

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第4742116号  
(P4742116)

(45) 発行日 平成23年8月10日(2011.8.10)

(24) 登録日 平成23年5月13日(2011.5.13)

(51) Int.Cl. F I  
**G06F 12/00 (2006.01)** G O 6 F 12/00 5 7 1 A  
**G06F 12/06 (2006.01)** G O 6 F 12/06 5 5 0 A

請求項の数 54 (全 21 頁)

(21) 出願番号	特願2008-140086 (P2008-140086)	(73) 特許権者	595168543 マイクロン テクノロジー, インク. アメリカ合衆国, アイダホ州 83716 -9632, ボイズ, サウス フェデ ラル ウェイ 8000
(22) 出願日	平成20年5月28日(2008.5.28)	(74) 代理人	100070150 弁理士 伊東 忠彦
(62) 分割の表示	特願2004-505842 (P2004-505842) の分割	(74) 代理人	100091214 弁理士 大貫 進介
原出願日	平成15年5月14日(2003.5.14)	(74) 代理人	100107766 弁理士 伊東 忠重
(65) 公開番号	特開2008-204487 (P2008-204487A)	(72) 発明者	ジェデロー, ジョーゼフ エム アメリカ合衆国 ミネソタ州 55126 ショアビュー レイランド・レイン 4 302
(43) 公開日	平成20年9月4日(2008.9.4)		
審査請求日	平成20年5月28日(2008.5.28)		
(31) 優先権主張番号	10/143,896		
(32) 優先日	平成14年5月14日(2002.5.14)		
(33) 優先権主張国	米国 (US)		

最終頁に続く

(54) 【発明の名称】 アウトオブオーダーDRAMシーケンサ

(57) 【特許請求の範囲】

【請求項1】

複数のメモリアクセスリクエストを本来の受信順序で少なくとも1つの要求元から受け取る入力キューと、

前記複数のメモリアクセスリクエストが構成可能であり、逐次的順序で前記複数のメモリアクセスリクエストを格納するシーケンスマトリックスと、

前記シーケンスマトリックスにおける複数のメモリアクセスリクエストのタイミングシーケンスを構成するシーケンシングユニットと、

前記シーケンシングユニットにより前記シーケンスマトリックスにおいて構成されたメモリアクセスリクエストのシーケンスにおいてコンフリクトまたは遅延が発生するか検出するコンフリクト検出器であって、前記シーケンスマトリックスの前記メモリアクセスリクエストの1つと、前記シーケンスマトリックスに構成される前記メモリアクセスリクエストのシーケンスの第2のメモリアクセスリクエストとの又は前記シーケンスマトリックスに以前に構成された複数の実行されていないメモリアクセスリクエストの1つとの比較の結果としてコンフリクト又は遅延が、検出されたコンフリクト又は遅延に係るメモリアクセスリクエストの前方への移動によって検出されると、前記シーケンシングユニットが前記シーケンスマトリックス内の前記リクエストの順序を再構成するように、前記シーケンシングユニットに検出されたコンフリクト又は遅延を通知するコンフリクト検出器と、

各コマンドの実行の間のアイドル時間を回避するような実行順序によりコマンドシーケンサによって構成される複数のコマンドが、各メモリアクセスリクエストがリード若しく

はライトコマンドの何れか及びデータコントロールコマンドを有する前記再構成されたメモリアクセスリクエストから前記コマンドシーケンサによって抽出されるよう構成される実行キューと、

前記実行キューからの前記メモリアクセスリクエストの前記実行順序による実行後、前記メモリアクセスリクエストの本来の受信順序を追跡し、前記メモリアクセスリクエストの本来の受信順序でリクエストデータを前記少なくとも1つの要求元に返すリードリターンキューと、

メモリから読み出されたリクエストデータを、該データが前記係るアクセスリクエストの要求元に返されるまで、保持するリターンデータバッファと、

を有するメモリコントローラであって、

【請求項2】

請求項1記載のメモリコントローラであって、さらに、

メモリアクセスリクエストを少なくともリードまたはライトコマンド及びデータコントロールコマンドにパース処理するコマンドパーサと、

前記実行キューの各アクセスリクエストに対して、前記データコントロールコマンドと前記リードまたはライトコマンドを構成するコマンドシーケンサと、

を有するメモリコントローラ。

【請求項3】

請求項2記載のメモリコントローラであって、さらに、

実行のため、前記実行キューから一度に少なくとも1つのコマンドを選択するコマンドセクタを有するメモリコントローラ。

【請求項4】

請求項3記載のメモリコントローラであって、

固定された時間インターバル中に選択されるコマンド数は、前記コマンドセクタに与えられるコマンド選択クロック信号に依存するメモリコントローラ。

【請求項5】

請求項1記載のメモリコントローラであって、

前記コンフリクト検出器は、前記複数の実行されていないメモリアクセスリクエスト若しくは前記シーケンスマトリックスにおける前記メモリアクセスリクエストから生じる潜在的なバンクビジー状態を検出することが可能であるメモリコントローラ。

【請求項6】

請求項1記載のメモリコントローラであって、

前記コンフリクト検出器は、前記複数の実行されていないメモリアクセスリクエスト若しくは前記シーケンスマトリックスにおける前記メモリアクセスリクエストから生じる潜在的なページコンフリクト状態を検出することが可能であるメモリコントローラ。

【請求項7】

請求項1記載のメモリコントローラであって、

前記コンフリクト検出器は、前記複数の実行されていないメモリアクセスリクエスト若しくは前記シーケンスマトリックスにおける前記メモリアクセスリクエストから生じる潜在的なページミス状態を検出することが可能であるメモリコントローラ。

【請求項8】

複数の要求元からリードリクエスト及び/またはライトリクエストからなる複数のメモリアクセスリクエスト信号を受け取り、該リクエストが前記要求元から受信された順序に対応するシーケンスで前記複数のメモリアクセスリクエスト信号を受け取るシーケンシングユニットと、

前記シーケンスにおいて時間的に近接した実行されていないリクエスト間の検出されたメモリアクセスのコンフリクトに基づき、前記複数のメモリアクセスリクエスト信号のシーケンスを再構成し、これにより前記メモリアクセスリクエストの実行が、検出されたコ

10

20

30

40

50

ンフリクトに係るメモリアクセスリクエストを前記シーケンスにおいて前方に移動することによって、前記再構成されたシーケンスにおいて実行されることを可能にする再構成ユニットと、

各コマンドの実行の間のアイドル時間を回避するような実行順序により構成される複数のコマンドを、各メモリアクセスリクエストがリード若しくはライトコマンド及びデータコントロールコマンドからなる前記再構成されたメモリアクセスリクエストから抽出し、前記複数のコマンドを実行する実行ユニットと、

実行されたリードリクエストを該リクエストの本来の受信順序に再順序付けし、これにより前記実行されたリードリクエストにより取得されたデータが前記再順序付けされたシーケンスにより前記要求元に返されることを可能にする再順序付けユニットと、

10

を有するメモリコントローラであって、

当該コントローラは、複数のメモリクロックスピードを有するメモリとインタフェースをとるよう構成されるメモリコントローラ。

【請求項 9】

請求項 8 記載のメモリコントローラであって、

前記再構成ユニットは、前記シーケンスにおける時間的に近接した実行されていないリクエスト間の検出されたバンクビジー状態を回避するため、前記メモリアクセスリクエストのシーケンスを再構成するメモリコントローラ。

【請求項 10】

請求項 8 記載のメモリコントローラであって、

前記再構成ユニットは、前記シーケンスにおける時間的に近接した実行されていないリクエスト間の検出されたバンクビジー状態中にメモリバンクがアクセス可能となるのを待機することにより生じる遅延を最小化するため、前記メモリアクセスリクエストのシーケンスを再構成するメモリコントローラ。

20

【請求項 11】

請求項 8 記載のメモリコントローラであって、

前記再構成ユニットは、前記シーケンスにおける時間的に近接した実行されていないリクエスト間の検出されたページコンフリクト状態を回避するため、前記メモリアクセスリクエストのシーケンスを再構成するメモリコントローラ。

【請求項 12】

請求項 8 記載のメモリコントローラであって、

前記再構成ユニットは、前記シーケンスにおける時間的に近接した実行されていないリクエスト間の検出されたページコンフリクト状態中にメモリバンクがアクセス準備されるのを待機することにより生じる遅延を最小化するため、前記メモリアクセスリクエストのシーケンスを再構成するメモリコントローラ。

30

【請求項 13】

請求項 8 記載のメモリコントローラであって、

前記再構成ユニットは、前記シーケンスにおける時間的に近接した実行されていないリクエスト間の検出されたページミス状態中にメモリバンクがアクセス準備されるのを待機することにより生じる遅延を最小化するため、前記メモリアクセスリクエストのシーケンスを再構成するメモリコントローラ。

40

【請求項 14】

複数のメモリアクセスリクエストを到着順に格納する入力回路と、

検出されたコンフリクトに係るメモリアクセスリクエストを前方に移動させることによって、前記格納されたメモリアクセスリクエストをメモリアクセスリクエストの実行順に再構成するメモリアクセスリクエスト再構成回路であって、前記格納されているメモリアクセスリクエストの順序を、前記格納されているメモリアクセスリクエストの順序において存在するメモリアクセスコンフリクトであって、各メモリアクセスリクエストがリード若しくはライトコマンド及びデータコントロールコマンドからなる前記複数のメモリアクセスリクエストの 1 つと第 2 の実行されていない格納されているメモリアクセスリクエ

50

トとの比較の結果として検出されるメモリアクセスコンフリクトの検出に基づき、前記実行順に再構成するメモリアクセスリクエスト再構成回路と、

各コマンドの実行の間のアイドル時間を回避するため、前記再構成されたメモリアクセスリクエストからのコマンドをコマンド実行順に構成するコマンドシーケンサと、

前記実行順によりメモリアクセスリクエストを出力する送信回路と、  
を有するメモリコントローラであって、

前記再構成回路は、第1クロックスピードにより動作し、

前記再構成回路により構成されるコマンドは、複数の異なるメモリクロックスピードの1つに従って実行される、メモリコントローラ。

【請求項15】

請求項14記載のメモリコントローラであって、

前記再構成回路は、前記実行されていない格納されているメモリアクセスリクエストから生じるバンクビジーコンフリクトの検出に基づき、前記格納されたメモリアクセスリクエストを再構成するメモリコントローラ。

【請求項16】

請求項15記載のメモリコントローラであって、

前記再構成回路は、前記検出されたバンクビジーコンフリクトを回避するよう前記格納されたメモリアクセスリクエストを再構成するメモリコントローラ。

【請求項17】

請求項15記載のメモリコントローラであって、

前記再構成回路は、前記メモリアクセスリクエストが該メモリアクセスリクエストの格納順に実行される場合、前記検出されたバンクビジーコンフリクト状態中にメモリバンクがアクセス可能になることを待機することにより生じる遅延を最小化するため、前記格納されたメモリアクセスリクエストを再構成するメモリコントローラ。

【請求項18】

請求項14記載のメモリコントローラであって、

前記再構成回路は、前記格納されたメモリアクセスリクエストの順序による前記実行されていない格納されているメモリアクセスリクエストから生じるページコンフリクト状態の検出に基づき、前記格納されたメモリアクセスリクエストを再構成するメモリコントローラ。

【請求項19】

請求項18記載のメモリコントローラであって、

前記再構成回路は、前記検出されたページコンフリクトを回避するよう前記格納されたメモリアクセスリクエストを再構成するメモリコントローラ。

【請求項20】

請求項18記載のメモリコントローラであって、

前記再構成回路は、前記格納されたメモリアクセスリクエストの順序により検出されたページコンフリクト状態の間に、メモリバンクがアクセス準備されるのを待機することにより生じる遅延を最小化するよう前記格納されたメモリアクセスリクエストを再構成するメモリコントローラ。

【請求項21】

請求項14記載のメモリコントローラであって、

前記再構成回路は、前記格納されたメモリアクセスリクエストの順序による前記実行されていない格納されているメモリアクセスリクエストから生じるページミス状態の検出に基づき、前記格納されたメモリアクセスリクエストを再構成するメモリコントローラ。

【請求項22】

請求項21記載のメモリコントローラであって、

前記再構成回路は、前記格納されたメモリアクセスリクエストの順序により検出されたページミス状態の間に、メモリバンクがアクセス準備されるのを待機することにより生じる遅延を最小化するよう前記格納されたメモリアクセスリクエストを再構成するメモリコ

10

20

30

40

50

ントローラ。

【請求項 2 3】

請求項 1 4 記載のメモリコントローラであって、さらに、  
前記格納されたメモリアクセスリクエストを前記実行順に実行する実行回路と、  
実行後、前記格納されたメモリアクセスリクエストの到着順を追跡する追跡回路と、  
を有するメモリコントローラ。

【請求項 2 4】

請求項 2 3 記載のメモリコントローラであって、  
前記追跡回路は、各メモリアクセスリクエストを前記到着順における該リクエストの位置に対応してタグを関連付けするメモリコントローラ。

10

【請求項 2 5】

請求項 2 3 記載のメモリコントローラであって、  
前記追跡回路は、前記メモリアクセスリクエストの実行により各自のメモリ位置から読み出されたデータを構成するバッファを有し、これにより、前記バッファに構成されたデータを前記本来の到着順により前記メモリコントローラから転送することを可能にするメモリコントローラ。

【請求項 2 6】

メモリアクセス処理の制御方法であって、  
受信シーケンスにおける複数のメモリアクセスリクエストを受信するステップと、  
前記複数のメモリアクセスリクエストの 1 つと第 2 の実行されていないメモリアクセス  
リクエストとを比較することによって、前記シーケンスにおける時間的に近接した実行され  
ていないリクエスト間のメモリアクセスコンフリクトを検出するステップと、

20

検出されたコンフリクトに係るメモリアクセスリクエストをシーケンスにおいて前方に  
移動させることによって、前記検出結果に基づき前記リクエストシーケンスを再構成され  
たシーケンスに再構成するステップと、

前記メモリアクセスリクエストを前記再構成された順序により実行するステップと、  
各メモリアクセスリクエストを少なくともリード若しくはライトコマンド及びデータコ  
ントロールコマンドにパース処理するステップと、

コマンドセレクトクロックスピードによる実行のため前記再構成されたシーケンスから  
一度に少なくとも 1 つのコマンドを選択するステップと、

30

を有し、

前記リクエストのシーケンスを再構成するステップはさらに、各コマンドの実行の間の  
アイドル時間を回避する実行順によりコマンドが実行されるように、前記再構成されたシ  
ーケンスの各リクエストに対して前記リード若しくはライトコマンド及び前記データコ  
ントロールコマンドを別々に構成し、

前記メモリアクセスコンフリクトを検出するステップはさらに、実行されていないメモ  
リアクセスリクエストから生じるコンフリクトを検出する方法。

【請求項 2 7】

請求項 2 6 記載の制御方法であって、

前記複数のメモリアクセスリクエストは、指定されたメモリアドレスにデータを書き込  
むためのライトリクエストと、指定されたメモリアドレスからデータを読み出すためのリ  
ードリクエストとを含む方法。

40

【請求項 2 8】

請求項 2 7 記載の制御方法であって、さらに、

前記リードリクエストの実行により前記指定されたメモリアドレスから取得されたデー  
タを、前記受信シーケンスにおける本来の順序によりメモリコントローラから出力するス  
テップを有する方法。

【請求項 2 9】

請求項 2 7 記載の制御方法であって、さらに、

実行後、前記メモリアクセスリクエストの受信シーケンスを追跡するステップと、

50

前記再構成されたシーケンスでの前記リードリクエストの実行により取得されるリクエストデータを前記リードリクエストの受信シーケンスに対応する順序により各自の要求元に転送するステップと、  
を有する方法。

【請求項 30】

請求項 29 記載の制御方法であって、

前記受信シーケンスを追跡するステップは、前記リクエストのシーケンスの再構成後、前記リードリクエストが本来受信されたシーケンスを示すよう各リードリクエストにタグを関連付ける方法。

【請求項 31】

請求項 29 記載の制御方法であって、

前記受信シーケンスを追跡するステップは、前記リクエストのシーケンスの再構成後、各リードリクエストの実行により取得されたデータが格納されるバッファ位置を割り当て、  
該割り当てられたバッファ位置は、前記リードリクエストが本来受信されたシーケンスを示す方法。

【請求項 32】

請求項 26 記載の制御方法であって、

前記検出するステップは、前記受信シーケンスの現在位置において最も最近受信されたメモリアクセスリクエストを実行することにより、コンフリクトが生じるか検出するよう実行される方法。

【請求項 33】

請求項 32 記載の制御方法であって、

前記検出するステップは、前記受信シーケンスの現在位置において最も最近受信されたメモリアクセスリクエストを実行することにより、バンクビジーコンフリクトが生じるか検出するよう実行される方法。

【請求項 34】

請求項 33 記載の制御方法であって、

前記再構成するステップは、前記バンクビジーコンフリクトを回避するよう実行される方法。

【請求項 35】

請求項 33 記載の制御方法であって、

前記再構成するステップは、検出されたバンクビジーコンフリクトの間にメモリバンクがアクセス可能になるのを待機することにより生じる遅延を最小化するよう実行される方法。

【請求項 36】

請求項 32 記載の制御方法であって、

前記検出するステップは、前記受信シーケンスの現在位置において最も最近受信されたメモリアクセスリクエストを実行することにより、ページコンフリクトが生じるか検出するよう実行される方法。

【請求項 37】

請求項 36 記載の制御方法であって、

前記再構成するステップは、前記ページコンフリクトを回避するよう実行される方法。

【請求項 38】

請求項 36 記載の制御方法であって、

前記再構成するステップは、検出されたページコンフリクトの間にメモリバンクがアクセス準備されるのを待機することにより生じる遅延を最小化するよう実行される方法。

【請求項 39】

請求項 26 記載の制御方法であって、

前記検出するステップは、前記受信シーケンスの現在位置において最も最近受信された

10

20

30

40

50

メモリアクセスリクエストを実行することにより、遅延が生じるか検出するよう実行される方法。

【請求項 4 0】

請求項 3 9 記載の制御方法であって、

前記検出するステップは、前記受信シーケンスの現在位置において最も最近受信されたメモリアクセスリクエストを実行することにより、ページミスが生じるか検出するよう実行される方法。

【請求項 4 1】

請求項 4 0 記載の制御方法であって、

前記再構成するステップは、検出されたページミスの中にメモリバンクがアクセス準備されるのを待機することにより生じる遅延を最小化するよう実行される方法。

10

【請求項 4 2】

請求項 2 6 記載の制御方法であって、

前記検出するステップは、前記受信シーケンスの現在位置において最初に受信されたメモリアクセスリクエストを実行することにより、コンフリクトが生じるか検出するよう実行される方法。

【請求項 4 3】

請求項 4 2 記載の制御方法であって、

前記検出するステップは、前記受信シーケンスの現在位置において最初に受信されたメモリアクセスリクエストを実行することにより、バンクビジーコンフリクトが生じるか検出するよう実行される方法。

20

【請求項 4 4】

請求項 4 3 記載の制御方法であって、

前記再構成するステップは、前記バンクビジーコンフリクトを回避するよう実行される方法。

【請求項 4 5】

請求項 4 3 記載の制御方法であって、

前記再構成するステップは、検出されたバンクビジーコンフリクトの中にメモリバンクがアクセス準備されるのを待機することにより生じる遅延を最小化するよう実行される方法。

30

【請求項 4 6】

請求項 4 2 記載の制御方法であって、

前記検出するステップは、前記受信シーケンスの現在位置において最初に受信されたメモリアクセスリクエストを実行することにより、ページコンフリクトが生じるか検出するよう実行される方法。

【請求項 4 7】

請求項 4 6 記載の制御方法であって、

前記再構成するステップは、前記ページコンフリクトを回避するよう実行される方法。

【請求項 4 8】

請求項 4 6 記載の制御方法であって、

前記再構成するステップは、検出されたページコンフリクトの中にメモリバンクがアクセス準備されるのを待機することにより生じる遅延を最小化するよう実行される方法。

40

【請求項 4 9】

請求項 4 2 記載の制御方法であって、

前記受信シーケンスの最初に受信されたメモリアクセスリクエストに関してコンフリクトが検出された場合、前記再構成するステップは、前記最初に受信されたメモリアクセスリクエストを前記シーケンスにおける最も最近に受信されたメモリアクセスリクエストに続く位置に一般に対応する前記シーケンスの終端に前記最初に受信されたメモリアクセスリクエストを移動する方法。

【請求項 5 0】

50

請求項 2 6 記載の制御方法であって、

前記検出するステップは、前記受信シーケンスの現在位置において最先に受信されたメモリアクセスリクエストを実行することにより、遅延が生じるか検出するよう実行される方法。

【請求項 5 1】

請求項 5 0 記載の制御方法であって、

前記検出するステップは、前記受信シーケンスの現在位置において最先に受信されたメモリアクセスリクエストを実行することにより、ページミスが生じるか検出するよう実行される方法。

【請求項 5 2】

請求項 5 1 記載の制御方法であって、

前記再構成するステップは、検出されたページミスの間にメモリバンクがアクセス準備されるのを待機することにより生じる遅延を最小化するよう実行される方法。

【請求項 5 3】

請求項 5 0 記載の制御方法であって、

前記受信シーケンスの最先に受信されたメモリアクセスリクエストに関してコンフリクトが検出された場合、前記再構成するステップは、前記最先に受信されたメモリアクセスリクエストを前記シーケンスにおける最も最近に受信されたメモリアクセスリクエストに続く位置に一般に対応する前記シーケンスの終端に前記最先に受信されたメモリアクセスリクエストを移動する方法。

【請求項 5 4】

プロセッサと、

前記プロセッサとデータをやりとりする少なくとも 1 つの記憶装置と、

前記プロセッサから前記少なくとも 1 つの記憶装置へのメモリアクセスリクエストを管理するメモリコントローラと、

を有するプロセッサシステムであって、

前記メモリコントローラは、

複数の要求元からリードリクエスト及び/またはライトリクエスト及びデータコントロールコマンドからなる複数のメモリアクセスリクエスト信号を受け取り、該リクエストが前記要求元から受信された順序に対応するシーケンスで前記複数のメモリアクセスリクエスト信号を受け取るシーケンシングユニットと、

メモリアクセスリクエスト信号と実行されていないメモリアクセスリクエスト信号とを再構成ユニットが比較することによって前記シーケンスにおいて時間的に近接した実行されていないリクエスト間において検出されたメモリアクセスのコンフリクトに基づき、検出されるコンフリクトを軽減するため前記シーケンスにおいてメモリアクセスリクエスト信号を前方に移動することによって、前記複数のメモリアクセスリクエスト信号のシーケンスを再構成し、これにより前記メモリアクセスリクエストの実行が前記再構成されたシーケンスにより実行されることを可能にする再構成ユニットと、

前記再構成されたシーケンスにより前記メモリアクセスリクエストを実行し、各メモリアクセスリクエストからのコマンドを、各コマンドの実行の間のアイドル時間を回避するよう構成する実行ユニットと、

実行されたリードリクエストを該リクエストの本来の受信順序に再順序付けし、これにより前記実行されたリードリクエストにより取得されたデータが前記再順序付けされたシーケンスにより前記要求元に返されることを可能にする再順序付けユニットと、

を有し、

前記コントローラは、複数のメモリクロックスピードの 1 つを有するメモリとインタフェースをとるよう構成されるシステム。

【発明の詳細な説明】

【技術分野】

【0001】

10

20

30

40

50

本発明は、有効なメモリ帯域幅を拡張させるためのメモリアクセス処理を制御するためのメモリコントローラの構成及び動作方法に関する。

【背景技術】

【0002】

大部分のコンピュータまたはデータ処理システムでは、メインとなるアクティブメモリ、すなわちランダムアクセスメモリ(RAM)は、ダイナミックランダムアクセスメモリ(DRAM)である。DRAMの構成は、一般に、複数のバンクの形式に構成された多数のメモリから構成される。各バンクは、各自が一意的なメモリアドレスに関連付けられたセルを有するメモリセルのレイに対応している。特に、バンク内の各メモリアドレスは、行アドレスと列アドレスにより指定され、各行アドレスはメモリページとして定義される。このため、各メモリページは、当該ページ内の異なる列指定に対応する複数のメモリ位置を含む。

10

【0003】

一連のアクセスリクエストを実行するとき、現在リクエストされているページが現在開かれている他のページを有するバンクと同じバンクにおいて検出される場合、このような状態は「ページコンフリクト(page conflict)」として知られ、この場合には、前に開かれていた当該ページがまず閉じられる、すなわち「プリチャージされる(precharged)」必要がある。プリチャージ後、リクエストされたページは開かれ、すなわち「アクティブ化(activated)」され、リード(read)またはライト(write)処理が実行される。現在リクエストされているページが開かれたページを持たないバンクにおいて検出される場合には、「ページミス(page miss)」が発生し、アクティブ化処理の実行が必要となる。現在のメモリアクセスリクエストが前のメモリアクセスリクエストからすでに開かれているページに対するものであるとき、「ページヒット(page hit)」が発生すると言われる。

20

【0004】

ページヒットリクエストとは相対的にページコンフリクトやページミスのメモリアクセスに対して実行されねばならない追加的処理のため、これら2つの処理を実行するのに要する時間は、ページヒットリクエストに対するものよりかなり大きなものとなる。マイクロプロセッサの技術進歩の初期段階では、リード及びライト処理のためのDRAMメモリページへのアクセスリクエストは、先入先出(first in first out)原則に基づき受け取り及び実行がなされていた。このような処理はかなり非効率なものとなる傾向があり、多数のページミスとコンフリクトを発生させ、プロセッサ及び/またはメモリコントローラのリソースをメモリページのプリチャージ及びアクティブ化に使用することが必要となる。

30

【0005】

最近では、メモリアクセスが優先度に基づくより先進的な処理方法が開発されてきた。このアクセスリクエストの優先度は、リクエストを送信する装置のタイプ、リクエストされるアクセスのタイプ、リクエストによるアクセスが所望されるメモリアドレスなどの様々な要因に基づくものであってもよい。しかしながら、優先度により厳密にメモリアクセスを提供する問題は、低い優先度のリクエストが許容できないくらい長期間にわたって拒絶アクセスとなってしまう可能性があるということである。

40

【0006】

さらに、新たな世代のコンピュータが進化するに従い、メモリクロック速度は大きく増大する。メモリクロック速度が増大するに従い、ページミスのメモリ処理の潜在的な発生及び時間的ペナルティ、バンクのビジーコンフリクト及び他のコンフリクトがますます深刻になる。特に、アクセスされた各メモリ位置との情報の転送に利用されるデータバスは、プリチャージ、アクティブ化、バンクが利用可能となるための待機などの間はアイドル状態となる。

【発明の開示】

【発明が解決しようとする課題】

50

## 【 0 0 0 7 】

従って、上記問題点を軽減する解法が必要とされる。特に、受け取ったメモリアクセスリクエストシーケンスの順序がコンフリクトの回避または軽減のため再構成可能である場合、メモリ処理効率は大きく向上するであろう。コンフリクトの回避または軽減を行うことにより、メモリデータバスのアイドルタイムが軽減または解消されるという点で、メモリデータバスはより効率的に利用され、これにより、メモリシステムのメモリ帯域幅を効果的に増大させ、従来可能とされたものより、より多くのメモリアクセストランザクションがより短い時間で発生することを可能にする。

## 【 0 0 0 8 】

本発明は、上述のようなコンフリクト及び遅延を最小化するため、逐次的に受け取った D R A M へのアクセスリクエストを再構成する一方、これらのリクエストが本来受け取られたものと同じ順序でリクエストされた情報を要求元ユニットに返すことにより、従来技術において特定された問題を解決しようとするものである。

10

## 【 0 0 0 9 】

本発明の課題は、有効なメモリ帯域幅を拡張させるためのメモリアクセス処理を制御するためのメモリコントローラの構成及び動作方法を提供することである。

## 【課題を解決するための手段】

## 【 0 0 1 0 】

本発明では、メモリアクセスリクエストは、メモリコントローラの入力キューに連続的に受け取られる。リクエストの入力キューへの受信後、シーケンスマトリックスが再構成され、逐次的なリクエスト間のコンフリクトまたは潜在的遅延がコンフリクト検出器により特定される。コンフリクト検出器は、データバスとの間のデータフローを最適化するため、メモリアクセスリクエストを再構成する。例えば、バンクビジー状態または他の遅延が逐次的に受け取られたメモリリクエストにおいてコンフリクト検出器により認識されると、メモリコントローラは、可能である場合にはコンフリクトまたは遅延を解消するため、あるいはそうでない場合には、遅延を最小化するため、保留中のメモリリクエストを実行する順序を再構成する。

20

## 【 0 0 1 1 】

ライトリクエストは、以前のリクエストとアドレッシングコンフリクトがない限り、アウトオブオーダー実行することができる。

30

## 【 0 0 1 2 】

再順序付けされたシーケンスは実行キューに保持され、各リクエストはもとのシーケンスにおけるその位置を示すためのタグが付けられ、これにより、返されたデータは入力メモリアクセスリクエストの順序に一致するようメモリコントローラにおいて適切に再順序付けすることができる。

## 【 0 0 1 3 】

コマンドセクタは、実行キューから実行対象の 1 以上のコマンドを選択する。コマンドセクタは、一定速度の D R A M シーケンサが複雑なクロックフェージング ( c l o c k p h a s i n g ) 処理を行うことなく複数のクロック速度とインタフェースをとることを可能にするインタフェースタイミング特性を含む。

40

## 【 0 0 1 4 】

リードリターンキューは、D R A M から取得されるリターンデータを追跡する。各リターンデータに関連付けられたタグに基づき、リードリターンキューは、もとの逐次的順序により読み出されたデータを各自の要求元に返す。具体的には、リードリクエストが実行されると、リターンデータに関連付けられたタグがメモリコントローラにおける最長の待ち時間を有するリードリクエストに対応する場合、リターンデータは当該データを要求したシステムユニットに返される。当該タグが最長の待ち時間を有するリードリクエストに関連付けられていない場合、より長い待ち時間を有するすべてのリードリクエストに対するリターンデータが各自の要求元に返されるまで、リターンデータはバッファに格納される。

50

## 【 0 0 1 5 】

あるいは、各リクエストは、受け取ったリクエストシーケンスに基づくバッファ位置に割り当てることができる。D R A Mから取得されたリターンデータが入力キューに受け取られた最先のリードリクエストに対応するバッファ位置に配置されるとき、当該データは適切な要求元に返される。そうでない場合、入力キューに受け取られたすべての以前のリードリクエストが実行されるまで、リターンデータはバッファに保持される。このようにして、リードリターンキューは、もとの受信順序によりバッファ位置からデータを返す。

## 【 発明の効果 】

## 【 0 0 1 6 】

本発明によると、有効なメモリ帯域幅を拡張させるためのメモリアクセス処理を制御するためのメモリコントローラの構成及び動作方法を提供することができる。

10

## 【 発明を実施するための最良の形態 】

## 【 0 0 1 7 】

説明の簡単化のため、本発明の好適な実施例が、ダイナミックランダムアクセスメモリ ( D R A M ) 装置に関して利用されるものとして以下において説明される。しかしながら、本発明が D R A Mに関する用途に限定されるものでないことは理解されるべきである。むしろ、本発明の方法とメモリコントローラは、スタティック R A M ( S R A M ) などの他のタイプのランダムアクセスメモリや、例えば、ファーストページモード D R A M ( F P M D R A M )、拡張データアウト D R A M ( E D O D R A M )、バースト E D O D R A M、同期 D R A M ( S D R A M )、ダブルデータレート D R A M ( D D R D R A M )、ランバス D R A M ( R D R A M ) などを含む様々な D R A Mの亜種と共に利用されてもよい。

20

## 【 0 0 1 8 】

図 1 は、入力キュー 1 2、コマンドパーサ 1 4、シーケンシングユニット 1 6、シーケンスマトリックス 1 8、コンフリクト検出器 2 0、コマンドシーケンサ 2 2、実行キュー 2 4、コマンドセクタ 2 6、入出力バッファ 2 8、リードリターンキュー 3 0 及びリターンデータバッファ 3 2 を有する本発明によるメモリコントローラ 1 0 を示す。

## 【 0 0 1 9 】

メモリアクセスリクエストがメモリコントローラ 1 0 に入力され、先入先出原則に基づき入力キュー 1 2 で受け取られる。その後、受け取ったリクエストは、メモリアドレス ( M A ) データ、チップセレクト ( C S ) コマンド ( アクセス対象のリクエストされたメモリバンクを示す )、行アドレスセレクト ( R A S ) コマンド、列アドレスセレクト ( C A S ) コマンド、及びリクエストがリードまたはライト処理であるかを示すライトイネーブル ( W E ) 状態などの各リクエスト信号から該当する情報を取得するため、コマンドパーサ 1 4 により逐次的に処理される。

30

## 【 0 0 2 0 】

その後、取得された情報は、シーケンスマトリックス 1 8 で受け取られるクロック信号に従って受信したメモリアクセスリクエストをシーケンスマトリックス 1 8 に配置するシーケンシングユニット 1 6 に与えられる。ここで、当該処理の開始にตอบสนองして、アクセスリクエストがそれらのシーケンシングユニット 1 6 での受信順序に従ってマトリックスにロードされる。

40

## 【 0 0 2 1 】

コンフリクト検出器 2 0 は、シーケンスマトリックス 1 8 における当該情報をモニタし、発生可能性があるコンフリクトや遅延に対して、マトリックスのリクエストシーケンスがシーケンスマトリックスにおける現在の順序により実行されるべきかチェックする。コンフリクト検出器 2 0 により検出されたコンフリクトまたは遅延は、シーケンシングユニット 1 6 に通知され、その後、特定されたコンフリクトや遅延によりメモリデータバスがアイドル状態となる時間を最小化あるいは解消するため、マトリックスにおけるリクエストの順序を再構成する。コンフリクト検出器 2 0 により検出される可能性のあるコンフリクトには、以下に限定するものではないが、例えば、ページコンフリクト、メモリバンク

50

が他のリードまたはライト処理を実行するのにビジーであるバンクビジー状態などが含まれる。コンフリクト検出器により特定される遅延は、必ずしも他のメモリアクセスリクエストの実行とコンフリクトするものではないが、メモリデータバスがアイドル状態の期間中に準備ステップの実行を要求する状態である。このような遅延には、例えば、ページコンフリクト、ページミスなどが含まれる。

**【 0 0 2 2 】**

一般に、リード及びライト何れのタイプのメモリアクセスリクエストも、シーケンスマトリックス 1 8 において再順序付けされてもよい。しかしながら、必要な場合には、再順序付けされたシーケンスが入力キューへのより早くに受け取られたリクエストとのアドレッシングコンフリクトを生成しない程度まで、あるいはマトリックスにおける他の任意のメモリアクセスリクエストに関連して該当するメモリ位置に格納された、または格納予定のデータに干渉しない程度まで、ライトリクエストは再構成されることが好ましい。

10

**【 0 0 2 3 】**

追加的なメモリアクセスリクエストがシーケンスマトリックス 1 8 に移される一方、シーケンシングユニット 1 6 によるコンフリクトの解消または遅延の低減を行うため再構成されると、当該シーケンスの先頭にあるリクエストは、再構成されたリクエストが実行を待機する転送バッファとして用いられる実行キュー 2 4 に移される。シーケンシングユニット 1 6 で利用されるコンフリクト解消処理に依存して、クロック信号による連続ベース、所定数のアクセスリクエストのバッチ、あるいは所定の蓄積サイズのリクエストの何れかに基づき、実行キュー 2 4 に移されるようにしてもよい。好ましくは、実行キュー 2 4 のクロック ( C L K ) は、シーケンスマトリックス 1 8 のロード処理を誘導するものと同じのクロック ( C L K ) である。

20

**【 0 0 2 4 】**

コマンドシーケンサ 2 2 は、データコントロールコマンドの挿入、及び/またはデータコントロールコマンドとあるメモリアクセスリクエストのリードまたはライトコマンドとの一時的分離が必要であるかもしれないため、シーケンスマトリックス 1 8 から実行キュー 2 4 に転送される各リクエストに関する様々なコマンドを調整する。例えば、実行キュー 2 4 に送信されるリードまたはライトリクエストがプリチャージ及び/またはアクティブ化処理を必要とする場合、当該プリチャージ及び/またはアクティブ化処理を開始するデータコントロールコマンドは、当該プリチャージ及び/またはアクティブ化コマンドと関連するリードまたはライトコマンドとの間に配置された他のアクセスリクエストに関する少なくとも 1 つの他のリードまたはライトコマンドと共に、実行キューにおける該当するリードまたはライトコマンドの前に配置される。

30

**【 0 0 2 5 】**

本発明の 1 つの効果は、このようにリードまたはライトコマンドとデータコントロールコマンドとを分離することにより実現される。具体的には、上記の例では、他のメモリアクセスリクエストに対するリードまたはライトコマンドは即座に実行可能である間、プリチャージ及び/またはアクティブ化処理は実行可能である。このため、メモリデータバスは、プリチャージ及び/またはアクティブ化処理の実行中、アイドル状態である必要はない。

40

**【 0 0 2 6 】**

各リードリクエストが実行キューに転送されると、各リクエストのもとの相対的配置を入力キュー 1 2 に受け取られたものとして特定するためのタグが、当該リクエストに対するデータコントロールコマンドに一時的に追加される。あるいは、各リードリクエストは、以降において詳述されるリードバッファ 3 2 の各自のバッファ位置に割り当てられるようにしてもよい。

**【 0 0 2 7 】**

コマンドセレクト 2 6 に与えられるコマンドセレクトクロック ( C M D C L K ) に従って、実行キュー 2 4 の先頭からの 1 以上のメモリアクセスリクエストが、適切な D R A M バンクにおいて実行のため選択される。例えば、コマンドセレクトクロック信号がシー

50

ケンスマトリックス 18 にリクエストがロードされるクロック速度の 4 倍である場合、キュー 24 の各クロック信号に対して 4 つのアクセスリクエストが実行キュー 24 から取り除かれる。この場合、コマンドセレクタ 26 は、キュー 24 から 1 度に選択された 4 つのアクセスリクエストのすべてが等しいインターバルで実行されるように、リクエスト選択プロセスを調整する。新たなリクエストがシーケンスマトリックス 18 に入力されるのと同じクロック速度での実行のため、コマンドが実行キューから選択される場合、コマンドセレクタ 26 はメモリコントローラ 10 から省略されてもよい。

**【 0 0 2 8 】**

入出力バッファ 28 は、各アクセスリクエストにおいて指定されるリードまたはライト処理中に使用される遷移バッファである。実行対象の現在のアクセスリクエストがライト処理である場合、選択されたメモリセルへの書き込み対象となるデータは一時的に入出力バッファ 28 に書き込まれる。同様に、リード処理における選択されたメモリセルから読み出されるデータは一時的に入出力バッファ 28 に格納される。

10

**【 0 0 2 9 】**

リクエストされたメモリアクセスの実行により、各リードリクエストがリードリターンキュー 30 に配置される。リードリターンキュー 30 は、リードリクエストの実行により、D R A M から読み出されたリクエストデータを管理し、それらが入力キューに受信された順序に従って、リクエストデータを各自の要求元に返す。D R A M から読み出されたデータは、返されたリクエストデータに関連付けされたタグまたは割り当てられたバッファ位置に依存して、要求元に直接転送されるか、あるいはリターンデータバッファ 32 に配置される。

20

**【 0 0 3 0 】**

図 2 を参照するに、リードリクエストの実行後、実行されたリードリクエストは、入出力バッファ 28 に一時的に保持されているリクエストにより取得されるデータと共に、リードリクエストキュー 30 に返される (ステップ 100)。ステップ 110 において、リードリクエストキュー 30 により、返されるリードリクエストが最も現在のタグまたは最も現在に割り当てられたバッファ位置に関連付けされていると判断されると、当該リクエストにより取得されるデータは、ステップ 120 において要求元に返され、ステップ 130 において現在タグ/バッファ位置情報が更新される。

**【 0 0 3 1 】**

タグまたはバッファ位置は、それがメモリコントローラ 10 における最長の待ち時間を有するリードリクエストに割り当てられているか、あるいは関連付けされている場合に「現在」のものとされる。リードリターンキュー 30 は、例えば、返されたデータがその要求元に転送されるごとに、現在タグまたはバッファ位置を表すカウント値をインクリメントすることにより、最も現在のタグまたはバッファ位置を追跡するかもしれない。

30

**【 0 0 3 2 】**

返されるリードリクエストが現在のものでないタグまたは割り当てられたバッファ位置を有する場合、この関連付けされたタグまたはバッファ位置が現在のものとなるまで、当該データはリードデータバッファ 32 に配置される (ステップ 140)。ステップ 140 において返されたデータをバッファ 32 に配置した後、あるいはステップ 130 において現在タグまたはバッファ位置を更新した後、リードリターンキュー 30 は、当該現在タグまたはバッファ位置に対応する返されたデータがリターンデータバッファ 32 において検出可能であるか判断する。「イエス」である場合、当該プロセスはステップ 120 に戻り、現在のリターンデータが当該データの要求元に転送され、現在タグまたはバッファ位置がステップ 130 において再び更新される。現在タグまたはバッファ位置に対応する返されたデータがバッファ 32 において検出されない場合、当該プロセスは、実行キューの次のリードリクエストの実行により取得されるリターンデータを受け取るため、ステップ 100 に戻る。図 2 に示される当該プロセスにより、リードデータは、リクエストが入力キュー 12 に受け取られた順序で各自の要求元に返される。

40

**【 0 0 3 3 】**

50

タグがリードリクエストの本来の受信順序を示すために使用される場合、タグは各リクエストのデータコントロールコマンドに一時的に挿入されるが、要求元に返されるリードデータに含まれてはいない。好ましくは、リードリターンキュー30及びリターンデータバッファ32の機能は、DRAMまたはその複数のものを一体化したもののクロック速度に対応するクロック速度に従って実行される。従って、リードリターンキュー30及びリターンデータバッファ32において受け取られるリターンリードリクエスト及びリターンデータは、現在タグ/バッファ位置の更新及びリターンデータの転送とそれぞれ協調されうる。

#### 【0034】

図3に示されるフローチャートを参照して、シーケンシングユニット16が検出されたコンフリクトまたは遅延によりメモリアクセスリクエストを再スケジューリングするのに利用するコンフリクト再順序付けプロセスの第1実施例が説明される。ステップ200において、パース対象の直近のメモリアクセスリクエストがシーケンスマトリックス18に配置される。ステップ210において、アクセスされることが所望されるメモリバンクの状態が、新たに届いたメモリアクセスリクエストがシーケンスマトリックス18のその現在の位置において実行される場合に、コンフリクトや遅延が発生するか判断するためチェックされる。

#### 【0035】

シーケンシングユニット16がこの新たに届いたリクエストに関するコンフリクトまたは遅延を検出する場合、シーケンシングユニット16は、ステップ240において、その現在の位置の前にあるメモリアクセスリクエストシーケンスの中で、より適切なタイミング位置が検出可能であるか判断する。具体的には、シーケンシングユニット16はまず、解消されていないコンフリクトまたは遅延が新たなアクセスリクエストの前のシーケンスに存在するか、そして存在する場合には、当該新たなアクセスリクエストが当該時間中にコンフリクトなく実行可能であるか判断する。保留されているコンフリクトまたは遅延が存在しない場合、シーケンシングユニット16は、この新たなアクセスリクエストが、以前にスケジューリングされたリクエストにおいて新たなコンフリクトまたは遅延を発生させることなく、マトリックスの任意のポイントにおいて再スケジューリングされてもよいかチェックする。そうでない場合、本プロセスはステップ230に移り、解消されない状態により新たなアクセスリクエストは現在のタイミングシーケンスの終わりに残される。適切なタイミング位置が当該シーケンスにおける前方に検出可能である場合、新たに届いたリクエストは当該位置においてシーケンスに挿入される(ステップ250)。その後、本プロセスはステップ200に戻り、次に入力されるメモリアクセスリクエストに対して繰り返される。

#### 【0036】

他方、新たに届いたアクセスリクエストに関してコンフリクトや遅延が検出されない場合、ステップ220において、コンフリクト検出器20は次に、新たに届いたリクエストの現在の位置の前に解消されていないコンフリクトまたは遅延がタイミングマトリックスに存在するか判断する。解消されていないコンフリクトまたは遅延が検出されると、本プロセスは上述のステップ240に移行する。既存のコンフリクトが検出されない場合、シーケンシングユニット16は、ステップ230においてシーケンスマトリックス18の現在の位置に当該リクエストをとどめ、その後ステップ200に戻り、次に入力されるメモリアクセスリクエストに対して本プロセスを繰り返す。

#### 【0037】

本実施例では、解消されていないコンフリクトや遅延は、シーケンシングユニット16の以降の繰り返しによって、新たなアクセスリクエストの到達により以降において解消されてもよいし、そうでなくともよい。コンフリクトや遅延を軽減する適切なリクエストが届かない場合、メモリアクセスリクエストは指定された順序で処理が継続されるが、解消されていないコンフリクトや軽減されていない遅延により、メモリデータバスラインの利用に対し非効率さがあるであろう。

10

20

30

40

50

## 【 0 0 3 8 】

図4に示されるフローチャートにおいて、シーケンシングユニット16により実行されるコンフリクト再順序付けプロセスの第2実施例が示される。他のメモリアクセスリクエストがシーケンスマトリックス18から実行キュー24に移されると、ステップ300において、シーケンスマトリックスの次のラインのアクセスリクエストが第1位置に移される。ステップ310において、コンフリクト検出器20は、コンフリクトや遅延がシーケンスマトリックスの先頭にあるアクセスリクエスト(すなわち、マトリックスのリクエストの中の最先のもの)に関して存在するか判断する。コンフリクトが検出されない場合、ステップ320において、アクセスリクエストは実行キュー24にわたされ、本プロセスはステップ300に戻る。

10

## 【 0 0 3 9 】

他方、コンフリクトまたは遅延がステップ310において検出されると、コンフリクト検出器20は、シーケンスマトリックスの次のアクセスリクエストに注目し、ステップ330において、当該リクエストがコンフリクトなく当該時間位置において実行可能であるか判断する。当該時間位置において当該アクセスリクエストをスケジューリングすることにより、コンフリクトまたは遅延が生じない場合、ステップ320においてリクエストは実行キュー24に送られ、本プロセスはステップ300に戻る。コンフリクトまたは遅延が検出されると、当該時間位置において適切に実行されてもよいリクエストが検出されるまで、本プロセスはステップ330を繰り返す。

## 【 0 0 4 0 】

本実施例では、シーケンスマトリックスの先頭にあるリクエスト及び以降のリクエストの何れかのメモリアクセスに関して存在するコンフリクトまたは遅延が検出される場合、適切なタイムスロットがこのリクエストに対して検出されるまで、拒絶されたリクエストはシーケンスのそれらの現在位置に残される。本プロセスにより、コンフリクトまたは遅延状態により実行キュー24に送信されるメモリアクセスは存在しない。また、各リクエストには待ち時間に基づく優先度が与えられ、当該リクエストのタイミングによりコンフリクト状態が生じない最先の可能なタイムスロットにおいて実行されるであろう。

20

## 【 0 0 4 1 】

図5において、図4を参照して上述された実施例の一変形が示される。ここでは、シーケンスマトリックスの先頭のアクセスリクエストの現在のタイミング位置に関してコンフリクトまたは遅延が検出されると、不適なリクエストは、その現在位置においてシーケンスに残されるのではなく、シーケンスの後方に送られる。具体的には、ステップ310においてコンフリクトが検出されると、ステップ420が実行され、コンフリクトまたは遅延を有するリクエストはシーケンスの後方に送られる。その後、コンフリクト検出器20は、次のラインのアクセスリクエストに移り、当該リクエストにコンフリクトまたは遅延が検出されるか判断する(ステップ430)。コンフリクトまたは遅延が検出されない場合、図4に示されたプロセスと同様に、当該リクエストは実行キュー24に送られる。コンフリクトまたは遅延が検出されると、本プロセスはステップ420に戻る。

30

## 【 0 0 4 2 】

図6を参照するに、本発明の動作により取得される結果を示すタイミングチャートが示される。本例では、リードリクエストRD<sub>0</sub>、RD<sub>1</sub>及びRD<sub>2</sub>がこの順序で入力キューにおいて受け取られる。ページコンフリクト状態がRD<sub>0</sub>に関して検出されており、RD<sub>1</sub>とRD<sub>2</sub>は共に他のメモリバンクにおけるページヒットであると仮定する。RD<sub>1</sub>及びRD<sub>2</sub>より以前にRD<sub>0</sub>が完了するよう実行されることが許可される場合には遅延が生じるため、RD<sub>0</sub>に対するプリチャージ処理の実行中、RD<sub>1</sub>及びRD<sub>2</sub>に対するリードメモリアクセスが実行されるようこれらのリクエストは再構成される。このようなコマンドの再構成により、RD<sub>0</sub>に対するプリチャージ処理の実行が終了するのを待機する間に、メモリデータバスがアイドル状態となる時間を最小化し、これにより、メモリデータバスのより効率的な利用が可能となる。

40

## 【 0 0 4 3 】

50

RD<sub>1</sub>及びRD<sub>2</sub>に対するデータがDRAMから読み出された後、RD<sub>0</sub>に対するアクティブ化処理が実行され、その後RD<sub>0</sub>に対するリードアクセスコマンドが実行される。リードデータ「1111」及び「2222」は、リード処理の実行順序によりリターンデータバッファに転送される。しかしながら、「0000」のデータが適切なメモリセルから取得された後、当該データはRD<sub>0</sub>の要求元に最初に戻される。その後、データ「1111」と「2222」がその順序で各自の要求元に返される。

【0044】

図7は、本発明のメモリコントローラ10を利用する一例となる処理システム900を示す。処理システム900は、ローカルバス904に接続された1以上のプロセッサ901を有する。メモリコントローラ10とプライマリバスブリッジ903がまたローカルバス904に接続される。処理システム900は、複数のメモリコントローラ10及び/または複数のプライマリバスブリッジ903を有するようにしてもよい。メモリコントローラ10とプライマリバスブリッジ903は、単一の装置906として一体化されてもよい。

【0045】

メモリコントローラ10はまた、1以上のメモリデータバス907に接続される。各メモリバスは、少なくとも1つのメモリ装置902を有するメモリコンポーネント908を受け入れる。メモリコンポーネント908は、メモリカードまたはメモリモジュールとして構成されてもよい。システム900において利用可能なメモリモジュールの例として、シングルインラインメモリモジュール(SIMM)やデュアルインラインメモリモジュール(DIMM)があげられる。メモリコンポーネント908は、1以上の追加的装置909を有するようにしてもよい。例えば、SIMMまたはDIMMでは、追加的装置909は、SPD(Serial Presence Detect)メモリなどの設定メモリであってもよい。

【0046】

メモリコントローラ10はまた、キャッシュメモリ905に接続されてもよい。キャッシュメモリ905は、処理システムの唯一のキャッシュメモリであってもよい。あるいは、プロセッサ901などの他の装置がまた、キャッシュメモリ905のキャッシュ階層を構成する複数のキャッシュメモリを有するようにしてもよい。処理システム900がバスマスタであったり、あるいはDMA(Direct Memory Access)をサポートする周辺装置やコントローラを有する場合、メモリコントローラ10は、キャッシュコヒーレンシプロトコルを実現するようにしてもよい。メモリコントローラ10が複数のメモリバス907に接続される場合、キャッシュメモリバス907はパラレルに動作してもよいし、あるいは異なるアドレス範囲が異なるメモリバス907にマッピングされてもよい。

【0047】

プライマリバスブリッジ903は、少なくとも1つの周辺バス910に接続される。周辺装置や追加的バスブリッジなどの各種装置が、周辺バス910に接続されてもよい。これらの装置は、格納コントローラ911、その他I/O装置914、セカンダリバスブリッジ915、マルチメディアプロセッサ918及びレガシー装置インタフェース920を有するようにしてもよい。プライマリバスブリッジ903はまた、1以上の特殊用途高速ポート922に接続されてもよい。パーソナルコンピュータでは、例えば、特殊用途ポートは、高パフォーマンスビデオカードと処理システム900とを接続するために利用されるAGP(Accelerated Graphics Port)であってもよい。

【0048】

格納コントローラ911は、格納バス912を介して1以上の記憶装置913と周辺バス910とを接続する。例えば、格納コントローラ911はSCSIコントローラであり、記憶装置913はSCSIDISKであってもよい。I/O装置914は、任意の種類の周辺装置であってもよい。例えば、I/O装置914は、イーサネット(登録商標)カードなどのローカルエリアネットワークインタフェースであってもよい。セカンダリバス

10

20

30

40

50

ブリッジは、他のバスを介して追加的装置と処理システムとのインタフェースをとるのに利用されてもよい。例えば、セカンダリバスブリッジは、USB (Universal Serial Port) 装置 917 と処理システム 900 とを接続するのに用いられる USB コントローラであってもよい。マルチメディアプロセッサ 918 は、サウンドカード、ビデオキャプチャカードまたは他の任意のタイプのマルチメディアインタフェースであってもよく、スピーカー 919 などの 1 つの追加的装置に接続されてもよい。レガシー装置インタフェース 920 は、旧式のキーボードやマウスなどの従来装置と処理システム 900 とを接続するのに利用される。

#### 【0049】

図 7 に示される処理システム 900 は、本発明が利用可能な単なる一例となる処理システムである。図 7 はパーソナルコンピュータやワークステーションなどの汎用コンピュータに特に適した処理アーキテクチャを示しているが、様々な用途における利用により適するものとするため、処理システム 900 に周知の改良が可能であるということは認識されるべきである。例えば、処理を要する多数の電子装置は、メモリコンポーネント 908 及び/または記憶装置 902 に接続される CPU 901 によるよりシンプルなアーキテクチャを用いて実現されてもよい。これらの電子装置には、以下に限定するものではないが、音声/映像プロセッサ及びレコーダ、ゲーム端末、デジタルテレビ装置、有線/無線電話、ナビゲーション装置 (GPS (Global Positioning System) 及び/または慣性ナビゲーションに基づくシステムを含む)、及びデジタルカメラ及び/またはレコーダが含まれてもよい。上記改良には、例えば、不要な構成要素の削除、特殊な装置または回路の追加、及び/または複数の装置の統合などが含まれてもよい。

#### 【0050】

本発明がその特定の実施例に関して説明されたが、他の多くの変形及び改良、そして他の利用が当業者には明らかになるであろう。従って、本発明は、ここでの特定の開示によってでなく、添付されたクレームによってのみ限定されることが好ましい。

#### 【図面の簡単な説明】

#### 【0051】

【図 1】図 1 は、本発明によるメモリコントローラのブロック図である。

【図 2】図 2 は、本来のリクエストの受信順序により、リードリクエストの再順序付けされた実行により取得されるリクエストデータを各自の要求元に返すプロセスを示すフローチャートである。

【図 3】図 3 は、現在のタイミングマトリックスにおけるコンフリクトまたは遅延の検出により、メモリコントローラ内のシーケンスロード論理ユニットにより実行されるコンフリクト再順序付けプロセスの第 1 実施例である。

【図 4】図 4 は、現在のタイミングマトリックスにおけるコンフリクトまたは遅延の検出により、メモリコントローラ内のシーケンスロード論理ユニットにより実行されるコンフリクト再順序付けプロセスの第 2 実施例である。

【図 5】図 5 は、現在のタイミングマトリックスにおけるコンフリクトまたは遅延の検出により、メモリコントローラ内のシーケンスロード論理ユニットにより実行されるコンフリクト再順序付けプロセスの第 3 実施例である。

【図 6】図 6 は、本発明の動作を示すタイミングチャートである。

【図 7】図 7 は、本発明のメモリコントローラが利用される処理システムのブロック図である。

#### 【符号の説明】

#### 【0052】

- 10 メモリコントローラ
- 12 入力キュー
- 14 コマンドパーサ
- 16 シーケンシングユニット
- 18 シーケンスマトリックス

10

20

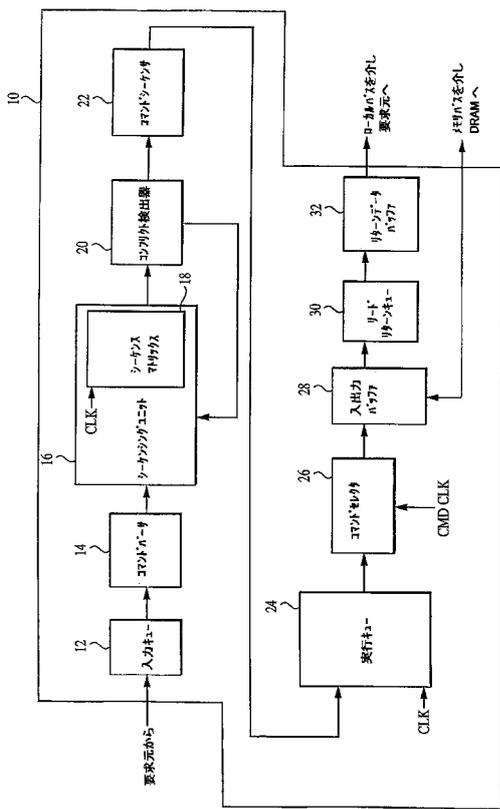
30

40

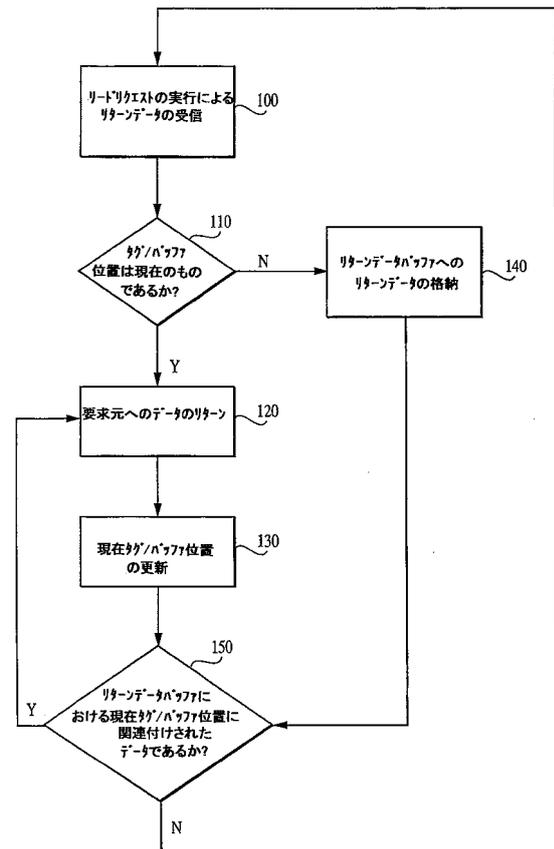
50

- 20 コンフリクト検出器
- 22 コマンドシーケンサ
- 24 実行キュー
- 26 コマンドセレクタ
- 28 入出力バッファ
- 30 リードリターンキュー
- 32 リターンデータバッファ
- 900 処理システム
- 901 プロセッサ
- 902 メモリ装置

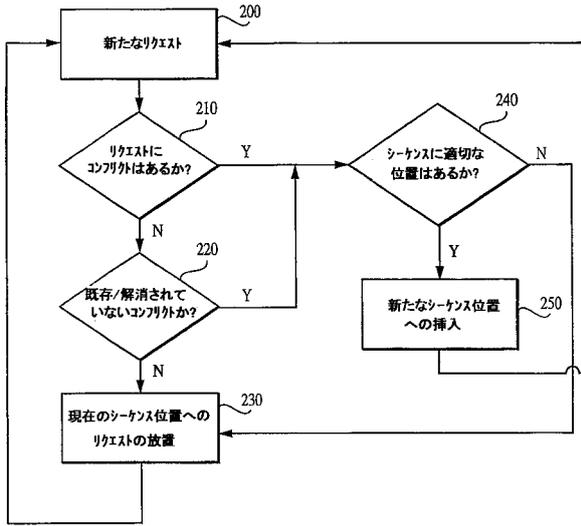
【図1】



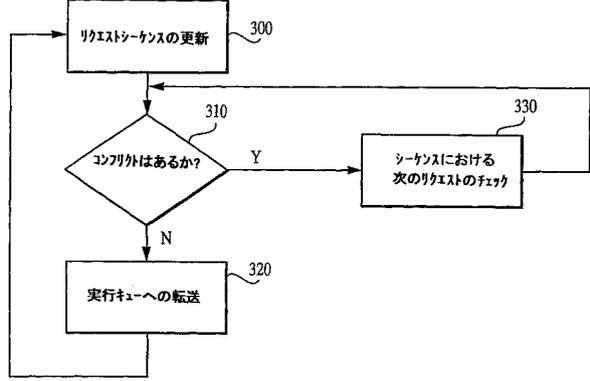
【図2】



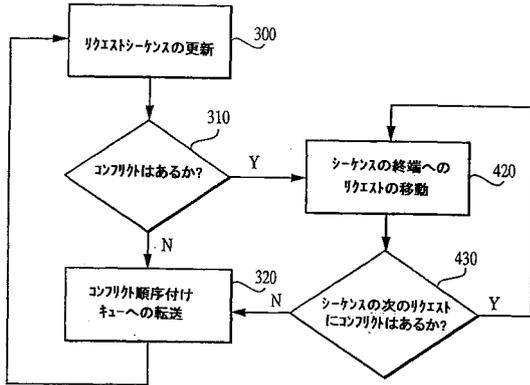
【図3】



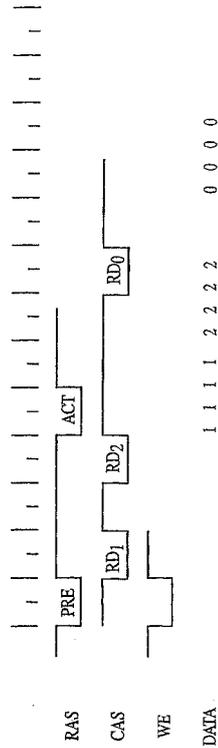
【図4】



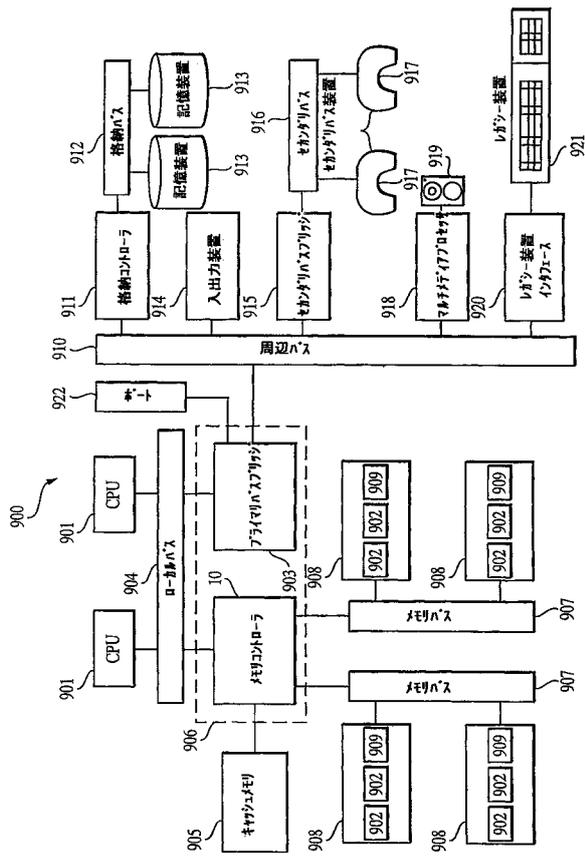
【図5】



【図6】



【図7】



---

フロントページの続き

審査官 鈴木 和樹

- (56)参考文献 特表2003-535380(JP,A)  
特開2001-154913(JP,A)  
特開平10-228417(JP,A)  
特開平3-263242(JP,A)  
国際公開第00/29959(WO,A1)  
米国特許第5630096(US,A)  
特開平6-161885(JP,A)

(58)調査した分野(Int.Cl., DB名)

G06F 12/00 - 12/06