

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号  
特許第6495327号  
(P6495327)

(45) 発行日 平成31年4月3日(2019.4.3)

(24) 登録日 平成31年3月15日(2019.3.15)

(51) Int.Cl.

F I

GO 6 F 12/00 (2006.01)

GO 6 F 12/06 (2006.01)

GO 6 F 12/00 5 7 1 B

GO 6 F 12/06 5 5 0 A

請求項の数 10 (全 21 頁)

(21) 出願番号	特願2016-560970 (P2016-560970)	(73) 特許権者	314015767
(86) (22) 出願日	平成27年4月6日 (2015.4.6)		マイクロソフト テクノロジー ライセン
(65) 公表番号	特表2017-511545 (P2017-511545A)		シング, エルエルシー
(43) 公表日	平成29年4月20日 (2017.4.20)		アメリカ合衆国 ワシントン州 9805
(86) 国際出願番号	PCT/US2015/024414		2 レッドモンド ワン マイクロソフト
(87) 国際公開番号	W02015/160541		ウェイ
(87) 国際公開日	平成27年10月22日 (2015.10.22)	(74) 代理人	100107766
審査請求日	平成30年3月19日 (2018.3.19)		弁理士 伊東 忠重
(31) 優先権主張番号	14/252,673	(74) 代理人	100070150
(32) 優先日	平成26年4月14日 (2014.4.14)		弁理士 伊東 忠彦
(33) 優先権主張国	米国 (US)	(74) 代理人	100091214
			弁理士 大貫 進介

最終頁に続く

(54) 【発明の名称】 メモリコントローラにおける細粒度帯域幅プロビジョニング

(57) 【特許請求の範囲】

【請求項 1】

メモリコントローラにおいて帯域幅プロビジョニングを実行するための方法であって、複数のマスタからバスファブリックを介してメモリ要求を受信するステップと、前記メモリ要求の各メモリ要求に割り当てられる優先順位レベルを決定するステップと

、メモリデバイスから受信された、最後に発行された要求に関する情報に基づいて、要求の帯域幅コストを算出するステップと、

メモリデータバス上で各マスタにより消費される、1つの要求当たりのメモリデータバス帯域幅の量を決定するステップと、

各マスタに割り当てられた優先順位クラスと、前記最後に発行された要求に関する前記情報に基づく、各要求の前記帯域幅コストと、各マスタにより消費される、1つの要求当たりのメモリデータバス帯域幅の量と、に基づいて、サービス品質(QoS)ロジックを適用して、前記メモリ要求のスケジュールを算出するステップであって、各マスタにより消費される、1つの要求当たりの前記メモリデータバス帯域幅は、最後に発行された要求に関する情報であって、該最後に発行された要求によりターゲットとされたページのページアドレス及びバンクと、該最後に発行された要求の要求タイプと、該最後に発行された要求が、ページヒットを有していたかページミス有していたか又はページ競合を有していたかと、を含む情報に基づいて算出される、ステップと、

前記スケジュールに基づいて、

前記メモリ要求をメモリコマンドに変換するステップと、  
メモリコマンドバスを介して前記メモリコマンドを少なくとも1つのメモリデバイスに送信するステップと、  
前記少なくとも1つのメモリデバイスから前記メモリデータバスを介して応答を受信するステップと、  
を含む方法。

【請求項2】

前記少なくとも1つのメモリデバイスは、ダイナミックランダムアクセスメモリ(DRAM)デバイスであり、前記メモリデータバスは、DRAMデータバスであり、前記メモリ要求は、DRAMメモリ要求であり、前記メモリコマンドは、DRAMメモリコマンドであり、前記メモリコマンドバスは、DRAMコマンドバスであり、

10

前記方法は、  
前記優先順位レベルに基づいて、複数の優先順位キューのうちの1つの優先順位キューに各DRAM要求を割り当てるステップ  
をさらに含む、請求項1記載の方法。

【請求項3】

前記複数の優先順位キューからマスタスケジューリングマルチプレクサに前記DRAM要求を送信するステップと、

各マスタに割り当てられた帯域幅の未使用部分を定期的にモニタリングするステップと、

20

前記帯域幅の前記未使用部分が予め定められたレベル以下に低下した場合に前記メモリコントローラにおける前記DRAM要求を限定するように前記スケジュールを算出するステップと、

をさらに含む、請求項2記載の方法。

【請求項4】

各マスタに割り当てられた帯域幅の未使用部分を定期的にモニタリングする前記ステップは、

QoS帯域幅プロビジョニングロジックを適用して、前記複数のマスタの各マスタに、前記DRAMデータバスにおける予め定められた量の帯域幅を割り当てるステップ

を含む、請求項3記載の方法。

30

【請求項5】

各マスタに割り当てられた帯域幅の未使用部分を定期的にモニタリングする前記ステップは、

QoSクレジットアキュムレータを適用して、前記複数のマスタの各マスタについてのクレジットをインクリメンタルに累積するステップであって、累積のレートは、前記QoS帯域幅プロビジョニングロジックにより各マスタに割り当てられた前記帯域幅に対応し、各マスタについての総クレジットは、予め定められた最大QoSクレジット値を超えない、ステップ

を含む、請求項4記載の方法。

【請求項6】

40

各マスタに割り当てられた帯域幅の未使用部分を定期的にモニタリングする前記ステップは、

QoSチャージ勘定ロジックを適用して、各DRAMメモリ要求により消費される前記DRAMデータバス上の帯域幅を算出するステップと、

前記QoSクレジットアキュムレータにおける、各マスタに対応する前記クレジットから、前記の算出された帯域幅を減算することにより、各マスタについてのクレジット残余を算出するステップと、

を含む、請求項5記載の方法。

【請求項7】

各DRAMメモリ要求により消費される前記DRAMデータバス上の帯域幅を算出する

50

ことは、最後に発行された要求に関する前記情報に基づいて、要求の前記帯域幅コストを算出することを含む、請求項 6 記載の方法。

【請求項 8】

スケジューリングロジックを適用して、前記帯域幅の未使用部分が予め定められたレベル以下に低下した場合に前記メモリコントローラにおける D R A M 要求を限定することは、

Q o S スケジューリングロジックを適用して、各 D R A M 要求の前記優先順位クラスに基づいて前記 D R A M 要求をスケジューリングし、D R A M 要求を発行したマスタについての、前記 Q o S クレジットアキュムレータにおける前記クレジット残余が、予め定められた最小 Q o S クレジット値未満である場合に、前記 D R A M 要求を限定するように前記スケジュールを算出する

10

ことにより達成される、請求項 7 記載の方法。

【請求項 9】

前記 D R A M 要求を、複数の D R A M デバイスのうちの少なくとも 1 つの D R A M デバイスに送信することは、

前記マスタスケジューリングマルチプレクサから D R A M プロトコルエンジンに前記 D R A M 要求を送信することと、

前記 D R A M プロトコルエンジンから前記複数の D R A M デバイスのうちの少なくとも 1 つの D R A M デバイスに前記 D R A M 要求を送信することと、

を含む、請求項 8 記載の方法。

20

【請求項 10】

複数のマスタからバスファブリックを介してメモリ要求を受信し、

前記複数のマスタの各マスタに割り当てられる優先順位レベルを決定し、

メモリデバイスから受信された、最後に発行された要求に関する情報に基づいて、要求の帯域幅コストを算出し、

メモリデータバス上で各マスタにより消費される、1 つの要求当たりのメモリデータバス帯域幅の量を決定し、

各マスタに割り当てられた優先順位クラスと、前記最後に発行された要求に関する前記情報に基づく、各要求の前記帯域幅コストと、各マスタにより消費される、1 つの要求当たりのメモリデータバス帯域幅の量と、に基づいて、サービス品質 ( Q o S ) ロジックを適用して、前記メモリ要求のスケジュールを算出し、ここで、各マスタにより消費される、1 つの要求当たりの前記メモリデータバス帯域幅は、最後に発行された要求に関する情報であって、該最後に発行された要求によりターゲットとされたページのページアドレス及びバンクと、該最後に発行された要求の要求タイプと、該最後に発行された要求が、ページヒットを有していたかページミスを有していたか又はページ競合を有していたかと、を含む情報に基づいて算出され、

30

前記スケジュールに基づいて、

前記メモリ要求をメモリコマンドに変換し、

メモリコマンドバスを介して前記メモリコマンドを少なくとも 1 つのメモリデバイスに送信し、

40

前記少なくとも 1 つのメモリデバイスから前記メモリデータバスを介して応答を受信する

よう構成されているメモリコントローラ。

【発明の詳細な説明】

【背景技術】

【0001】

ダイナミックランダムアクセスメモリ ( D R A M ) デバイスは、コンピュータ、スマートフォン、タブレット、ゲームコンソール、及びテレビジョンを含む多種多様な電子デバイス内で見つけられる。D R A M デバイスは、それらの電子デバイス上で動作しているアプリケーションが、電子デバイス内の不揮発性ストレージ (例えば、フラッシュメモリや

50

ハードドライブ)から命令及びデータを、こうした命令が電子デバイスの中央処理装置(CPU)又は他の固定ハードウェアユニットにより実行される前に、一時的に記憶させることを可能にする。

#### 【0002】

効率性の理由から、DRAMデバイスは、電子システムにおいて、DRAMメモリデバイスへの読み取りトラフィック又は書き込みトラフィックを生成し得る多くの異なるマスタデバイス又はIP(intellectual-property)ロジックブロックにより共用される。各マスタデバイスは、しばしばバスファブリック及びメモリコントローラを介して、メモリ要求をDRAMデバイスに発行する。メモリコントローラは、これらのメモリ要求をスケジューリングし、これらのメモリ要求がDRAMデバイスにどのように送信されるかを決定する役割を担う。DRAMデバイスのアクセス待ち時間、したがって、DRAMデバイスの実効利用帯域幅は、メモリコントローラがメモリ要求をいかに効率的且つ実効的にスケジューリングできるかに依存する。このスケジューリングが非効率的である場合、過度の帯域幅変動が、ビデオゲームにおけるフレームレート低下やオーディオフレームの欠落に起因するサウンドポッピング(sound popping)といった、システムのハードウェア及びソフトウェアアプリケーションの不具合を生じさせ得る。これらの不具合を防止するための一般的なアプローチは、メモリ要求がどのようにスケジューリングされるかを決定するサービス品質(QoS)ロジックを利用すること、及び、できる限り多くの既知の帯域幅変動シナリオ下でアプリケーションに不具合が生じないことを確実にするための広範なテストを実行すること、である。

#### 【0003】

従来技術において、QoSロジックソリューションは、主として、優先順位付け及び速度制限という2つのメカニズムに依拠していた。優先順位付けメカニズムは、システムによってより高い優先順位が割り当てられたマスタからの要求を、より低い優先順位が割り当てられたマスタからの要求よりも優先させる。一方、低優先順位のマスタの不足を防止するために、速度制限メカニズムは、高優先順位のマスタが、システムによってそれらに割り当てられた帯域幅の量を超えるのを防止するために使用される。

#### 【0004】

しかしながら、これらの従来のQoSロジックソリューションは、マスタデバイスにより認識されるような、DRAMデバイスの実効帯域幅の大幅な変動を依然として可能にしてしまう。したがって、従来のQoSロジックソリューションを利用する電子システム用に開発されたソフトウェアアプリケーションは、帯域幅変動が広範にテストされている特定のタイプのDRAMに制限されていた。より高速であれより低速であれ異なるタイプのDRAMデバイスが、システムにおいて使用されるならば、QoSロジックソリューションは、さらなる帯域幅変動を受け、ソフトウェアアプリケーションは、正しく動作しないであろう。通常は長い製品ライフサイクルを有するゲームコンソールの例においては、この問題は、以前に開発されたゲームが、アップデートされたハードウェア上で正しく動作しないようになることをおそれて、コンソールメーカーが、ゲームコンソールのライフサイクルの後期の間に、より高速でより低価格のタイプのDRAMに切り替えることを妨げさせてしまう。

#### 【発明の概要】

#### 【0005】

細粒度(fine-grained)QoSロジックを適用するためのシステム及び方法が提供される。本システムは、メモリコントローラを含み得る。メモリコントローラは、複数のマスタからバスファブリックを介してメモリアクセス要求を受信するよう構成されている。メモリコントローラは、複数のマスタの各マスタの優先順位クラスを決定し、さらに、データバス上で各マスタにより消費されるデータバス帯域幅の量を決定する。複数のマスタの各マスタに割り当てられた各マスタに割り当てられた優先順位クラスと、各マスタにより消費されるデータバス帯域幅の量と、に基づいて、メモリコントローラは、細粒度QoSロジックを適用して、メモリアクセス要求のスケジュールを算出する。このスケジュール

に基づいて、メモリコントローラは、メモリアクセス要求をメモリアクセスコマンドに変換し、メモリコマンドバスを介してメモリアクセスコマンドをメモリデバイスに送信し、メモリデバイスからメモリデータバスを介して応答を受信する。

【0006】

この発明の概要は、発明を実施するための形態において以下でさらに説明されるコンセプトのうち選択されたコンセプトを簡略化された形で紹介するために提供されている。この発明の概要は、特許請求される主題の主要な特徴又は必要不可欠な特徴を特定することを意図するものではないし、特許請求される主題の範囲を限定するために使用されることを意図するものでもない。さらに、特許請求される主題は、本開示のいずれかの部分に記載されているいずれかの問題又は全ての問題を解決する実施形態に限定されるものではない。

10

【図面の簡単な説明】

【0007】

【図1】一実施形態に従った、細粒度QoSロジックを実行して、DRAMデータバスの帯域幅をプロビジョニングするよう構成されているシステムの概略図。

【図2】DRAMデータバス上でDRAM要求により消費される帯域幅の量を算出するよう構成され得る、図1のシステムの細粒度QoSロジックの概略図。

【図3】マスタ、バスファブリック、及びDRAMメモリコントローラが、システムオンチップとして形成されている、細粒度QoSロジックを実行して、DRAMデータバスの帯域幅をプロビジョニングするよう構成されている第2の実施形態のシステムの概略図。

20

【図4】一実施形態に従った、DRAMメモリコントローラにおいて細粒度帯域幅プロビジョニングを実行するための方法のフローチャート。

【図5】図4の方法のステップを達成することができるサブステップを例示するフローチャート。

【図6】図4の方法のステップを達成することができるサブステップを例示するフローチャート。

【図7】システムのさらなるコンポーネントを示す概略図。

【発明を実施するための形態】

【0008】

ダイナミックランダムアクセスメモリ(DRAM)等のメモリへのアクセスを制御するためのシステム及び方法が、上述した問題に対処するために、本明細書で開示される。

30

【0009】

現代の電子システムにおいては、しばしばマスタデバイスと呼ばれるCPU及び他のハードウェアデバイスが、メモリコントローラ及び様々なバスを介して、システムのDRAMに要求を発行することが一般的である。メモリコントローラは、様々なキューとマルチプレクサとを使用して、マスタデバイスから発行された要求をスケジューリングすることができ、要求を実行する最も効率的な方法を決定するために、品質サービス(QoS)ロジックを適用することができる。

【0010】

図1は、メモリコントローラ12を含む一実施形態のシステム10を示している。メモリコントローラ12は、複数のマスタ16からバスファブリック18を介してダイナミックランダムアクセスメモリ(DRAM)要求14を受信し、複数のマスタ16の各マスタに割り当てられる優先順位レベルを決定し、DRAMデータバス30上で各マスタにより消費されるDRAMデータバス帯域幅48の量を決定し、各マスタ16に割り当てられた優先順位レベルと、各マスタ16により消費される帯域幅48の量と、に基づいて、細粒度QoSロジック26を適用して、DRAM要求14のスケジュールを算出し、このスケジュールに基づいて、DRAM要求14をDRAMコマンド42に変換し、DRAMコマンドバス34を介してDRAMコマンド42を少なくとも1つのDRAMデバイス32に送信し、少なくとも1つのDRAMデバイス32からDRAMデータバス30を介して応答を受信する、よう構成されている。マスタ16は、例えば、ビデオゲームコンソールの

40

50

CPU及びGPUであり得、ビデオゲームコンソールのユーザにより現在プレイされているビデオゲームに関連するデータをDRAMデバイス32から読み込むこと又はDRAMデバイス32に書き込むことを要求している最中であり得る。

#### 【0011】

要求14がマスタ16により発行されると、システム10は、バスファブリック18を介して要求を送信するよう構成されている。バスファブリックが、図1において、マスタ16とメモリコントローラ12との間に配置された1つのファブリックとして示されている。しかしながら、バスファブリック18は、マスタ16からの全てのトラフィックパターンに対応するのに必要とされる多数のバスセグメントを含み得る。図3は、バスファブリックが、速度リミッタを含むバスセグメント60及び速度リミッタを含まないバスセグメント62を含む複数のバスセグメントに分割されている第2の実施形態のシステム10を示している。図1に戻ると、バスファブリック18は、QoS優先順位割り当てロジック20を適用して、マスタ16の各マスタに割り当てられた優先順位レベルに基づいて要求14の各要求に優先順位レベルを割り当てよう構成され得る。優先順位レベルは、システム10上で動作するソフトウェアアプリケーションにより、マスタ16の各マスタに割り当てられ得る。そのような割り当ては、例えば、システム10の起動のたびに1回、アプリケーションがシステム10上で実行されるときのアプリケーション境界上で、又は、システム10上の仮想環境の各起動時に、生じ得る。マスタ16は、帯域幅変動に敏感であり得るリアルタイムマスタデバイスであって、帯域幅変動が過度になった場合にユーザ体験の低下を生じさせ得るリアルタイムマスタデバイスを含み得る。そのようなリアルタイムマスタデバイスには、システム10上で動作するソフトウェアアプリケーションにより、より高い優先順位レベルが割り当てられ得る。

#### 【0012】

要求14の各要求に優先順位レベルを割り当てた後、バスファブリック18は、要求をメモリコントローラ12に送信するよう構成され得る。メモリコントローラ12は、さらに、この優先順位レベルに基づいて、複数の優先順位キュー22のうちの1つの優先順位キューに各DRAM要求14を割り当て、複数の優先順位キュー22の各優先順位キュー内のDRAM要求14をスケジューリングするよう構成され得る。

#### 【0013】

要求14をスケジューリングするために優先順位キュー22により使用され得るスケジューリングパラメータを理解するために、DRAMデバイス32の構造及び動作の説明に一時的に移ることが助けになる。DRAMデバイス32は、「ページ」のアレイとして構成され得る。ここで、各ページは、複数のメモリセル（例えば、1ページ当たり8キロビットと16キロビットとの間）を含む。ページは、行及び列状に配置され、さらに、バンクにグループ化され得る。ここで、各バンクは、数万のページを含む。DRAMデバイス32内のページは、要求14のうちの1つの要求によりアクセスされたか否かに応じて、オープン状態であるか又はクローズ状態であるかのいずれかであり得る。システム10は、システム10が起動されたときにはDRAMデバイス32の全てのバンク内の全てのページがクローズ状態であるように、構成され得る。DRAM要求14のうちDRAMデバイス32のバンク内のクローズ状態のページへのアクセスを要求する1つのDRAM要求が発行されたとき、このDRAM要求14は、最初に「ページミス (page miss)」をもたらすことになり、メモリコントローラ12は、「ページオープン操作」を実行して、ターゲットとされたページをオープンする必要がある。ページオープン操作が生じている間、メモリコントローラ12は、そのバンク内のいかなる他のページのコンテンツにもアクセスすることができない。なぜならば、メモリコントローラ12は、通常、どの時点においても各バンク内に1つのオープン状態のページしか許容しないよう構成されているからである。結果として、ページオープン操作中、メモリコントローラ12は、他のバンク内のページをターゲットとする要求14を探すよう構成され得る。

#### 【0014】

ページがオープン状態であるとき、その同じページにアクセスすることを要求する後続

10

20

30

40

50

の要求 1 4 は、最初にメモリコントローラ 1 2 内の「ページヒット (page hit)」をもたらしことになり、したがって、ページオープン操作が実行されることなく完了し得る。したがって、メモリコントローラ 1 2 は、「同一ページ収集 (same page gathering)」を実行するよう構成され得る。「同一ページ収集」において、同じページへのアクセスを要求する要求 1 4 は、ページオープン操作の必要性を低減させるために、一緒にグループ化される。

#### 【 0 0 1 5 】

後続の要求が、最後に発行された要求と同じバンク内の異なるページをターゲットとした (すなわち、最後の要求が、現在の要求の前に完了した) 場合、いわゆる「ページ競合」が発生し、メモリコントローラ 1 2 は、要求 1 4 によりターゲットとされたページを、ページオープン操作によってオープンする前に、現在オープン状態のページを、「ページクローズ操作」によってクローズしなければならない。ページオープン操作及びページクローズ操作の両方が、D R A M データバス 3 0 上の帯域幅を消費する。D R A M データバス 3 0 上の帯域幅消費が、所与の要求 1 4 を完了するのに必要とされるバスサイクルの数として測定され得る。バスサイクルの数、したがって、1 回の操作当たりの D R A M データバス 3 0 上で消費される帯域幅 4 8 の量が、D R A M デバイス 3 2 用に使用される特定のタイプのメモリのタイミングパラメータにより決定される。しかしながら、使用されるメモリのタイプにかかわらず、D R A M データバス 3 0 上の帯域幅消費 4 8 は、要求 1 4 がページヒットを有するかページミスをするか又はページ競合を有するかに応じて変わる。

#### 【 0 0 1 6 】

さらに、D R A M 要求 1 4 は、読み取り要求又は書き込み要求のいずれかであり得る。要求 1 4 が読み取り要求である場合、メモリコントローラ 1 2 は、D R A M コマンドバス 3 4 上で「読み取りコマンド」を送信して、予め定められた数のクロックサイクルの後に D R A M データバス 3 0 上で「読み取りデータ」を受信する「読み取り操作」を実行するよう構成されている。要求 1 4 が書き込み要求である場合、メモリコントローラ 1 2 は、D R A M コマンドバス 3 4 上で書き込みコマンドを送信して、予め定められた数のクロックサイクルだけ待機した後に D R A M データバス 3 0 上で書き込みデータを送信する書き込み操作を実行するよう構成されている。しかしながら、要求 1 4 が書き込み要求であり、最後に発行された要求が読み取り要求であった場合、メモリコントローラ 1 2 は、データバス 3 0 の使用における競合から生じるデータハザード (すなわち、データの処理を誤るリスク) を回避するために、「読み取りから書き込みへのバスターンアラウンド (リードツーライトバスターンアラウンド: read to write bus turnaround)」を実行するよう構成されている。例えば、書き込み操作が開始されたときに、読み取りデータが D R A M データバス 3 0 上でまだ返ってきている場合、送信する書き込みデータと受信する読み取りデータとの間のバス衝突が発生し得る。書き込み操作中におけるそのような発生を回避するために、書き込み操作を開始する前に、メモリコントローラ 1 2 は、前の読み取りデータが D R A M データバス 3 0 上で返されていないことを確認するよう構成され得る。競合が存在するときには、メモリコントローラ 1 2 は、「読み取りから書き込みへのバスターンアラウンド」を実行するよう構成され得る。「読み取りから書き込みへのバスターンアラウンド」において、書き込みデータを D R A M デバイス 3 2 に送信する前にデータバス 3 0 上で読み取りデータが返されることを可能にするために、書き込みコマンドが遅延される。同様に、要求 1 4 が読み取り要求であり、最後に発行された要求が書き込み要求であった場合、メモリコントローラ 1 2 は、読み取りデータが D R A M デバイス 3 2 から D R A M データバス 3 0 を介してメモリコントローラ 1 2 に返される前に書き込みデータが D R A M デバイス 3 2 のメモリに書き込まれることを可能にするために、読み取りコマンドを遅延させる「書き込みから読み取りへのバスターンアラウンド (ライトツーリードバスターンアラウンド: write to read bus turnaround)」を実行するよう構成されている。「読み取りから書き込みへのバスターンアラウンド」操作及び「書き込みから読み取りへのバスターンアラウンド」操作の両方は、複数のバスサイクル、したがって

、D R A Mデータバス30上の帯域幅量を消費する。したがって、要求14を完了するのに必要とされるD R A Mデータバス30上で消費される帯域幅48の量は、メモリコントローラ12によりD R A Mデバイス32に発行されている現在の要求14の要求タイプ(すなわち、読み取り又は書き込み)と、最後に発行された要求14の要求タイプと、の両方に依存し得る。この帯域幅消費を最小にするために、メモリコントローラ12は、「読み取り/書き込みグループ化」を実行するよう構成され得る。「読み取り/書き込みグループ化」において、要求14は、「読み取りから書き込みへのバスターンアラウンド」操作及び「書き込みから読み取りへのバスターンアラウンド」操作の必要性を低減させるために、要求タイプに基づいてグループ化される。

【0017】

10

さらに、要求14が読み取り要求である場合、この要求14を発行するマスタ16は、D R A Mデバイス32にまだ書き込まれていないデータを要求している最中であり得る。データが書き込まれる前にこの要求14を発行するのを回避するために、要求14のページアドレスが、全ての優先順位キュー22内の全ての書き込み要求14のページアドレスと比較されなければならない。このプロセスは、アドレス競合解決(address conflict resolution)と呼ばれ、データ整合性を確実にするために使用され得る。合致がある場合、当該の読み取り要求14は、全ての前の書き込み要求14が完了するまで待機しなければならない。

【0018】

以下に示す表1は、要求されたページの位置、要求タイプ、及び、所与の要求14が生成された時点におけるD R A Mデバイス32の状態(すなわち、前に完了した要求タイプ)に基づく、所与の要求14のいくつかの一般的なケースの帯域幅コストの例を示している。

20



【表 1】

前の DRAM要求	現在のDRAM要求					
	同じページ		異なるページ			
	読み 取り	書き 込み	同じバンク		異なるバンク	
			読み 取り	書き 込み	読み取り	書き込み
読み取り	BL/2	BL/2 + tR2W	tPRE + tACT + BL/2	tR2W + tPRE + tACT + BL/2	<ul style="list-style-type: none"> <li>● ターゲットとされたページとは異なるページがオープン状態である場合: tPRE + tACT + BL/2</li> <li>● ターゲットとされたページがクローズ状態である場合: tACT + BL/2</li> <li>● ターゲットとされたページがオープン状態である場合: BL/2</li> </ul>	<ul style="list-style-type: none"> <li>● ターゲットとされたページとは異なるページがオープン状態である場合: tR2W + tPRE + tACT + BL/2</li> <li>● ターゲットとされたページがクローズ状態である場合: tACT + BL/2</li> <li>● ターゲットとされたページがオープン状態である場合: BL/2</li> </ul>
書き込み	BL/2 + tW2R	BL/2	tW2R + tPRE + tACT + BL/2	tPRE + tACT + BL/2	<ul style="list-style-type: none"> <li>● ターゲットとされたページとは異なるページがオープン状態である場合: tW2R + tPRE + tACT + BL/2</li> <li>● ターゲットとされたページがクローズ状態である場合: tACT + BL/2</li> <li>● ターゲットとされたページがオープン状態である場合: BL/2</li> </ul>	<ul style="list-style-type: none"> <li>● ターゲットとされたページとは異なるページがオープン状態である場合: tPRE + tACT + BL/2</li> <li>● ターゲットとされたページがクローズ状態である場合: tACT + BL/2</li> <li>● ターゲットとされたページがオープン状態である場合: BL/2</li> </ul>

表 1 : バスサイクルの観点での、1つの要求当たりの帯域幅コスト

【 0 0 1 9 】

上記表 1 において、変数 B L は、D R A M デバイス 3 2 のバースト長を示す。B L / 2

10

20

30

40

50

は、DDR<sub>n</sub>、低電力DDR<sub>n</sub>(LPDDR<sub>n</sub>)、又はグラフィックスDDR<sub>n</sub>(GDDR<sub>n</sub>)等の任意のダブルデータレートメモリデバイスについてDRAMデバイス32から読み取られるデータ又はDRAMデバイス32に書き込まれるデータを搬送するDRAMデータバス30がビジーとなる継続時間である。変数tACTは、ページをオープンする帯域幅コストであり、上述したようにページをアクティブ化するのに要する時間である。tPREは、ページをクローズする帯域幅コストであり、上述したようにオープン状態のページをクローズするのに要する時間である。tW2Rは、上述したように書き込みから読み取りへのバスターンアラウンド操作を実行することにより、書き込み操作に続く読み取り操作のためにDRAMデータバス30を遷移させるのに必要とされる、DRAMデータバス30のサイクルの数として表されるコストである。tR2Wは、上述したように読み取りから書き込みへのバスターンアラウンド操作を実行することにより、読み取り操作に続く書き込み操作のためにDRAMデータバス30を遷移させるのに必要とされる、DRAMデータバス30のサイクルの数として表されるコストである。重要なことは、異なるDRAMメモリデバイス(例えば、DDR3、LPDDR3、又はGDDR3)は、これらのパラメータについて異なる名前を使用することがあるが、本開示で説明するコンセプトは、上記DRAMデバイスのいずれにも適用されることに留意することである。以下の段落において詳細に説明するリフレッシュ操作は、表1には含まれていないことに留意されたい。なぜならば、リフレッシュ操作は、マスタ16のオーバーヘッドコストではなく、DRAMデバイス32のオーバーヘッドコストと考えられるからである。

#### 【0020】

表1は、システム10の1つの可能な実施形態を表している。システム10は、表1に列挙されるパラメータに限定されるものではなく、QoSスキームの最終目的に応じて、他の表が、所望の結果を達成するために本明細書で説明する原理に従って構成され得る。

#### 【0021】

メモリコントローラ12は、DRAMデータバス30上の帯域幅消費を最小にするために、上記方法に従って、優先順位キュー22内のDRAM要求14をスケジューリングするように構成され得る。各優先順位キュー22は、複数のキュー46を含み得る。各優先順位レベルについての優先順位キュー22の数及びサイズ、並びに、優先順位キュー22の各優先順位キュー内のキュー46の数は、実装に応じて変わり得る。図1においては、N個のマスタデバイス16及びM個の優先順位キュー22が示されている。所与の優先順位レベルについての優先順位キュー22の各優先順位キュー内のキュー46は、少なくとも1つの読み取りキューと1つの書き込みキューとを含み得る。メモリコントローラ12は、複数の優先順位キュー22の各優先順位キュー内の要求14をスケジューリングするように構成され得る。スケジュールは、マスタ依存であり、アドレス競合解決、同一ページ収集、読み取り/書き込みグループ化、及び、要求14の各要求に割り当てられた優先順位レベル等のパラメータに基づき得る。

#### 【0022】

メモリコントローラ12は、さらに、DRAM要求14をマスタスケジューリングマルチプレクサ24に送信し、各マスタ16に割り当てられた帯域幅の未使用部分を定期的にモニタリングし、帯域幅の未使用部分が予め定められたレベル以下に低下した場合にメモリコントローラ12におけるDRAM要求14を調整する(throttle)よう構成され得る。メモリコントローラ12は、上述したスケジューリングパラメータに基づいて、優先順位キュー22から最良の要求14の候補を選択した後、DRAM要求14をマスタスケジューリングマルチプレクサ24に送信するよう構成され得る。

#### 【0023】

マスタスケジューリングマルチプレクサ24は、DRAMデータバス30における予め定められた量の帯域幅を、複数のマスタ16の各マスタに割り当てるよう構成されている細粒度QoS帯域幅プロビジョニングロジック28を含む細粒度QoSロジック26を適用するよう構成され得る。細粒度QoSロジック26は、DRAMデバイス32の帯域幅を、有効(useful)QoS帯域幅及びオーバーヘッド帯域幅という2つのカテゴリに分割

するよう構成され得る。オーバーヘッド帯域幅は、D R A Mデバイス32の適切な動作を確実にするためにメモリコントローラ12により消費される帯域幅であり得、リフレッシュ操作について消費される帯域幅を含み得る。リフレッシュ操作中、メモリコントローラ12は、D R A Mデバイス32の全てのバンクへの全ての読み取り操作及び書き込み操作を中断し、全てのオープン状態のページをクローズし、D R A Mデバイス32内の全てのページのコンテンツをリフレッシュするためにリフレッシュコマンドを発行するよう構成されている。リフレッシュ操作が完了した後、メモリコントローラ12は、読み取り操作及び書き込み操作を再開するよう構成されている。有効QoS帯域幅は、D R A Mデバイス32に対して読み取り操作及び書き込み操作等の操作を実行するためにマスタ16により消費される帯域幅48を含み得る。メモリコントローラ12は、細粒度QoS帯域幅プロビジョニングロジック28により、予め定められた量の有効QoS帯域幅をマスタ16に割り当てるよう構成され得る。マスタ16の各マスタに割り当てられる帯域幅の量は、システム10上で動作するソフトウェアアプリケーションにより決定され得る。そのような割り当ては、例えば、システム10の起動のたびに1回、アプリケーションがシステム10上で実行されるときのアプリケーション境界上で、又は、システム10上の仮想環境の各起動時に、生じ得る。

#### 【0024】

メモリコントローラ12は、各要求14についてD R A Mデータバス30上で各マスタ16により消費される帯域幅48を算出し、細粒度QoS帯域幅プロビジョニングロジック28により各マスタ16に割り当てられた有効QoS帯域幅の量から、算出された帯域幅48を減算することにより、各マスタ16についての帯域幅の未使用部分を決定するよう構成され得る。メモリコントローラ12は、要求14を完了するのに必要とされる、D R A Mデータバス30のバスサイクルの数の観点で、要求14により消費される帯域幅48を測定するよう構成され得る。要求14を完了するのに必要とされるバスサイクルの数は、要求タイプ(すなわち、読み取り又は書き込み)、D R A Mデバイス32内の、要求14によりターゲットとされたページが、現在オープン状態であるか又はクローズ状態であるか、及び、要求14によりターゲットとされたページが、最後に発行された要求によりターゲットとされたページと同じバンク内に位置するか否か等の、要求14の様々な特性に応じて変わり得る。

#### 【0025】

メモリコントローラ12の細粒度QoSロジック26は、さらに、QoSクレジットアキュムレータ(QoS credit accumulator)36を含むよう構成され得る。QoSクレジットアキュムレータ36は、複数のマスタ16の各マスタについてのクレジットをインクリメンタルに累積するよう構成され得る。累積のレートは、細粒度QoS帯域幅プロビジョニングロジック28により各マスタに割り当てられた帯域幅に対応し得、各マスタ16についての総クレジットは、予め定められた最大QoSクレジット値を超えることができない。例えば、D R A Mデータバス30における毎秒1ギガビットの帯域幅が、所与のマスタ16に割り当てられ、D R A Mデータバス30が、200MHz(毎秒2000000000サイクル)のクロック周波数で動作している場合、QoSクレジットアキュムレータ36は、当該のマスタ16について指定される、1サイクル当たり5バイトに値するクレジットを累積する。マスタ16が、要求14を発行する場合、要求14により消費される帯域幅48は、QoSクレジットアキュムレータ36における、このマスタ16に対応するクレジットから減算される。マスタ16が、要求14を発行しない場合、QoSクレジットアキュムレータ36は、このマスタ16についての総クレジットが最大QoSクレジット値に達するまで、1サイクル当たり5バイトのレートでクレジットを累積し続ける。システム10は、システム10上で動作するソフトウェアアプリケーションが最大QoSクレジット値を決定することを可能にするよう構成され得る。

#### 【0026】

細粒度QoSロジック26は、D R A Mデータバス上で各D R A Mメモリ要求により消費される帯域幅48を算出し、QoSクレジットアキュムレータにおける、各マスタに対

10

20

30

40

50

応するクレジットから、算出された帯域幅を減算することにより、各マスタについてのクレジット残余 (credit balance) を算出するよう構成され得る細粒度 QoS チャージ勘定ロジック (fine-grained QoS charge accounting logic) 38 をさらに含むよう構成され得る。図 2 は、細粒度 QoS チャージ勘定ロジック 38 の動作に含まれ得るプロセスの概略図を示している。図示されるように、細粒度 QoS チャージ勘定ロジック 38 は、優先順位キュー 22 から、現在キューされている要求に関する、現在の要求の情報 50 を受信するよう構成され得る。現在の要求の情報 50 は、優先順位キュー 22 からマスタスケジューリングマルチプレクサ 24 に現在送信されている要求 14 によりターゲットとされたページのページアドレス及びバンク、並びに要求 14 の要求タイプ (すなわち、読み取り又は書き込み) を含み得る。加えて、細粒度 QoS チャージ勘定ロジック 38 は、さらに、DRAM プロトコルエンジン 44 から、最後に発行された要求の情報 52 を受信するよう構成され得る。最後に発行された要求の情報 52 は、DRAM デバイス 32 に最後に送信された要求 14 によりターゲットとされたページのページアドレス及びバンク、並びに要求 14 の要求タイプ (すなわち、読み取り又は書き込み) を含み得る。最後に発行された要求の情報 52 は、DRAM デバイス 32 に最後に発行された要求 14 が、ページヒットを有していたかページミスをしたか又はページ競合を有していたかをさらに含み得る。細粒度 QoS チャージ勘定ロジック 38 は、さらに、現在の要求の情報 50 及び最後に発行された要求の情報 52 の一方又は両方を使用して、要求 14 により DRAM データバス 30 上で消費される帯域幅 48 を算出し、QoS クレジットアキュムレータ 36 における、この要求 14 を発行したマスタに対応するクレジットから、算出された帯域幅の量を減算するよう構成され得る。細粒度 QoS チャージ勘定ロジック 38 はまた、DRAM プロトコルエンジン 44 から受信されたグローバルページステータス情報 58 を使用するよう構成され得る。グローバルページステータス情報 58 は、DRAM デバイス 32 内の全てのページの現在のステータスを含み得る。細粒度 QoS チャージ勘定ロジック 38 は、グローバルページステータス情報 58 を使用して、DRAM デバイス 32 に発行されるならば、どの要求 14 が、ページヒット、ページミス、又はページ競合を有することになるかを判定することができる。細粒度 QoS チャージ勘定ロジック 38 は、さらに、QoS クレジットアキュムレータ 36 における総クレジットから、要求 14 により消費される帯域幅 48 を減算する前に、リフレッシュにより消費されるグローバルなオーバーヘッド帯域幅 56 と、リフレッシュについての 1 クロックサイクル当たりのクレジットゲイン (credit gain) 54 と、を考慮に入れるよう構成され得る。システム 10 は、有効 QoS 帯域幅のみをマスタ 16 に割り当てるよう構成され得るので、細粒度 QoS チャージ勘定ロジック 38 は、リフレッシュ操作により消費される帯域幅又は他の形態のオーバーヘッド帯域幅消費を補償するために、1 クロックサイクル当たりのクレジットゲインを提供するよう構成され得る。QoS チャージ勘定ロジック 38 が、最後に発行された要求 14 により消費された帯域幅 48 を減算した後、細粒度 QoS クレジットアキュムレータ 36 は、さらに、マスタ 16 の各マスタについての、QoS クレジットアキュムレータ 36 における現在の残余を、以下で説明する QoS スケジューリングロジック 40 に送信するよう構成され得る。

【 0 0 2 7 】

図 1 に戻ると、細粒度 QoS ロジック 26 は、さらに、各 DRAM 要求の優先順位クラスに基づいて DRAM 要求 14 をスケジューリングし、DRAM 要求 14 を発行したマスタ 16 についての、QoS クレジットアキュムレータ 36 におけるクレジット残余が、予め定められた最小 QoS クレジット値未満である場合に、DRAM 要求 14 を調整するよう構成されている細粒度 QoS スケジューリングロジック 40 を含むよう構成され得る。細粒度 QoS スケジューリングロジック 40 は、細粒度 QoS チャージ勘定ロジック 38 から、キューされている要求 14 の各要求に対応する、マスタ 16 の各マスタについての現在のクレジット残余を受信するよう構成され得る。細粒度 QoS スケジューリングロジック 40 は、さらに、最高優先順位レベルと、予め定められた最小 QoS クレジット値以上のクレジット残余と、を有する要求 14 を選択するよう構成され得る。要求 14 が、最

小QoSクレジット値未満のクレジット残余を有する場合、細粒度QoSスケジューリングロジック40は、その要求14の優先順位レベルにかかわらず、その要求14を調整するよう構成され得る。複数の要求14が、同じ優先順位レベルと、最小QoSクレジット値以上のクレジット残余と、を有する場合、スケジューリングは、ラウンドロビン方式で、又は、別の適切なデフォルトスケジューリングアルゴリズムに従って、実行され得る。全ての利用可能な要求14が、最小QoSクレジット値未満のクレジット残余を有する場合、最高優先順位レベルを有する要求14が選択され得る。システム10は、システム10上で動作するソフトウェアアプリケーションが最小QoSクレジット値を決定することを可能にするよう構成され得る。

#### 【0028】

メモリコントローラ12は、さらに、マスタスケジューリングマルチプレクサ24からDRAM要求14を受信し、DRAMコマンドバス34を介してDRAM要求14をDRAMデバイス32に送信するよう構成されているDRAMプロトコルエンジン44を含むよう構成され得る。DRAMプロトコルエンジン44は、細粒度QoSロジック26により作成されたスケジュールに基づいて、マスタスケジューリングマルチプレクサ24により選択されたDRAM要求14を受信するよう構成され得る。メモリコントローラ12は、さらに、DRAM要求14をDRAMプロトコルエンジン44に送信する前に、DRAM要求14をDRAMコマンド42に変換するよう構成され得る。DRAMプロトコルエンジン44は、さらに、DRAMデバイス32から応答を受信し、最後に発行された要求14に関する情報を細粒度QoSチャージ勘定ロジック38に送信するよう構成され得る。任意的に、DRAMプロトコルエンジン44は、DRAM要求14が成功裡に完了したことをメモリコントローラ12に通知するための割り込みの形態であり得る応答を、DRAMデバイス32から受信するよう構成されてもよい。DRAMプロトコルエンジン44は、さらに、図2に示され説明した最後に発行された要求の情報52を細粒度QoSチャージ勘定ロジック38に送信するよう構成され得る。この情報は、要求14の要求タイプ（すなわち、読み取り又は書き込み）、要求14によりターゲットとされたページのページアドレス及びバンク、並びに、要求14がページヒットを有していたかページミス（ページ競合）を有していたかを含み得る。

#### 【0029】

図3は、細粒度QoSロジックを実行して、DRAMデータバスの帯域幅をプロビジョニングするよう構成されている第2の実施形態のシステム100を示している。システム100は、システムオンチップ（SOC）101を含み、マスタ16、バスファブリック18、及びDRAMメモリコントローラ12が、SOC101上に形成されており、DRAMメモリコントローラ12が、DRAMデバイス32と通信する。DRAMデバイス32は、SOC101上に形成されることもあるし、独立したデバイスとして形成されることもある。システムオンチップ上に形成されている様々なコンポーネントは、上述したものと同様に機能し、簡潔さのために再度説明しない。上述したように、この実施形態においては、バスファブリックは、速度リミッタを含むバスセグメント60及び速度リミッタを含まないバスセグメント62を含む複数のバスセグメントに分割されている。システムオンチップ101は、バスファブリック上の最少固定量の帯域幅が各マスタのために予約されることを確実にしつつ、各マスタにより消費される最大帯域幅を制限することにより、各マスタがこの固定量を超えないことを確実にするために、速度リミッタを利用することが理解されよう。これは、スケジューリング競合を回避させ、速度制限されたマスタからDRAMデバイスへの効率的なアクセスを促進させる。バスファブリックのうち速度制限されていないバスセグメント（例えば、バスセグメントM）のみを経由する新たなマスタが、このようなシステムに追加される場合、このマスタは、バスファブリック上の過度の帯域幅を消費し得る、且つ/又は、固定量の予約された帯域幅を有するという予測の下で設計された他のマスタの性能を劣化させ得る。したがって、速度リミッタは、何らかのスケジューリング制御を実施するが、バスセグメントM上には速度リミッタが構成されない。特に速度制限されていないバスセグメントMを通過するDRAM要求は、スケジ

10

20

30

40

50

ューリングマルチプレクサ24により適用される細粒度QoSスケジューリング及び優先順位レベルから恩恵を受けることができ、これは、バスセグメントMを介して接続されるマスタによる予約された帯域幅の想定外の消費を防止する。これは、全てのマスタの性能を向上させることができる。

#### 【0030】

次に図4に移ると、メモリコントローラにおいて細粒度帯域幅プロビジョニングを実行するための方法400のフローチャートが示されている。以下に記載の方法は、図1及び図2を参照して上述したシステム10のハードウェア上で実行されてもよいし、他の適切なハードウェア上で実行されてもよい。本明細書に記載の方法が実行され得る適切なハードウェアは、ラップトップパーソナルコンピュータ、デスクトップパーソナルコンピュータ、スマートフォン、ビデオゲームコンソール、タブレットコンピューティングデバイス、スマートテレビジョン等を含むことが理解されよう。本方法のいくつかの実装においては、図4における破線内のステップは省かれてもよいが、他の実装においては、破線内のステップが含まれる。

10

#### 【0031】

方法400は、402において、複数のマスタからバスファブリックを介してダイナミックランダムアクセスメモリ(DRAM)要求を受信することを含み得る。404において、方法400は、DRAM要求の各DRAM要求に割り当てられる優先順位レベルを決定することを含み得る。

#### 【0032】

502において、方法400は、この優先順位レベルに基づいて、複数の優先順位キューのうちの1つの優先順位キューに各DRAM要求を割り当ててることを含み得る。

20

#### 【0033】

506において、方法400は、DRAM要求をマスタスケジューリングマルチプレクサに送信することを含み得る。

#### 【0034】

406において、方法400は、DRAMデータバス上で各マスタにより消費されるDRAMデータバス帯域幅の量を決定することを含み得る。508において、これは、各マスタに割り当てられた帯域幅の未使用部分を定期的にモニタリングすることにより少なくとも部分的に達成され得ることが示されている。

30

#### 【0035】

408において、方法400は、各マスタに割り当てられた優先順位クラスと、各マスタにより消費されるDRAMデータバス帯域幅の量と、に基づいて、細粒度サービス品質(QoS)ロジックを適用して、DRAM要求のスケジュールを算出することを含み得る。510において、このスケジュールは、帯域幅の未使用部分が予め定められたレベル以下に低下した場合にメモリコントローラにおけるDRAM要求を調整するように算出され得ることが示されている。

#### 【0036】

410において、方法400は、ステップ408において作成されたスケジュールに基づいて、DRAM要求をDRAMコマンドに変換することを含み得る。412において、方法400は、DRAMコマンドバスを介してDRAMコマンドを少なくとも1つのDRAMデバイスに送信することを含み得る。414において、方法400は、少なくとも1つのDRAMデバイスからDRAMデータバスを介して応答を受信することを含み得る。

40

#### 【0037】

図5は、図4の方法400のステップ508、ステップ510、及びステップ412を達成することができる例示的なサブステップを例示しているフローチャートを示している。方法400のステップ508は、サブステップ602~608を実行することにより達成され得る。602において、方法は、細粒度QoS帯域幅プロビジョニングロジックを適用して、複数のマスタの各マスタに、DRAMデータバスにおける予め定められた量の帯域幅を割り当ててることを含み得る。604において、方法は、QoSクレジットアキュ

50

ムレータを適用して、複数のマスタの各マスタについてのクレジットをインクリメンタルに累積することを含み得る。ここで、累積のレートは、細粒度QoS帯域幅プロビジョニングロジックにより各マスタに割り当てられた帯域幅に対応し、各マスタについての総クレジットは、予め定められた最大QoSクレジット値を超えない。606において、方法は、細粒度QoSチャージ勘定ロジックを適用して、各DRAMメモリ要求により消費されるDRAMデータバス上の帯域幅を算出することを含み得る。608において、方法は、QoSクレジットアキュムレータにおける、各マスタに対応するクレジットから、算出された帯域幅48を減算することにより、各マスタについてのクレジット残余を算出することを含み得る。

【0038】

10

方法400のステップ510は、サブステップ610により達成され得る。サブステップ610に示されるように、細粒度スケジューリングロジックを適用すること、及び、帯域幅の未使用部分が予め定められたレベル以下に低下した場合にメモリコントローラにおけるDRAM要求を調整することは、細粒度QoSスケジューリングロジックを適用して、各要求の優先順位クラスに基づいてDRAM要求をスケジューリングし、DRAM要求を発行したマスタについての、QoSクレジットアキュムレータにおけるクレジット残余が、予め定められた最小QoSクレジット値未満である場合にDRAM要求を調整するようにスケジュールを算出することにより、達成される。

【0039】

方法400のステップ412は、サブステップ612～614により達成され得る。612において、方法は、マスタスケジューリングマルチプレクサからDRAMプロトコルエンジンにDRAM要求を送信することを含み得る。614において、方法は、DRAMプロトコルエンジンから複数のDRAMデバイスのうちの少なくとも1つのDRAMデバイスにDRAM要求を送信することを含み得る。

20

【0040】

図6は、図4の方法400のステップ406を達成することができる例示的なサブステップ650、652を例示しているフローチャートを示している。650において、方法は、DRAMプロトコルエンジンから、最後に発行された要求に関する情報を受信することを含み得る。652において、方法は、最後に発行された要求に関する情報に基づいて、各DRAMメモリ要求により消費されるDRAMデータバス上の帯域幅を算出すること

30

【0041】

一時的に図1に戻ると、システム10は、システム10よりも少ない総メモリ帯域幅を有する異なるシステム上で動作するよう元々設計された1以上のソフトウェアプログラムの性能をモデル化するためにも使用され得る。そうするために、マスタ16からの要求14の一部が、こうしたソフトウェアプログラムの実行から生じたものとしてタグ付けされ得る。メモリコントローラ12は、こうしたソフトウェアプログラムが、元々設計された対象のシステム上で動作しているならばマスタ16が受けたであろう帯域幅及びQoSを算出することができる。次いで、メモリコントローラ12は、この算出を用いて、帯域幅及びQoSを、システム10上の要求14に割り当てることができる。したがって、コンピューティングデバイスは、さらに、複数のマスタからのメモリ要求の一部を、このコンピューティングデバイスよりも少ない総メモリ帯域幅を有するシステム上で動作するよう元々設計された1以上のソフトウェアプログラムの実行から生じたものとしてタグ付けし、1以上のソフトウェアプログラムが、より少ない総メモリ帯域幅を有する上記システム上で動作しているならば複数のマスタが受けたであろう帯域幅及びQoSを算出するよう構成され得る。さらに、コンピューティングデバイスは、これら帯域幅及びQoSを、複数のマスタからのメモリ要求に割り当て、結果として、より少ない総メモリ帯域幅を有する上記システム上での1以上のソフトウェアプログラムの性能をモデル化するよう構成され得る。

40

【0042】

50

上述したシステム及び方法は、種々の有用な産業上の用途を有する。例えば、第1の用途は、より高速なメモリタイプが、より低速なメモリタイプと同じ帯域幅変動をもたらすことを可能にすることである。例えば、システムは、例えば、2133Mbpsのクロック周波数で動作するダブルデータレート3（DDR3）メモリを使用するように設計されたビデオゲームコンソールであり得る。ビデオゲームコンソールのライフサイクル中に、このメモリを、例えば、2400Mbpsのクロック周波数で動作するダブルデータレート4（DDR4）メモリに変更することが、（例えば、性能又はコスト上の理由で）有利なことがある。ビデオゲームコンソールのライフサイクル中にそのような変更がなされると、DDR3とともに使用されるよう設計されたソフトウェアが、DDR4とともに適切に動作しないリスクが存在するようになる。しかしながら、上述した細粒度QoSロジックは、DDR4に対するプロビジョニングが、DDR3に対するプロビジョニングと同じであるように、DRAMデータバス上の帯域幅の量が各マスタデバイスにプロビジョニングされることを可能にし得る。したがって、細粒度QoSロジックは、DDR4等の新たな世代のメモリデバイスの帯域幅変動が、DDR3等の前の世代のメモリデバイスと実質的に同様であるように、新たな世代のメモリデバイスを制限し、そうする際、さらなる帯域幅変動を受けることなく、前の世代のメモリデバイス用に設計されたソフトウェアが、後の世代のメモリデバイスを使用するシステム上で機能することを可能にし得る。

10

#### 【0043】

上述したシステム及び方法の別の例示的なユースケースシナリオは、さらなるマスタデバイスがシステムに追加されたときにマスタデバイスが受ける帯域幅変動の量を低減させることである。システムが、例えば、ビデオゲームコンソールである場合、ビデオゲームコンソールが発売され、ソフトウェアがそのビデオゲームコンソール用に設計された後であっても、さらなるマスタが追加されることがある。さらなるマスタデバイスは、さらなる帯域幅変動を望ましくない形でシステムにもたらすことがある。しかしながら、細粒度QoSロジックを使用して、マスタデバイスを分離させて、上述した細粒度QoSパラメータ及び優先順位クラスに基づいて、論理的方法で、マスタデバイスから生成された要求をスケジューリングすることができ、それにより、そうでなければ新たなマスタデバイスを追加することにより生じたであろうさらなる帯域幅変動からシステムを保護することができる。これを達成するために、マスタデバイスに、DRAMデータバス上の固定量の帯域幅を割り当てることができ、全てのマスタデバイスをサポートするのに十分な帯域幅がある限り、各マスタに割り当てられた帯域幅は、新たなマスタが追加されたときにも、変更されないままである。これは、さらなるマスタが追加される前と同じようにマスタデバイスが挙動することを確実にするために必要とされるテストの量の著しい低減をもたらす。そのような用途において、細粒度QoSロジックを使用して、システム内のマスタデバイスの各マスタデバイスについて固定量の帯域幅を予約し、優先順位クラス及び細粒度QoS技術を使用して、既存のマスタの固定量の予約された帯域幅と干渉することから、新たなマスタにより消費される帯域幅を制限する。

20

30

#### 【0044】

第3の例は、上述したシステム及び方法を使用して、物理DRAMデバイスを、多くの論理的に独立したDRAMデバイスにパーティショニングすることである。マスタデバイスを、これらの論理的に独立したDRAMデバイスに割り当てることができ、それにより、他のパーティションに割り当てられた他のマスタデバイスにより引き起こされる帯域幅変動から、マスタデバイスを保護することができる。そのようなパーティションが作成されると、他のパーティション上のマスタデバイスを妨げることなく、マスタデバイスを追加及び除去することができる。

40

#### 【0045】

DRAMメモリデバイスのコンテキストにおいて上述したシステム及び方法は、スタティックランダムアクセスメモリ（SRAM）、フラッシュメモリ、磁気ディスクメモリ、テープメモリ、並びに、他の種類の揮発性メモリデバイス及び不揮発性メモリデバイスを含む他のタイプのメモリデバイスに適用されてもよいことが理解されよう。したがって、

50



上述したDRAMデータバスは、メモリデータバスとすることができ、上述したDRAMメモリ要求は、メモリ要求とすることができ、上述したDRAMコマンドは、メモリコマンドとすることができ、上述したDRAMコマンドバスは、メモリコマンドバスとすることができる。これらの様々なメモリコンポーネントは、この段落において上記で列挙されたメモリデバイス等の特定のタイプのメモリデバイスとともに動作するよう構成される。

【0046】

図7は、上述したシステム及び方法を実装することができるコンピューティング環境を示している。図7に示されるように、コンピューティングシステム700は、プロセッサ720と不揮発性メモリ722と揮発性メモリ724とを有するホストシステム718を含み得るだけでなく、表示サブシステム706、通信サブシステム708、及び入力サブシステム710を含み得、これらは全て、通信バスサブシステム702を介してリンクされる。

10

【0047】

システム10は、1以上のパーソナルコンピュータ、サーバコンピュータ、タブレットコンピュータ、ホームエンターテイメントコンピュータ（ケーブルセットトップボックス、デジタルビデオレコーダ、ビデオサーバ等）、ネットワークコンピューティングデバイス、ゲームデバイス（ゲームコンソール等）、モバイルコンピューティングデバイス、モバイル通信デバイス（例えば、スマートフォン）、及び/又は他のコンピューティングデバイスの形態をとることができる。

【0048】

20

ホストシステム718のプロセッサ720は、不揮発性メモリ722及び揮発性メモリ724の諸部分を使用して命令を実行するよう構成される物理デバイスである。例えば、プロセッサは、1以上のアプリケーション、サービス、プログラム、ルーチン、ライブラリ、オブジェクト、コンポーネント、データ構造、又は他の論理的構造体の一部である命令を実行するよう構成され得る。そのような命令は、タスクを実行するために、データ型を実装するために、1以上のコンポーネントの状態を変換するために、技術的效果を達成するために、又は所望の結果に達するために、実装され得る。

【0049】

不揮発性メモリ722は、プロセッサ720により実行可能な命令を保持するよう構成される1以上の物理デバイスを含む。不揮発性メモリ722は、フラッシュ、EPROM、EEPROM等といった不揮発性メモリを含み得、処理中に有限の期間の間命令を保持し、コンピューティングシステムに電力が供給されていないときでも、命令を保持し続ける。揮発性メモリ724は、上述したDRAMデバイス32等のDRAMや、処理中に有限の期間の間命令を保持するが、通常は、コンピューティングシステムに電力が供給されなくなった後には命令を記憶し続けることはないRAMを含み得る。

30

【0050】

プロセッサ720、永続的メモリ722、及び揮発性メモリ724の態様は、1以上のハードウェアロジックコンポーネントと一緒に統合されてもよい。そのようなハードウェアロジックコンポーネントは、例えば、フィールドプログラマブルゲートアレイ（FPGA）、特定用途プログラム向け集積回路/特定用途向け集積回路（ASIC/ASIC）、特定プログラム向け標準品/特定用途向け標準品（SSP/ASSP）、システムオンチップ（SOC）、及びコンプレックスプログラマブルロジックデバイス（CPLD）を含み得る。

40

【0051】

表示サブシステム706は、永続的メモリ722又は揮発性メモリ724により保持されているデータの視覚的表現を提示するために使用され得る。この視覚的表現は、グラフィカルユーザインタフェース（GUI）の形態をとることができる。本明細書に記載の方法及びプロセスは、不揮発性メモリ722又は揮発性メモリ724により保持されているデータを変化させ、したがって、不揮発性メモリ722又は揮発性メモリ724の状態を変換させるので、表示サブシステム706の状態も同様に、元となるデータの変化を視覚

50

的に表現するように変換され得る。表示サブシステム 706 は、実質的に任意のタイプの技術を利用する 1 以上の表示デバイスを含み得る。そのような表示デバイスは、共有筐体において、プロセッサ 720 及び / 又は不揮発性メモリ 722 若しくは揮発性メモリ 724 と結合されてよいし、そのような表示デバイスは、周辺表示デバイスであってもよい。

【0052】

入力サブシステム 710 は、キーボード、マウス、タッチスクリーン、又はゲームコントローラ等の 1 以上のユーザ入力デバイスを含んでもよいし、このような 1 以上のユーザ入力デバイスとインタフェースをとってもよい。いくつかの実施形態において、入力サブシステムは、選択されたナチュラルユーザ入力 (NUI) コンポーネントを含んでもよいし、選択された NUI コンポーネントとインタフェースをとってもよい。そのようなコンポーネントは、統合されることもあるしペリフェラルであることもあり、入力アクションの変換及び / 又は処理は、オンボード又はオフボードで扱われ得る。例示的な NUI コンポーネントは、音声認識及び / 又は発話認識のためのマイクロフォン；マシンビジョン及び / 又はジェスチャ認識のための赤外線カメラ、カラーカメラ、立体視カメラ、及び / 又はデプスカメラ；動き検出及び / 又は意図認識のためのヘッドトラッカ、アイトラッカ、加速度計、及び / 又はジャイロスコープ；及び、脳活動を評価するための電場検知コンポーネントを含み得る。

【0053】

通信サブシステム 708 は、システム 10 を 1 以上の他のコンピューティングデバイスに通信可能に接続するよう構成され得る。通信サブシステム 708 は、1 以上の異なる通信プロトコルに適合する有線通信デバイス及び / 又は無線通信デバイスを含み得る。非限定的な例として、通信サブシステム 708 は、無線電話網、又は、有線ローカルエリアネットワーク、有線ワイドエリアネットワーク、無線ローカルエリアネットワーク、若しくは無線ワイドエリアネットワークを介する通信用に構成され得る。いくつかの実施形態において、通信サブシステムは、コンピューティングシステム 700 が、インターネット等のネットワークを介して他のデバイスとの間でメッセージを送信及び / 又は受信することを可能にし得る。

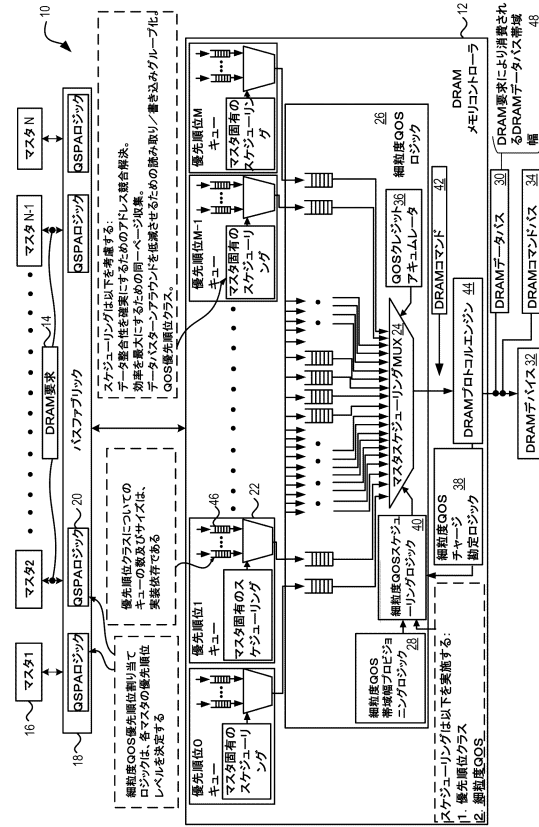
【0054】

多数の変形が可能であるので、本明細書に記載の構成及び / 又はアプローチは、本質的に例示であり、そのような特定の実施形態又は例は、限定とみなされるべきではないことが理解されよう。本明細書に記載の特定のルーチン又は方法は、任意の数の処理方策のうちの 1 以上を表し得る。したがって、図示及び / 又は説明した様々な動作は、図示及び / 又は説明した順番で実行されることもあるし、他の順番で実行されることもあるし、並列に実行されることもあるし、省かれることもある。同様に、上述したプロセスの順番も、変更されることがある。

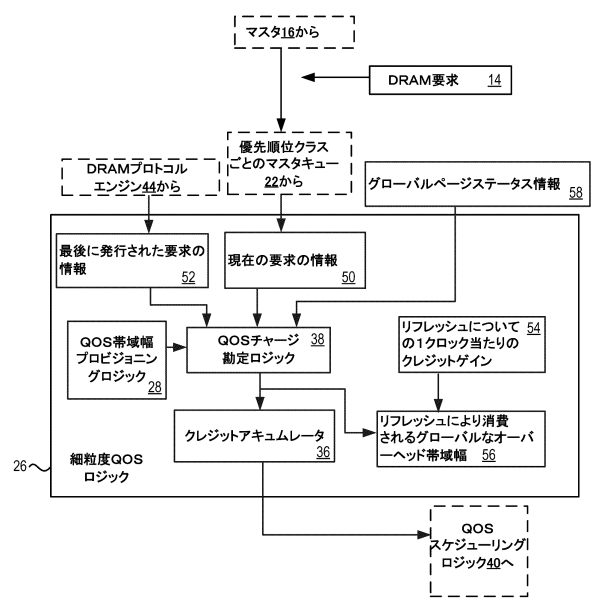
【0055】

本開示の主題は、様々なプロセス、システム、及び構成、並びに、本明細書で開示した他の特徴、機能、動作、及び / 又は特性の全ての新規で非自明なコンビネーション及びサブコンビネーションに加えて、それらのありとあらゆる均等なものを含む。

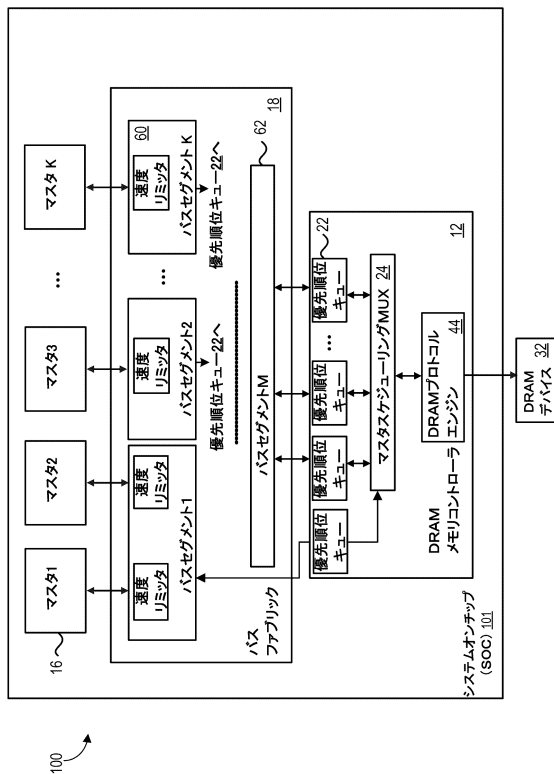
【図1】



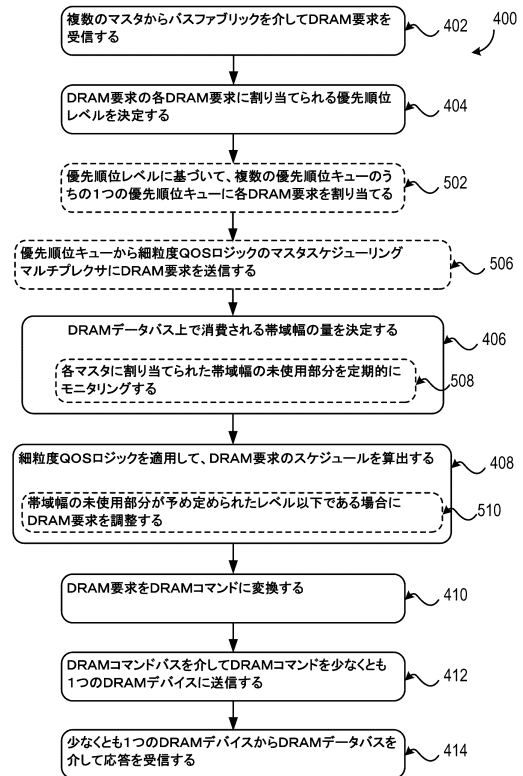
【図2】



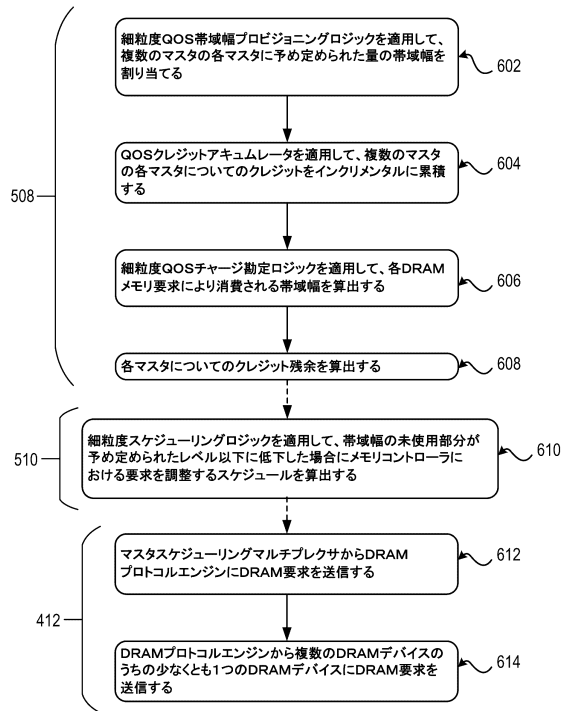
【図3】



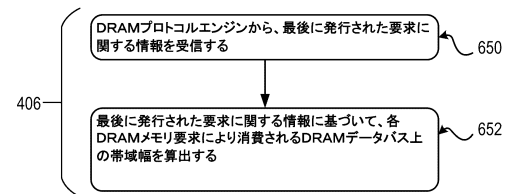
【図4】



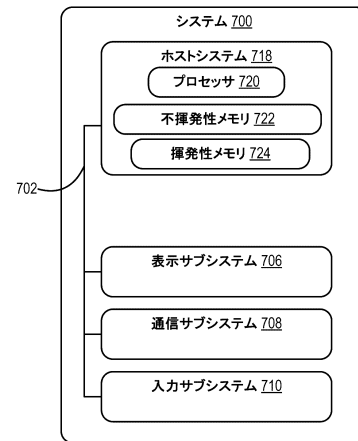
【図 5】



【図 6】



【図 7】



## フロントページの続き

- (72)発明者 コーク, ノン, トアイ  
アメリカ合衆国 98052-6399 ワシントン州 レッドモンド ワン マイクロソフト  
ウェイ マイクロソフト テクノロジー ライセンシング, エルエルシー エルシーイー - インタ  
ーナショナル パテンツ (8/1172) 内
- (72)発明者 カリー, スーザン  
アメリカ合衆国 98052-6399 ワシントン州 レッドモンド ワン マイクロソフト  
ウェイ マイクロソフト テクノロジー ライセンシング, エルエルシー エルシーイー - インタ  
ーナショナル パテンツ (8/1172) 内
- (72)発明者 アンドリュース, ジェフリー  
アメリカ合衆国 98052-6399 ワシントン州 レッドモンド ワン マイクロソフト  
ウェイ マイクロソフト テクノロジー ライセンシング, エルエルシー エルシーイー - インタ  
ーナショナル パテンツ (8/1172) 内
- (72)発明者 セル, ジョン  
アメリカ合衆国 98052-6399 ワシントン州 レッドモンド ワン マイクロソフト  
ウェイ マイクロソフト テクノロジー ライセンシング, エルエルシー エルシーイー - インタ  
ーナショナル パテンツ (8/1172) 内
- (72)発明者 ボー, ケヴィン  
アメリカ合衆国 98052-6399 ワシントン州 レッドモンド ワン マイクロソフト  
ウェイ マイクロソフト テクノロジー ライセンシング, エルエルシー エルシーイー - インタ  
ーナショナル パテンツ (8/1172) 内

審査官 篠塚 隆

- (56)参考文献 特表2005-505857(JP, A)  
特表2005-508550(JP, A)  
米国特許出願公開第2007/0113023(US, A1)

- (58)調査した分野(Int.Cl., DB名)  
G06F 12/00  
G06F 13/16  
G06F 12/06