

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第5498505号
(P5498505)

(45) 発行日 平成26年5月21日(2014.5.21)

(24) 登録日 平成26年3月14日(2014.3.14)

(51) Int.Cl.

F I

G O 6 F 13/362 (2006.01)

G O 6 F 13/362 5 1 0 A

請求項の数 15 (全 18 頁)

(21) 出願番号	特願2011-541110 (P2011-541110)	(73) 特許権者	510000633
(86) (22) 出願日	平成20年12月19日(2008.12.19)		エスティー-エリクソン、ソシエテ、アノ ニム
(65) 公表番号	特表2012-512472 (P2012-512472A)		スイス国ブラン-レーズアト、シュマン、 デュ、シャン-デーフィュー、39
(43) 公表日	平成24年5月31日(2012.5.31)	(74) 代理人	100076428
(86) 国際出願番号	PCT/EP2008/068039		弁理士 大塚 康德
(87) 国際公開番号	W02010/069398	(74) 代理人	100112508
(87) 国際公開日	平成22年6月24日(2010.6.24)		弁理士 高柳 司郎
審査請求日	平成23年12月7日(2011.12.7)	(74) 代理人	100115071
			弁理士 大塚 康弘
		(74) 代理人	100116894
			弁理士 木村 秀二
		(74) 代理人	100130409
			弁理士 下山 治

最終頁に続く

(54) 【発明の名称】 データバースト間の競合の解決

(57) 【特許請求の範囲】

【請求項1】

データバースト間の競合を解決する方法であって、
前記競合するデータバーストのうちの1つである第1のデータバーストを選択するステップと、

ここで、前記競合するデータバーストの各々は要求に対する応答である；
選択された前記第1のデータバーストのクリティカル部分の長さを判定するステップと

、
選択された前記第1のデータバーストの前記クリティカル部分を処理するステップと、
前記競合するデータバーストのうちの1つである第2のデータバーストを選択するステップと、

選択された前記第2のデータバーストのクリティカル部分の長さを判定するステップと

、
前記第1のデータバーストの非クリティカル部分を処理する前に前記第2のデータバーストの前記クリティカル部分を処理するステップと

を有することを特徴とする方法。

【請求項2】

前記クリティカル部分の前記長さを判定するそれぞれのステップは、
該当する選択されたデータバーストに含まれたタグを読み出すステップを有し、前記タグが前記該当する選択されたデータバーストの前記クリティカル部分の前記長さを示すこ

10

20

とを特徴とする請求項 1 に記載の方法。

【請求項 3】

該当する選択されたデータバーストの前記クリティカル部分の前記長さを示す前記タグを変更するステップと、

前記変更されたタグを含む前記該当する選択されたデータバーストを転送するステップと

を更に含むことを特徴とする請求項 2 に記載の方法。

【請求項 4】

前記クリティカル部分の前記長さを判定するそれぞれのステップは、

該当する選択されたデータバーストの前記クリティカル部分の処理が完了したことを示すメッセージを受信するステップを含むことを特徴とする請求項 1 に記載の方法。

10

【請求項 5】

前記クリティカル部分の前記長さを判定するそれぞれのステップは、

該当する選択されたデータバーストの優先順位に基づいて前記該当する選択されたデータバーストの前記クリティカル部分の前記長さを判定するステップと、

前記データバーストの前記クリティカル部分の前記長さを示すタグを前記該当するデータバーストに含めるステップと

を含むことを特徴とする請求項 1 に記載の方法。

【請求項 6】

前記クリティカル部分の前記長さを判定するそれぞれのステップは、

該当する選択されたデータバーストが検索される記憶場所に基づいて前記該当する選択されたデータバーストの前記クリティカル部分の前記長さを判定するステップを含むことを特徴とする請求項 1 に記載の方法。

20

【請求項 7】

バス調停装置において実行され、

それぞれのソースから前記競合するデータバーストを受信するステップを先行ステップとして含むことを特徴とする請求項 1 乃至 6 のいずれか 1 項に記載の方法。

【請求項 8】

データソースにおいて実行され、

データに対する複数の要求を受信するステップと、

データに対するそれぞれの要求に回答して前記競合するデータバーストを形成するステップとを先行ステップとして含むことを特徴とする請求項 1 乃至 6 のいずれか 1 項に記載の方法。

30

【請求項 9】

該当する選択されたデータバーストの前記クリティカル部分が処理されている場合、前記競合するデータバーストのうちの他の前記クリティカル部分を処理する間、前記該当する選択されたデータバーストの前記非クリティカル部分を待機させるステップと、

前記該当する選択されたデータバーストの前記非クリティカル部分が待機させられている間、処理される多数の他のデータバーストのカウントを維持するステップと、

前記カウントが所定の閾値に到達する場合に前記該当する選択されたデータバーストの前記非クリティカル部分を優先させるステップとを含むことを特徴とする請求項 1 乃至 8 のいずれか 1 項に記載の方法。

40

【請求項 10】

データ転送システムにおいてデータバースト間の競合を解決する調停装置であって、

前記競合するデータバーストのうちの1つである第 1 のデータバーストを選択するステップと、

選択された前記第 1 のデータバーストのクリティカル部分の長さを判定するステップと

、

ここで、前記競合するデータバーストの各々は要求に対する応答である；

選択された前記第 1 のデータバーストのクリティカル部分の長さを判定するステップと

50

、
選択された前記第 1 のデータバーストの前記クリティカル部分を処理するステップと、前記競合するデータバーストのうちの 1 つである第 2 のデータバーストを選択するステップと、

選択された前記第 2 のデータバーストのクリティカル部分の長さを判定するステップと

、
前記第 1 のデータバーストの非クリティカル部分を処理する前に前記第 2 のデータバーストの前記クリティカル部分を処理するステップとを実行することにより、前記データバースト間の競合を解決するように構成されることを特徴とする調停装置。

【請求項 1 1】

受信したデータバーストに対するポイントを格納する少なくとも 1 つのメモリを備えることを特徴とする請求項 1 0 に記載の調停装置。

【請求項 1 2】

データ転送システムにおいてデータ要求を形成する方法であって、
 データバーストのクリティカル部分の長さを示すタグを前記データバーストに対する要求に含めるステップを有することを特徴とする方法。

【請求項 1 3】

データバーストのクリティカル部分の長さを示すタグを前記データバーストに対する要求に含むことにより、データ要求を形成するように構成されることを特徴とする組込みシステムに関連付けられたデータ要求装置。

【請求項 1 4】

複数のバスコンポーネントにより相互接続された複数のホストと、複数のメモリリソースとを備える組込みシステムにおいてデータ要求を処理する方法であって、

前記複数のホストのうちの 1 つである前記ホストにおいて、前記要求されたデータが受信されるべきメモリリソースを識別するデータ要求を形成し、及び、前記データバーストのクリティカル部分の長さを示すタグを前記データ要求に含め、

前記識別されたメモリリソースにおいて、前記要求されたデータを含むバーストランザクションを形成し、及び、前記データバーストのクリティカル部分の長さを示す第 2 のタグを前記バーストランザクションに含め、

前記バスコンポーネントのうちのそれぞれ 1 つと関連付けられた少なくとも 1 つの調停装置において、前記第 2 のタグの値に基づいて前記バーストランザクションを分割し、及び前記データバーストの非クリティカル部分より高い優先順位を前記データバーストの前記クリティカル部分に与える

ステップを有することを特徴とする方法。

【請求項 1 5】

複数のデータ要求装置と、複数のデータソースと、前記データ要求装置及び前記データソースを相互接続するバスファブリックとを備える組込みシステムであって、前記バスファブリックが複数のバスコンポーネントを備え、前記バスコンポーネントがそれぞれの調停装置と関連付けられており、各調停装置が、

バスコンポーネントに関連付けられた調停装置によって、前記競合するデータバーストのうちの 1 つである第 1 のデータバーストを選択し、

ここで、前記競合するデータバーストの各々は要求に対する応答である；

選択された前記第 1 のデータバーストのクリティカル部分の長さを判定し、

選択された前記第 1 のデータバーストの前記クリティカル部分を処理し、

前記競合するデータバーストのうちの 1 つである第 2 のデータバーストを選択し、

選択された前記第 2 のデータバーストのクリティカル部分の長さを判定し、

前記第 1 のデータバーストの非クリティカル部分を処理する前に前記第 2 のデータバーストの前記クリティカル部分を処理する

こととによりデータバースト間の競合を解決するように構成される組込みシステム。

【発明の詳細な説明】

10

20

30

40

50

【技術分野】**【0001】**

本発明は、電子システムにおいて装置間で転送されるデータバースト間の競合を解決する方法及びそのようなシステムで使用される装置に関する。

【背景技術】**【0002】**

種々のコンポーネントバスから構成されたバスマトリックス又はバスファブリックにより全てが相互接続された種々のプロセッサ等の多数のマスタ装置、及び、種々のメモリ等の多数のリソース装置が存在する組込みシステムは既知である。組込みシステムが動作するために、データは、一般に多数のデータワードを含むデータバーストにおいて、これらの装置間で転送されなければならない。メモリ素子等のリソース又はバスファブリックのコンポーネントバス等のリソースを使用したいという多くの要求が同時に発生することは、そのようなシステムにおいて一般的である。

10

【0003】

そのような場合、或るトランザクションは、1つ又は複数の他のトランザクションより優先されなければならないため、システムの全体の性能が損なわれる。

【発明の概要】**【発明が解決しようとする課題】****【0004】**

米国特許第5,668,975号公報において、メモリブロックからのデータに対して多くの要求があるような特定の例において調停に対する1つの手法が説明される。この文献で説明された方法において、要求されたデータ転送の各々は、クリティカルワードと1つ以上の非クリティカルワードとに分裂する。その後、各クリティカルワードは各非クリティカルワードより優先され、クリティカルワード及び非クリティカルワードはそれぞれの優先順位で処理される。

20

【0005】

しかし、この文献は、要求間の調停が多数のポイントで必要とされる可能性がある上述したような組込みシステムにおいて発生する更なる問題に対する解決方法を全く提供しない。

【課題を解決するための手段】

30

【0006】

本発明の第1の態様によると、データバースト間の競合を解決する方法が提供される。競合するデータバーストのうちの第1のデータバーストが選択され、第1の選択されたデータバーストのクリティカル部分の長さが判定される。次に、選択されたデータバーストのクリティカル部分が処理される。競合するデータバーストのうちの第2のデータバーストが選択され、第2の選択されたデータバーストのクリティカル部分の長さが判定される。第2の選択されたデータバーストのクリティカル部分は、選択されたデータバーストの非クリティカル部分の前に処理される。

【0007】

本発明の第2の態様によると、第1の態様の方法に従って動作する調停装置が提供される。

40

【0008】

本発明の第3の態様によると、データ転送システムにおいてデータ要求を形成する方法が提供される。この方法によると、データバーストのクリティカル部分の長さを示すタグがデータバーストに対する要求に含まれる。

【0009】

本発明の第4の態様によると、第3の態様の方法に従って動作するデータホストが提供される。

【0010】

本発明の第5の態様によると、複数のバスコンポーネントにより相互接続された複数の

50

ホストと、複数のメモリリソースとを備える組込みシステムにおいてデータ要求を処理する方法が提供される。ホストにおいて、データ要求が形成され、前記データ要求は、要求されたデータが受信されるべきメモリリソースを識別し且つデータバーストのクリティカル部分の長さを示すタグを更に含む。識別されたメモリリソースにおいて、要求されたデータを含むバーストランザクションが形成され、バーストランザクションは、データバーストのクリティカル部分の長さを示す第2のタグを含む。前記バスコンポーネントのうちそれぞれ1つと関連付けられた少なくとも1つの調停装置において、バーストランザクションは第2のタグの値に基づいて分裂し、データバーストの非クリティカル部分より高い優先順位がデータバーストのクリティカル部分に与えられる。

【0011】

10

本発明の第6の態様によると、複数のデータ要求装置と、複数のデータソースと、データ要求装置及びデータソースを相互接続するバスファブリックとを備える組込みシステムが提供され、バスファブリックが複数のバスコンポーネントを含む。バスコンポーネントは、各々が第1の態様の方法に従って動作する調停装置の形態でそれぞれの調停装置と関連付けられている。

【0012】

従って、本発明の実施形態により、ランザクションがランザクションのクリティカル部分に対して最小の遅延で実行されるようになり、クリティカル部分の長さを考慮されるようになる。

【0013】

20

従って、本発明の実施形態において、メタデータは、システムが重要なパラメータ及び性能をより適切に制御し且つ最適化できることを目標とした複雑なシステムにおけるデータランザクションにおいて要求に追加される。ランザクションのクリティカル部分の長さを示すタグを追加することにより、リアルタイムで設定及び構成を継続的に更新する必要性がバスファブリックから除去される。要求装置は、ファブリックを予めプログラムしなければならないのではなく、このメタデータをその要求にタグ付けする。従って、ファブリックは、特定の詳細でプログラムされるのではなく、リアルタイムで適切に選択するためにタグにおいてメタデータを使用して、全てのランザクションに適用する「ポリシー」を与えられる。ポリシーは、必要に応じてより遅い時間フレームで変更されるが詳細に述べるように変更される必要はないため、システムの負担が軽くなる。また、種々の設計間で機能又はサブシステムを移動させる場合、それは、構成及び検証時間を判定することが減少して再利用性が向上することを意味する。

30

【図面の簡単な説明】

【0014】

【図1】本発明の一態様に係る組込みシステムを示す概略ブロック図である。

【図2】図1に示されたようなシステムにおける命令メモリを示し、ランザクションの可能な形態を示す概略図である。

【図3】図2の命令メモリと関連付けられた第1のランザクションを示す図である。

【図4】図2の命令メモリと関連付けられた第2のランザクションを示す図である。

【図5】本発明の一態様に係るバス調停装置を示す概略図である。

40

【図6】図5のバス調停装置において実行された方法を示すフローチャートである。

【図7】図6の方法を実行した結果を示す図である。

【図8】図6の方法を実行した結果を示す別の図である。

【図9】図6の方法を実行した結果を更に示す図である。

【図10】図1のシステムにおける第1の装置において実行された方法を示すフローチャートである。

【図11】図10の方法に従って生成されたデータ要求を示す概略図である。

【図12】図1のシステムにおける装置において実行された別の方法を示すフローチャートである。

【図13】図1のシステムにおける装置において実行された更に別の方法を示すフローチャートである。

50

ャートである。

【発明を実施するための形態】

【0015】

図1は、本発明が適用される一般的な組込みシステムを示す。図1に示されたシステムは単なる一例として提示され、本発明が種々の多くの他のシステムに適用されるため、システムの詳細は限定するものではないことが理解されるだろう。

【0016】

更に詳細には、図1は、例えば移動通信装置に存在し得る組込みシステム10を示す。システム10は、例えば他から受信したデータに対してコードを実行し且つ動作を実行する処理装置の形態であってもよい多数のデータマスタ、すなわちホストを含む。図1に示されたように、これらのデータマスタは、アプリケーションCPU20、マルチメディアDSP22、マルチメディア(MM)ハードウェア(HW)アクセラレータ24、ダイレクトメモリアクセス(DMA)システムブロック28を介して接続された周辺スイート26として一括して示された装置、アクセスCPU30、アクセスDSP32及びモデムハードウェア34である。システム10は、他のハードウェア(HW)ブロック36~38を更に含んでもよい。各ホストがソフトウェア又はハードウェアのサブコンポーネントの形態で多数のデータリクエストを含んでもよいことは、理解されるだろう。

【0017】

データ及び命令(instruction)は、共にメモリソリューション48と呼ばれる外部メモリインタフェース(EMIF)46、外部メモリ44及び外部メモリインタフェース46を介して接続されたROM40、RAM42及び外部メモリ44等の種々のメモリ素子に格納される。ここでも、あらゆるメモリ素子又はメモリマップド装置が多数のデータソースを含んでもよいことは、理解されるだろう。

【0018】

これらの種々のブロックは、バスファブリック、すなわちバスマトリックス50により相互接続される。すなわち、全てのブロックが接続される単一のバスはない。バスファブリック50は、図1でアプリケーションバスマトリックス52、アクセスバスマトリックス54、共通バスマトリックス56及びシステムメモリバス58として識別された多数のバスコンポーネントを含む。

【0019】

この例示的な例において、アプリケーションCPU20、マルチメディアDSP22、マルチメディア(MM)ハードウェア(HW)アクセラレータ24及びダイレクトメモリアクセス(DMA)システムブロック28は、それぞれのバスセグメントによりアプリケーションバスマトリックス52に接続される。ダイレクトメモリアクセス(DMA)システムブロック28、アクセスCPU30、アクセスDSP32及びモデムハードウェア34は、それぞれのバスセグメントによりアクセスバスマトリックス54に接続される。ROM40、RAM42及び外部メモリインタフェース(EMIF)46は、それぞれのバスセグメントによりシステムメモリバス58に接続される。アプリケーションバスマトリックス52、アクセスバスマトリックス54及びシステムメモリバス58に加えてダイレクトメモリアクセス(DMA)システムブロック28及び他のハードウェア(HW)ブロック36~38は、それぞれのバスセグメントにより共通バスマトリックス56に接続される。従って、組込みシステム10のコンポーネントブロックのうちのいずれかの2つの間の接続は、多数のバスセグメントを通過しなければならなくなる。特に、アプリケーションバスマトリックス52、アクセスバスマトリックス54及びシステムメモリバス58の各々が多数のそれぞれのバスセグメントにより共通バスマトリックス56に接続されるため、接続を可能にするバスが多く存在する。

【0020】

従って、この例において、リソースが共有されるため、要求されたデータ転送間で使用可能なリソースに対する競合が存在する可能性のある多くのポイントがあることを実証している。例えば多数のデータバーストは、同時に特定のバス又はバスコンポーネントへの

10

20

30

40

50

アクセスを要求し得る。別の例として、外部メモリは、同時に多数のソースからデータに対する多くの要求を受信してもよい。そのような競合の解決方法を以下に更に詳細に説明する。

【0021】

バス又はバスコンポーネントにおけるデータ競合(data contention)は、例えばバス又はバスコンポーネントと関連付けられたバス調停装置を使用して解決される。例えば図1には、アクセスバスマトリックス54内に含まれた第1のバス調停装置60、及び、共通バスマトリックス56内に含まれた第2のバス調停装置62を示している。図1は、外部メモリインタフェース46の一部を形成するメモリコントローラ(MC)64を更に示し、メモリコントローラ64は調停装置(AD)66を含む。

10

【0022】

本明細書において、データ又は命令を処理ブロックに提供するのが遅延することにより、システム全体又はシステムの一部の待ち時間が増加するキャッシュベースのシステムを参照して本発明を説明する。

【0023】

図2は、このように示された本発明の実施形態におけるデータの構造を示す。

【0024】

特に、図2は、命令メモリ80を示し、データを含むメモリの構造が実質的には同一であってもよいことが理解される。命令メモリ80は、32ビット幅であり、8行のグループで命令を格納する。8行のグループの各々は、キャッシュラインを含む32バイトのブロックを形成する。従って、0x00001020の開始アドレスを含むキャッシュラインは、0x00001020、0x00001024、0x00001028、0x0000102C、0x00001030、0x00001034、0x00001038及び0x0000103Cの開始アドレスを含む命令メモリラインから形成される。この例示的な実施形態において、メモリ要求は常に完全なキャッシュラインを使用する。

20

【0025】

図2は、場所0x0000102Aでキャッシュミスがある状況を示す。すなわち、これは、メモリから要求される第1のワードである。上述したように、この記憶場所を含むキャッシュライン82全体が要求されなければならない。

【0026】

図3は、このキャッシュラインが標準的な非ラッピングライン84として要求される状況を示す。すなわち、全てのキャッシュラインは記憶場所の順序で要求される。示されたレイテンシ期間は、キャッシュラインの開始からキャッシュミスの場所までの期間である。

30

【0027】

図4は、このキャッシュラインが最初にクリティカルワードを含むラッピングバースト86として要求される状況を示す。すなわち、全てのキャッシュラインは上述したように要求されるが、アドレス0x00001028で開始するクリティカルワードはメモリから要求された最初のワードである。従って、レイテンシ期間は、クリティカルワードの開始からキャッシュミスの場所までの期間だけであるため、この場合は減少する。ラッピングバーストにおいて、開始ワードからキャッシュラインの終わりまでのバーストの部分は命令を実行するうえでクリティカルであるが、キャッシュラインの開始からクリティカルワードの開始までのラップされた部分は一般にクリティカルではない。

40

【0028】

従って、特に本発明は、キャッシュミス又はダーティデータラインのライトバックのいずれかを主メモリに提供するためにシステムに対してバーストランザクション要求を頻繁に生成するキャッシュに基づくCPUの必要性を満たすシステムに関するが、それに限定されない。そのようなシステムにおいて、特にデータ及びコードの双方に対するキャッシュラインフィルにおける待ち時間を減少することにより、性能は向上する。バーストが最初のクリティカルワードとして配置されると仮定すると、クリティカルであるバースト

50

の量はプロセッサに依存したままである。最初のクリティカルワードの目的は、プロセッサが最短のレイテンシで必要なものを提供することである。データの場合、それは16のバーストの1つ又は2つのワードであってもよく、コードの場合、それはより多くのワードであってもよい。

【0029】

プロセッサデータランザクションの場合、処理される特定の値に対する要望は2つのことを意味する。1つ目は命令毎にほんの少量が必要とされることであり、2つ目は全ての命令が外部データを必要とするとは限らないことである。その結果、データのバーストに対する必要性は、多くの命令に及ぶ。これは、バーストにおいて、最初のワードが一般にクリティカルであるが、バーストにおける後続データがそれほどクリティカルではないこと、すなわち介在する命令を実行することにより、後続データがより緩和された速度で読み出されるかあるいは書き込まれ、且つ最初のワードとして優先される必要がなくなることを意味する。

10

【0030】

命令の場合、これらは逐次迅速に実行され、その速い速度により性能が向上する。しかし、コードの実行は、コードにおいて頻繁に発生する可能性のあるジャンプ及びブランチにより変化する。その結果、命令におけるキャッシュミスに対するバーストランザクションは、クリティカルワードの最初のバーストにおいて、クリティカルであるのが最初のワードだけでなく、バーストにおける多数の連続したワードであることを意味する。コードの連続した特性を考慮すると、クリティカルワードの数は、バーストがラップし且つより少なくともよいポイントで最大となり、コードにおいてジャンプ又はブランチの頻度により判定される。この場合、クリティカル分裂ポイントは、バーストがクリティカルワードの前からのデータワードを含むバーストを完成するようにラップするポイントに配置される。

20

【0031】

以下に更に詳細に説明されるように、キャッシュフィル又はライトバック以外のバーストを生成する他の種類のプロセッサの場合、データがクリティカルでなくなるポイントは、他の基準に基づいて判定される。

【0032】

図5は、本発明の一実施形態に従ってバス調停装置の一般的な形態を示す概略図である。例えばバス調停装置100は、図1に示されたバス又はコンポーネントバスのいずれかと関連付けられてもよく、競合するデータバーストが例えば多数のソースから受信されるシステムにおいてあらゆるポイントに配置されてもよく、データバーストがバス等の共有リソースにアクセスするのを制御する。種々の競合するデータ要求が処理されるべき方法を判定するために、実質的に同様の装置が共有メモリリソースと関連付けられたメモリコントローラの一部として実現されることは、理解されるだろう。

30

【0033】

バス調停装置100は、マルチプレクサ110を介して調停装置に多重化され、それぞれポート0、ポート1、ポート2及びポート3と呼ばれた4つの入力ポート102、104、106、108を有するものとして図5に示される。第1のメモリ112及び第2のメモリ114は、マルチプレクサ110に接続される。以下に更に詳細に説明される調停制御処理を実行するアービタ116は、マルチプレクサ110に更に接続される。データへのポイントは、アービタ116において実行された処理に従って第1のメモリ112及び第2のメモリ114に格納される。カウンタ118はアービタ116と関連付けられる。第1のメモリ112及び第2のメモリ114は、更なるマルチプレクサ120を介して出力バス122に接続される。この例示的な例において、出力バス120がバス(又はバスへの接続)であってもよい場合、バス調停装置100は、バスへのアクセスを要求するデータバースト間の競合を解決できる。

40

【0034】

図6は、アービタ116において実行され、本発明の一態様に従って1つの例示的な処

50

理を示すフローチャートである。図 6 に示されたステップのうちのいくつかを省略し、且つ/あるいは他のステップを含み且つ/あるいは示されたステップの順序を変更する多くの他のある程度類似した処理が実行可能であることは、理解されるだろう。

【 0 0 3 5 】

図 6 に示された処理のステップ 1 4 0 において、多数のトランザクションは、バス調停装置 1 0 0 により受信されるか、あるいは受信されている。ステップ 1 4 2 において、これらのトランザクションのうちの 1 つは、例えば何らかの適当な機構によりトランザクションに割り当てられた優先順位に基づいて選択される。

【 0 0 3 6 】

ステップ 1 4 4 において、選択されたトランザクションは従来通り続行することを許可され、他の待ち状態のトランザクションは待機させられる。

10

【 0 0 3 7 】

ステップ 1 4 6 において、選択されたトランザクションは進行中であり、アービタ 1 1 6 は、シーケンスをプリエンプト (preempt) するのが可能なワード、すなわちトランザクションをクリティカル部分と非クリティカル部分とに分裂させることが可能なワードを判定する。これが達成される種々の方法を以下に更に詳細に説明する。しかし、このポイントにおいて、アービタ 1 1 6 は、自身のいくつかの解析を実行することにより又は他の場所で実行された解析に依存し且つトランザクションの一部を形成するタグを読み出すことにより、クリティカル部分の長さ、すなわちトランザクションが分裂してもよいポイントを判定してもよい。

20

【 0 0 3 8 】

ステップ 1 4 8 において、値は、選択されたトランザクションと関連付けられたカウンタ 1 1 8 のうちの 1 つにロードされる。

【 0 0 3 9 】

ステップ 1 5 0 において、何らかの待ち状態のトランザクションがあるか、すなわちクリティカル部分が開始されていないトランザクションが進行中であるか、あるいは全ての待ち状態のトランザクションのクリティカル部分が完了したか又は進行中であるかを判定する。クリティカル部分が進行していない待ち状態のトランザクションがあることがステップ 1 5 0 において判定される場合、ステップ 1 5 2 に進み、これらの待機中のトランザクションのうちの一方が例えばトランザクションに割り当てられた優先順位に基づいて選択される。待機中のトランザクションを選択することは、トランザクションの切り替えが実行される際に更なる待ち時間が発生することなく処理が継続することを意味する。

30

【 0 0 4 0 】

ステップ 1 5 4 において、現在実行中のトランザクションの最後のクリティカルワードに到達したかを判定する。このステップは、最後のクリティカルワードが処理されるまで繰り返される。

【 0 0 4 1 】

現在実行中のトランザクションの最後のクリティカルワードに到達した際、このトランザクションは待機させられる。例えば、トランザクションの未処理の非クリティカル部分は、バス調停装置 1 0 0 に格納されてもよい。あるいは、バス調停装置 1 0 0 は、待機信号をトランザクションのソースに再送し、そのソースがトランザクションの非クリティカル部分を送出するのを遅延させるか又は遅延後にトランザクションの非クリティカル部分を再送させる。

40

【 0 0 4 2 】

処理は、ステップ 1 5 6 において、ステップ 1 5 2 において選択された待機中のトランザクション、例えば最も優先順位の高いトランザクションに切り替えられる。

【 0 0 4 3 】

そのように切り替えられる場合、ステップ 1 4 8 のあらゆる性能にロードされたあらゆるカウンタ値はステップ 1 5 8 において 1 ずつ減少し、これらのカウンタ値のうちのいずれかがそれによりゼロに到達したかをステップ 1 6 0 においてテストする。ゼロに到達し

50

た場合、ステップ162において、そのカウンタ値と関連付けられたトランザクションの優先順位は上がる。例えば、そのトランザクションが全ての待機中のトランザクションの中で最も高い優先順位を自動的に与えられてもよいため、この機構は、トランザクションが待機させられる最長時間を確立する。

【0044】

ステップ162において1つ又は複数の優先順位を調整した後、あるいはカウンタ値がゼロに到達しなかったことをステップ160において判定した後、ステップ144に戻り、ステップ152において選択された待機中のトランザクションが進行する。

【0045】

受信した全てのクリティカル部分が完了したかあるいは進行中であることがあらゆる回復のステップ150において判定される場合、処理は、受信したトランザクションの非クリティカル部分の進行に移り、特に、優先順位の最も高い待機中のトランザクションが選択されるステップ164に進む。

【0046】

その後、クリティカルワードが依然として進行中であるかを判定するステップ166に進む。進行中である場合、ステップ154に戻り、上述したように継続する。しかし、非クリティカル部分が既に進行中である場合、現在実行中のトランザクションの最後のワードに到達したかを判定するステップ168に進む。このステップは、最後のクリティカルワードが処理されるまで繰り返される。

【0047】

現在実行中のトランザクションの最後のワードに到達した場合、待機させられている何らかのトランザクションがあるかを判定するステップ170に進む。待機させられているトランザクションがない場合、ステップ180に進んで終了する。

【0048】

待機中のトランザクションがある場合、処理は、ステップ172において、ステップ164において選択された待機中のトランザクション、例えば最も優先順位の高いトランザクションに切り替えられる。

【0049】

上述したように、そのように切り替えられる場合、ステップ148のあらゆる性能にロードされたあらゆるカウンタ値はステップ174において1ずつ減少し、これらのカウンタ値のうちのいずれかがそれによりゼロに到達したかをステップ176においてテストする。ゼロに到達した場合、ステップ178において、そのカウンタ値と関連付けられたトランザクションの優先順位は上がる。例えば、そのトランザクションが全ての待機中のトランザクションの中で最も高い優先順位を自動的に与えられてもよいため、この機構は、トランザクションが待機させられる最長時間を確立する。

【0050】

ステップ178において1つ又は複数の優先順位を調整した後、あるいはカウンタ値がゼロに到達しなかったことをステップ176において判定した後、ステップ144に戻り、ステップ172において進行が切り替えられたトランザクションが更に進行する。

【0051】

従って、全ての既存のトランザクションは優先順位で完了する。また、既存のトランザクションが提供されている間に発生する全ての新しいトランザクションは、上述したようにキューに追加され且つ評価される。

【0052】

この例示的な実施形態において、全てのトランザクションのクリティカル部分が全ての非クリティカル部分より高い優先順位を与えられると仮定される。しかし、トランザクションのクリティカル部分の長さが判定されると、そのトランザクションの非クリティカル部分は、例えば一般にそのトランザクションに与えられた優先度を考慮して、少なくとも1つの他のトランザクションのクリティカル部分より高い優先度を与えられる。

【0053】

10

20

30

40

50

上述したように、全てのトランザクションは異なるプロセッサからのものであると仮定される。不可能ではないが、同一のプロセッサから2つのトランザクションに介在するのは望ましくないため、更なるステップはこの手段を回避するようにする。

【0054】

図7は、上述のシステムが実際に動作する方法を示し、図5は、処理のあるポイントにおけるバス調停装置100の状況を示す。特に、図5は、トランザクション130-1及び130-2が、それぞれポート1 104及びポート2 106上でバス調停装置100により受信されており、トランザクション130-0及び130-3が、それぞれポート0 102及びポート3 108上でバス調停装置100によりまもなく受信される状況を示す。

10

【0055】

この場合、トランザクション130-1が受信され且つそれがクリティカル部分C-1及び非クリティカル部分S-1から構成されることが判定されており、トランザクション130-2が受信され且つそれがクリティカル部分C-2及び非クリティカル部分S-2から構成されることが判定されている。図7の135aにおいて示されたように、クリティカル部分C-1、C-2に対するポインタはメモリ112に配置され、非クリティカル部分S-1、S-2に対するポインタはメモリ114に配置される。

【0056】

データは、格納されてもよいが、この図示した例では格納されない。接続は、進行可能になるまで待機させられる。

20

【0057】

この例示的な例において、トランザクション130-2がトランザクション130-1より優先順位の高いことが判定されているため、図5は、バス122上に最初に出力されるトランザクション130-2のクリティカル部分C-2を示す。

【0058】

図5に示した例において、トランザクション130-3は、トランザクション130-2のクリティカル部分C-2が完了する前に入力ポート108上に到達し、トランザクション130-0はその後まもなく到達する。これらのトランザクションは共にトランザクション130-1より優先度が高いと判定されるため、トランザクション130-3、130-0のクリティカル部分C-3、C-0の長さを判定した後、これらのトランザクションのクリティカル部分及び非クリティカル部分へのポインタはメモリ112、114に格納され、トランザクション130-3の到達後のメモリコンテンツが図7において135bで示され、トランザクション130-0の到達後及びクリティカル部分C-2の完了後のメモリコンテンツが図7において135cで示される。トランザクション130-1のクリティカル部分C-1の後にバス122上で次に出力されるのはこれらのC-3、C-0であり、これらのクリティカル部分の各々の完了後のメモリ112、114のコンテンツが、図7においてそれぞれ135d、135e、135fで示される。これらの後に非クリティカル部分S-2、S-3、S-0、S-1が同一の優先順位で行われる。

30

【0059】

図8及び図9は、2つのトランザクションが共通バスバスを介して調停ポイントを通している簡略化された例を使用して競合解決システムが動作してもよい方法を示す。

40

【0060】

図8は、データがそれぞれのデータバス、データバスA¹及びデータバスB¹を介して2つの異なるソース220、222から第1のバス調停ポイント224に渡されている状況を示す。次に2つのデータトランザクションは、共通バスバス226を介して更なるバス調停ポイント228に渡される。バス調停ポイント228において、2つのデータトランザクションは、再分割され、それぞれのデータバス、データバスA²及びデータバスB²上でそれぞれのプロセッサ230、232に送出される。

【0061】

第1のプロセッサ230に対する第1のトランザクション234は、クリティカル部分

50

236及び非クリティカル部分238から構成される。第2のプロセッサ232に対する第2のトランザクション240は、クリティカル部分242及び非クリティカル部分244から構成される。

【0062】

図8は、246において上述のプリエンティブ調停方式の動作を示し、第2のトランザクション240のクリティカル部分242の後に第1のトランザクション234のクリティカル部分236が最初に送出され、次に第1のプロセッサ230の非クリティカル部分238及び第2のトランザクション240の非クリティカル部分244が送出される。

【0063】

データがプロセッサ230、232に到達する時間は、データパス、データパスA²及びデータパスB²に対するデータワードの水平位置により図8に示される。従って、第1のトランザクション234のクリティカル部分236が第1のプロセッサ230に到達する時間と比較して、第2のトランザクション240のクリティカル部分242は、遅延248後に第2のプロセッサ232に到達することが分かる。

【0064】

比較として、図8は、250において上述の従来調停方式の動作を比較として更に示し、全ての第2のトランザクション240の後に全ての第1のトランザクション234が最初に送出される。そのような状況において、第1のトランザクション234のクリティカル部分236が第1のプロセッサ230に到達する時間と比較して、第2のトランザクション240のクリティカル部分242は、全ての第1のトランザクション234が第1

【0065】

従って、上述のプリエンティブ調停方式が使用される場合、第2のトランザクション240のクリティカル部分242の遅延はより短くなる。これは、第1のプロセッサ230の非クリティカル部分238がより長い遅延を被ることによるものであるが、このより長い遅延によりプロセッサ自体の動作に遅延が生じることは決してない。

【0066】

この理由を図9で示す。図9は、第1のトランザクション234及び第2のトランザクション240がメモリ命令を含む状況を示す。

【0067】

260において示されたように、第1のプロセッサ230(CPU A)の実行スレッドは、キャッシュミスが発生し、メモリから命令が取り出されることを要求し、第1のトランザクションのクリティカル部分236はできるだけ最短の遅延後に取り出される。同様に、第2のプロセッサ232(CPU B)の実行スレッドは、262においてキャッシュミスに到達し、メモリから命令が取り出されることを要求し、第2のトランザクションのクリティカル部分242は、第1のトランザクションのクリティカル部分236が取り出されたらすぐ取り出される。

【0068】

上述したように、図9に更に示されたが、第2のトランザクションのクリティカル部分242が取り出された際、第1のトランザクションの非クリティカル部分238は第1のプロセッサに送信され、次に第2のトランザクションの非クリティカル部分244は第2のプロセッサに送信される。

【0069】

第1のプロセッサ(CPU A)上で動作する実行スレッドは、264において示されたように、プロセッサが第1のトランザクションのクリティカル部分236を受信したらすぐ最高速度で再開し、第1のトランザクションの非クリティカル部分238を受信する際の更なる遅延により、この実行スレッドが遅延することはない。一方、第2のプロセッサ(CPU B)上で動作する実行スレッドは、266において示されたように、プロセッサが第2のトランザクションのクリティカル部分242を受信したらすぐ最高速度で再

10

20

30

40

50

開するため、これを受信する際の遅延が減少することは非常に有利である。

【0070】

上述したように、非クリティカル部分の前にトランザクションのクリティカル部分を処理することには利点があり、一方のトランザクションのクリティカル部分が処理され、次に第1のトランザクションの非クリティカル部分が処理される前に他方のトランザクションのクリティカル部分が処理されるようにトランザクションを分裂させることは、全体的に見てシステムにとって利点がある。

【0071】

この利点を最大限にするために、クリティカル部分の長さを可能な限り正確に識別できる必要があり、これは種々の方法で実行される。

10

【0072】

図10に示された第1の手段において、クリティカル部分の長さは、ホスト又は関連付けられたキャッシュにおいて統計的に判定される。ステップ280において示されたように、プリエンブションポイント、すなわちクリティカル部分の直後のポイントが判定される。次に、ステップ282において、トランザクション要求はプリエンブション値でタグ付けされる。

【0073】

従って、図11に例として示されたように、データバーストの前に送出されたトランザクション要求284は、一般に従来通りであるため更に詳細に説明しないが、それがデータ読み出し動作を要求しているかあるいはデータ書き込み動作を信号伝送しているかを示すフラグ286と、バーストのサイズ/種類に関する表示288と、アドレス値290とを含む。更にトランザクション要求284は、プリエンブション値タグ292を含む。このタグ値はバーストのクリティカル部分の長さを直接示し、これは、バスファブリックにおけるバスノード又はメモリリソース自体において調停装置により読み出され、且つデータバーストを分裂すべきポイントを判定するプリエンブション機構により使用される。トランザクション要求がバーストの長さの表示を既に含む場合、タグ値は、絶対項で又はバースト長の端数としてクリティカル部分の長さを示す。

20

【0074】

タグ値は、いくつかの方法のうちの1つでホスト又はキャッシュにより導出される。例えばタグ値は、コンパイル時に事前に判定され且つコードに追加された固定値であり、次にアドレス領域内でトランザクションのグループに対するタグを設定するために使用される。別の例として、タグ値は、動作を最適化するためにソフトウェアの使用率を測定し且つタグを設定するメモリ管理ユニット(MMU)又はスヌープロジック等のモニタ処理により判定される。L1キャッシュ上のスヌープロジックを含むマルチコアプロセッサにおいて、ロジックは、事前にL1キャッシュ間に出力されたかあるいは移動したラインに対するプリエンブション値を能動的に維持するために使用される。別の例として、タグ値は、クリティカルワードの最初の機構のラッピングポイントから導出される。すなわち、ラッピングポイントの前の全てのトランザクションは、クリティカル部分であると考えられる。

30

【0075】

ホストがこのようにトランザクションのクリティカル部分の長さの表示でデータ要求をタグ付けする場合、リソースはこのタグを使用する。第1に、リソースは、このデータトランザクションと他のデータトランザクションとの間のあらゆる競合を解決するためにタグ値を使用し、上述したようにトランザクションの非クリティカル部分の前にトランザクションのクリティカル部分を処理する。第2に、リソースは、このタグ又はその変形バージョンをホストに返送されるデータを含む。リソースにおいて追加されたタグは、バスファブリックの調停ポイントにおいて読み出され、トランザクションにおいて分裂ポイントを判定する。ここでも、調停ポイントは、図6を参照して説明した方法で又は別の方法でこの分裂ポイントを使用する。

40

【0076】

50

クリティカル部分のサイズのある推定値に基づいて値が予め適用されるため、この方法は静的である。

【 0 0 7 7 】

図 1 2 に示された第 2 の手段において、クリティカル部分の長さは、データが検索されているメモリ素子と関連付けられたメモリコントローラ等のリソースにおいて静的に判定され、ホストプロセッサ又は CPU に返送された検索されたデータは、プリエンブション値でタグ付けされる。ステップ 3 0 0 において示されたように、プリエンブションポイント、すなわちクリティカル部分の直後のポイントが判定される。次に、ステップ 3 0 2 において、返送されたデータはプリエンブション値でタグ付けされる。

【 0 0 7 8 】

プリエンブション値は、そのメモリ領域に対してホスト CPU により設定された予めプログラムされた静的な値であるか、あるいは高度なメモリコントローラによるモニタリング及び最適化に基づく。メモリコントローラによる更なる手段は、低い優先順位及び高い優先順位で単一の要求からのバーストを 2 つのより小さなバーストに分裂させることである。

【 0 0 7 9 】

リソースにおいて追加されたタグは、バスファブリックの調停ポイントにおいて読み出され、トランザクションにおいて分裂ポイントを判定する。既存のメモリコントローラ及びバスファブリックは、この方法を使用するように構成される。

【 0 0 8 0 】

図 1 3 に示された第 3 の手段において、クリティカル部分の長さは、既存のバスファブリック内の調停のポイントにおいて分散されて判定される。

【 0 0 8 1 】

ステップ 3 1 0 において、調停ポイントは、トランザクションにおいてプリエンブションポイントを判定し、ステップ 3 1 2 において、調停ポイントは、プリエンブションポイントに基づいて調停を実行する。この調停は、例えば図 6 に示された方法に従って実行される。

【 0 0 8 2 】

例えば、アービタにおける各バス入力は、優先順位の値が調停に対して割り当てられるのと同じ方法で、ホストによりプログラムされた所定のプリエンブション値を割り当てられてもよい。ホストは、トラフィックを最適化するようにアプリケーションにより要求されたようにこの値を設定してもよく、値は、バスファブリックにおいてそのパス上で全てのバストランザクションに対して固定される。

【 0 0 8 3 】

リソースが競合するシステムの 1 つの特定のポイントにおける競合解決システムの動作を例としてそのように説明した。

【 0 0 8 4 】

しかし、図 1 に示されたような組込みシステムにおいて競合解決システムが多くのポイントで適用されることは、理解されるだろう。特に、クリティカル部分の長さを示すタグがトランザクションのソース又はシステムの 1 つの競合ポイントにおいて適用される場合、このタグは、トランザクションがポイントからポイントに渡されるのに伴い含まれる。しかし、トランザクションを受信するあらゆるコンポーネント又はトランザクションを受信するあらゆる競合ポイントにおけるあらゆる調停装置は、適当にタグを変更する。

【 0 0 8 5 】

トランザクションがクリティカル部分と非クリティカル部分とに分裂する例を参照して競合解決システムをこれまで説明したが、トランザクションを 3 つ以上の部分に更に分裂させられる。例えば、上述の基準に基づいて、トランザクションは、クリティカル部分と、より小さなクリティカル部分と、非クリティカル部分とに分裂し、これらの 3 つの部分は適切に処理される。

10

20

30

40

【図1】

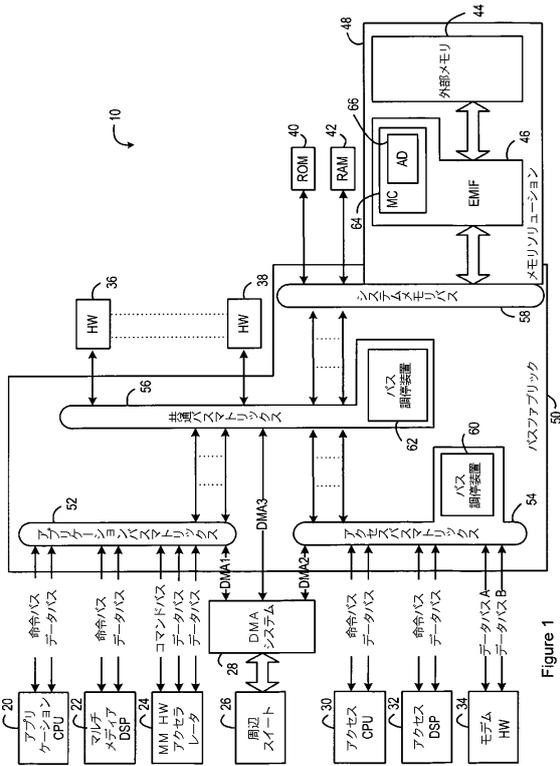


Figure 1

【図2】

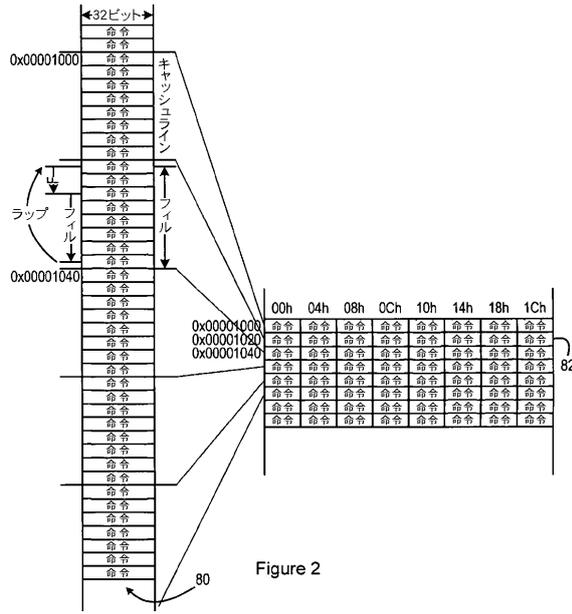


Figure 2

【図3】

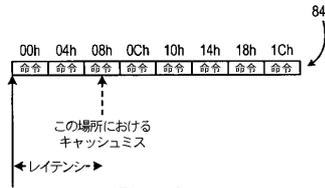


Figure 3

【図4】

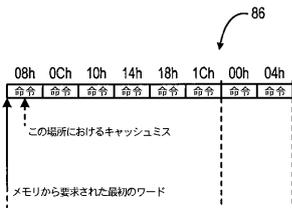


Figure 4

【図5】

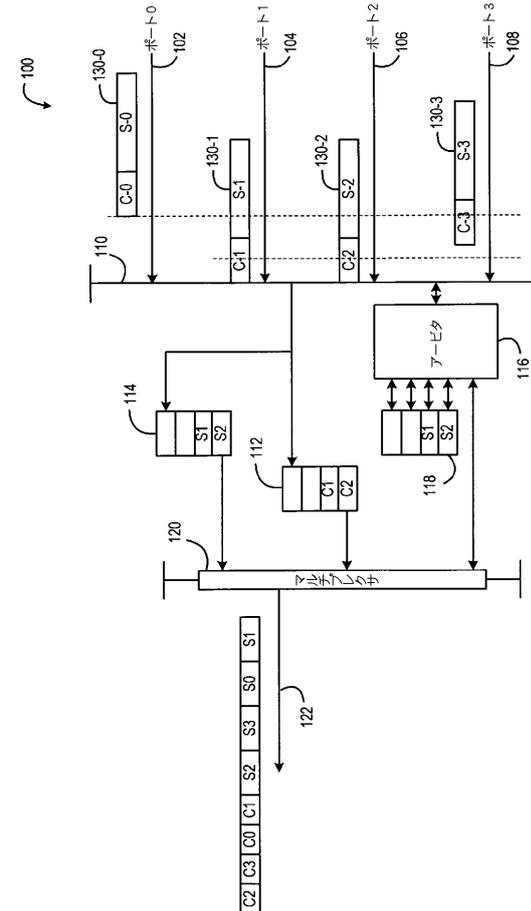


Figure 5

【図 6】

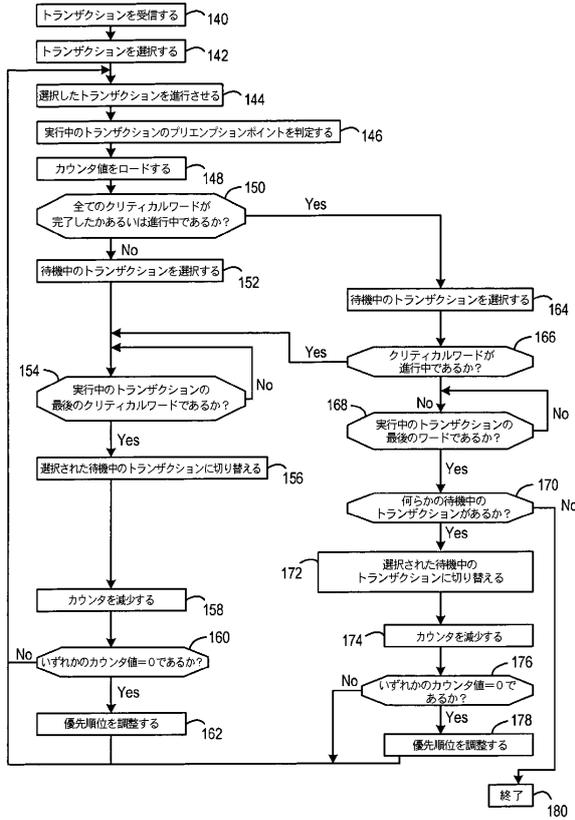


Figure 6

【図 7】

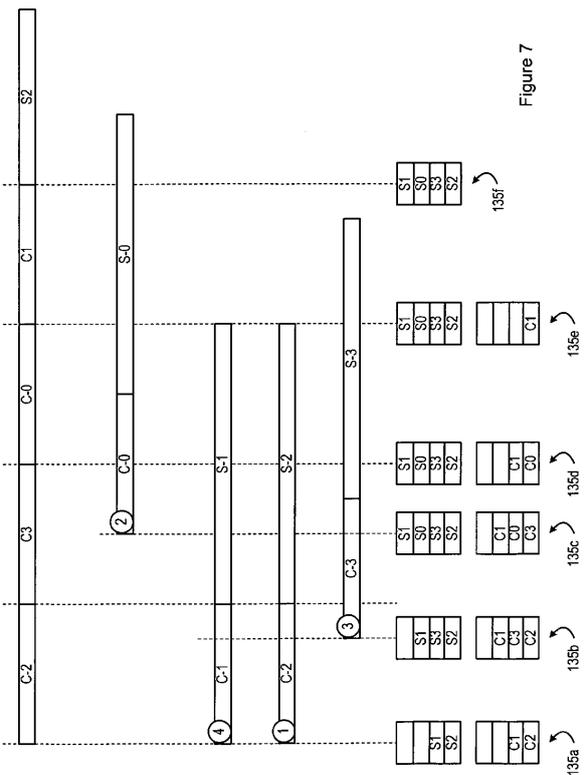


Figure 7

【図 8】

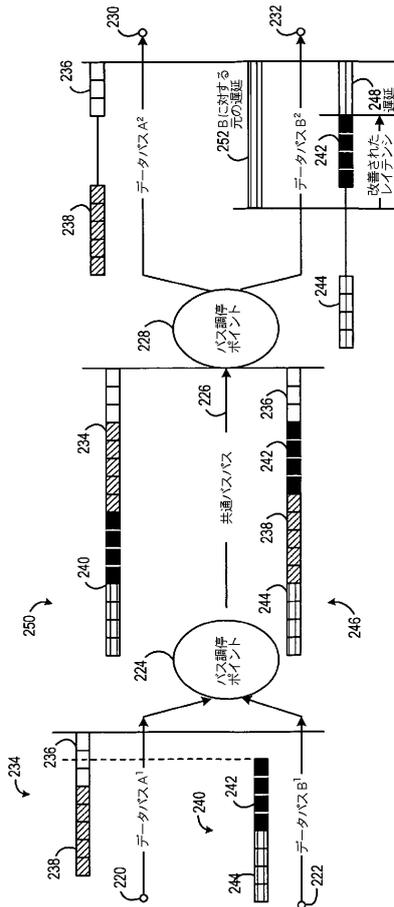


Figure 8

【図 9】

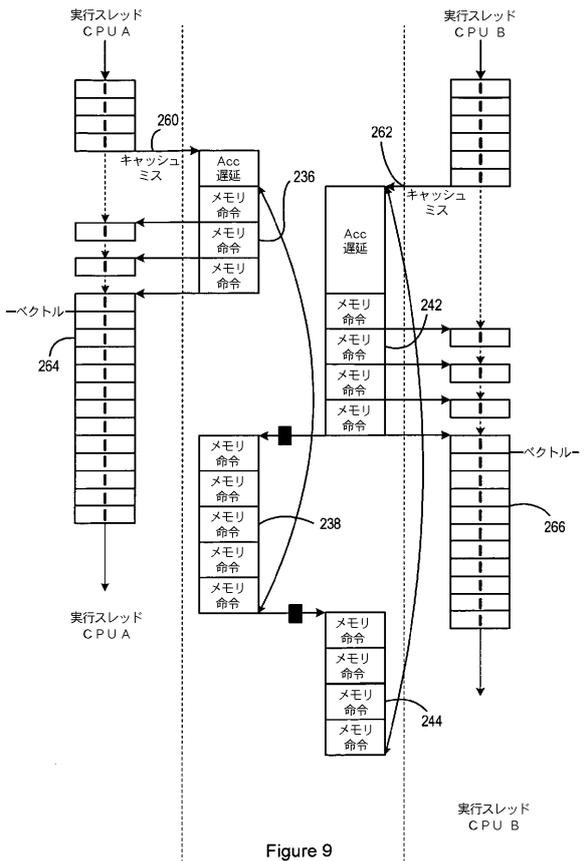


Figure 9

【図10】

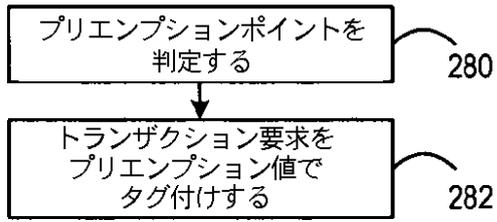


Figure 10

【図11】

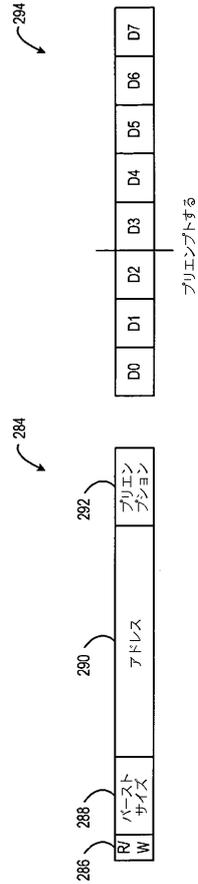


Figure 11

【図12】

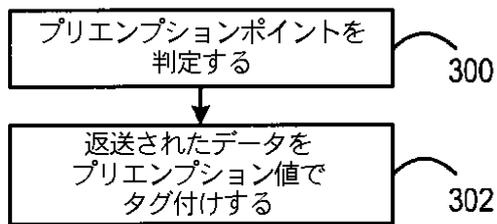


Figure 12

【図13】

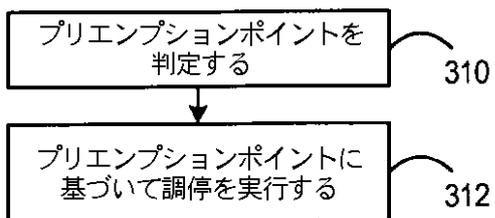


Figure 13

フロントページの続き

(72)発明者 ネイラー, ローワン ナイジェル
イギリス国 ベーキングストーク ハンプシャー アールイー 2 1 8 ティーユー, ダウンスラ
ンド ロード, リサ コート 4

審査官 木村 貴俊

(56)参考文献 特開平07 - 006122 (JP, A)
特開2000 - 285087 (JP, A)
特開2002 - 222162 (JP, A)

(58)調査した分野(Int.Cl., DB名)
G06F 13/20 - 13/42