウェア、ソフトウェア、ファームウェア、又はそれらの組合せも言う。しばしば、別個なものとして表されるモジュール境界は一般的に変化し、潜在的には重なる。例えば、第1及び第2のモジュールは、ハードウェア、ソフトウェア、ファームウェア、又はそれらの組合せを共有しながら、潜在的に或る独立したハードウェア、ソフトウェア、又はファームウェアを保ってよい。一実施例で、用語「ロジック」の使用は、例えばトランジスタや抵抗等のハードウェア、及び例えばプログラム可能なロジックデバイス等の他のハードウェアを含む。なお、他の実施例で、「ロジック」は、また、例えばファームウェア又はマイクロコード等の、ハードウェアと一体化したソフトウェア又はコードを含む。

30

【0113】

ここで使用される「値」は、数、状態(ステート)、論理状態、又は2進論理状態のあらゆる既知の表現を含む。しばしば、論理レベル、論理変数、又は論理値の使用は、“1”及び“0”と呼ばれることがある。これは容易に2進論理状態を表す。例えば、“1”は高論理レベルに言及し、“0”は低論理レベルに言及する。一実施例で、トランジスタ又はフラッシュセル等の記憶セルは、単一の論理値又は複数の論理値を保持する能力を有してよい。なお、コンピュータシステムでの値の他の表現が使用されている。例えば、10進数“10”は、また、“1010”の2進値及び16進文字“A”として表されてもよい。従って、「値」は、コンピュータシステムで保持され得る情報のあらゆる表現を含む。

40

【0114】

更に、「状態(ステート)」は値又は値の一部によって表され得る。一例として、論理“1”等の第1の値はデフォルト又は初期状態を表すことができ、一方、論理“0”等の第2の値は非デフォルト状態を表すことができる。更に、用語「リセット」及び「設定(セット)」は、一実施例で、夫々、デフォルトの及び更新された値又は状態を言う。例えば、デフォルト値は、潜在的に、高論理値、すなわちリセットを有し、一方、更新される値は、潜在的に、低論理値、すなわちセットを有する。留意すべきは、値の如何なる組合せも、あらゆる数の状態を表現するために利用される点である。

50

【 0 1 1 5 】

前述の方法、ハードウェア、ソフトウェア、ファームウェア又はコードの実施例は、処理要素によって実行可能である機械アクセス可能な又は機械読出可能な媒体に格納されている命令又はコードを介して実施されてよい。機械アクセス可能／読出可能な媒体は、コンピュータ又は電子システム等の機械によって読出可能な形式で情報を提供（すなわち、記憶及び／又は送信）するあらゆるメカニズムを有する。例えば、機械アクセス可能な媒体には、静的 R A M（ S R A M）又は動的 R A M（ D R A M）等のランダムアクセスメモリ（ R A M）、 R O M、磁気若しくは光記憶媒体、フラッシュメモリデバイス、電気記憶デバイス、光記憶デバイス、音響記憶デバイス、又は他の形態の伝播信号（例えば、搬送波、赤外線信号、デジタル信号）記憶デバイス等がある。例えば、機械は、例えば搬送波等の伝播信号で送信される情報を保持することができる媒体から伝播信号を受信することにより記憶媒体にアクセスしてよい。

10

【 0 1 1 6 】

本明細書の全体を通して“一実施例”又は“実施例”との言及は、その実施形態に関連して記載される特定の特徴、構造、又は事項が本発明の少なくとも 1 つの実施例に含まれることを意味する。このように、本明細書の全体を通して様々な場所でのフレーズ“一実施例で”又は“実施例で”の出現は、必ずしも全て同じ実施形態に言及しているわけではない。更に、特定の特徴、構成、又は事項は、1 又はそれ以上の実施例で何らかの適切な方法で組み合わせられてよい。

【 0 1 1 7 】

以上、詳細な記載が特定の具体的な実施形態を参照して与えられた。なお、様々な改良及び変形が、添付の特許請求の範囲に記載される本発明の幅広い精神及び適用範囲から外れることなく行われてよいことは明らかである。従って、明細書及び図面は、限定の観点よりむしろ例示の観点で受け止められるべきである。更に、実施例及び他の例となる用語の使用は、必ずしも同じ実施例又は同じ例に言及しているわけではなく、潜在的に同じ実施例はもちろん異なる個別の実施例に言及していることがある。

20

【符号の説明】

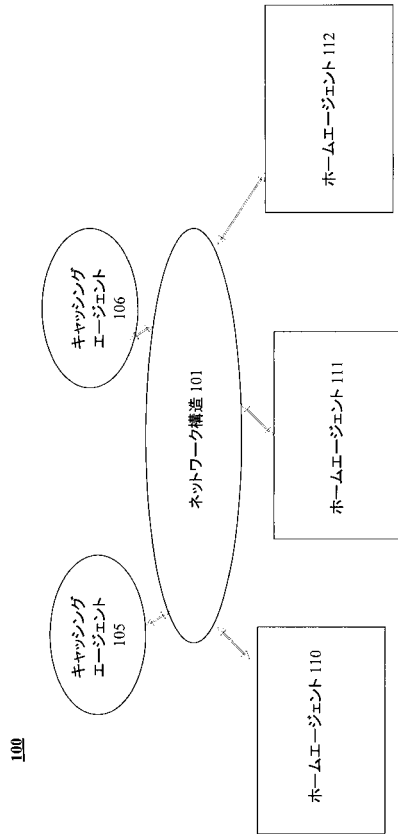
【 0 1 1 8 】

- 1 0 0 相互接続アーキテクチャ
- 1 0 1 ネットワーク構造
- 1 0 5 , 1 0 6 キャッシングエージェント
- 1 1 0 ~ 1 1 2 ホームエージェント
- 2 0 5 , 2 1 0 プロセッサ
- 2 0 6 , 2 0 7 , 2 1 1 , 2 1 2 処理要素（ P E ）
- 2 0 8 , 2 1 3 リソース
- 2 0 9 , 2 1 4 メモリ
- 2 2 0 ~ 2 2 2 ポイント・ツー・ポイントリンク
- 2 3 0 チップセット
- 2 6 1 ~ 2 6 6 相互接続及び I / O デバイス
- 3 0 2 a , 3 0 2 b 物理レイヤ
- 3 0 3 a , 3 0 3 b リンクレイヤ
- 3 0 4 a , 3 0 4 b ルーティングレイヤ
- 3 0 5 a , 3 0 5 b トランスポートレイヤ
- 3 0 6 a , 3 0 6 b プロトコルレイヤ
- 3 5 1 , 3 5 2 双方向差動シグナリング対

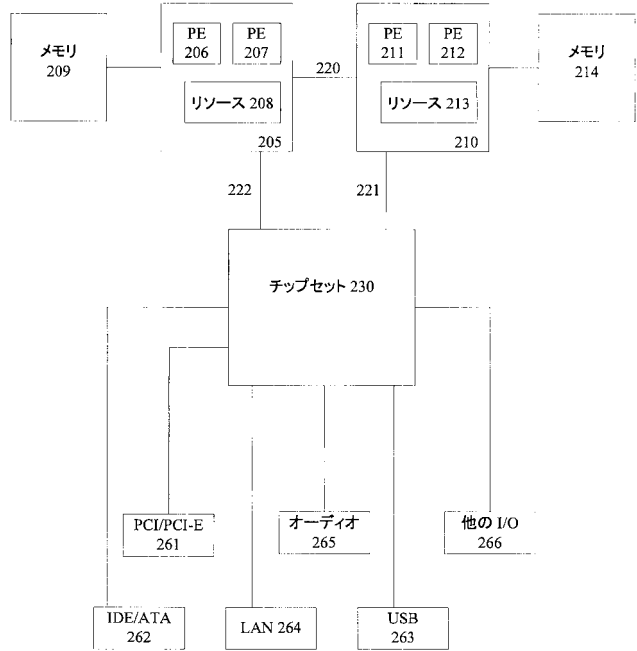
30

40

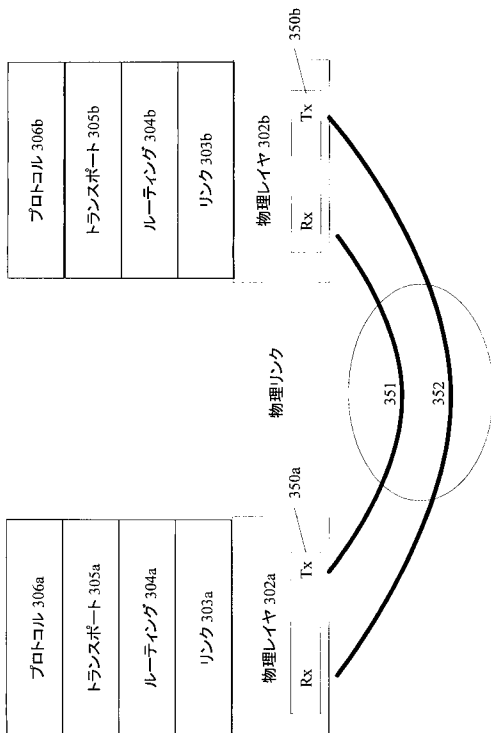
【図 1】



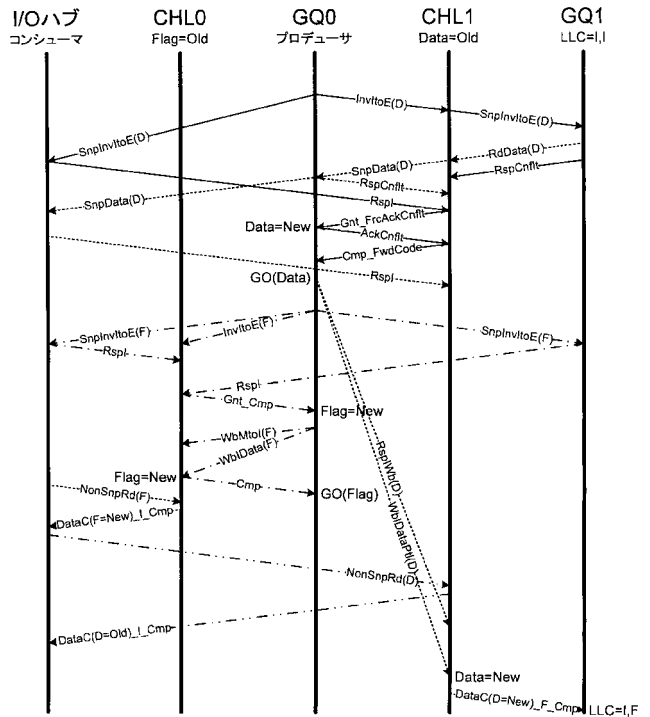
【図 2】



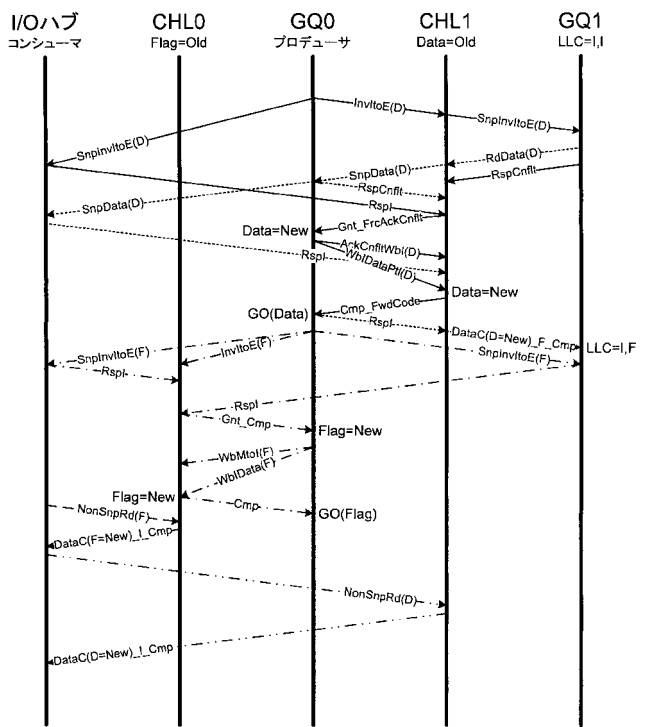
【図 3】



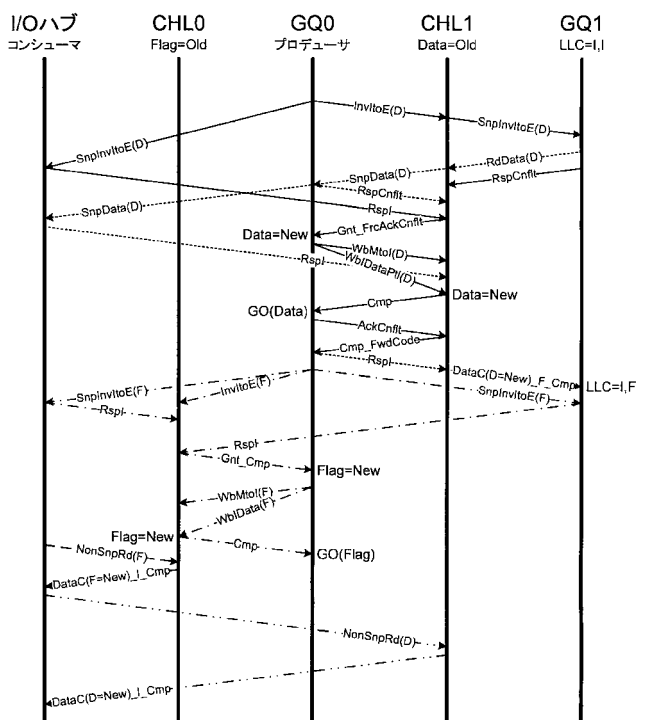
【図 4 a】



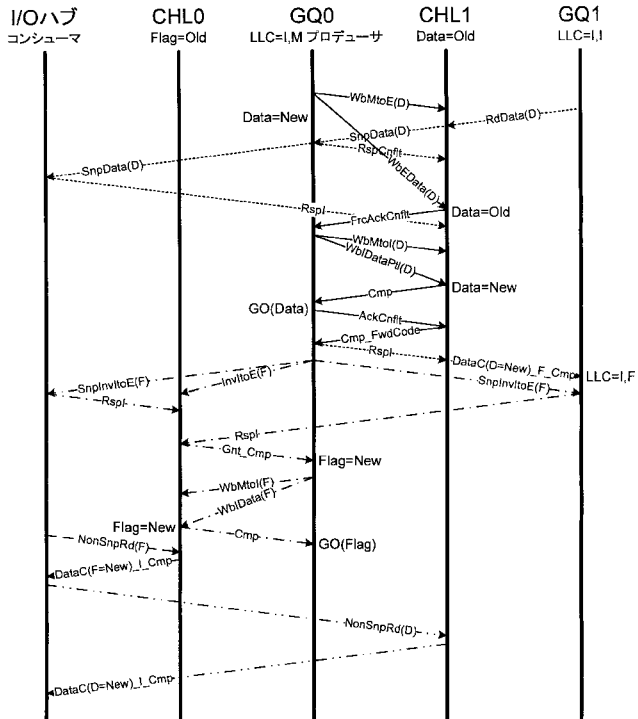
【 図 5 a 】



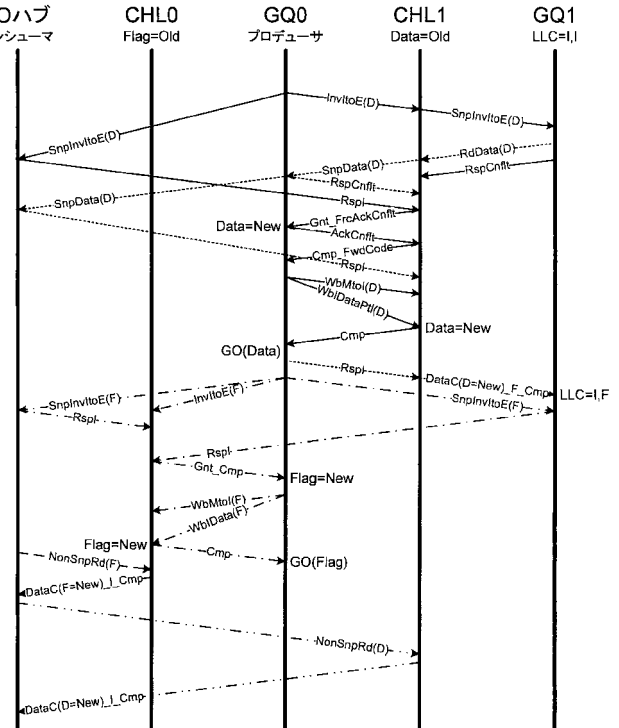
【 図 6 a 】



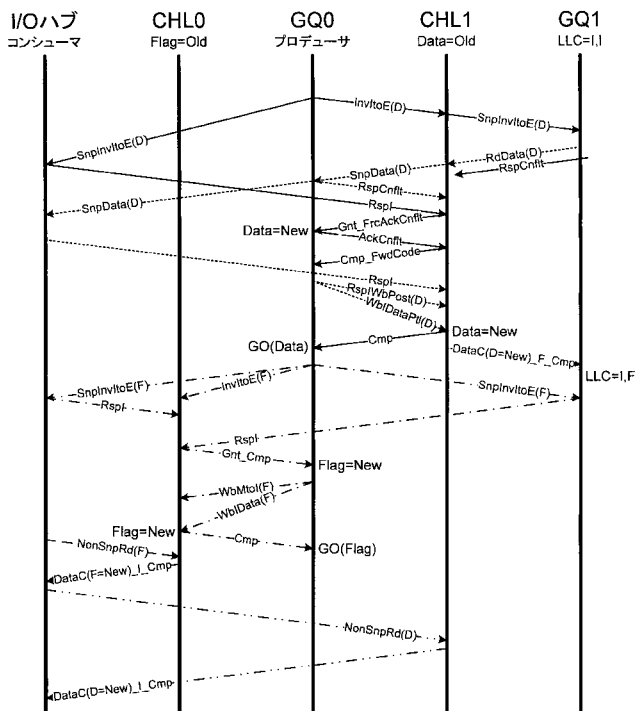
【図 6 b】



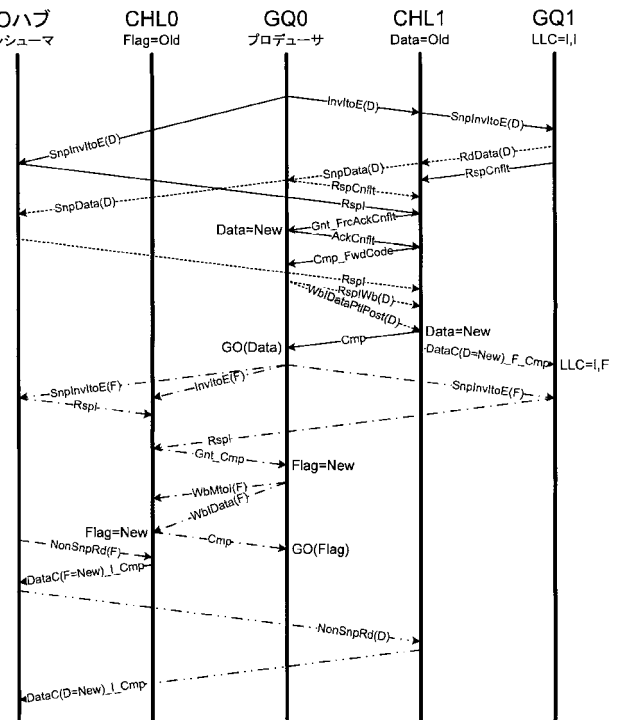
【図 6 c】



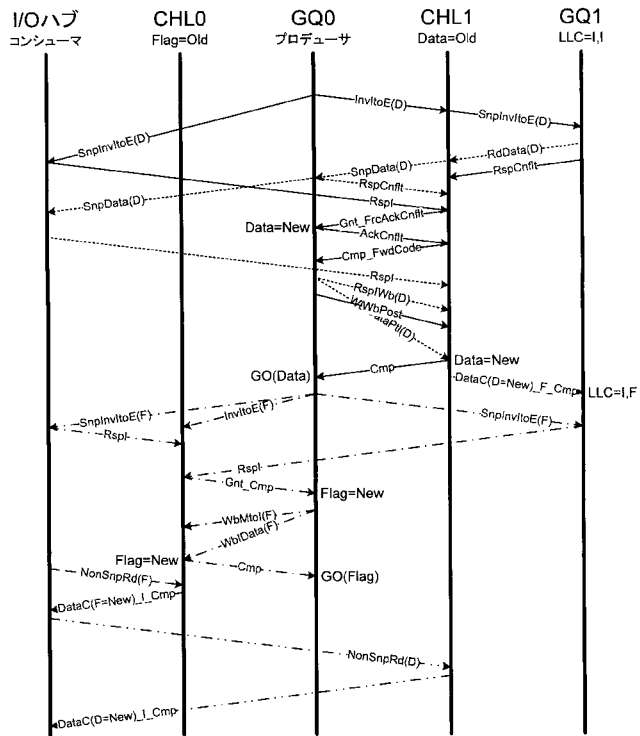
【図 7 a】



【図 7 b】



【図 7 c】



フロントページの続き

(51)Int.Cl.	F I	テーマコード(参考)
	G 0 6 F 13/42	3 1 0
	G 0 6 F 12/08	5 0 1 C
	G 0 6 F 12/08	5 1 9 E

(72)発明者 チン - ツン チョウ
アメリカ合衆国 9 4 3 0 6 カリフォルニア州 パロアルト サウス コート 3 6 1 6

(72)発明者 ロバート ジェイ . サフラネク
アメリカ合衆国 9 7 2 2 9 オレゴン州 ポートランド ノースウェスト ネカニカム ウェイ
5 8 1 6

F ターム(参考) 5B005 JJ01 KK14 MM01 NN76 PP11
5B077 DD02

【外国語明細書】

2010027048000001.pdf

2010027048000002.pdf

2010027048000003.pdf

2010027048000004.pdf