(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号

特許第4516395号
(P4516395)

(45)発行日　平成22年8月4日(2010.8.4)　(24)登録日　平成22年5月21日(2010.5.21)

(51)Int.Cl.		F I		
G06F 13/38	(2006.01)	G06F	13/38	310D
G06F 12/02	(2006.01)	G06F	12/02	540
G06F 12/06	(2006.01)	G06F	12/06	522C
H04L 13/08	(2006.01)	G06F	12/06	550B
		H04L	13/08	

請求項の数 10　外国語出願　(全 37 頁)

(21)出願番号　特願2004-286229 (P2004-286229)
(22)出願日　平成16年9月30日 (2004.9.30)
(65)公開番号　特開2005-174286 (P2005-174286A)
(43)公開日　平成17年6月30日 (2005.6.30)
審査請求日　平成19年9月28日 (2007.9.28)
(31)優先権主張番号　10/699315
(32)優先日　平成15年10月31日 (2003.10.31)
(33)優先権主張国　米国(US)

(73)特許権者　596092698
アルカテル－ルーセント　ユーエスエー　インコーポレーテッド
アメリカ合衆国　０７９７４　ニュージャーシィ，マレイ　ヒル，マウンテン　アヴェニュー　６００－７００
(74)代理人　100064447
弁理士　岡部　正夫
(74)代理人　100085176
弁理士　加藤　伸晃
(74)代理人　100106703
弁理士　産形　和央
(74)代理人　100096943
弁理士　臼井　伸一

最終頁に続く

(54)【発明の名称】リンク・リスト・プロセッサを持つメモリ管理システム

(57)【特許請求の範囲】

【請求項１】

リンク・リスト・データ・ファイルを処理するように適合されたメモリ管理システムを操作する方法であって、前記システムは複数の高速小容量メモリおよび低速大容量メモリを備え、前記高速メモリは第１のデータ転送速度を持ち、前記大容量メモリは前記第１のデータ転送速度よりも遅い第２のデータ転送速度を持ち、前記システムはさらに、前記メモリによりリンク・リストの読み書きの要求を生成するためのアクセス・フロー・レギュレータを備え、該方法が、

前記アクセス・フロー・レギュレータから前記高速メモリに書き込み要求を送信することにより、前記高速メモリへのリンク・リストの書き込みを開始するステップ、

前記リンク・リストの先頭バッファおよび末尾バッファおよび少なくとも１つの中間バッファを前記高速メモリに書き込むステップ、及び

前記高速メモリから前記大容量メモリに前記少なくとも１つの中間バッファを転送する一方で、先頭バッファおよび末尾バッファを前記高速メモリ内に残すステップ

からなる方法。

【請求項２】

リンク・リスト・データ・ファイルを処理するように適合されたメモリ管理システムを操作する方法であって、前記システムは複数の高速小容量メモリおよび低速大容量メモリを備え、前記高速メモリは第１のデータ転送速度を持ち、前記大容量メモリは前記第１のデータ転送速度よりも遅い第２のデータ転送速度を持ち、前記システムはさらに、前記メ

モリによりリンク・リストの読み書きの要求を生成するためのアクセス・フロー・レギュレータを備え、
指定されたリンク・リストの読み出し要求を前記アクセス・フロー・レギュレータから前記指定されたリンク・リストのバッファを含む前記高速メモリに送信するステップ、
前記指定されたリンク・リストの先頭バッファを読み込むステップ、
前記リンク・リストの前記少なくとも１つの中間バッファを前記大容量メモリから前記高速メモリのうちの１つに転送するステップ、
前記１つの高速メモリに転送された前記中間バッファを前記指定されたリンク・リストの交換先頭バッファとして指定するステップ、
前記１つの高速メモリから前記指定されたリンク・リストの前記中間バッファを読み出すステップ、及び
前記指定されたリンク・リストの前記読み出されたバッファを前記アクセス・フロー・レギュレータに送信するステップ
からなる方法。
【請求項３】
請求項１又は２の方法であって、
前記システムを前記アクセス・フロー・レギュレータからの複数の要求に対するリンク・リストを同時処理するように動作させるステップ、
前記システムを前記複数の高速メモリのうちの異なるメモリに格納されているリンク・リストのバッファを処理するように動作させるステップ、
前記システムを、末尾バッファを新しいリンク・リストに書き込まれた第１のバッファとして書き込むように動作させるステップ、及び
リンク・リストの先頭バッファを最初に読み込むステップ
をさらに含む方法。
【請求項４】
請求項１又は２の方法において、前記システムはさらに、複数の状態コントローラ（１８０４）を備え、それぞれの状態コントローラは前記複数の高速メモリのうちの対応するメモリに対する個別のコントローラであり、前記システムはさらに、前記アクセス・フロー・レギュレータを前記状態コントローラと接続する要求バス（１８０２）を備え、読み出し要求を送信する前記ステップが、
前記アクセス・フロー・レギュレータを動作させ、前記読み出し要求を受信すべきアイドル状態の高速メモリを選択するステップ、
前記読み出し要求を前記アクセス・フロー・レギュレータから前記要求バスを介して前記選択された高速メモリに対して個別の状態コントローラに送信するステップ、
前記状態コントローラを、前記選択された高速メモリの現在の占有レベルを判別するように動作させるステップ、
前記現在の占有レベルが所定のレベル以下の場合に前記要求を前記高速メモリに送信するステップ、及び
前記選択された高速メモリの前記現在の占有レベルが前記所定のレベルを超えた場合に前記大容量メモリへの接続を要求するステップ
を含み、
前記システムがさらに、バックグラウンド・アクセス・マルチプレクサ、および前記状態コントローラを前記マルチプレクサと接続するアクセス・バスを備え、前記システムがさらに、前記マルチプレクサを前記大容量メモリと接続するバスを備え、前記方法が、前記マルチプレクサを、
前記大容量メモリへの接続の要求を前記状態コントローラから受信し、
複数の要求状態コントローラのどれかに対し前記大容量メモリへのアクセスを許可するかを決定するように動作させるステップ、
前記１つの要求状態コントローラを前記大容量メモリに接続するステップ、
前記１つの高速メモリから前記大容量メモリにデータを転送する際の前記大容量メモリ

のオペレーションを制御するステップ、及び
前記アクセス・バスを介して前記状態コントローラから前記マルチプレクサに前記リンク・リストのバッファを転送するステップを含む方法。
【請求項５】
請求項１又は２の方法において、前記大容量メモリから前記バッファを転送する前記ステップが、
リンク・リストの中間バッファを前記大容量メモリから前記高速メモリに、前記高速メモリのデータ転送速度に実質的に等しいデータ転送速度を持つバースト・モードで転送するステップ、
前記読み出されたバッファを前記高速メモリに格納するステップ、
その後、前記高速メモリから前記リンク・リストのバッファを読み出し、前記アクセス・フロー・レギュレータに転送するステップ、
バッファを既存のリンク・リストに書き込むのに、前記高速メモリから前記大容量メモリに前記既存のリンク・リスト既存の末尾を転送することにより書き込むステップ、及び
新しいバッファを前記既存のリンク・リストの新しい末尾バッファとして前記高速メモリに書き込むステップ
を含む方法。
【請求項６】
請求項４記載の方法において、前記状態コントローラを動作させる前記ステップが、さらに、
複数のリンク・リストのバッファを同時に受け取るステップ、
前記アクセス・フロー・レギュレータに送られる複数のリンク・リストのバッファを分離するステップ、
前記アクセス・フロー・レギュレータによって受け取られた複数のアクセスを前記高速メモリに送るステップ、
前記アクセス・フロー・レギュレータからそれぞれの受信された要求に応答し、前記状態コントローラに個別の高速メモリの現在の占有レベルを判別するステップ、
前記現在の占有レベル以下の場合、前記関連する高速メモリに前記アクセスを送るステップ、
前記現在の占有レベルを超えた場合、前記要求をバッファリングするよう前記アクセス・フロー・レギュレータに信号を送るステップ、
前記高速メモリからバッファをバースト・モードで前記大容量メモリに転送するのを制御するステップ、
前記大容量メモリからバッファを前記高速メモリに転送するのを制御するステップ、
転送が要求されたときに前記大容量メモリがアイドル状態かどうか判別するステップ、
アイドル状態の場合前記バッファを前記大容量メモリに送るステップ、及び
前記大容量メモリが使用中の場合、前記転送をバッファリングするステップ
を含む方法。
【請求項７】
請求項４の方法において、前記マルチプレクサを動作させる前記ステップが、さらに、
複数のビッディング高速メモリのどれに対し前記大容量メモリへのアクセスを許可するかを決定するステップ、
他のバッファの要求を前記１つの高速メモリにバッファリングするステップ、
前記大容量メモリからのバッファの転送先の高速メモリに対するＩＤを決定するステップ、及び
前記大容量メモリから前記識別された高速メモリにバースト・モードで前記バッファを転送するのを制御するステップ
を含む方法。
【請求項８】
請求項１又は２の方法であって、さらに、

それぞれの前記高速メモリの使用中／アイドル状態を示す、それぞれの高速メモリに固有の信号を発生するステップ、
それぞれの発生した信号を前記アクセス・フロー・レギュレータに送るステップ、
前記アクセス・フロー・レギュレータを、前記高速メモリによるリンク・リストの読み書きの要求を受け取るように動作させるステップ、
要求を受け取ったことに対して応答して前記アクセス・フロー・レギュレータを動作させることにより、前記高速メモリにより生成される前記使用中／アイドル状態信号を読み出すステップ、
前記読み出しに応答して前記アクセス・フロー・レギュレータを、前記いくつかの高速メモリのうちアイドル状態のメモリを識別するように動作させるステップ、
前記アクセス・フロー・レギュレータを、データ・ファイルの読み書きの要求を前記アイドル状態の１つの高速メモリに送るように動作させるステップ
からなる方法。

【請求項９】
リンク・リストのデータ・ファイルを処理するように適合されているメモリ管理システム（１８００）であって、前記システムが、
複数の高速小容量メモリ（１８０３－１、１８０３－２、１８０３－３、１８０３－４、１８０３－５、１８０３－６、１８０３－７、１８０３－８）および低速大容量メモリ（８０６－１、８０６－２、８０６－３）であって、前記高速メモリは第１のデータ転送速度を持ち、前記大容量メモリは前記第１のデータ転送速度よりも遅い第２のデータ転送速度を持つメモリ、
前記メモリ（１８０３、１８０６）によるリンク・リストの読み書きの要求を生成するためのアクセス・フロー・レギュレータ（１８０１）、
書き込み読み出し要求を前記複数の高速メモリ（１８０６－１、１８０６－２、１８０６－３、１８０６－４、１８０６－５、１８０６－６、１８０６－７、１８０６－８）のうちのアイドル状態のメモリに送ることにより前記メモリ内のリンク・リストの書き込みを開始するための装置（１８１９）、
前記リンク・リストの先頭バッファおよび末尾バッファおよび少なくとも１つの中間バッファを前記高速メモリに書き込むための装置（１９１９）、
前記高速メモリから前記大容量メモリに前記リンク・リストの前記少なくとも１つの中間バッファを転送し、その一方で前記リンク・リストの先頭バッファおよび末尾バッファを前記高速メモリに残すための装置（１８０４、１８０８、１８１０）、
その後前記アクセス・フロー・レギュレータからの前記リンク・リストの読み込みの要求を前記高速メモリに送るための装置（１８０２）、
前記リンク・リストの先頭バッファを前記高速メモリの１つに読み込むための装置（１８０４）、
前記リンク・リストの前記少なくとも１つの中間バッファを前記大容量メモリから前記高速メモリに転送するための装置（１８０８、１８１０、１８０４）、
前記高速メモリ内の転送されたバッファを新しい先頭バッファとして指定するための装置（１８１０）、
その後前記先頭バッファおよび前記末尾バッファだけでなく前記中間バッファを前記高速メモリから読み出すための装置（１８０４）、及び
前記リンク・リストの前記読み出されたバッファを前記アクセス・フロー・レギュレータに送信するための装置（１８０２）
を備えるメモリ管理システム。

【請求項１０】
請求項９のメモリ管理システムであって、さらに、
それぞれの高速メモリの現在の使用中／アイドル状態を示す、それぞれの高速メモリに固有の信号を発生するための前記高速メモリ（１８０３）を備える装置（１８０４）、
前記信号を前記アクセス・フロー・レギュレータに送るための装置（１８０２）、

前記アクセス・フロー・レギュレータ（１８０１）により、前記高速メモリによるリンク・リストの読み書きの要求を受け取る装置（１８１４）、
前記アクセス・フロー・レギュレータ（１８０１）により、前記要求を受け取ったことに応答して、前記使用中／アイドル状態信号を読み込む装置（１８２１）、
前記アクセス・フロー・レギュレータ（１８０１）により、前記読み出しに対する応答して前記高速メモリのそれぞれの現在の使用中／アイドル状態を判別するための装置（１８２１）、及び
前記アクセス・フロー・レギュレータ（１８０１）により、前記メモリの１つが現在アイドル状態かどうかを判別したことに応答して、前記高速メモリによるリンク・リストの読み書きの要求を許可する装置（１８２１、１８０４）
を備えるメモリ管理システム。

【発明の詳細な説明】

【技術分野】

【０００１】

本発明は、通信ネットワーク用のメモリ管理ファシリティに関するものであり、特に、ネットワーク・ノードでのブロッキングを減らすことによりネットワークのトラフィック処理能力を最適化するファシリティに関するものである。本発明は、さらに、ネットワーク・メモリ要素へのアクセス権の争奪時間を短縮することによりトラフィックの流れを改善するメモリ管理ファシリティに関するものである。本発明は、さらに、情報の格納を高速小容量メモリと低速大容量メモリとに割り当てることにより処理および争奪時間を短縮するための配置に関するものである。

【背景技術】

【０００２】

ネットワーク・トラフィック処理能力を高めるために、マルチノード通信ネットワークをアクティブに管理する方法が知られている。予想されるトラフィックを適切に処理するため各ノードに十分なファシリティを置いたネットワークが設計されている。これは、通常のトラフィック量をこなすために必要なファシリティだけでなく、あまり頻繁には発生しないピーク・トラフィックを経済的に実現可能な範囲で処理するための追加ファシリティを設置することを含む。通信ネットワークは、通常、理論上はありえると思われるが、あったとしても発生がまれな、トラフィック・ピークを処理するために必要な数のファシリティを備えるような設計ではない。

【０００３】

マルチノード通信ネットワークでは、ネットワークが全体として十分なレベルのトラフィックを処理するように設計されているとしてもトラフィック・ブロッキングが発生する場合がある。このブロッキングは、トラフィックの分布が不均等であることから来るもので、ネットワーク・ノードの一部または全部が法外なレベルのトラフィック発生によりオーバーロード状態になるのである。ネットワーク・ノードは、それがネットワーク接続要求の要求先である送信先ノードである場合にオーバーロード状態になる可能性がある。ネットワーク・ノードは、さらに、リンクを介して要求された送信先ノードに接続され、上流のノードから送信先ノードに送られる要求を受信した場合にオーバーロード状態になる可能性がある。不均等なトラフィック分布によるノードのオーバーロードを最小限に抑えるためネットワークにトラフィック・シェービング・ファシリティを追加する方法が知られている。これらのトラフィック・シェーピング・ノードでは、各ノードでのトラフィックだけでなく、各ノードにより発生する接続要求も監視する。遠隔地のノードの輻輳は、送信側ノードがすでにオーバーロード状態にある遠隔地のノードへのアクセスに対して発生する可能性のある要求の数を絞ることにより防止される。

【０００４】

マルチノード・ネットワークおよびそのトラフィック・シェーピング・ファシリティを利用すると、ネットワークは十分に低いレベルのブロッキングで通常のトラフィック・レベルを処理することができる。しかし、ネットワーク・トラフィックを管理し、制御する

ために必要なファシリティは、複雑で、高価であり、必要な処理オペレーションが複雑であることからネットワークのトラフィック・スループットが低下する。これらのファシリティでは、プロセッサにより制御されるリンク・リスト・エンジンをノードの入力と出力に装備し、着信および送信トラフィックをバッファリングする。リンク・リスト・エンジンのオペレーションでは、リンク・リスト・エンジン内の争奪に関する問題を最小限に抑えるために必要な複雑なデータ処理オペレーションを伴う。これらの争奪問題の複雑さゆえに、ネットワーク全体のトラフィック処理能力が低下する。

【０００５】

オーバーロードは、重いトラフィックが何回も発生したときに、各ノードでリンク・リスト・バッファが枯渇することで発生することがある。これにより、パケットが破棄され、システムのパフォーマンスがひどく低下することがある。バッファのオーバーロードは、バッファ・サイズが不適切であったり、着信トラフィックを処理するのに不十分な速度のバッファを使用したりして、生じる。システム設計者は、低速な大容量のバッファを使用するのか、それとも高速だが小容量のバッファを使用するのかの選択に直面している。低速な大容量のバッファだと、パケットが喪失してネットワーク・トラフィックの流れが阻害される。高速小容量バッファもまた、バッファのオーバーロードを引き起こし、重いバースト処理時に利用できるバッファが不足してパケットが破棄される。

【０００６】

これら両方のタイプのバッファに関連する根本の問題は、同じファシリティを使用して複数のアクセスが発生した場合に争奪問題が生じることにある。これは、例えば、特定のメモリ・バンクの読み書きアクセスに関して複数回のアクセスを受け取ったときに発生する。このような状況の下で、一方のアクセスは成功するが、他方のアクセスは、要求されたメモリ・バンクが利用可能になるのを待つ。この成功したアクセスに関連する呼び出しは、適切に処理されるが、遅延したアクセスに関連する呼び出しは、破棄されるか、または不適切な処理になる。

【０００７】

ＲＡＭメモリ・バンクに対するアクセスの争奪は、不適切な数のＲＡＭメモリ・バンクを使用すること、および／またはメモリ・バンクを処理するために用意されている争奪ファシリティのせいで発生する。いくつかの争奪ファシリティは、アクセスが処理される速度を制限するアルゴリズムおよびプロセスに依存する。このような従来技術の配置の１つでは、アクセスからアクセスまでの間約２５０ナノ秒の最小遅延時間を要するアルゴリズムを使用している。これは、第１のメモリ・パンクへのアクセスの割り当ての後アクセスを受け取るために第２のＲＡＭメモリ・バンクが利用可能かどうかを判別するための機能を用意していないため著しい制限である。したがって、アクセスの処理に要する２５０ナノ秒の時間間隔では、システムのスループットは、ＲＡＭメモリ・バンクが使用可能かどうかに関係なく、最大毎秒４，０００，０００回のアクセスの処理に制限される。既存の争奪の配置に関連する他の問題として、それらの多くは、複雑で、高価で、高トラフィック・レベルを処理するのには不適切なロジック・ファシリティを使用することが挙げられる。

【発明の開示】

【発明が解決しようとする課題】

【０００８】

本発明は、ＲＡＭメモリ・バンクの数を増やす第１の可能な実施例により、これらの争奪問題を解消する。これはそれ自体で、争奪の可能性を減じる。本発明の持つ第２の特徴は、それぞれのＲＡＭメモリ・バンクに、状態コントローラと呼ばれる関連する制御要素を備えることである。この状態コントローラは、ＲＡＭメモリ・バンクとアクセス要求の受信に使用されるシステム・バスとの間のインターフェースである。すべてのアクセス要求は、ノードによって生成されたすべてのアクセス要求を受け取るアクセス・フロー・レギュレータによりシステム・バスに適用され、ＲＡＭメモリ・バンクがアクセス要求の処理に利用できるかどうかを判別し、指定されたＲＡＭメモリ・バンクが現在使用中であれ

ばアクセス要求をバッファに入れ、アイドル状態のときにアクセス要求をＲＡＭメモリ・バンクに関連付けられた状態コントローラに適用する。ＲＡＭメモリ・バンクが使用中の場合、状態コントローラはアクセス・フロー・レギュレータに、ＲＡＭメモリ・バンクが現在他のアクセスの処理で使用中であり、当分の間さらにアクセス要求を処理するのには使用できないことを示す信号を送る。

【０００９】

書き込みアクセス要求の場合、アクセス・フロー・レギュレータは、アクセス要求をＲＡＭメモリ・バンクに送ろうとしたときにすべての状態コントローラをスキャンする。その際に、その関連するＲＡＭメモリ・バンクまたは格納のための空き領域のないＲＡＭメモリ・バンクについて現在使用中信号を発生している状態コントローラを即座にバイパスする。アクセス・フロー・レギュレータは、使用中のまたは完全に使い果たされたＲＡＭメモリ・バンクおよびその状態コントローラをハイパスし、格納領域用に利用可能なバッファがあるアイドルＲＡＭメモリ・バンクにアクセス要求を送る。

【００１０】

ＲＡＭメモリ・バンクのメモリは、高速だが比較的記憶容量の小さいタイプである。それぞれのＲＡＭメモリ・バンクは、そのバンク宛のそれぞれのアクセス要求を高速に処理することができる。アクセス・サイクルが完了すると、その状態コントローラは、ＲＡＭの状態を示すアクセス・フロー・レギュレータへの使用中信号を除去する。使用中信号が除去されるとすぐに、アクセス・フロー・レギュレータは、ＲＡＭメモリ・バンクが新たなアクセス要求の処理に利用可能になったことを知る。

【００１１】

本発明の他の可能な実施例は、ＲＡＭメモリ・バンクが使用中である短い時間間隔でのみ持続するＲＡＭメモリ・バンク使用中信号の使用オペレーションである。この信号を送ることは、ＲＡＭメモリ・バンクを具現化するＲＡＭデバイスの速度によってのみ制限される速度でアクセス要求を処理するのに都合がよく処理することができる争奪配置に含まれる。この争奪配置は、アクセスを処理可能な速度が不可欠な時間間隔を課すことで制限されるか、または実現される争奪ロジックの複雑さにより制限される。

【００１２】

本発明を具現化する争奪ファシリティを使用すると、ダイナミックＲＡＭメモリ・バンク・ファシリティが動作可能な最大速度は、争奪配置に内在する任意の制限ではなく使用されるＲＡＭデバイスの速度によってのみ制限される。本発明を具現化する高速ダイナミックＲＡＭメモリ・バンク・ファシリティは、パイプライン方式で動作させることで、光ファイバ伝送ファシリティのバス速度で到着するパケットを処理することができる。本発明により実現される争奪配置では、プロセッサにより制御されるリンク・リスト・エンジンが高トラフィック・レベルの処理時に最低の輻輳の着信および送信トラフィックを処理することができる速度が高まる。

【課題を解決するための手段】

【００１３】

本発明の一態様は、リンク・リスト・データ・ファイルを処理するように適合されたメモリ管理システムを操作する方法であり、前記システムは複数の高速小容量メモリおよび低速大容量メモリを備え、前記高速メモリは第１のデータ転送速度を持ち、前記大容量メモリは前記第１のデータ転送速度よりも遅い第２のデータ転送速度を持ち、前記システムはさらに、前記メモリによりリンク・リストの読み書きの要求を生成するためのアクセス・フロー・レギュレータを備え、前記方法は、

【００１４】

前記アクセス・フロー・レギュレータから前記高速メモリに書き込み要求を送信することにより、前記高速メモリへのリンク・リストの書き込みを開始する工程と、

前記リンク・リストの先頭バッファおよび末尾バッファおよび少なくとも１つの中間バッファを前記高速メモリに書き込む工程と、

前記高速メモリから前記大容量メモリに前記少なくとも１つの中間バッファを転送する

一方で、先頭バッファおよび末尾バッファを前記高速メモリ内に残す工程とを含む。

【００１５】

本発明の他の態様は、リンク・リスト・データ・ファイルを処理するように適合されたメモリ管理システムを操作する方法であり、前記システムは複数の高速小容量メモリおよび低速大容量メモリを備え、前記高速メモリは第１のデータ転送速度を持ち、前記大容量メモリは前記第１のデータ転送速度よりも遅い第２のデータ転送速度を持ち、前記システムはさらに、前記メモリによりリンク・リストの読み書きの要求を生成するためのアクセス・フロー・レギュレータを備え、前記方法は、

【００１６】

指定されたリンク・リストの読み取り要求を前記アクセス・フロー・レギュレータから前記指定されたリンク・リストのバッファを含む前記高速メモリに送信する工程と、

前記指定されたリンク・リストの先頭バッファを読み込む工程と、

前記リンク・リストの前記少なくとも１つの中間バッファを前記大容量メモリから前記高速メモリのうちの１つに転送する工程と、

前記１つの高速メモリに転送された前記中間バッファを前記指定されたリンク・リストの交換先頭バッファとして指定する工程と、

前記１つの高速メモリから前記指定されたリンク・リストの前記中間バッファを読み出す工程と、

前記指定されたリンク・リストの前記読み出されたバッファを前記アクセス・フロー・レギュレータに送信する工程を含む。

好ましくは、前記方法はさらに、

前記システムを前記アクセス・フロー・レギュレータからの複数の要求に対するリンク・リストを同時処理するように動作させる工程と、

前記システムを前記複数の高速メモリのうちの異なるメモリに格納されているリンク・リストのバッファを処理するように動作させる工程と、

前記システムを、末尾バッファを新しいリンク・リストに書き込まれた第１のバッファとして書き込むように動作させる工程と、

リンク・リストの先頭バッファをまず最初に読み込む工程を含む。

【００１７】

好ましくは、前記システムはさらに、複数の状態コントローラを備え、それぞれの状態コントローラは前記複数の高速メモリのうちの対応するメモリに対する個別のコントローラであり、前記システムはさらに、前記アクセス・フロー・レギュレータを前記状態コントローラと接続する要求バスを備え、読み取り要求を送信する前記工程は、

前記アクセス・フロー・レギュレータを、前記読み取り要求を受信すべきアイドル状態の高速メモリを選択するように動作させる工程と、

前記読み取り要求を前記アクセス・フロー・レギュレータから前記要求バスを介して前記選択された高速メモリに対して個別の状態コントローラに送信する工程と、

前記状態コントローラを、前記選択された高速メモリの現在の占有レベルを判別するように動作させる工程と、

前記現在の占有レベルが所定のレベル以下の場合に前記要求を前記高速メモリに送信する工程と、

前記選択された高速メモリの前記現在の占有レベルが前記所定のレベルを超えた場合に前記大容量メモリへの接続を要求する工程を含み、

前記システムはさらに、バックグラウンド・アクセス・マルチプレクサ、および前記状態コントローラを前記マルチプレクサと接続するアクセス・バスを備え、前記システムはさらに、前記マルチプレクサを前記大容量メモリと接続するバスを備え、前記方法は、前記マルチプレクサを、

前記大容量メモリへの接続の要求を前記状態コントローラから受信し、

複数の要求状態コントローラのどれに対し前記大容量メモリへのアクセスを許可するかを決定するように動作させる工程と、

前記１つの要求状態コントローラを前記大容量メモリに接続する工程と、
前記１つの高速メモリから前記大容量メモリにデータを転送する際の前記大容量メモリのオペレーションを制御する工程と、
前記アクセス・バスを介して前記状態コントローラから前記マルチプレクサに前記リンク・リストのバッファを転送する工程とを含む。

【００１８】

好ましくは、前記大容量メモリから前記バッファを転送する前記工程は、
リンク・リストの中間バッファを前記大容量メモリから前記高速メモリに、前記高速メモリのデータ転送速度に実質的に等しいデータ転送速度を持つバースト・モードで転送する工程と、
前記読み出されたバッファを前記高速メモリに格納する工程と、
その後、前記アクセス・フロー・レギュレータに転送するため前記高速メモリから前記リンク・リストのバッファを読み出す工程と、
バッファを既存のリンク・リストに書き込むのに、前記高速メモリから前記大容量メモリに前記既存のリンク・リスト既存の末尾を転送することにより書き込む工程と、
新しいバッファを前記既存のリンク・リストの新しい末尾バッファとして前記高速メモリに書き込む工程とを含む。

【００１９】

好ましくは、前記状態コントローラを動作させる前記工程は、
複数のリンク・リストのバッファを同時に受け取る工程と、
前記アクセス・フロー・レギュレータに送られる複数のリンク・リストのバッファを分離する工程と、
前記アクセス・フロー・レギュレータによって受け取られた複数のアクセスを前記高速メモリに送る工程と、
前記アクセス・フロー・レギュレータからそれぞれの受信された要求に応答し、前記状態コントローラに個別の高速メモリの現在の占有レベルを判別する工程と、
前記現在の占有レベル以下の場合、前記関連する高速メモリに前記アクセスを送る工程と、
前記現在の占有レベルを超えた場合、前記要求をバッファリングするよう前記アクセス・フロー・レギュレータに信号を送る工程と、
前記高速メモリからバッファをバースト・モードで前記大容量メモリに転送するのを制御する工程と、
前記大容量メモリからバッファを前記高速メモリに転送するのを制御する工程と、
転送が要求されたときに前記大容量メモリがアイドル状態かどうか判別する工程と、
アイドル状態の場合前記バッファを前記大容量メモリに送る工程と、
前記大容量メモリが使用中の場合、前記転送をバッファリングする工程とを含む。
好ましくは、前記マルチプレクサを動作させる前記工程は、
複数のビッディング高速メモリのどれに対し前記大容量メモリへのアクセスを許可するかを決定する工程と、
他のバッファの要求を前記１つの高速メモリにバッファリングする工程と、
前記大容量メモリからのバッファの転送先の高速メモリに対するＩＤを決定する工程と、
前記大容量メモリから前記識別された高速メモリにバースト・モードで前記バッファを転送するのを制御する工程とを含む。

【００２０】

好ましくは、前記方法は、さらに、
それぞれの前記高速メモリの使用中／アイドル状態を示す、それぞれの高速メモリに固有の信号を発生する工程と、
それぞれの発生した信号を前記アクセス・フロー・レギュレータに送る工程と、
前記アクセス・フロー・レギュレータを、前記高速メモリによるリンク・リストの読み

書きの要求を受け取るように動作させる工程と、
要求を受け取ったことに対する応答として前記アクセス・フロー・レギュレータを、前記高速メモリにより生成される前記使用中／アイドル状態信号を読み取るように動作させる工程と、
前記読み取りに対する応答として前記アクセス・フロー・レギュレータを、前記高速メモリのうちアイドル状態のメモリを識別するように動作させる工程と、
前記アクセス・フロー・レギュレータを、データ・ファイルの読み書きの要求を前記アイドル状態の１つの高速メモリに送るように動作させる工程とを含む。

【００２１】

本発明の他の態様は、リンク・リスト・データ・ファイルを処理するように適合されたメモリ管理システムを備え、前記システムは、
複数の高速小容量メモリおよび低速な大容量メモリであって、前記高速メモリは第１のデータ転送速度を持ち、前記大容量メモリは前記第１のデータ転送速度よりも遅い第２のデータ伝送速度を持つメモリと、
前記メモリによるリンク・リストの読み書きの要求を生成するためのアクセス・フロー・レギュレータと、
書き込み読みだし要求を前記複数の高速メモリのうちのアイドル状態のメモリに送ることにより前記メモリ内のリンク・リストの書き込みを開始するための装置と、
前記リンク・リストの先頭バッファおよび末尾バッファおよび少なくとも１つの中間バッファを前記高速メモリに書き込むための装置と、
前記高速メモリから前記大容量メモリに前記リンク・リストの前記少なくとも１つの中間バッファを転送し、その一方で前記リンク・リストの先頭バッファおよび末尾バッファを前記高速メモリに残すための装置と、
その後前記アクセス・フロー・レギュレータからの前記リンク・リストの読み込みの要求を前記高速メモリに送るための装置と、
前記リンク・リストの先頭バッファを前記高速メモリの１つに読み込むための装置と、
前記リンク・リストの前記少なくとも１つの中間バッファを前記大容量メモリから前記高速メモリに転送するための装置と、
前記高速メモリ内の転送されたバッファを新しい先頭バッファとして指定するための装置と、
その後前記先頭バッファおよび前記末尾バッファだけでなく前記中間バッファを前記高速メモリから読み出すための装置と、
前記リンク・リストの前記読み出されたバッファを前記アクセス・フロー・レギュレータに送信するための装置を備える。

【００２２】

好ましくは、前記メモリ管理システムはさらに、
それぞれの高速メモリの現在の使用中／アイドル状態を示す、それぞれの高速メモリに固有の信号を発生するための前記高速メモリを備える装置と、
前記信号を前記アクセス・フロー・レギュレータに送るための装置と、
前記高速メモリによるリンク・リストの読み書きの要求を受け取るための前記アクセス・フロー・レギュレータを備える装置と、
前記要求を受け取ったことに対する応答として、前記使用中／アイドル状態信号を読み込むため前記アクセス・フロー・レギュレータを備える装置と、
前記読み取りに対する応答として前記高速メモリのそれぞれの現在の使用中／アイドル状態を判別するための前記アクセス・フロー・レギュレータを備える装置と、
前記メモリの１つが現在アイドル状態かどうかを判別したことに対する応答として、前記高速メモリによるリンク・リストの読み書きの要求を許可する前記アクセス・フロー・レギュレータを備える装置とを備える。
本発明のこれらの態様および他の態様は、図面とともに詳細な説明を読むとよく理解できるであろう。

【発明を実施するための最良の形態】
【００２３】
図１の説明
本発明は、マルチノード通信ネットワークのトラフィック・スループットを高めるための機能強化されたメモリ・インターフェースを備える。この機能強化されたメモリ・インターフェースは、図１に示されているような通信ネットワークへのパケット化された情報の放出を制御するトラフィック・シェーピング要素を具現化する。図１のネットワークは、互いに通信するノードと呼ばれるスイッチング要素を相互接続している。ノードは、Ａ、Ｂ、Ｃ、Ｄ、Ｅ、Ｆ、およびＧと指定されており、リンク１からリンク８で指定されている別々のリンクにより接続される。図１のノードでは、内向きポートから外向きポートにトラフィックを分散するネットワークを定義している。
【００２４】
図２および３の説明
図２は、図１のノードを具現化する機器を開示している。図１は、ノードＡとノードＢを相互接続する単一経路（リンク１）を開示している。図１の各ノードは、着信リンクと送信リンクにより他のノードと接続されている。ノードＡは、ノードＡの着信リンクによりノードＢから受信し、ノードＡの送信リンクを使ってノードＢに送信する。
【００２５】
図２は、その着信リンクおよび送信リンクを含むそれぞれのノードの詳細を開示している。経路２１８は、ノードの送信リンクであり、経路２２３は、ノードの着信リンクである。図２は、ノードを定義する機器を示しており、図２の左半分では、ノードは着信リンク２２３、スプリッタ２２２、および各スプリッタと複数のポート２０１から２０５のそれぞれとを相互接続する経路２２１を含む。ポート２０１はその右側で外向きリンク２１８に接続され、他のノードに及ぶ。図２の左側に示されている機器は、外向きポート２０１から２０５を含んでおり、これらにより、ノードはリンク２１８に接続され、５つの外向きポート２０１から２０５を介して５つの異なるノードに及ぶようにできる。
【００２６】
図２の右側は、外向きポート２０２を具現化する機器を詳しく例示している。ポート２０２は、複数のリンク・リスト・キュー２１５、各キューに個別の制御ロジック１８００を備える。制御ロジックは、その後図１８に詳しく例示されており、リンク・リスト情報を処理し、マルチプレクサ２１３に届けるのに必要な機器を備える。５つのキュー２１５は、経路２３１によりマルチプレクサ２１３に接続されている複数の制御ロジック要素１８００のうちのそれぞれの要素により処理される。マルチプレクサ２１３は、経路２３１を、図１のリンク１から８のいずれかに対応する送信リンク２１８に接続する。図２のノードは、最大のパフォーマンスを発揮できるように出力キュー機能を備える。いかなる時点でも、外向きポート２０１～２０５のそれぞれについて、キュー２１５のうちの１つがマルチプレクサ２１３により選択され、キュー２１５からのパケットが送信リンク２１８に送られる。所定のポートに対するキュー１８００の選択は、ネットワークで使用されるトラフィック・シェーピング・アルゴリズムによって異なる。そこで図１の複数のノードでは、ノードＡなどの複数のノードのうちの１つにより処理される小さなコミュニティとの通信を望んでいると仮定する。オーバーロードは、ノードＡへのリンクを介してノードＡへの資源のトラフィック争奪として発生しうる。パケットはバッファリングされ、かかわっているリンクが処理されると伝送を完了できる。
【００２７】
この高トラフィック・シナリオは図３に詳しく示されており、ノードＡは要求のあるノードＡであり、暗色のリンク１、２、３、４、５、および６は要求のあるノードＡへの可能な回線経路にサービスを提供するリンクを表す。
【００２８】
ノードＢは、自分自身、およびノードＢおよび／またはノードＣをトラバースしノードＡを取得するトラフィックを提供する３つのノード（Ｃ、Ｅ、およびＦ）のあわせて４つ

のノードのトラフィックを持つ。ノードＢは、ノードＡへの１つのリンク（リンク１）のみ持つ。ノードＧおよびＤは、完全リンク容量のトラフィックを供給することができる。したがって、リンク１はオーバーロードされる。ノードＢは、高争奪時にノードＡに向かうトラフィックをバッファリングしなければならない。したがって、トラフィックは、争奪の少ない状況に徐々に向かうにつれ送信されるようになる。

【００２９】

高争奪間隔が十分長く持続すると、ノードＢのバッファはオーバーフローする可能性がある。これを防止するために、トラフィック・シェーピング・アルゴリズムにより、リンクへのトラフィックの放出を調整することで、ネットワーク全体にわたってバッファが、要求のあるノードに向かう大きなトラフィックを排出し、それをいつでも吸収するようにできる。例えば、ノードＥおよびＦは、それぞれ、リンク５および６を満たすだけの十分なパケットがあるとしても、リンク５および６にトラフィックを放出しない。その後、もはや、それらのパケットを送信することはないが、そうする際に、ノードＢはオーバーロードを回避し、ネットワークはより少ないパケットを破棄することになる。トラフィック・シェーピングは、前向きのフロー制御とみなすことができる。

【００３０】

トラフィック・シェーピングは、正常動作させるためにはハイパフォーマンスのバッファリングを必要とする。このバッファリングは、ハードウェアのリンク・リスト・プロセッサに左右される。ハードウェアのリンク・リスト・プロセッサは、メモリを効率よく利用し、バッファを着信パケットに動的に割り当て、保持されているデータが正常に外向きリンクに転送された後バッファを回収する。

【００３１】

図４～６の説明

リンク・リスト・バッファは、バッファリング情報に使用される本発明に基づく。初期化されたシステムにおいて、すべてのメモリは複数の汎用バッファに分けられる。各バッファには、そのバッファによって格納される内容４０１、および次のバッファへのポインタ４０２用の領域が確保される。これは、図４に示されている。

【００３２】

初期化時に、すべてのバッファは、前のバッファのポインタ・フィールドを次のバッファのアドレスに設定することにより１つにつながれる。これは、自由リストと呼ばれ、図５に示されている。

【００３３】

通信システムでは、これらの汎用バッファに１つずつ情報を書き込み、書き込まれたバッファを、何らかの特定の機能について情報を格納するキューにリンクする。システムの初期化が終わった後、すべてのキューは空である。それらのキューの長さは０であり、先頭と末尾はＮＵＬＬを指す。図６の特定のキューについて情報が到着すると、汎用バッファが自由リストから取り出され、情報が埋め込まれ、キューのリストに追加される。末尾ポインタは、キューの追加された要素のアドレスに変更され、キューの長さカウントは１つ増やされる。キューから情報が読み出されると、そのキューのリストの先頭にあるバッファの内容が読み出され、リストの先頭はリスト内の次のバッファに移される。キューの長さカウントも１つ減らされる。

【００３４】

図６のキューＡでは、先頭バッファに内容Ｑがあり、末尾バッファに内容Ｚがあり、キューＢでは、先頭バッファに内容Ｎがあり、末尾バッファに内容Ｇがあるが、キューＣのバッファでは、先頭バッファも末尾バッファも内容ＨＨが入っている。このシステムは、１１個のバッファおよび３つのキューを持つ。リンク・リスト・バッファリング・システムの重要な特徴の１つに、バッファ割り当てが完全に動的であることが挙げられる。自由リストが空になっていない限り、バッファのキューへのどのような分配も許される。例えば、キューＡは１１個のバッファすべてを持ち、その後しばらくしてから、キューＡは４つのバッファ、キューＢは４つのバッファ、そしてキューＣは３つのバッファを持つよう

にできる。その後しばらくして、すべてのキューは空になり、すべてのバッファが再び、自由リストに戻される。

【００３５】

リンク・リスト・バッファリングは、特に、通信アプリケーションに適しているのは、全能力で動作している所定のファシリティに対し、到着する情報が一定の速度でバッファを消費するからである。しかし、このファシリティは、多くの場合、複数の流れまたはストリームに関連する情報を多重化しなければならない。リンク・リストは、処理およびその後の外向き伝送媒体への多重化のためは内向き情報を逆多重化し、編成する効率のよい手段となっている。処理されたバッファは、送られた後、自由リストを通じてリサイクルすることができる。

【００３６】

図２のノードＢへの内向きの合計トラフィックが毎秒１０個のバッファを消費するが、ノードＢはさらに毎秒１０個のバッファを送信すると仮定する。このノードは、バッファが埋められるやいなや空にされるので、バランスがとれている。バッファリングの観点からは、どの流れが入って来るのか、出て行くのかは問題でない。例えば、ノードＣからノードＢへの内向きトラフィックの向かい先がノードＧまたはノードＡであるかは問題でない。ある時点において、リンク１をノードＡに送るキューは、書き込み済みであり、リンクをノードＧに送るキューは空でありえるが、その後、しばらくしてから、第１のグループは空になり、第２のグループは満杯になることがありえる。流れ全体がバッファ容量と一致する限り、システム完全性は維持される。しかしながら、トラフィック全体は固定された最大速度で媒体上を出入りするが、多くの流れはサポートされている通信媒体で多重化することができる。バッファの管理は柔軟性が高いため、一時的な機器状態が緩和される。

【００３７】

リンク・リスト・バッファは、トラフィック・シェーピングのシナリオでは有用である。ノードはそれぞれ、ある程度の全体的スループットに対応できる。トラフィック・シェーピングの実装により、バッファリングがノードに入り込むが、それは、シェーピング・プロファイルの条件を満たすため、いくつかの外向きリンクに向かう内向きトラフィックを一定時間バッファリングしてから送信しなければならないからである。コネクションレス型パケット指向ネットワークでは、内向きパケットの宛先は、そのパケットが到着してからでないとわからない。特定のストリームまたは流れは全体的な最大速度を持ちえるが、所定の流れにより情報が爆発的にノードに送られる可能性がある。そのバースト送信時に、流れのバーストはリンク容量全体を消費する可能性があるため、他のノードを宛先とするリンクによりサポートされるパケットの流れは定義からアイドル状態である。

【００３８】

例として、図３のネットワークを再度考察する。ノードＢは、３つのリンク１、４、および７に接続されている。リンク４上のトラフィックは、ノードＡまたはノードＧのいずれかを宛先とすることができる。ノードＡを宛先とする一連のパケットは、ノードＡからノードＢへのストリームまたは流れをなす。ノードＧを宛先とする一連のパケットは、ノードＧからノードＢへのストリームまたは流れをなす。

【００３９】

図７の説明

図７は、リンク４上の考えられる内向きパケット・シーケンスを説明するタイミング図を示している。

ノードＡを宛先とする４つのパケットのバーストで、自由リストから切り離されたバッファはすべてリンク１をサポートするキューに追加される。このバーストが延長した場合、自由リストから取り出されたバッファはこのキューに流れ込む。しかし、いかなるときも、次の内向きパケットは、ノードＡまたはノードＧのいずれかを宛先とすることができる。次の内向きパケットの宛先は、確認されるまでわからない。したがって、効率のよいパケット交換機器のオペレーションには、バッファの自由度の高い動的な割り当てが必要

である。４つのパケットのバースト中に、ノードＧへのリンク７をサポートするキューは、追加バッファを一切受け取らないが、バーストはノードＡを宛先とするトラフィック専用であるため、それらのキューは、それ以上バッファを必要としないので内向きトラフィックを持たない。したがって、効率のよいパケット交換機器のオペレーションには、バッファの自由度の高い動的な割り当てが行えれば十分である。

【００４０】

リンク・リストのオペレーション

リンク・リスト・エンジンは、半導体メモリをＲＡＭ記憶装置として採用している。バッファの追加または削除のためリンク・リストを操作する方法では、バッファ・メモリおよび関連するリンク・リスト・テーブル・メモリの両方に対し一連の読み書きを行う必要がある。リンク・リスト・テーブル・メモリは、リンク・リスト・プロセッサによりサポートされる各リンク・リストへの先頭および末尾のルックアップ・テーブルを含む静的構造である。例えば、流れに読み出しまたは書き込みトラフィックが含まれる場合、リンク・リスト・プロセッサは、最初に、流れの番号を使用して、注目しているリストの先頭および末尾のアドレスを検索する。注目しているリンク・リストのそれらのアドレスが判明すると、プロセッサはそのリストに対し所定のオペレーションを実行することができる。バッファがリンク・リストに追加される場合、空バッファが図６の自由リストの先頭から取り出され、自由リストの先頭がそのリスト内の次の空バッファとなるように書き直される。自由リストの新しい先頭のアドレスは、書き込みのためちょうど取り出されたばかりのバッファのリンクに含まれる。そのバッファの内容が書き込まれ、リンク・リストの末尾のバッファ内のリンク・フィールドは、書き込まれたばかりのバッファのアドレスを書き込まれる。その後、テーブル・メモリは、新しい末尾アドレスを書き込まれる。新しいバッファをリンク・リストに書き込むプロセスでは、テーブル・メモリは読み出しと書き込みを維持し、バッファ記憶装置では２つの書き込みを維持する。バッファをリンク・リストから読み出すプロセスでは、テーブル・メモリは読み出しと書き込みを維持し、バッファ記憶装置では読み出しおよび書き込みを維持する。バッファ記憶装置への書き込みは、空にされたバッファが自由リストに再度連結されなければならなくなると発生する。

【００４１】

リンク・リスト・プロセッサはバッファ・メモリにランダムにアクセスする

このプロセスの重要な側面として、メモリへのアクセスがランダムであるという性質が挙げられる。ランダム化にはいくつかの要因がかかわっている。リンク・リストが通信ファシリティからのトラフィックをバッファリングする場合、一連のアクセスはそのファシリティによって伝送されるトラフィックに完全に左右される。Ｅｔｈｅｒｎｅｔ（登録商標）などのコネクションレス型ネットワークでは、ファシリティに到着するパケットは、その旅先で多数のキューを宛先としている可能性がある。将来のパケットに対する宛先キューは、一般的には予想できない。到着がランダムであるという性質から、所定のキュー内のアドレスはスクランブルされる。出て行くパケットの伝送は制御下にあるが、ここでもまた、ランダム化にネットワーク状態が絡んでいる。例えば、送信ファシリティで多数のキューを多重化するものとする。ときどき、それらのキューすべてが、一部にかかわることもあれば、１つにかかわることもあれば、何にもかかわらない場合もある。輻輳は、外向きファシリティの遠端フロー制御により、または外向きファシリティを宛先とするトラフィックが発生する多数の内向きファシリティにより、引き起こされることがある。

【００４２】

ランダム化にかかわる第２の著しい要因は自由リストである。自由リストにかかわる要因は、送信ファシリティへのバッファ伝送の順序に完全に依存する。しかし、外向き伝送は予測不可能な条件に左右される。したがって、トラフィックの状態により、自由リスト上の空バッファのアドレス・シーケンスがランダム化される。

【００４３】

リンク・リスト・プロセッサをバッファ管理の目的に使用する通常のシステムが１、２秒の間著しい負荷の下で動作すると、バッファ・メモリへのアクセスは、相関関係を完全

に欠いている。

【００４４】

バッファ・メモリのアクセス・パラメータおよびメカニクス

リンク・リスト処理ではメモリに対し、事前に作成されたスクリプトを使って一連のアクセスが実行されるため、リンク・リスト・プロセッサのパフォーマンスは、それらのアクセスの管理の仕方に大きく依存する。しかし、メモリ・コンポーネントの可用性およびアクセス配置などの最終的なバッファリング要件により、リンク・リスト・プロセッサに対するメモリ・システム設計が制約される。

【００４５】

リンク・リスト・プロセッサが中継交換アプリケーションで使用される場合、アタッチされたファシリティの長さおよびその容量がメモリ・システムの設計に課せられる。例えば、ＳＯＮＥＴネットワークではケーブル切断に関する標準報告間隔は６０ミリ秒である。しかし、この標準は、一般にあるネットワーク・ケーブル配線路に対しては制約的であり、報告間隔は、さらに標準的には、２００～３００ミリ秒である。これは、数千キロメートルのケーブル配線長に対応する。最小サイズのパケットをバースト伝送する場合、単一のＯＣ－７６８ファイバは、２３００万個を超えるパケットを３００ミリ秒で送信する。光ファイバ・ケーブルのトラフィック伝達ファイバは、１００本またはそれ以上の異なる個別の素線を持ちうる。したがって、そのような１つのケーブルを終端するシステムは、継ぎ目なくケーブル切断から回復できるように数十億個のパケットの順序に基づいてバッファリングする必要がある。

【００４６】

知られているハードウェア・リンク・リスト・エンジンの基本的な問題

ハードウェア・リンク・リスト処理をサポートするメモリ・サブシステムは、大きく、高速であり、多数のキューをサポートし、クロックの任意のエッジに基づき任意のキューでメモリを操作し現在利用可能なトラフィック内での瞬間移動に対応できなければならない。ランダム・アクセス・メモリは、リンク・リスト・プロセッサのメモリを適当に操作できるが、十分高速にサイクル動作できないか、または十分に大きくないため最大容量ファシリティのバッファリングを行えない。

【００４７】

市販されている同期ダイナミック・メモリ（ＳＤＲＡＭ）パッケージである２種類のメモリは、最大１メガビットの記憶域を持つが、そのランダムな読み書きサイクル時間は約６０ナノ秒である。ＯＣ－７６８中継器にはメガビットのオーダーの記憶装置が必要であるため、ＳＤＲＡＭのサイズが好適であるが、最小サイズのパケットをバースト送出する全負荷状態のＯＣ－７６８中継器では、パケットは４０ナノ秒おきに到着する。したがって、市販のＳＤＲＡＭは遅すぎてＯＣ－７６８ファシリティには使えない。

【００４８】

市販の同期スタティック・メモリ（ＳＳＲＡＭ）は、約２ナノ秒でランダムな読み書きアクセス・サイクルを実行し、待ち時間は４ナノ秒である。これは、少数のＯＣ－７６８ファシリティと制御オーバーヘッドを処理できる十分な速さである。しかし、ＳＳＲＡＭは１６メガビットを超える容量のものは市販されていない。ＯＣ－７６８ファシリティのバッファリングを適切に行うためには約９０個のＳＳＲＡＭデバイスが必要であろう。そのような多数のＳＳＲＡＭデバイスによって発生する熱量は問題になる。

【００４９】

結論として、ハードウェア・リンク・リスト・プロセッサにより形成される利用可能なメモリの基本的な問題は、大容量で低速（ＳＤＲＡＭ）か、高速だが非常に小容量（ＳＳＲＡＭ）のいずれかであるという点である。大容量でかつ高速なハードウェア・リンク・リスト・プロセッサが得られるような妥協点はない。

【００５０】

リンク・リスト・プロセッサの設計改善

本発明の改善されたリンク・リスト・プロセッサは、大容量高密度バッファだけでなく

、高速高性能バッファをどのようにしたら得られるかという問題に対する解を具現化する。これは、新規性のある方法で、メモリへのアクセスのストリームのランダム性に依存することにより実現される。

【００５１】

これまで、ＳＤＲＡＭを使用するメモリ・サブシステムの争奪の問題により、システムは、利用可能なＳＤＲＡＭのパイプライン化された特徴を使用することによりアクセスをオーバーラップするのではなく、各連続するメモリ・サイクルが完了するのを待っていた。これにより、システムはかなり高速なバス・サイクル・レートで動作することが可能である。本発明では、ＲＡＭメモリを、固有のランダム読み書きアクセス速度、つまり数メガヘルツではなく、そのポート速度、つまり、数百メガヘルツで動作させることにより、争奪問題を解消する。本発明の他の特徴は、バンク化ＲＡＭメモリへのランダム・アクセス・ストリームが存在する場合に、バンクの個数が増えると、次のアクセスがすでに使用中のバンクに向かう確立が減じるので、より多くのメモリ・バンクが使用されるというものである。

【００５２】

図８および９の説明

ダイナミックＲＡＭは、複数のバンクとともに単一パッケージにパッケージングされている。バンクとは、別々にアドレス指定可能な１つの記憶ユニットのことである。バンクは物理パッケージ上のピンなどの入出力資源を共有するので、複数のバンクが同時に複数のアクセス要求を処理することができる。処理できる未解決要求の最大数は、クロック速度および同期ＳＤＲＡＭデバイスの構造によって決まる。近似的に、ＳＤＲＡＭ内の１バンクへのアクセスが完了するのに４クロック・サイクルを要し、パッケージ内に４つまたはそれ以上のバンクがある場合、処理に４回のアクセスが同時に含まれる可能性がある。第１のアクセスを完了するのにかかわる４クロックのそれぞれの立ち上がりエッジで新しいアクセスが有効にされる。

【００５３】

図８は、円８０３の中に４つのバンク８１０、８１１、８１２、および８１３を示している。これは、１つの可能なＳＤＲＡＭの中の４つのバンクを表している。漏斗８０２および蛇口８０４は、ＳＤＲＡＭの中にあるものにアクセスするための共有制御およびデータ・バス資源を表す。４つのアクセス要求Ａ、Ｂ、Ｃ、Ｄ（８０１）が、ＳＲＡＭ８０３の漏斗８０２に入る状況を示している。

【００５４】

アクセス要求Ａ、Ｂ、Ｃ、またはＤのうち一度に１つだけが漏斗８０２に入ることができる。各バンク８１０～８１３は、アクセスに他と同じだけ時間を要する。バンクが要求Ａを開始し、その後、他のバンクが要求Ｂを開始すると、要求Ａに対する結果は、要求Ｂからの結果の前に蛇口８０４から現れる。しかし、しばらくの間、１つのバンクが要求Ａについて動作し、他のバンクは同時に要求Ｂについて動作する。

【００５５】

アクセスは、バンクの任意の組合せにより処理することができる。これらは、すべて、同じバンクにより処理するか、または異なるバンクにより処理することができる。いくつかのアクセスでは、そのアクセス自体のバンクとなるが、他のアクセスではバンクを共有する。アクセスのいくつかのグループでは、バンクがまったく使用されていない可能性がある。例えば、図９に示されているように、アクセスＡおよびＢは、同じバンク８１０により処理されるが、アクセスＣは、バンク８１２により処理され、アクセスＤは、バンク８１３により処理される場合がある。アクセスのグループを利用可能なバンクに分配するプロセスは、パーティション分割と呼ばれる。バンクの観点から、アクセスの回数は重要な情報であるが、それは、すべてのアクセスが一様な特質を持つからである。したがって、パーティションは、バンクへのアクセスの会計といえる。例えば、｛４，０，０，０｝は、４つのアクセスが単一バンクを占有し、他の３つのバンクは占有されていないことを意味する。図９では、パーティションは｛２，１，１，０｝である。

【００５６】
図１０～１３の説明
図１０の同期ダイナミックＲＡＭ（ＳＤＲＡＭ）は、４独立バンク・メモリ・アーキテクチャである。バンク毎に、オペレーションにはアクセス待ち時間、書き込みオペレーションの場合にピンからメモリ・アレイへの情報の送出、または読み出しオペレーションの場合にメモリ・アレイからピンへの情報の送出が伴う。また、メモリ内部のセンス・アンプが次の読み出しまたは書き込みサイクルを準備できるようにプリチャージ間隔も必要である。４つのバンクのそれぞれが利用可能である。４つのバンクはそれぞれ、専用のセンス・アンプを備えており、ＳＤＲＡＭの制御ポートおよびデータ・ポートについてのみアクセスの争奪が発生する。
【００５７】
図１０および１１は、制御バス８０１がＳＤＲＡＭのバンク１および２に向けられた活動を受け入れていることを示している。アクセス・コマンド「Ａ」と図１１の関連する読み出しコマンド「Ｒ」との間に待ち時間がある。また、読み出しコマンドとデータが利用できるまでの間にも待ち時間がある。さらに、アクセス・コマンドＡ１とＡ２との間に待ち時間がある。図１１では、１０ナノ秒のクロック・サイクルは、便宜上使用されている。全体的なサイクル時間は８０ナノ秒である。ＳＤＲＡＭの４つすべての利用可能なバンクは、このときにアクセスできるが、どのバンクもこの時間間隔内では１回しかアクセスできない。このような待ち時間があるため、ＳＤＲＡＭは、パイプライン深さが４段であると言われる。衝突インスタンスの条件が満たされるためには、さらに８０ナノ秒のメモリ・サイクルが必要である。衝突インスタンスは、使用中ＳＤＲＡＭメモリ・バンクへのアクセス・フロー・レギュレータの到着として定義される。例えば、図１１のクロック・サイクル２でバンク１に対するアクセス要求が到着した場合、クロック・サイクル９までメモリには適用できないが、それは、バンク１に対する要求Ａがクロック・サイクル１で到着しているからである。
【００５８】
図１２は、ＳＤＲＡＭ　１２０３をバッファ記憶域として使用するハードウェア・リンク・リスト・プロセッサ１２０１およびサポートされている流れ毎に先頭、末尾、およびカウントを保持するテーブル記憶域用に小さな同期スタティックＲＡＭ　１２０７を示している。ＳＤＲＡＭ　１２０３は、４つのＳＤＲＡＭ　１２１０、１２１１、１２１３、および１２１４を備える。リンク・リスト・プロセッサ１２０１では、クロック１、３、５、１１、および１５について図１３に示されているように５つのアクセス要求のストリームが発生すると仮定する。第１の３つのアクセスＡ１、Ａ２、およびＡ３は、異なるバンクに向かうのでパイプライン化することができるが、クロック１１での第４のアクセスＡ２は、クロック３のアクセスＡ２を処理する使用中バンク２に進むので遅延させなければならない。
【００５９】
一般に、ランダムなアクセス・ストリームが存在する場合に発生する争奪の平均量は、４バンクＳＤＲＡＭに対する基準としてみなすことができる。バンクは入出力ファシリティのみを共有するので、４バンクＳＤＲＡＭへの１ブロック分のアクセスに対し重み付きの平均アクセス時間を計算するほうが簡単である。例えば、パーティション｛３，１，０，０｝を考察する。アクセスに（Ａ、Ｂ、Ｃ、Ｄ）のラベルを付けた場合、可能なグループ分けとして、｛（Ａ，Ｂ，Ｃ），（Ｄ）｝，｛（Ａ，Ｂ，Ｄ），（Ｃ）｝，（（Ａ，Ｃ，Ｄ），（Ｂ）｝，および｛（Ｂ，Ｃ，Ｄ），（Ａ）｝がある。選択された２つのバンクは、｛１，２｝，｛１，３｝，｛１，４｝，｛２，３｝，｛２，４｝，または｛３，４｝とすることができる。
【００６０】
３つからなる１つのパーティションと１つからなる１つのパーティションを２つのバンクにマッピングする２つの方法がある。例えば、パーティション｛（Ａ，Ｂ，Ｃ），（Ｄ）｝を｛１，２｝にマッピングしようとした場合、バンク１を持つ（Ａ，Ｂ，Ｃ）および

バンク２を持つ（Ｄ）、またはバンク１を持つ（Ｄ）およびバンク１を持つ（Ａ，Ｂ，Ｃ）のいずれかがありえる。４＊６＊２＝４８であり、４つのアクセスを配置する方法は２５６通りあるため、パーティション｛３、１、０、０｝が生じる確率は、０．１８７５である。漂遊アクセスは、２つの衝突するアクセスの内側でパイプライン化することができ、したがって、完了するクロックの数は８＊３＝２４となる。

【表１】

パーティション	確率	完了するクリック	完了までの重み付き平均時間
4,0,0,0	0.0156	32	0.5
3,1,0,0	0.1875	24	4.5
2,2,0,0	0.1406	19	2.6714
2,1,1,0	0.5625	16	9.0
1,1,1,1	0.0938	14	1.3125

表１：４つのトランザクションの４メモリへの配置

【００６１】

表１は、それぞれの可能な分配について示されている関連するデータとともにアクセスを４メモリ・バンクに分配する様々な可能な方法に関係するデータを例示している。表１の第１欄では、様々なバンクへのアクセスが可能な様々な可能なパーティションの一覧を示している。表１の最上行は、第１のバンクが４つのアクセスを受け取り、残りバンクは何も受け取らないことを示している。第１欄２行目の分配は、３、１、０、０である。第１欄の下の行の分配は、４つのアクセスすべてが平等に各バンクに分配されていることを示す。第２欄は、その行の各分配の確率を示す。第３欄は、その機能を完了するのに要するクロック・サイクル数を示している。第４欄は、完了までの重み付き平均時間を示す。第１行については、第１のバンクへの４つすべてのアクセスの分配の確率は０．０１５６である。これは、完了までの重み付き平均時間が０．５であれば完了までに３２クロック・サイクルを要する。下の行は、１，１，１，１の分配、０．０９３８の確率を持ち、重み付き平均時間１．３１２５で完了までに１４クロック・サイクルを要する。この評価から、最も確率の高い分配は２、１、１、０であり、確率は０．５６２５であることがわかる。最もよい可能な分配は、下の行に示されており、確率は０．０９３８である。

【００６２】

表１の４つのバンクについて、４つのアクセスを完了するまでの合計重み付き平均時間は１７．９８クロックである。この数字は、４バンク・メモリへのアクセスの可能なすべての組合せについて完了までの重み付き時間の合計である。２つの１０ナノ秒システム・クロックのパイプライン・クロックおよび１６ビット幅バスを仮定すると、４バンクＳＤＲＡＭに対する平均持続可能スループットは、１．２４Ｇビット／秒であるが、それは各メモリのトランザクションが、それぞれにつき２クロックに対し１６ビットを要するからである。争奪オーバーヘッドは２８．５％である。この値は、アクセス完了までの平均時間とアクセス完了までの可能な最短時間との差をアクセス完了までの可能な最短時間で割るという計算で求められる。例えば、アクセス完了までの平均時間が５０ナノ秒とし、アクセス完了までの可能な最短時間が４０ナノ秒であったと仮定する。すると、オーバーヘッドは２５％になる。

図１３および１４は、争奪状態でないアクセスと争奪状態であるアクセスとの差を例示している。

【００６３】

図１４の説明

頻繁な争奪は設計の決定に影響を及ぼす。図１４のタイミング図を考察する。ここで、メモリ・バンクへのアクセス・バスはスロット付きである。それぞれのアクセスには、読み出し間隔と書き込み間隔が設定されている。これらのアクセスは、メモリの制御バスのアイドル時間を最小限に抑えるためできる限り緊密にパックする必要がある。

【００６４】

このようにしてアクセスを構造化することは、争奪でメモリの平均読み書きサイクル時間が実質的に増大するため意味のあることである。平均読み書きサイクル時間は最大読み書きサイクル時間程度なので、メモリの読み書きサイクル時間が最大となるようにシステムを設計するとより効率的である。少ない読み書きサイクルを使用して効率および動作速度を高めることは、争奪を管理するために必要なハードウェアのコストに見合わない。

【００６５】

メモリは、バス速度で定義された速度でアクセス要求を受け付けることができるが、それは内部パイプライン設計によりそうするようにできるからである。しかし、争奪率２８．５％では、争奪の少ないシステムとは反対に、争奪率がキューの深さに指数関数的に関係しているため、このシステムにおけるキュー操作が劇的に増える。様々なアクセス時間をサポートするこのキュー操作および状態制御装置は、一定の最大アクセス時間を予想するものと比べてかなり念入りである。例えば、一定の最大アクセス時間を持つ設計では、キューを必要とせず、全体にわたる状態制御はかなり単純である。アクセス時間が頻繁に変わる設計では、メモリを使用しているマシン内に複数のキュー、さらに争奪が発生したときにアプリケーションからこのマシンを起動、停止できるより精巧な状態ロジックを必要とする。

【００６６】

図１５および１６の説明

図１５に示されているように、８バンク分のＳＤＲＡＭがあると仮定する。この場合、メモリは、図１３に示されているＳＤＲＡＭのサイズの半分のブロックにパーティション分割される。例えば、図１３のバンク１を宛先とするアクセスは、今度は、図１６のバンク１またはバンク２のいずれかに入る。図１３のバンク２を宛先とするアクセスは、図１６のバンク３とバンク４との間に分割される、などとなる。図１３では、バンク２への第２のアクセスは、バンク２への第１のアクセスとの争奪状態を引き起こし、遅延が発生する。図１６では、これらのアクセスのうちの第１のものがバンク３に向かうが、それらのアクセスのうち第２のものはバンク４に向かう。

【００６７】

図１３および１６を比較すると、争奪の解消により、図１３のクロック１３（Ａ４）とクロック１６（Ｒ４）との間のアイドル・ギャップが閉じられることがわかる。図１６では、データ・バスは、クロック７からクロック１６へ連続的に占有される。争奪は、アクセスのランダムな内向きストリームが与えられると、異なる組合せで発生することがある。争奪は、８つのバンクでは解消されないが、バンクの数が多くなるほど争奪の発生は減少する。

【表２】

バンク	完了までの重み付き平均時間	争奪オーバーヘッド	最大持続可能スループット（Gビット／秒）
4	17.98	28.5%	1.24
5	17.06	21.9%	1.31
6	16.47	17.6%	1.36
7	16.06	14.7%	1.39
8	15.77	12.6%	1.42
9	15.55	11.1%	1.44
10	15.38	9.9%	1.45
11	15.24	8.9%	1.47
12	15.12	8.0%	1.48
13	15.03	7.4%	1.49
14	14.95	6.8%	1.5
15	14.88	6.3%	1.5
16	14.82	5.9%	1.51

表2：バンクの個数に対する4回アクセス完了までの重み付き平均時間

【００６８】

表２は、バンクの数を増やした場合に完了までの重み付き平均時間がどのように減少するかを示している。最大持続可能スループットでは再び、１０ナノ秒のシステム・クロックと１６ビット幅のデータ・バスを仮定している。

【００６９】

パイプライン深さも、パフォーマンスに影響を与える。例えば、８バンク分のメモリがあるが、パイプラインの中には２段しかない場合、２回アクセス完了までの重み付き平均時間は６．２５クロックである。この場合のオーバーヘッドは４．２％であり、最大持続可能スループットは１．５１Ｇビット／秒である。争奪率は５％に低下するものとする。その後、平均アクセス時間は３．５ナノ秒である。これは、かなりバス・サイクル時間に近い。この結果は、図１４および図１６に示されているアクセス制御配置を比較してみるとわかる。図１４では制御バス活動の密度が高い、つまり、図１４の場合よりも１クロック当たりのランダム読み書きアクセスが多いということがわかる。

【００７０】

図１７および１８の説明

本発明では、独立連携状態コントローラ１８０４はＲＡＭパッケージのそれぞれのバンク１８０３に割り当てられている。それぞれのバンクは、独立にサイクル動作し、その状態コントローラ１８０４を通じて、他のＲＡＭバンク１８０３にあわせよく調整された形で結果を出すことができる。さらに、状態コントローラ１８０４の流れにより、ときおり発生する争奪のためにアクセス要求を保持するアクセス・フロー・レギュレータ１８０１のキューを制御する。これにより、アクセス要求の破棄が防止される。状態コントローラ１８０４は、さらに、バンク１８０３のリフレッシュなどのハウスキーピング機能を独立に管理する。状態コントローラ１８０４は、独立している。状態コントローラ１８０４を使用すると、ＲＡＭバンク１８０３との間のバックグラウンド・バースト転送を、他のＲＡＭバンク１８０３のフォアグラウンドのアクセス活動と同時に行うようにできる。このため、ＲＡＭバンク１８０３からリンク・リストの中間部をリモートＲＡＭ１８０６に格納することができる。これにより、リンク・リストの先頭および末尾のみがＲＡＭバンク１８０３に残る。例えば、図６のキュー５０６を見ると、バッファＱおよびＺは、ＲＡＭバンク１８０３内のどこかに置かれるが、バッファＤおよびＲは、リモートＲＡＭ１８０６内に格納される。リンク・リストの中間をリモートで格納できれば、開示されているシステムでは、市販のパッケージＲＡＭを使用して任意のサイズのリストをサポートすることができる。リンク・リストの大部分をリモートＲＡＭ１８０６にリモートで格納することができる場合、ＦＰＧＡに埋め込まれているＲＡＭ１８０３を先頭と末尾に使用することができる。状態コントローラ１８０４は、先頭と末尾を保持するＲＡＭ１８０３と組み合わせる。この設計は、ＲＡＭ　１８０３と異なるパッケージに常駐する状態コントローラよりも効率がよい。ＲＡＭ　１８０３と状態コントローラとが同じ場所にある場合は、リストの先頭と末尾を格納するため技術を選択することになる。この選択結果は、少数のリンク・リスト・キュー用のオンボード・レジスタ、適度の量のリンク・リスト・キューのスタティックＲＡＭ　１８０３、または大量のリンク・リスト・キューのダイナミックＲＡＭ　１８０６である。

【００７１】

状態コントローラ１８０４のブロック図が図１７に示されている。

状態コントローラ１８０４は、フォアグラウンド・ポートから流れ込む情報をゲートするアービトレーションおよびシーケンシング・ロジックが適用され、フォアグラウンドまたはバックグラウンド転送で使用中の場合に新しい着信活動を行わないようにＲＡＭメモリ・バンク１８０３をガードする。さらに、状態コントローラは、ＲＡＭメモリ・バンク１８０３内の状態を監視し、リモートＲＡＭ１８０６との相互作用が発生する場合にそれを判別する。状態コントローラ１８０４は、図１８に示されているように、バンクグラウンド・アクセス・マルチプレクサ１８０８、リモートＲＡＭ　１８０６、およびアクセス・フロー・レギュレータとともにシステム内に組み込まれる。

【００７２】

図１７の状態コントローラ１８０４は、関連するＲＡＭメモリ・バンク１８０３、アクセス・フロー・レギュレータ１８０１、およびバックグラウンド・アクセス・マルチプレクサ１８０８の間のインターフェースとして機能する。これらの要素への接続は、図１７に詳しく示されている。状態コントローラ１８０４は、マルチプレクサ１７０２ならびにアービトレーションおよびシーケンシング・ロジック要素７０３を備える。低位側では、マルチプレクサ１７０２は、経路１７１０および１７１１に接続され、図１８のバス１８０９－１～１８０９－９の一部となる。この経路の上で、マルチプレクサは読み出しおよび書き込みオペレーションで経路１７１０を介して関連するＲＡＭメモリ・バンクとデータを交換する。経路１７１１は、マルチプレクサ１７０２を介して状態コントローラ１８０４が関連するＲＡＭメモリ・バンク１８０３のオペレーションを制御することを可能にする双方向制御経路である。ＲＡＭデータ経路１７１０は、マルチプレクサを介してデータ経路１７０４またはバックグラウンド・アクセス・マルチプレクサ１８０８に届くデータ経路、またはデータ経路１７０５およびバス１８０２を介してアクセス・フロー・レギ

ュレータ１８０１に接続することができる。これらのデータ経路は、読み出しオペレーションと書き込みオペレーションの両方で使用することができる。

【００７３】

マルチプレクサ１７０２の底部のＲＡＭ制御経路１７１１は、経路１７１２ならびにアービトレーションおよび制御ロジック要素１７０３を介して経路１７０７および１７０６に接続されている。マルチプレクサの経路１７１１は、一度に経路１７０７および１７０６の一方のみに接続可能である。経路１７０６に接続された場合、さらに経路１８１０におよび、読み出しと書き込み両方のオペレーションでバックグラウンド・アクセス・マルチプレクサ１８０８および関連するリモートＲＡＭ　１８０６のオペレーションを制御する。経路１７１１が要素１７０３を介して経路１７０７に接続された場合、さらに、バス１８０２を介してアクセス・フロー・レギュレータ１８０１に及ぶ。アービトレーションおよびシーケンシング・ロジック要素１７０３は、読み出しおよび書き込みの両方のオペレーションで状態コントローラ１８０４とのデータ交換後、アクセス・フロー・レギュレータ１８０１を制御するために必要な情報処理機能およびロジック機能を備える。アービトレーションおよびシーケンシング・ロジック１７０３はさらに、バス１７０６および１８１０を介して、バックグラウンド・アクセス・マルチプレクサ１８０８と通信し、リモートＲＡＭ１８０６がＲＡＭメモリ・バンク１８０３からデータを受け取ったときのオペレーションだけでなくリモートＲＡＭ　１８０６がデータを状態コントローラ１８０４に送信し、状態コントローラに関連付けられたＲＡＭメモリ・バンクに入力するオペレーションを制御する。

【００７４】

状態コントローラ１８０４は、関連するＲＡＭメモリ・バンク１８０３と、およびアクセス・フロー・レギュレータ１８０１と、およびバックグラウンド・アクセス・マルチプレクサ１８０８を介してリモートＲＡＭ　１８０６と制御およびデータを交換する４つの高水準機能を備える。これら４つの高水準機能について次に説明する。

【００７５】

図１７の状態コントローラ１８０４によって実行される第１の機能は、読み出しまたは書き込み要求が発生したときにアクセス・フロー・レギュレータ１８０１からの情報の転送に関連する内向きアクセス要求シーケンスを開始し、制御し、そうする際に、関連するＲＡＭメモリ・バンク１８０３を制御し、書き込み要求ではデータをＲＡＭメモリ・バンク１８０３に書き込み、アクセス・フロー・レギュレータ１８０１からの読み出し要求ではＲＡＭメモリ・バンク１８０３からデータを読み出すことである。

【００７６】

図１７の状態コントローラにより実行される第２の機能は、関連するＲＡＭメモリ・バンク１８０３内のバッファ満杯レベルを検出するトリガ信号に応答することである。このトリガは、関連するＲＡＭメモリ・バンク内のバッファが十分に消費されたか枯渇したことを示す。関連するＲＡＭメモリ・バンク内のバッファが十分に消費されている場合、リモートＲＡＭ　１８０６への書き込みがトリガされる。関連するＲＡＭメモリ・バンク内のバッファが十分に枯渇している場合、リモートＲＡＭ　１８０６からの読み出しがトリガされる。

【００７７】

状態コントローラ１８０４により実行される第３の機能は、関連するＲＡＭメモリ・バンク１８０３からリモートＲＡＭ　１８０６への転送を開始し、管理し、さらに、リモートＲＡＭ　１８０６からＲＡＭメモリ・バンク１８０３に戻る逆の方向のデータ転送も管理することである。

【００７８】

状態コントローラ１８０４により実行される第４の機能は、マルチプレクサ１７０２からの信号を待ち、マルチプレクサ１７０２からその信号を受信した後リモートＲＡＭ　１８０６との間の転送を開始することである。

【００７９】

マルチプレクサ１７０２により実行される他の機能は、複数のＲＡＭメモリ・バンク１８０３が同時にリモートＲＡＭ　１８０６へのアクセス権を要求している場合にリモートＲＡＭ　１８０６へのアクセス権をどのビッディングＲＡＭメモリ・バンク１８０３に与えるかを選択することである。マルチプレクサ１７０２により実行される他の機能は、リモートＲＡＭ　１８０６とビッディングＲＡＭメモリ・バンク１８０３との間のオペレーションに関連する転送およびスケジューリング機能を、前記転送の間に依存関係がある場合に、開始することである。このような依存関係は、メモリ・システムとの間のストリーミング化されたアクセスから生じることがある。

【００８０】

マルチプレクサ１７０２により実行されるさらに他の機能は、ＲＡＭメモリ・バンク１８０３を制御し、リモートＲＡＭ　１８０６からの書き込み入力の方向を決めることである。マルチプレクサ１７０２は、リモートＲＡＭ　１８０６へのアクセス権を与え、リモートＲＡＭ　１８０６とＲＡＭメモリ・バンク１８０３との間の情報の経路を設定する。

【００８１】

表２は、ＲＡＭバンク１８０３争奪があると、従来のＳＤＲＡＭに基づくシステムのパフォーマンスが制限される場合があることを示している。図１４の説明から、この制限は厳しいため、システムはＲＡＭメモリ・バンク争奪を回避して設計されていることがわかる。図１４の設計では、ＲＡＭバンクの利用可能性を予想することに基づく争奪とは反対にＲＡＭメモリの全サイクル時間の時間間隔についてアクセスを構造化することによりすべてのサイクルでの争奪を回避している。つまり、ＲＡＭメモリ・バンクは設計上、最適な速度よりも遅い速度で動作するということである。図１２および１３は、バンクの利用可能性を予想するシステムを説明しているが、争奪が発生した場合のために実質的によけいなロジックを必要とする。この実装の問題は、争奪によりパフォーマンスが奪われ、必要なハードウェアを追加したとしても十分なパフォーマンス向上がないという点である。

【００８２】

図１５および１６は、ＲＡＭメモリ・バンクの数を増やして争奪回数を減らすことによるオペレーションの向上を示している。これには、ハードウェアのロジックを増やす必要がある。パフォーマンスの向上とハードウェアの増大との関係は、ハードウェアを受け入れられるだけ追加するとパフォーマンスの違いが目立ってわかるというような関係である。多数のＲＡＭメモリ・バンクへのアクセスを調整するには、図１７に示されているような専用の状態ロジックが必要であり、ハードウェア資源が消費される。これらの資源は、市販のＦＰＧＡの中から見つけられる。パフォーマンスを最大にするためには、マスカブル・データ・バスが好ましい。バースト・データを滑らかに吸収するために広くなければならず、また最小クオンタムのデータを格納しやすくするためにマスク可能でなければならない。このようにして構成可能なメモリは、市販のＦＰＧＡで利用可能である。しかし、このメモリは、前述のように、３００ミリ秒分のＯＣ－７６８オプション・ファイバのバッファリングなど、実質的なバッファリング・ジョブには不十分な小さな量でしか利用できない。さらに、集積回路内では利用可能な領域の量は限られている。ＲＡＭメモリに使用される領域は、状態コントローラ・ロジックには使用できず、その逆もそうである。しかし、ＲＡＭバンクを増やすと、パフォーマンスが向上し、各バンクは専用の状態コントローラを持たなければならない。したがって、バッファリング要件を満たすこととシステムのパフォーマンスとの間には食い違いがある。

【００８３】

このコンフリクトに対する解決策は、ＦＰＧＡ　ＲＡＭバンク１８０３に載せるメモリの量を制限することである。リンク・リストの先頭および末尾のみがアクセスされ、リンク・リストの中間の要素は、リストの先頭に移動するまで、常にアイドル状態である。したがって、リンク・リストの中間要素をボードから移動して離し、ＦＰＧＡ　ＲＡＭバンク１８０３からリモートＲＡＭ　１８０６に移すと、パフォーマンスを向上させることができる。

【００８４】

この妥協策を実装するシステムは、図１７および１８に示されている。
この解決策は、経済的な理由からも賢明な方法である。ＲＡＭメモリの構成可能なバンクは、１ビット当たりの価格が市販のＳＤＲＡＭに比べてかなり高い。リンク・リストの中間要素を保持するリモートＲＡＭ　１８０６がアクセスされていなければ、そのメモリに関連するサイクル時間ペナルティはない。最終的には、リモートＲＡＭ　１８０６にアクセスして、リストの中間への要素の格納および取得を行わなければならない。しかし、ＦＰＧＡ上のメモリの多数のバンクには上述の構成可能な性質があるため、ＳＤＲＡＭＲＡＭバンク１８０３のバースト・モードと互換性のある設計が可能である。メモリの多数のバンクからなるシステムのスループットをＳＤＲＡＭに一致させ、バランスのとれた設計とすることも可能である。これにより、ＳＤＲＡＭ　１８０３をバースト・モードで動作させ、データ・バス上でクロックと同期がとられているデータとともにサイクル時間をパイプライン化して、ＳＤＲＡＭサイクル時間のコストを最小限に抑えることができる。これで、同様に、メモリはメモリが当初設計された動作モードに戻る。

【００８５】
リンク・リスト・バッファは、バックグラウンド・アクセス・マルチプレクサ１８０８を介してリモートＲＡＭ　１８０６に置いたり、取り出したりされる。このため、フォアグラウンド・アクセスを、この追加トラフィックに関係なく、進められる。これは、表２に例示されている確率論的モデルが図１８に示されているように利用可能なフォアグラウンド・バスに依存しているため重要である。フォアグラウンド・バス１８１０で進行中のリモートＲＡＭ　１８０６へのバックグラウンド転送をブロックすると、表２を得るために使用されるモデルはかなり込み入ったものとなる。したがって、パフォーマンスが低下する。バックグラウンド・アクセス・バス１８１０が示されている。

【００８６】
図１８は、本発明を具現化するリンク・リスト・エンジン１８００を開示している。リンク・リスト・エンジン１８００は、ポート１８１７および１８１８に接続されている着信経路１８１２および送信経路１８１３を持つ通信システム１８１１に接続されていることが示されている。このシステムは、プロセッサ１８１４を備え、さらに、経路１８１９を介してアクセス・フロー・レギュレータ１８０１にまで及ぶ経路１８１５および１８１６も含む。動作すると、システム１８１１はリンク・リスト・エンジン１８００のメモリを使用して読み出しおよび書き込みオペレーションを実行し、動作中にポート１８１７および１８１８で必要とするデータを格納する。

【００８７】
アクセス・フロー・レギュレータ１８０１およびバス１８０２は、複数の状態コントローラ１８０４に接続されるが、それぞれ複数のＲＡＭメモリ・バンク１８０３のうちの１つと関連付けられている。アクセス・フロー・レギュレータ１８０１は、ＲＡＭメモリ・バンク１８０３の１つに情報を格納するよう要求する書き込み要求をシステム１８１１から受け取る。アクセス・フロー・レギュレータ１８０１は、それらのアクセス要求を受け取って格納し、データを関連するＲＡＭメモリ・バンク１８０３に入力するため様々な状態コントローラ１８０４に選択的に分配する。このプロセスは、アクセス・フロー・レギュレータ１８０１がシステム１８１１から読み出し要求を受け取ったときにメモリ読み出しオペレーションで逆の方向に動作し、このプロセスにより、状態コントローラ１８０４を介して、要求されたデータを含むＲＡＭメモリ・バンク１８０３が読み出され、状態コントローラ１８０４およびバス１８０２を介して、システム１８１１にデータを送信するアクセス・フロー・レギュレータに適用される。

【００８８】
ＲＡＭメモリ・バンク・デバイス１８０３は、ＲＡＭメモリ・バンク１８０３から情報を受け取り、その情報がＲＡＭメモリ・バンクですぐに必要ない場合に格納しておくことができるリモートＲＡＭ　１８０６により増強された比較的小さなメモリ記憶容量を持つ高速な要素である。リモートＲＡＭ１８０６は、動作中、バックグラウンド・アクセス・マルチプレクサ１８０８およびパス経路１８１０により補助され、マルチプレクサおよび

パス経路はそれぞれ一意的な状態コントローラ１８０４および関連するＲＡＭメモリ・バンク１８０３にまで及んでいる。この手段により、満杯または空になりつつあるＲＡＭメモリ・バンク１８０３は、このことを関連する状態コントローラ１８０４に通知し、このコントローラはさらに、バス経路８１０を介してバックグラウンド・アクセス・マルチプレクサ１８０８と通信することができる。マルチプレクサ１８０８は、状態コントローラ１８０４がＲＡＭメモリ・バンク１８０３から情報を読み出して、リモートＲＡＭ　１８０６に転送し、その情報が出所のＲＡＭメモリ・バンク１８０３によって再び必要になるまで一時的記憶域に置く動作を補助する。このときに、ＲＡＭメモリ・バンク１８０３は、状態コントローラ１８０４に、リモートＲＡＭ１８０６がＲＡＭメモリ・バンクによって必要になりそうな情報を含んでいることを通知する。バックグラウンド・アクセス・マルチプレクサ１８０８および状態コントローラ１８０４を連携動作し、それにより、リモートＲＡＭ１８０６の適切な部分は情報を読み出して出所のＲＡＭバンク１８０３に送り返す。リモートＲＡＭ１８０６は、比較的低速の大容量メモリ要素であり、ＲＡＭメモリ・バンク１８０３からオーバーフローする情報を効率よく格納するか、または情報をアンダーフローになったＲＡＭメモリ・バンク１８０３に供給することができる。

【００８９】

本発明の一態様は、図１８の方法および装置を含み、そこでは、ＲＡＭメモリ・バンク１８０３への書き込みオペレーションは状態コントローラ１８０４を介してリンク・リスト情報をＲＡＭメモリ・バンク１８０３に書き込む工程と、満杯状態に近づくまでＲＡＭメモリ・バンク１８０３への書き込みオペレーションを継続する工程と、状態コントローラ１８０４およびバックグラウンド・アクセス・マルチプレクサ１８０８を介してＲＡＭメモリ・バンク１８０３から新規に受信された情報の一部を同時にリモートＲＡＭ　１８０６に読み出しながらアクセス・フロー・レギュレータ１８０１からＲＡＭメモリ・バンクに追加情報を書き込むことを継続する工程とにより実行される。この情報は、その後出所のＲＡＭメモリ・バンク１８０３により必要になるまでリモートＲＡＭ１８０６内に残される。

【００９０】

図１８のシステムは、ＲＡＭメモリ・バンク１８０３からのデータの読み出しを実行するが、そのために、要求されたデータを含むＲＡＭメモリ・バンク１８０３と関連付けられた状態コントローラ１８０４に信号を送る工程と、状態コントローラ１８０４およびバス１８０２を介して選択されたデータの読み出し結果をアクセス・フロー・レギュレータ１８０１に戻すことを開始する工程と、選択されたＲＡＭメモリ・バンク１８０３の読み出しを継続し、それと同時に、それらの読み出しで注目するデータの選択されたＲＡＭメモリ・バンク１８０３が尽きてしまったこと、およびその後の読み出しオペレーションで必要となる注目するデータの一部が現在、リモートＲＡＭ　１８０６に格納されていることを判別する工程とを使用する。リモートＲＡＭ１８０６からプリフェッチ読み出しを開始し、情報を要求される前に出所のＲＡＭメモリ・バンク１８０３に転送して戻す工程と、選択されたＲＡＭメモリ・バンク１８０３の読み出しを継続し、必要ならば、リモートＲＡＭ１８０６から読み出し中のＲＡＭメモリ・バンク１８０３へのデータの読出しを継続する工程とを使用する。このオペレーションは、読み出しオペレーションについてアクセス・フロー・レギュレータ１８０１により要求された情報全部が履行されるまで継続される。

【００９１】

ＲＡＭメモリ・バンク１８０３の高速メモリ・デバイスおよびリモートＲＡＭ　１８０６の低速大容量メモリは連携動作し、高速ＲＡＭメモリ・バンク１８０３の容量を超えるデータを格納する。リモートＲＡＭ　１８０６は、書き込み中のＲＡＭメモリ・バンクからの書き込みオペレーションでこのデータを受け取るが、その一方で、ＲＡＭメモリ・バンク１８０３はアクセス・フロー・レギュレータ１８０１から高速でさらにデータを受信し続けることができる。このプロセスは、ＲＡＭメモリ・バンク１８０３が最初に高速で読み出されると逆方向に読み出しオペレーションで動作し、ＲＡＭメモリ・バンクで必要

とされリモートＲＡＭ　１８０６に格納されるデータは、ＲＡＭメモリ・バンク１８０３に注目する情報が枯渇すると、プリフェッチ方式でＲＡＭメモリ・バンク１８０３に転送して戻される。読出しオペレーションは継続し、リモートＲＡＭ１８０６内の関連情報すべてを高速ＲＡＭメモリ・バンク１８０３に転送して戻すことができ、アクセス・フロー・レギュレータ１８０１により要求された情報全部が状態コントローラ１８０４およびバス１８０２を介して高速なＲＡＭメモリ・バンク１８０３によりアクセス・フロー・レギュレータ１８０１に送り返されるまで高速データ転送速度で読み出される続ける。または、状態コントローラ１８０４およびＲＡＭメモリ・バンク１８０３は、バックグラウンドでの取得をトリガした前回の読み出しオペレーションと別の書き込みオペレーションを続行することができる。プリフェッチはトリガにより自動実行されるため、特定の状態コントローラ１８０４およびＲＡＭメモリ・バンク１８０３がバックグラウンド転送で占有されている間、他の状態コントローラ１８０４およびＲＡＭメモリ・バンク１８０３は自由に、このバックグラウンド・オペレーションとは無関係にオペレーションを実行できる。

【００９２】

本発明の他の態様では、状態コントローラ１８０４を使用して経路１８２０を介して電位をアクセス・フロー・レギュレータ１８０１に印加し、それぞれの状態コントローラ１８０４に関連付けられているＲＡＭメモリ・バンク１８０３が既存の読み出しまたは書き込みオペレーションで現在使用中かどうか、または新しい読み出しまたは書き込みオペレーションの受信された要求に利用可能かどうかを示す。ＲＡＭメモリ・バンク・デバイス１８０３のメモリ要素は光バスの高速データ転送速度で動作するので、ＲＡＭメモリ・バンク１８０３は、光ファイバ・バスに適した速度で読み出しまたは書き込みオペレーションを実行することができる。その結果、経路１８２０を経由して状態コントローラにより使用中信号がアクセス・フロー・レギュレータ１８０１に印加され、関連するＲＡＭメモリ・バンクが利用可能である、または利用可能でないことを示す。この信号は、読み出しまたは書き込みオペレーションを実行するためにＲＡＭメモリ・バンク１８０３デバイスに必要な数ナノ秒間しか持続しない。したがって、状態コントローラ１８０４によりこの使用中／アイドル電位が経路１８２０に印加される場合、争奪ファシリティによりアクセス・フロー・レギュレータ１８０１およびその要素１８２１はＲＡＭメモリ・バンク１８０３の使用中／アイドル状態を監視することができる。このため、読み書きでの任意の所定の最小時間間隔をＲＡＭメモリ・バンク１８０３に課す従来技術の複雑なロジック回路要素または争奪配置による争奪の遅延は解消される。この手段により、アクセス・フロー・レギュレータ１８０１、経路１８２０、および状態コントローラ１８０４は、光ファイバ・バスのナノ秒転送速度で動作する効率的で高速な争奪ファシリティを実現する。この改善された高速な争奪配置では、ポート１８１７および１８１８に関連付けられた着信および送信キューはＲＡＭメモリ・バンク１８０３の高速要素により処理されるためより高速にデータを交換し処理することができるので、システム１８１１により処理されるデータのスループットを高めることができる。したがって、図１８のリンク・リスト・エンジン１８００は、光ファイバ・リンクと同じ速度でポート１８１７および１８１８により必要とされるデータ・キューの読み書きを実行することができる。

【００９３】

図１９の説明

図１９は、５つのバッファ１から５を含む標準的なリンク・リストを開示している。これら５つのバッファのリンク・リストは、図１８の同じＲＡＭバンク１８０３上には格納されない。その代わりに、これら５つのバッファは、５つの別々のバンク１８０３にランダムに格納される。それぞれのバッファは、システムによって格納され処理される物理的情報またはデータを含む第１の部分を持つ。各バッファの下位部分は、リンク・リストの次のバッファを格納するＲＡＭバンクへのリンク・フィールド・アドレスを含む。図１９のバッファ１は、上位部分に物理的情報を格納し、下位部分に０１００のリンク・フィールド・アドレスを格納する。バッファ１を格納するＲＡＭバンクのアドレスは、バッファ１の右に示されているように０００／００である。

【００９４】
リンク・リストのバッファ２は、バッファ１のリンク・フィールドで指定されているようにＲＡＭアドレス０１／００に格納される。バッファ２の最上位部分は、物理的情報（データ）を含む。下位部分は、ＲＡＭバンクＩＤおよびリンク・リストのバッファ３を格納する場所を指定する０１００１のリンク・フィールド・アドレスを含む。
【００９５】
バッファ３は、バッファ２のリンク・フィールドによって指定されているようにＲＡＭバンク・アドレス０１０／０１に格納される。バッファ３のリンク・フィールドは、リンク・リストの第４のバッファの場所を指定する１１０１０のアドレスを含む。
【００９６】
第４のバッファは、バッファ３のリンク・フィールドによって指定されているようにＲＡＭアドレス１１０／１０に格納される。同様に、リンク・リストの第５のバッファおよび最後のバッファは、バッファ４のリンク・フィールドによって指定されているようにＲＡＭアドレス１０１／１１に格納される。バッファ５の最上位部分は、使用できる自由リストの先頭バッファであることを示す。
【００９７】
リンク・リスト上のバッファは、図１８の別のＲＡＭバンク上にランダムに格納される。このランダム性は、システムのデータ処理および制御オペレーションを効率よく行う上で望ましく、また必要である。このランダム性は、本発明の争奪ファシリティの所望のオペレーションを実現するために特に必要であり有用である。
【００９８】
図２０および２１の説明
図２０は、ＲＡＭメモリ・バンク１８０３内のリンク・リスト・キューの先頭がどのように枯渇し、大容量ＲＡＭメモリ１８０６のアクセスを必要とするかを示している。図に示されているリンク・リスト・キューは、それぞれの順序で、内容Ｑ、Ｆ、Ｒ、およびＺを含むバッファを持つ。図２１では、クロック期間Ａ１の第１のアクセスで、ＲＡＭ１８０３から図２１のクロック期間７および８の内容Ｑを保持するアドレスを読み出す。しかし、アクセスＡ１により、使用するキューの先頭が高速メモリ１８０３から取り出され、リモートＲＡＭ　１８０６に次の要素Ｆが格納される。したがって、アクセスＡ１により、アクセスＡ１にかかわるＲＡＭバンク１８０３の状態コントローラ１８０４を通じてクロック期間２でバックグラウンド要求Ｒｑ１がトリガされる。このリモート・メモリ１８０６アクセス要求は、バックグラウンド・アクセス・マルチプレクサ１８０８により処理され、図２１のクロック期間３でリモート・メモリ・アクセスＡ１Ｂによって返される、リモートＲＡＭ　１８０６はＲＡＭバンク１８０３へのアクセスの流れと同時に発生することに注意されたい。データＤ１Ｂは、クロック期間９および１０でリモートＲＡＭ　１８０６から消費される。このデータはリンクされたリスト要素Ｆである。要素Ｆは、アドレスがＲＡＭバンク５内にある自由リストから取り出された空要素に書き込まれ、アクセスＡ５はクロック・サイクル１１で始まる。リンクは全体を通して保存される。したがって、キューの先頭は、要素Ｆを含み、再び、ハイパフォーマンスＲＡＭバンク１８０３～１８０５に置かれる。要素Ｑを保持しているＲＡＭバンク１８０３～１８０５および内容Ｆを以前保持していたリモート・メモリ要素は、それぞれの自由リストに戻される。
ハイパフォーマンスＲＡＭ　１８０３へのアクセスが無相関であるという性質は温存され、表２に至るモデルはそのままとなるだけでなく、アクセス制御のギャップを作ることがないことにより図２０に示されているように、より効率的に利用できるのではないかという期待が持続する。ギャップは争奪によって生じるという前の議論を思い出そう。相関のあるトラフィック、つまり同じバンクへの順次アクセスが争奪を引き起こすということである。
【００９９】
図２２の説明
図２２は、バッファが図１８の様々なＲＡＭバンク１８０３－１から１８０３－８（こ

れ以降ＲＡＭバンク１８０３）にランダムに格納されるリンク・リストの読み出しを要求するアクセス・フロー・レギュレータ１８０１によって開始される読み出し要求を実行する本発明のプロセスを開示している。アクセス・フロー・レギュレータ１８０１が読み出しオペレーションを要求するプロセッサ１８１４から命令を受け取ったときに工程２２０１でプロセスが開始する。リンク・リストの個別バッファが一度に１つずつ順次読み出され、複数のＲＡＭバンク１８０３のうち様々なバンクにランダムに格納される。各バッファの読み出しには、アクセス・フロー・レギュレータ１８０１による別々の読み出し要求が必要である。

【０１００】

第１の読み出し要求は、工程２２０２で受け取られ、工程２２０３に進み、そこで、ＲＡＭバンク１８０３から読み出される要素の個数に関してしきい値を超えたかどうかを判別する。すでに述べたように、リンク・リストの読み出しでは、先頭バッファが格納されているＲＡＭバンク１８０３からリストの初期バッファ（先頭）を読み出す必要がある。リンク・リストの実行の残り部分で、リンク・リストの中間バッファを読み出す必要があり、それらをリモートＲＡＭ　１８０６に格納し、また取り出してＲＡＭバンク１８０３に格納しなければならない。リモートＲＡＭ　１８０６からＲＡＭバンク１８０３にバッファを戻す効率よい転送には、そのような複数の要求をバックグラウンド・アクセス・マルチプレクサ１８０３に適用し、次に、その工程の実行時に効率のためリモートＲＡＭ　１８０６に適用する必要がある。このような理由から、しきい値検出要素２２０３を用意し、個別に一度に１つずつではなく同時にそのような複数の要求をリモートＲＡＭ１８０６に送るようにする。

【０１０１】

最初に、要素２２０３がしきい値を超えていないことを判別すると仮定しよう。この場合、リモートＲＡＭ　１８０６は即座にアクセスされず、プロセスは工程２２０４に進み、リンク・リストの第１のバッファ（先頭）の読み出し要求により識別されたＲＡＭバンクを読み出す。このバッファの位置は、読み出され、一時的に格納され、プロセスは工程２２０５に進み、そこで、読み出された情報をアクセス・フロー・レギュレータ１８０１に返す。次に、このプロセスは、工程２２０６に進み、ＲＡＭバンク１８０３がアクセス・フロー・レギュレータ１８０１から次のアクセス要求を受け取る準備ができていることを示す。末尾バッファも、リンク・リストがただ１つのバッファで構成されている場合と同じように読み出される。

【０１０２】

次に、要素２２０３は、工程２２０２の新着読み出し要求により、読み出しオペレーションにＲＡＭバンク１８０３内で利用可能なバッファの数に関してしきい値を超えたかどうかを判別すると仮定しよう。この場合、工程２２１１では、先頭バッファを読み出し、工程２２２０にさらに進み、工程２２１１の読み出し情報をアクセス・フロー・レギュレータ１８０１に送信する。次に、工程２２２１では、読み出すＲＡＭ　１８０６内のリストの先頭付近にある中間バッファの読み出しを要求する。これは、リストの先頭の要求を含む。そうする際に、工程２２２１は、リストの新しい先頭バッファの読み出し要求を、指定されたＲＡＭバンク１８０６を処理するバックグラウンド・アクセス・バス１８１０に出す。次に、工程２２２２では、リモートＲＡＭ　１８０６からリストの複数の中間バッファを取り出す。次にこのプロセスは、工程２２２３に進み、リモートＲＡＭ　１８０６が別のアクセスの実行準備ができていることを示す。このプロセスはさらに、工程２２１２に進み、そこで、ＲＡＭバンク１８０３に、指定されたＲＡＭバンク１８０３に書き込まれた工程２２２２中に読み出されたリモートＲＡＭ　１８０６の情報を書き込む。この情報は、指定されたＲＡＭバンク１８０３内のリストの新しい先頭バッファの形成を含む。このプロセスは、次に、工程２２０５に進み、情報をアクセス・フロー・レギュレータ１８０１に送る。このプロセスは、次に、工程２２０６に進み、読み取り要求の完了を表し、またＲＡＭバンク１８０３において次のアクセス要求の準備ができていることを示す。

【０１０３】
図２３の説明
図２３は、アクセス・フロー・レギュレータ１８０１から受け取った書き込み要求を実行するために必要な工程を開示している。このプロセスは、工程２３０１から始まり、そこで、アクセス・フロー・レギュレータ１８０１は書き込み要求をバス１８０２に送る。工程２３０２では、書き込み要求を図１８の状態コントローラ１８０４に入力する。工程２２０３では、リストの末尾付近の書き込まれた要素に関してしきい値を超えたかどうかを判別し、そのしきい値を超えていない場合（新しいリストの場合はそうである）、リストの最後のバッファ（末尾）がＲＡＭバンク１８０３に書き込まれる。その後、このプロセスは、工程２３０５に進み、そこで、リモートＲＡＭ１８０６へのアクセス要求が必要でないことをアクセス・フロー・レギュレータ１８０１に知らせる。次に、このプロセスは、工程２３０６に進み、ＲＡＭバンク１８０３において次のアクセスの実行準備ができていることを示す。

【０１０４】
工程２３０３が書き込み要求のしきい値を超えていることを判別すると仮定しよう。この場合、プロセスは工程２３１１に進み、ＲＡＭバンク１８０３に保持されているリストの末尾を取り出す。そこで、プロセスは工程２３２１に進み、工程２３１１で取り出された末尾バッファをバックグラウンド・アクセス・バス１８１０に出し、リモートＲＡＭ１８０６に入れる。次に、工程２３２２で、リンク・フィールドをリモートＲＡＭ　１８０６内の最後から２番目の最終バッファから工程２３２１で書き込まれたバッファの位置に更新するが、それは、最後から２番目のバッファのリンク・フィールド内に示されているバッファが、工程２３１１および２３２１により、ＲＡＭバンク１８０３からリモートＲＡＭ　１８０６に位置が変更されているためである。工程２３２３では、書き込まれたばかりのリモートＲＡＭ　１８０３はアクセスの準備が整っていることを示す。

【０１０５】
工程２３２１と同時に、工程２３１２では、空のバッファがリンク・リストの末尾に書き込まれ、ＲＡＭバンク１８０３に入る。工程２３２１で書き込まれたバッファへのポインタは、工程２３１２で書き込まれた空のバッファのリンク・フィールドに書き込まれる。このプロセスは、次に、工程２３０６に進み、書き込み要求の完了を表し、またＲＡＭバンク１８０３において次のアクセス要求の準備ができていることを示す。

【０１０６】
図２４の説明
リンク・リスト・ファイルを処理するためのメモリ管理ファシリティについて説明した。本発明の他の可能な実施形態では、開示されているメモリ管理ファシリティはさらに、高速ＲＡＭバンク・メモリ１８０３およびリモート・メモリ１８０６を使用して、リンク・リスト型でないデータ・ファイルを処理することもできる。これは、図２４および２５の処理工程について説明される。

【０１０７】
以下では、図１８の読み出しオペレーションが、高速小容量ＲＡＭバンク１８０３にアクセス・フロー・レギュレータ１８０１から受け取った情報を格納するような方式で動作するように構成されているＲＡＭバンク１８０３およびリモートＲＡＭ　１８０６とともに説明されている図２４の処理工程を説明する。リモートＲＡＭ　１８０６は、大きなファイル用の情報を格納する際のオーバーフローとして使用される。このプロセスは、工程２４０１から始まり、工程２４０２に進み、そこで、アクセス・フロー・レギュレータ１８０１は読み出し要求をバス１８０２に送り、ＲＡＭバンク１８０３に格納されているその情報を要求する。

【０１０８】
工程２４０３で、取り出されるファイルのサイズがＲＡＭ　１８０３の現在の記憶容量を超えるかどうかを判別する。しきい値を超えていなかった場合、プロセスは工程２４０４に進み、そこで、ＲＡＭバンク１８０３は要求されたファイルを読み出し、アクセス・

フロー・レギュレータ１８０１に送り返す。次に、このプロセスは、工程２４０５に進み、状態コントローラ１８０４およびバス１８０２を介してＲＡＭバンク１８０３から読み出された要求情報をアクセス・フロー・レギュレータ１８０１に返す。アクセス・フロー・レギュレータ１８０１は、情報を受け取り、それをプロセッサ１８１４に受け渡し、その際にコントローラでは要求された情報に関連する機能を使用する。次に、このプロセスは、工程２４０６に進み、ＲＡＭバンク１８０３が次のアクセス要求を受け取る準備ができていることをアクセス・フロー・レギュレータ１８０１にアドバイスする。

【０１０９】

要素２４０３で、読み出されるファイルのサイズがしきい値を超えたと判定された場合、プロセスは工程２４１０に進み、そこで、ＲＡＭバンク１８０３にあると思われるファイルの一部を読み出す。このプロセスは、さらに工程２４１１に進み、そこで、読み出された情報がアクセス・フロー・レギュレータ１８０１に返される。プロセスは工程２４１２に進み、リモートＲＡＭ１８０６からの読み出し要求をバックグラウンド・アクセス・バス１８１０に出し、このバスから、バックグラウンド・アクセス・マルチプレクサ１８０８を通じて、要求されたリモートＲＡＭ　１８０６に要求が送られる。

【０１１０】

その後、このプロセスは、工程２４１３に進み、リモートＲＡＭ　１８０６から要求された情報を取り出す。次に、工程２４１５で、工程２４１３で取り出された情報をリモートＲＡＭ　１８０６からＲＡＭバンク１８０３に送り、そこに格納する。次に、このプロセスは、工程２４１４および２４０６に進むが、これらの工程は両方とも、ＲＡＭバンク１８０３が他のアクセス要求を受け取る準備ができていることをアクセス・フロー・レギュレータ１８０１に知らせる。

【０１１１】

図２５の説明

図２５では、ＲＡＭバンク１８０３にすでに存在しているファイルに追加する書き込み要求を説明している。ＲＡＭバンク１８０３およびリモートＲＡＭ　１８０６は連携動作し、図２４の読み出しオペレーションについて説明されている方法と類似の方法でアクセス制御レギュレータ１８０１から受け取った大容量データを格納する。ＲＡＭバンク１８０３は、ファイルに対し既存のが選択された量のデータを格納し、大容量ファイルの残り部分はオーバーフローされ、リモートＲＡＭ　１８０６に書き込まれる。

【０１１２】

このプロセスは工程２５０１から始まり、要素２５０２に進み、そこで、アクセス制御レギュレータ１８０１から受け取った書き込み要求を解析し、書き込む要求されたファイルのサイズがＲＡＭバンク１８０３に格納することが可能な容量を超えているかどうかを判別する。しきい値を超えていない場合、要素２５０２により工程２５０４では、選択されたＲＡＭバンク１８０３にすでに存在している関連ファイルに追加データを書き込む。その後、このプロセスは要素２５０５に進み、そこで、要求されたファイルがＲＡＭバンク１８０３に書き込まれたこと、およびリモートＲＡＭ１８０３への書き込みが不要であることを通知する肯定応答をアクセス制御レギュレータ１８０１に送り返す。

【０１１３】

しきい値を超えたと要素２５０２が判定した場合、プロセスは工程２５１１に進み、ＲＡＭバンク１８０３に格納されているファイルの格納済み部分を読み出す。さらに、工程２５２２に進み、工程２５１１で取り出されたファイルのこの部分をリモートＲＡＭバンク１８０６に書き込むために必要なオペレーションを開始する。工程２５２３では、リモートＲＡＭ　１８０６において再びアクセスの準備ができていることを示す。工程２５１２では、書き込みポインタをＲＡＭバンク１８０３に書き込み、他の部分がＲＡＭ　１８０３に書き込まれるファイルの残り部分を含むリモートＲＡＭ　１８０６のアドレスを関連付けられるようにする。その後、このプロセスは、工程２５０６に進み、ＲＡＭバンク１８０３において他のアクセス要求を受け取る準備ができていることを示す。

【０１１４】

結末

ネットワーク・トラフィック・シェーピングには、柔軟なバッファリング機能が必要である。優先度の高い流れは、他の優先度の低い流れに優先する。例えば、リアルタイム・トラフィックの需要は、ファイル転送などの通常のデータ・トラフィックの需要よりも高い。しかし、トラフィックの一瞬一瞬の特性はランダムである。次に入って来るパケットは、どの流れにも送ることができる。例えば、再び図３を考察しよう。リンク１上でノードＡに向かう内向きパケットのストリームは、リンク２またはリンク３のいずれかへ外向きとなる可能性がある。リンク１上で内向きのトラフィックは、リンク２に向かう数千個もの連続するパケットを伝送し、リンク３に向かう単一のパケットを伝送し、その後、数千個の連続パケットが再び、リンク２に向けられる。パケットを破棄しないこと、またケーブル切断などの大惨事からの復旧をサポートするという要件に加えて、遅延時間を変更できる、スケジュールされた外向きトラフィック・シェーピング・アルゴリズムをサポートしていなければならない。したがって、トラフィックの着信と送信の両方の性質は複雑であり、柔軟なバッファリングを必要とする。

【０１１５】

内向きストリームと外向きストリームに対するバッファ処理のバランスをとるために最も効率のよい配置は、ハードウェアによるリンク・リスト・エンジンである。しかし、ハードウェアによるリンク・リスト・エンジンの現在の実装は、大容量で低速か、または小容量で高速のいずれかである。本発明の改良されたリンク・リスト・エンジンは、安価にギガバイト・クラスのバッファ記憶容量が得られ、市販の半導体メモリの最大スループットで動作するため、従来のリスト・エンジンよりも優れている。したがって、本発明の改良されたリンク・リスト・エンジンは、最新技術のものと同じ大容量であり、実質的に高速化されている。速度がアップすることは、改良されたリンク・リスト・エンジンでは大容量（つまり、ＯＣ－７６８）ファイバ回線をサポートしており、同じ規模のハードウェア資源、つまりＦＰＧＡおよびＳＤＲＡＭを利用できるため、トラフィック・シェーピング・アプリケーションには魅力的である。改良されたリンク・リスト・エンジンは、バッファがギガバイト程度の深さをとれるため、大容量の回線をサポートするようにできる。

【０１１６】

本発明では、市販のダイナミック・メモリのバースト・モードをサポートしている。このサポートは、本発明の２つの面で明らかである。第１に、市販のダイナミックＲＡＭへのアクセスが連続アクセスであり、続いている場合、本発明では、市販のダイナミック・メモリのデータ・ピンを占有し、取り出された情報を読み出すか、または格納すべき情報を書き込み、そうしながら、次の連続アクセスを開始する。第２に、本発明では、市販のダイナミックＲＡＭに書き込まれるバッファおよび市販のダイナミックＲＡＭから読み出したバッファを状況に応じてキャッシュする。

【０１１７】

本発明では、現在のアクセス要求に対するデータ・ピンの処理と次のアクセス要求の開始を同時に行うことにより、市販のダイナミックＲＡＭのデータ・バスのデータ・バス・ラッチ機能を利用する。この機能を使用すると、データ・バスの速度で、通常、１６６メガヘルツ以上で、ダイナミック・メモリのロー全体を読み出すか、または書き込むことができる。このローは、長さは１２８ビットとすることができるが、ダイナミック・メモリ上で利用できるデータ・バス・ピンは、８本だけである。したがって、メモリは内部的に、読み出されるメモリのローをラッチし、この情報を一度に８ビットずつピンに送出するか、または一度に８ビットずつ内向きデータをラッチしてから、ロー全体を内部メモリ・アレイに書き込む。本発明では、このようにしてアクセスをオーバーラップさせることにより、データ・バス上にデータを出しっぱなしにすることで、市販のダイナミック・メモリからの連続読み出しが図１８のＲＡＭバンク１８０３などの本発明のＲＡＭバンクと互換性のある速度で実行されるようにする。これは、多数の長いバッファのリストに連続してアクセスする場合に有用である。このような場合、市販のダイナミック・メモリからストリーミングで情報を出力しなければならない。市販のダイナミック・メモリへのこのよ

うな拡張された一連のアクセス時にパフォーマンスを維持するために、市販のダイナミック・メモリとのインターフェースの定常状態パフォーマンスは他のメモリと整合性がとれていなければならない。

【０１１８】

書き込まれるバッファと読み出されたバッファをキャッシュすることにより、市販のダイナミック・メモリを効率よく使用することができる。市販のダイナミック・メモリにはロー毎にアクセスするのが最も効率的である。しかし、ローは、１２８ビット以上のオーダーと、通常大きい。格納されるまたは取り出される個々のバッファは、１６ビットのみで構成できる。したがって、十分な情報が集まり完全なローを構成するまでバッファをキャッシュした場合、市販のダイナミック・メモリへの書き込みアクセスにより、完全なローが転送され、これが最も効率的である。同様に、読み出し後、完全なローがキャッシュされた場合、市販のメモリに対しロー毎に１回アクセスするだけでよく、最も効率的である。

【０１１９】

したがって、書き込みおよび読出しをキャッシュすると、実質的なパフォーマンスの向上が得られるが、これは、市販のダイナミック・メモリに対するアクセス回数が少なくて済み、アクセス回数を減らすことで、市販のダイナミック・メモリに対する争奪が少なくなるためである。

【０１２０】

本発明では、多数の同時リンク・リストをサポートしている。多数の同時リンク・リストのうちの任意のリンク・リストとのそれぞれの相互作用は、多数の同時リンク・リストの残りとは完全に独立している。

【０１２１】

添付の請求項では、ＲＡＭバンク１８０３を高速メモリと、またリモート・メモリ１８０６を大容量メモリと特徴付けている。

【０１２２】

上記の説明では、本発明の可能な実施例を開示している。当業者であれば、請求項に規定されているとおり文字通りに、または均一者の原則に従って、本発明を侵害する他の実施形態を設計することが可能であり、また設計するであろうことは予想される。

【図面の簡単な説明】

【０１２３】

【図１】マルチノード・ネットワークを開示する図である。

【図２】１つのノードを含むハードウェア要素を開示する図である。

【図３】ブロッキングのある図１のマルチノード・ネットワークを開示する図である。

【図４】リンク・リスト・バッファの配置を開示する図である。

【図５】リンク・リスト・バッファの配置を開示する図である。

【図６】リンク・リスト・バッファの配置を開示する図である。

【図７】図３のノード上の仮説的なトラフィック状態を開示する図である。

【図８】複数のＲＡＭメモリ・バンクで処理されるネットワークによるアクセス要求の処理を開示する図である。

【図９】複数のＲＡＭメモリ・バンクで処理されるネットワークによるアクセス要求の処理を開示する図である。

【図１０】制御バスおよびデータ・バスに接続されている４つのＲＡＭメモリ・バンクを開示する図である。

【図１１】図８および９に示されているように図１０のＲＡＭメモリ・バンクがアクセス要求を処理するプロセスを説明するタイミング図である。

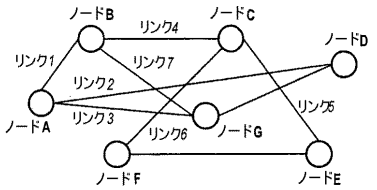
【図１２】４つのＲＡＭメモリ・バンクを持つリンク・リスト・プロセッサにより制御されるメモリ・システムを開示する図である。

【図１３】図１２のシステムのオペレーションを例示するタイミング図である。

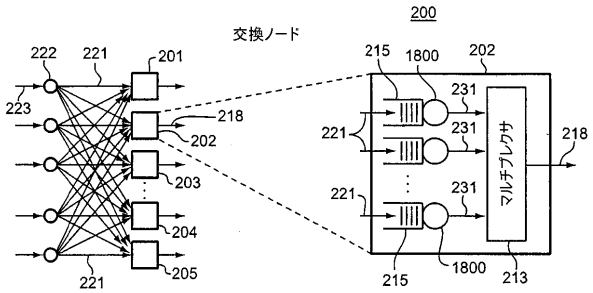
【図１４】図１２のシステムの他のオペレーションを例示する他のタイミング図である。

【図１５】８つのＲＡＭメモリ・バンクを持つプロセッサにより制御されるリンク・リスト処理システムを開示する図である。
【図１６】図１５のシステムのオペレーションを例示するタイミング図である。
【図１７】図１８の状態コントローラ１８０４の要素を開示する図である。
【図１８】本発明を具現化するプロセッサにより制御されるＲＡＭメモリ・システムを開示する図である。
【図１９】本発明によるリンク・リスト・バッファの配置を開示する図である。
【図２０】バッファをリンク・リストの新しい先頭として作成することを例示する読み出しオペレーションを開示する図である。
【図２１】図１８のシステムのオペレーションを例示するタイミング図である。
【図２２】本発明のオペレーションを例示する流れ図である。
【図２３】本発明のオペレーションを例示する流れ図である。
【図２４】本発明のオペレーションを例示する流れ図である。
【図２５】本発明のオペレーションを例示する流れ図である。

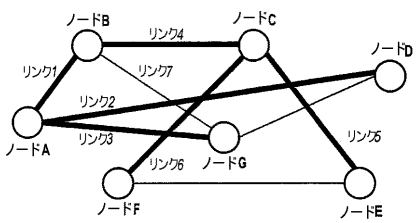
【図１】



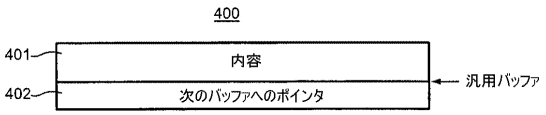
【図２】



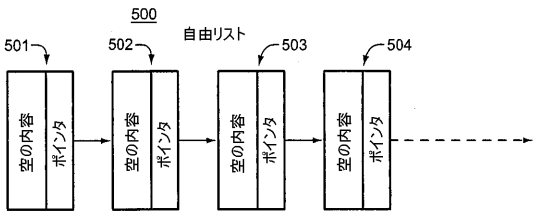
【図３】



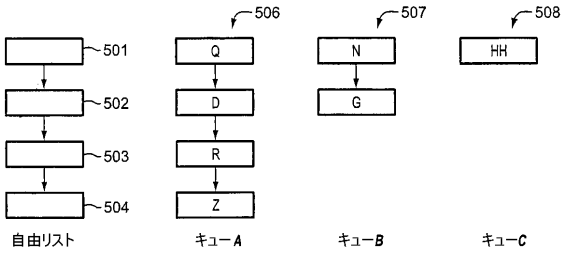
【図４】



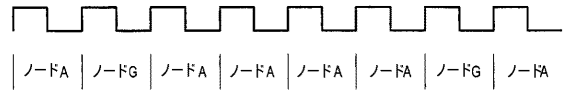
【図５】



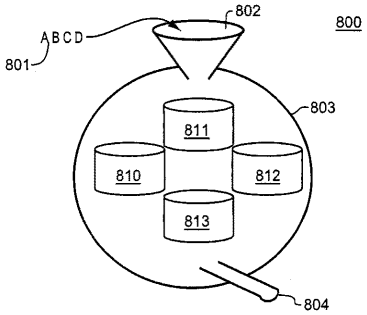
【図６】



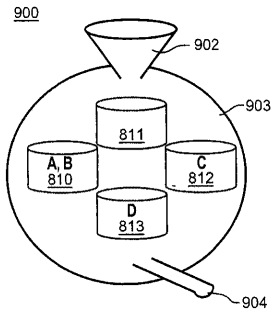
【図７】



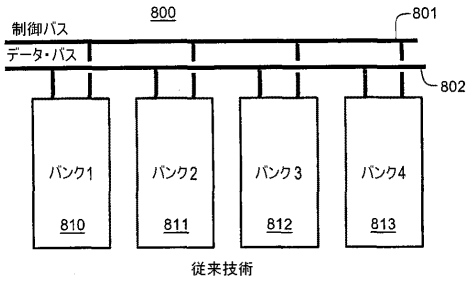
【図８】



【図９】

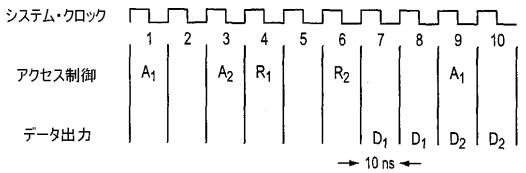


【図１０】

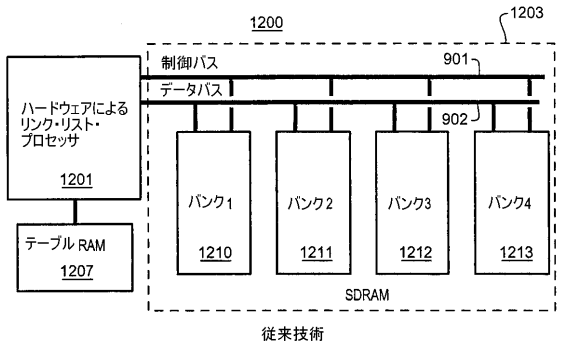


従来技術

【図１１】

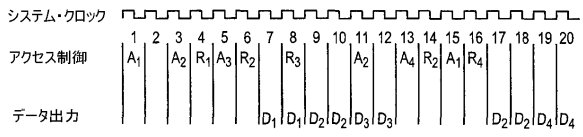


【図１２】

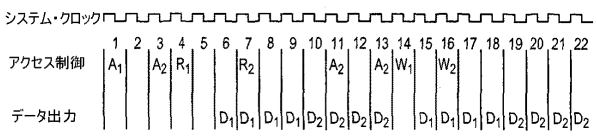


従来技術

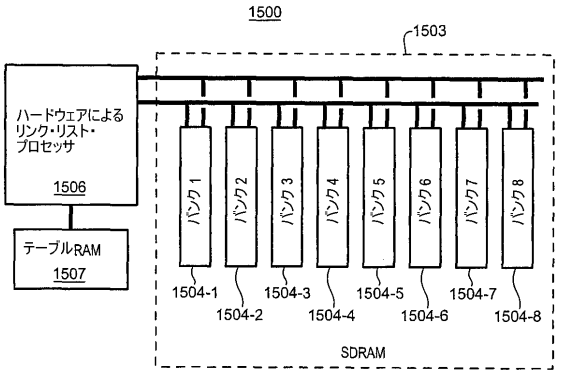
【図１３】



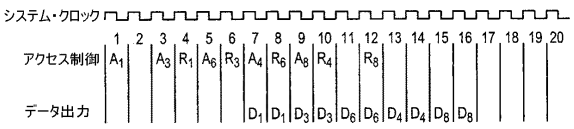
【図１４】



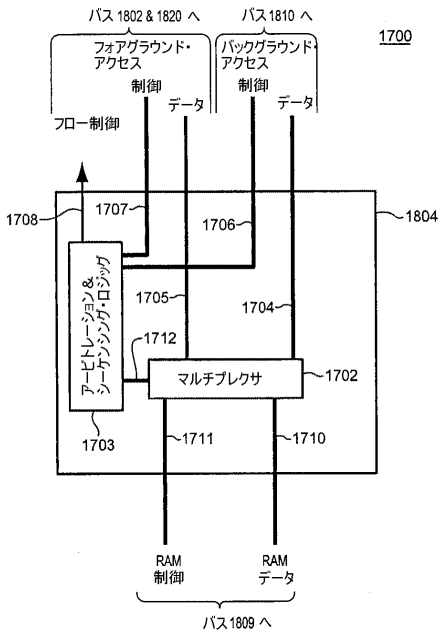
【図１５】



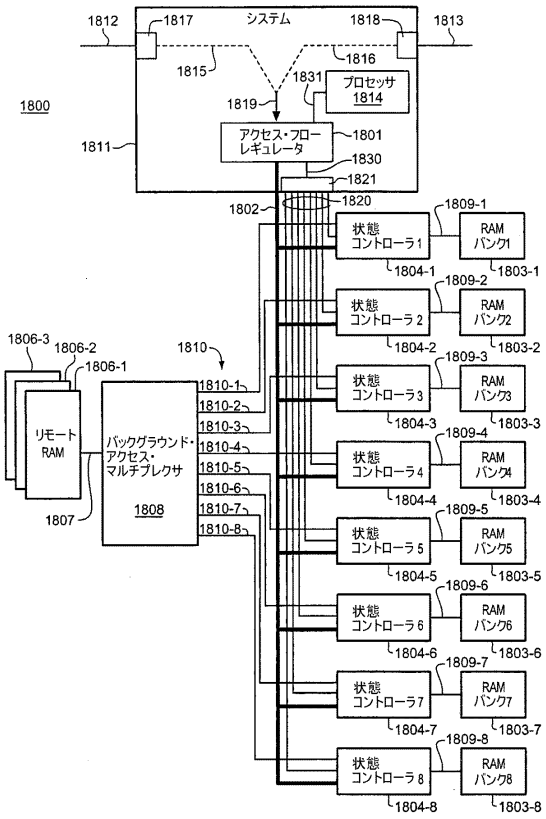
【図１６】



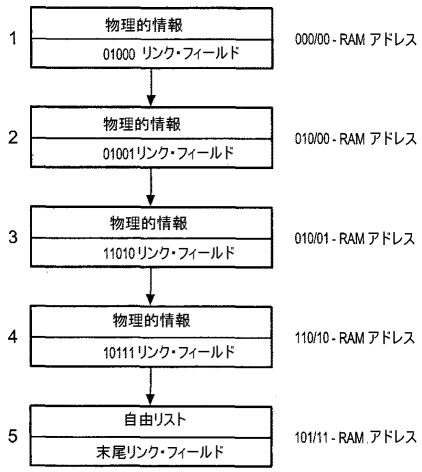
【図１７】



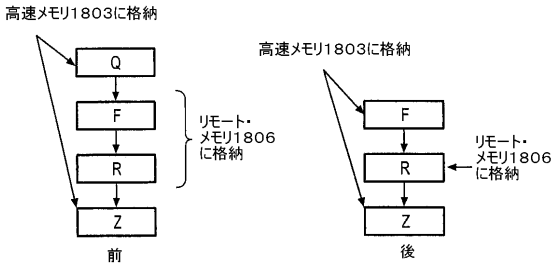
【図１８】



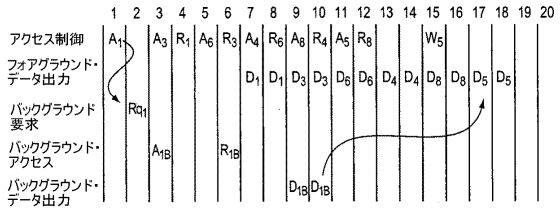
【図１９】



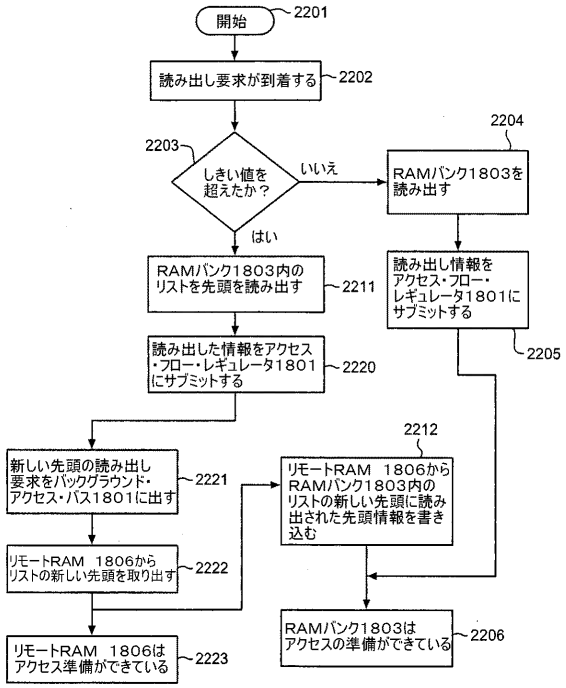
【図２０】



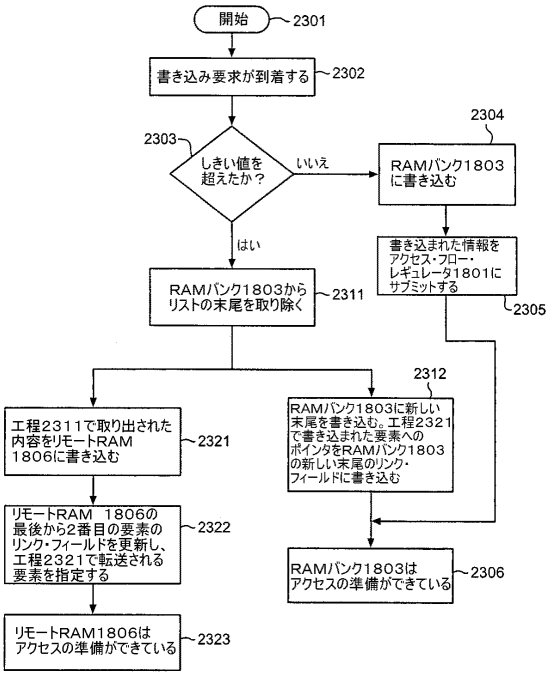
【図２１】



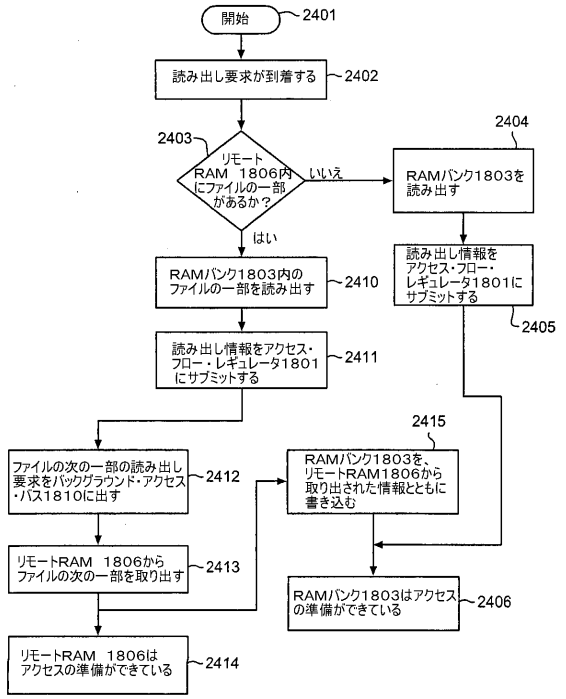
【図２２】



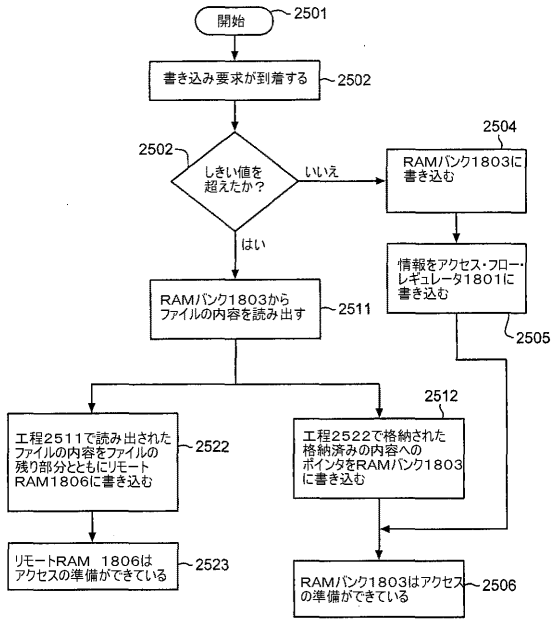
【図２３】



【図２４】



【図２５】



フロントページの続き

(74)代理人 100101498
弁理士 越智 隆夫
(74)代理人 100096688
弁理士 本宮 照久
(74)代理人 100104352
弁理士 朝日 伸光
(74)代理人 100128657
弁理士 三山 勝巳
(72)発明者 ピーター ジェー. ジーヴァース
アメリカ合衆国 ６０５４０ イリノイス, ネイパーヴィル, ブリドルスパー ドライヴ ４９８

審査官 坂東 博司

(56)参考文献 米国特許第６７９８７７７（ＵＳ，Ｂ１）
米国特許出願公開第２００１／０００８１５（ＵＳ，Ａ１）
特開平１１－３５９６（ＪＰ，Ａ）
特開２００２－２８７７５７（ＪＰ，Ａ）
特表２００４－５２３０１７（ＪＰ，Ａ）
特開平１０－２７６２２０（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｇ０６Ｆ １３／３８
Ｇ０６Ｆ １２／０２
Ｇ０６Ｆ １２／０６
Ｈ０４Ｌ １３／０８