

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第4599172号

(P4599172)

(45) 発行日 平成22年12月15日(2010.12.15)

(24) 登録日 平成22年10月1日(2010.10.1)

(51) Int.Cl.

F I

G 0 6 F 12/02 (2006.01)

G 0 6 F 12/02 5 4 O

G 0 6 F 12/02 5 1 O M

請求項の数 17 (全 13 頁)

(21) 出願番号	特願2004-566587 (P2004-566587)	(73) 特許権者	504107339
(86) (22) 出願日	平成15年12月19日(2003.12.19)		エミュレックス デザイン アンド マニ
(65) 公表番号	特表2006-513493 (P2006-513493A)		ュファクチュアリング コーポレーション
(43) 公表日	平成18年4月20日(2006.4.20)		アメリカ合衆国 9 2 6 2 6 カリフォル
(86) 国際出願番号	PCT/US2003/040967		ニア州 コスタ メサ スーザンストリー
(87) 国際公開番号	W02004/063850		ト 3 3 3 3
(87) 国際公開日	平成16年7月29日(2004.7.29)	(74) 代理人	100078282
審査請求日	平成18年11月14日(2006.11.14)		弁理士 山本 秀策
(31) 優先権主張番号	10/340,078	(74) 代理人	100062409
(32) 優先日	平成15年1月9日(2003.1.9)		弁理士 安村 高明
(33) 優先権主張国	米国 (US)	(74) 代理人	100113413
			弁理士 森下 夏樹

最終頁に続く

(54) 【発明の名称】 フリーバッファプールを使用することによるメモリの管理

(57) 【特許請求の範囲】

【請求項 1】

第 1 の数のエントリーを有するバッファインデックスの第 1 フリーリストを共有メモリに記憶する工程であって、各バッファインデックスは共有メモリ中のバッファのアドレスに対応する、工程と、

巡回待ち行列状態である、第 2 の数のエントリーを有するバッファインデックスの第 2 フリーリストを、バッファインデックスの第 1 フリーリストからバッファインデックスを予め取り出すことによって、第 2 メモリに記憶する工程であって、該第 2 の数のエントリーは、該第 1 の数のエントリーより少ない、工程と、

第 1 プロセッサから第 1 バッファ割り当てコマンドを受信する工程であって、該第 1 バッファ割り当てコマンドは該共有メモリ内のバッファのプールに関連するレジスタアドレスを含む、工程と、

該第 2 フリーリスト内のフリーバッファに対応するバッファインデックスの利用可能性に基づき、バッファがバッファプール内で利用可能であるかを判定する工程と、

バッファが利用可能であると判定された場合、該バッファを該第 1 プロセッサに割り当てる工程と

を含むことを特徴とする方法。

【請求項 2】

前記第 2 フリーリストからのバッファインデックスを、バッファの割り当て解除に続き、前記第 1 フリーリストに書き戻す工程をさらに含むことを特徴とする、請求項 1 記載の方

10

20

法。

【請求項 3】

バッファコンテキスト情報を前記第 2 メモリに記憶することをさらに含み、前記バッファコンテキスト情報はベースプールアドレス、可変バッファサイズおよび可変バッファの数のうちの少なくとも 1 個を含み、該バッファコンテキスト情報は、前記バッファがバッファプール内で利用可能であることを判定する工程において用いられることを特徴とする、請求項 1 記載の方法。

【請求項 4】

前記バッファプール中のバッファのサイズに基づき、バッファインデックス値またはバッファポインタアドレスを決定することをさらに含むことを特徴とする、請求項 3 記載の方法。

10

【請求項 5】

前記バッファプール中のバッファの数に基づき、バッファインデックス値またはバッファポインタアドレスを決定することをさらに含むことを特徴とする、請求項 3 記載の方法。

【請求項 6】

前記割り当てる工程は、

前記割り当てられたバッファに対応するバッファポインタアドレスを前記第 1 プロセッサに送ることをさらに含むことを特徴とする、請求項 1 記載の方法。

【請求項 7】

バッファプールに関連する前記アドレスは、該バッファプールに関連するアドレス範囲内のアドレスを含むことを特徴とする、請求項 6 記載の方法。

20

【請求項 8】

前記方法は、共有メモリに参照カウント値を記憶することをさらに含み、該参照カウント値は、バッファにアクセスするプロセッサの数に対応することを特徴とする、請求項 1 記載の方法。

【請求項 9】

第 2 プロセッサからバッファインクリメントコマンドを受信する工程であって、該バッファインクリメントコマンドは、前記割り当てられたバッファに関連するアドレスを含む工程と、

該割り当てられたバッファの参照カウントを更新する工程と
をさらに含むことを特徴とする、請求項 8 記載の方法。

30

【請求項 10】

前記バッファプール内の複数のバッファの要求を特定する割り当てコマンドを受信する工程と、

該バッファプール内に、少なくとも 2 個のバッファを割り当てる工程と
をさらに含むことを特徴とする、請求項 8 記載の方法。

【請求項 11】

複数のプロセッサのうちの少なくともひとつから、割り当て解除コマンドを受信する工程と、

前記割り当てられたバッファの参照カウント値を更新する工程と
をさらに含むことを特徴とする、請求項 8 記載の方法。

40

【請求項 12】

システムバスと、

該システムバスに接続された複数のプロセッサと、

該複数のプロセッサからデータを送受信するために、該システムバスに接続され、バッファのプールおよび参照カウント値のアレイを含む、共有メモリであって、参照カウントアレイ内の各参照カウント値は、バッファプール内のバッファに対応する、共有メモリと

、
共有メモリ内に記憶された第 1 の数のエントリーを有するバッファインデックスの第 1 フリーリストであって、該第 1 フリーリスト内の各バッファインデックスは該バッファプ

50

ール内のバッファに対応する、第1フリーリストと、

該複数のプロセッサのうちの少なくとも1つからバッファ割り当てコマンドを受信し、プロセッサにバッファ割り当て応答を送るために、該システムバスに接続されたバッファマネージャと、

を含む、システムであって、

該バッファマネージャは、該複数のプロセッサの1つからのバッファ割り当てコマンドに
応答して、該バッファプールからのバッファを割り当て、かつ、共有メモリへ/からデ
ータを送受信するために作動し、バッファマネージャは、バッファインデックスの該第1
フリーリスト内の第1の数のエントリーより少ない第2の数のエントリーを有するバッ
ファインデックスの第2フリーリストを局部メモリに記憶する局部メモリと、該複数のプロ
セッサから受信されたコマンドを受信および復号し、かつ、該局部メモリに記憶されたバ
ッファの該第2フリーリストにアクセスし、該参照カウンタレイ内の参照カウンタを更新
するために作動するコマンド復号論理ブロックとを含むことをすることを特徴とするシ
ステム。

10

【請求項13】

前記局部メモリに記憶されたプールコンテキスト情報であって、プールコンテキスト情報
は前記共有メモリに規定されたバッファプールに対応し、該プールコンテキスト情報は、
プールベースアドレス、バッファプールのサイズおよびバッファプールにおけるエントリ
ーの数のうちの少なくとも1個を含む、プールコンテキスト情報をさらに含むことを特徴
とする、請求項12記載のシステム。

20

【請求項14】

復号論理は、前記プールコンテキスト情報に基づいて、バッファインデックスおよびバッ
ファポインタアドレスを決定するように動作する論理をさらに含むことを特徴とする、請
求項13記載のシステム。

【請求項15】

機械により実行されたとき、

各々が共有メモリ中のバッファのアドレスに対応するバッファインデックスであって、第
1の数のエントリーを有するバッファインデックスの第1フリーリストを該共有メモリに
記憶し、

該第1の数のエントリーより少ない第2の数のエントリーを有するバッファインデック
スの第2フリーリストを、該第2のフリーリスト上に記憶されるべきバッファインデック
スの該第1フリーリストからバッファインデックスを予め取り出すことによって、第2メ
モリに記憶し、

30

該共有メモリ内のバッファのプールに関連するレジスタアドレスを含む第1バッファ割
り当てコマンドを第1プロセッサから受信し、

該第2フリーリスト内のフリーバッファに対応するバッファインデックスの利用可能性
に基づき、バッファが該バッファプール内で利用可能であるかを判定し、

バッファが利用可能であると判定された場合、該バッファプール内のバッファを該第1
プロセッサに割り当てる、

という結果を生じる指示を記憶した機械読み取り可能な記憶媒体。

40

【請求項16】

機械により実行されたとき、

バッファインデックスを、前記第2フリーリストからバッファの割り当て解除に続き、
前記第1フリーリストに書き戻す、

という結果を生じる指示をさらに含むことを特徴とする、請求項15記載の機械読み取り
可能な記憶媒体。

【請求項17】

機械により実行されたとき、

前記バッファプール内のバッファのサイズに基づき、バッファインデックス値またはバ
ッファポインタアドレスを決定する、

50

という結果を生じる指示をさらに含むことを特徴とする、請求項 15 記載の機械読み取り可能な記憶媒体。

【発明の詳細な説明】

【技術分野】

【0001】

本出願は、メモリの管理に関する。

【背景技術】

【0002】

(背景)

マルチプロセッサシステムは、共有メモリを含むことができる。すなわち、同一のメモリに対して、システム内の 2 個以上のプロセッサがアクセス（読み出しまたは書き込み）することができる。共有メモリは、論理的にバッファに区分することができる。

【発明の開示】

【課題を解決するための手段】

(説明)

図 1 は、プロセッサ 20 - 29 と、メモリ 30 と、局部メモリ 41 を有するバッファマネージャ (BMGR) とを含むマルチプロセッサシステム 10 を示す。プロセッサ 21 - 29、メモリ 30 および BMGR 40 はそれぞれ、システムバス 50 に接続している。作動しているとき、プロセッサ 20 - 29 および BMGR 40 はそれぞれ、メモリ 30 にアクセスすることができる。例えば、メモリ 30 に対して、データの読み込みおよび / または書き込みを行うことができる。一実施形態において、メモリ 30 はバッファプール 60、70 に論理区分され、各プールは同一サイズのバッファのセットを含む。本例において、バッファ 61 - 65 はプール 60 に含まれ、バッファ 71 - 73 はプール 70 に含まれる。またメモリ 30 は、バッファインデックス 91 - 95 を保持するバッファインデックスアレイ 90 と、R_CNT 101 - 105 を保持する参照カウンタ (R_CNT) アレイ 98 とを含む少なくとも 1 個のバッファ情報アレイ 80 を記憶している。本実施形態において、バッファインデックス 91 - 95 および R_CNT 101 - 105 は、バッファプール 60 のバッファ 61 - 65 にそれぞれ対応する。バッファインデックスアレイ 90 は、バッファインデックス 91 - 95 の「フリーリスト」と呼ぶことができる。以下に説明するように、バッファインデックス 91 - 95 のフリーリストおよび R_CNT 101 - 105 は、プール内でバッファの割り当ておよび割り当て解除を管理するために、BMGR 40 によって使用される。BMGR 40 は、バッファインデックスアレイ 90 からのバッファインデックスの各フリーリストのサブセットを、局部メモリ 41 に保存する。BMGR 40 は、プール割り当て / 割り当て解除レジスタ 34 のセットを含み、各割り当て / 割り当て解除レジスタは、メモリ 30 のバッファプールに対応する。システム 10 の動作中、プロセッサは、メモリ 30 のバッファプールに対応する割り当て / 割り当て解除レジスタ 34 を特定する BMGR 40 に読み取りコマンドを送ることにより、バッファ割り当てを要求することができる。これに対応して、BMGR は、要求を行ったプロセッサに対し、バッファのバッファポインタアドレスを送る。バッファポインタアドレスは、局部メモリ 41 に記憶されたバッファインデックスのフリーリストのバッファインデックスに基づいている。

【0004】

本明細書で説明するように、一実施形態において、バッファインデックスアレイ 90 からのバッファインデックスの各フリーリストのサブセットだけが、局部メモリ 41 に記憶される。このようにして、BMGR 40 の局部メモリ 41 のサイズを減少させることができる。局部メモリ 41 に記憶されたフリーリストインデックスは、いわゆるフリーリスト巡回待ち行列に記憶される。巡回待ち行列とは、データ（またはアドレス）を待ち行列の始めに上書きすることができるよう待ち行列が巡回するようになっており、このときデータ（またはアドレス）が待ち行列の第 1 配置から始まり最後部配置に至るまで連続する場所に記憶されるようになっている待ち行列のことをいう。システム 10 の一実施形態に

10

20

30

40

50

において、フリーリスト巡回待ち行列は、2個のポインタ、「先頭」ポインタおよび「最後尾」ポインタにより参照される。先頭ポインタは、割り当て可能な局部メモリ41内の次のバッファインデックスに指定するために使用され、最後尾ポインタは、メモリ30に書き戻すことができる局部メモリ41内の次のバッファインデックスに指定するために使用される。

【0005】

図2は、バッファを要求を行ったプロセッサ21-29に割り当てるためにBMGR40が行うことができるプロセス100を示す。プロセス100の実行中、BMGR40は、プロセッサ21-29からのコマンドを待っているアイドル状態(110)にある。受信した割り当てコマンド(112)に関して、プロセス100で、バッファが割り当て可能かどうか(例えば、バッファインデックスがBMGR40フリーリスト待ち行列上で使用可能か)について判定する(115)。バッファインデックスが使用可能であると判定されると、BMGR40は、割り当てられたバッファに対応するバッファポインタアドレスを第1プロセッサに送り(120)、先頭ポインタをインクリメントし(122)、フリーリスト待ち行列上の次のバッファインデックスを指し示す。バッファインデックスが利用不可であると判定されると、BMGR40は、ヌルポインタ(例えば、ゼロ値)を第1プロセッサに返す(116)。プロセス100には、より多くのバッファインデックスを予め取り出し、局部メモリ41におけるフリーリスト巡回待ち行列上のバッファインデックスを十分な数に維持する必要があるかを判定すること(125)と、より多くのバッファインデックスが必要と判定された(125)場合に、メモリ30から追加のバッファインデックスを予め取り出すこと(126)とが含まれる。

【0006】

プロセス100を実行することにより、2個以上のプロセッサ21-29は、バッファプール内の特定のバッファに共通にアクセスすることができる。本実施形態において、第1プロセッサが割り当てバッファを有する場合、第1プロセッサは、第2プロセッサに割り当てバッファへのアクセスをさせることができる。例えば、第1プロセッサは、割り当てバッファのアドレスを第2プロセッサに送ることができる。第2プロセッサが割り当てバッファへアクセスを開始するとほぼ同時に、第2プロセッサは、バッファインクリメントコマンドをBMGR40に送ることができる。さらにプロセス100を参照すると(図2参照)、受信した(130)バッファインクリメントコマンドに関して、BMGR40は、ポインタアドレスに対応する適切なバッファプールを判定し(135)、R_CNTを読み込み、'n'を追加し(例えば、R_CNTをインクリメントし)、割り当てバッファに対応するR_CNTを書き戻す(140)。第1および第2プロセッサは、引き続き、割り当てバッファにアクセスすることができる。プロセッサが割り当てバッファにアクセスする必要がなくなれば、当該プロセッサは割り当て解除コマンドをBMGR40に送ることができる。受信した(150)割り当て解除コマンドに関して、BMGR40は、割り当て解除コマンドに対応するバッファプールを決定し(155)、対応するR_CNTをメモリ30から読み出し(157)、R_CNTから'n'を減算(157)した(例えば、R_CNTをデクリメントした)後、対応するR_CNTが0であるかを判定し(160)、対応するR_CNTが0であると判定された場合、プロセス100は、先の判定(160)において対応するR_CNTが1であるかを判定し(165)、フリーリストの最後尾ポインタをインクリメントし(175)、フリーリストがバッファインデックスの書き戻しを必要とするかを判定し(18)、局部メモリ41のバッファインデックスのフリーリストからバッファインデックスを共有メモリ30に書き戻し(185)、コマンドを待つ(110)ために帰還する。

【0007】

このようにバッファを管理することにより、プロセッサは、1個の割り当てコマンド(例えば、1個の読み出しコマンド)によりバッファ割り当てを要求することができ、1個の割り当て解除コマンド(例えば、1個の書き込みコマンド)によりバッファ割り当て解除を要求することができる。また、プロセッサに共有バッファ情報のオーバヘッドを管理

10

20

30

40

50

させることを要求することなく、複数のプロセッサ間でバッファを共有することができる。さらに、このバッファ管理方法により、プロセッサは、B M G R 4 0 が要求を処理する間、システムバスおよび/またはプロセッサを停止させることなく、バッファの割り当てまたは割り当て解除を要求することができる。さらに、プロセス 1 0 0 には、共有メモリ 3 0 からバッファインデックスを予め取り出し、局部メモリ 4 0 へ記憶することが含まれる。従って、プロセッサからの割り当て要求が連続すれば、プロセッサからの要求があったときのみバッファインデックスを取り出すときに要求する場合より短い時間で処理することができる。これに対して、バッファが割り当て解除されるに従って、B M G R 4 0 は、バッファインデックスを、フリーリストからバッファインデックスアレイ 9 0 に書き戻すことができる。

10

【 0 0 0 8 】

バッファ割り当てコマンドを、バッファプールの割り当て/割り当て解除コマンドレジスタ 3 4 への読み込みとして実行することができる。バッファマネージャ 4 0 は、バッファポインタアドレスを含む要求プロセッサへ書き込みコマンドを送ることにより、割り当てコマンドに対応する。バッファ割り当て解除コマンドは、バッファプールの割り当て/割り当て解除コマンドレジスタ 3 4 への書き込みコマンドとして実行することができる。

【 0 0 0 9 】

一実施形態において、割り当て/割り当て解除双方のコマンドについて、1 個の割り当て/割り当て解除レジスタを指定することができる。一実施形態において、割り当て/割り当て解除レジスタ 3 4 は、各レジスタがメモリ 3 0 のバッファプールに対応する 3 2 個のレジスタとして実施される。しかしながら、これより多いまたは少ないバッファプールおよび対応する数の割り当て/割り当て解除レジスタを設けることが可能である。

20

【 0 0 1 0 】

一実施形態において、各バッファプールは対応するベースアドレスを有し、本例においては、p o o l 6 0 _ b a s e および p o o l 7 0 _ b a s e が、各プール 6 0、7 0 内の第 1 バッファのメモリ 3 0 内の配置を識別する。また、各バッファプールは、プールの特性を識別するために対応する変数を含むことができる。例えば、“b u f f _ s i z e X”変数は、プールにおける各バッファのサイズ(例えば、メモリ配置の数)を示すために使用することができ、“#_b u f f X”変数は、プールに含まれるバッファの数を示すために使用される。システム 1 0 の作動中、b u f f _ s i z e X 変数および #_b u f f X 変数は、アレイ 9 0 に記憶されたバッファインデックス値を算出するために B M G R 4 0 により使用され、また、バッファインデックス値から対応するバッファポインタ値を決定するために B M G R 4 0 により使用される。

30

【 0 0 1 1 】

より詳細には、バッファポインタアドレスを含むコマンドにより、バッファにアクセス(例えば、読み出しおよび書き込み)することができる。しかしながら、B M G R 4 0 上のフリーリスト巡回待ち行列は、バッファインデックスを記憶するために使用される。従って、バッファをプロセッサに割り当てると、フリーリストからのバッファインデックスは、要求プロセッサへ返されるバッファポインタアドレスに変換する。一実施形態において、メモリ 3 0 内におけるバッファインデックスのバッファポインタアドレスおよび該バッファを含むバッファプールの関連する変数(例えば b u f f _ s i z e X および #_b u f f x)を使用して、プール内のバッファのバッファインデックスを決定する。一例として、第 1 バッファのバッファインデックスは、バッファのバッファポインタアドレスを、該バッファを含むプールの b u f f _ s i z e X 変数で割ることにより、決定された値と同値に設定することができる。一方、バッファのバッファポインタアドレスは、そのバッファインデックス値に、該バッファを含むプールの関連する b u f f _ s i z e X 値を掛けることにより、決定することができる。

40

【 0 0 1 2 】

各バッファプールの数および特性は、プログラム化することができる。例えば、各バッファプールの特性は、システム初期化シーケンス中、またはシステム動作中に設定される

50

。バッファプールおよびその特性は、システム 10 の作動前に初期化し、作動中には原則として静止状態に維持する。システム 10 の作動中にバッファプールの特性を変更するために、プールに対するすべての動作を停止し、バッファプール特性値をリセットした後、該バッファを取り扱う動作を再開することができる。

【0013】

システム 10 の一実施形態において、BMGR 40 は、プロセッサ 21 - 29 に対してコマンドを送受信し、メモリ 30 に対してデータを送受信するバスインターフェース 42 を含む。また、バッファマネージャ 40 は、インターフェース 42 からの受信コマンドを復号する復号論理 44 を含み、バッファインデックスからのバッファポインタアドレスを決定する。

10

【0014】

一実施形態において、システム 10 の作動中、BMGR 40 は、「プールコンテキスト」情報、例えば、BMGR 40 に管理された各バッファプールに関する情報を、局部メモリ 41 に記憶する。プールコンテキスト情報には、例えば、プールベースアドレス、およびバッファプールに関連する変数 `buff_size X` および `#_buff X` が含まれる。プールコンテキスト情報は、フリーリス上のバッファインデックスを、バッファプール内のバッファにアクセスするとき、プロセッサにより利用可能なバッファポインタアドレスに変換するために、復号ロジック 44 により使用することができる。

【0015】

一実施形態において、プロセッサは、1 個のコマンドにより、複数のバッファの割り当てまたは割り当て解除を要求することができる。例えば、プロセッサは、1 個のバッファを特定するコマンドまたは 4 個のバッファを特定するコマンドを送出することができる。他の例として、プロセッサは、`'n'` 値を特定するバッファ割り当てコマンドを送出することができるが、この `n` 値はプロセッサによって要求されたバッファの数に対応する。これに対応して、BMGR 40 は、各ポインタがプール内において割り当てられたバッファの配置に対応する複数のバッファポインタを、要求を出したプロセッサに返すことができる。

20

【0016】

一実施形態において、複数のバッファインデックスは、1 個のバス 50 コマンドをもつ BMGR 40 により、(バス 50 上のアクティビティ量を減少させるために) 予め取り出される。例えば、4 個以上のバッファインデックスを BMGR 40 により割り当てたときはいつでも、事前取り出しコマンドをマスター FIFO 44 c に配置する。マスターインターフェース制御 44 b は、フリーリスト 90 から 4 個の追加バッファインデックスを要求するメモリ 30 に読み出しコマンドを送ることができる、BMGR 40 は、メモリ 30 から受信したとき、これらの追加バッファインデックスをバッファプールのフリーリスト待ち行列に記憶する。

30

【0017】

プロセス 100 は、プロセス 100 の動作中に `R_CNT` に適用されるインクリメント値および/またはデクリメント値として使用される `MAX_R_CNT` 値を選択的に含むことができる。例えば、アクション (127)、(135) および (140) により、`MAX_R_CNT` 値は、複数のバッファを割り当て解除するときに使用することができる。

40

【0018】

プロセス 100 の一実施形態において、割り当てに利用できるプール内にバッファがないと BMGR 40 が判定した (115) 場合、ヌルポインタ (すなわち、0 の値) が要求を発したプロセッサに返される。このとき、要求を発したプロセッサは、他のバッファ割り当てコマンドを引き続いて送ることができる。図 2 をさらに参照すると、フリーリスト上に利用できるバッファがないとプロセス 100 で判定した (115) 場合、BMGR 40 は、要求を発したプロセッサにヌルポインタを送り (116)、追加のバッファインデックスがメモリ 30 から必要とされているかを判定し (125)、追加のバッファインデックスがバッファマネージャ 40 により必要とされていると判断した場合、メモリ 30 か

50

らバッファインデックスを取り出す (1 2 6)。

【 0 0 1 9 】

一実施形態において、バッファがプロセッサに割り当てられる前に、メモリ 3 0 に記憶された R_C N T 値を最初に 1 の値に設定する。それ故、バッファの割り当て中は、対応する R_C N T は インクリメント されない。これにより、システム 1 0 により実行されるバスコマンド周期数を減少させることができる。バッファの割り当て解除を行うか、または R_C N T 値を インクリメント するために、プロセッサはバッファが割り当てられたバッファプールについて知る必要はない。より詳細には、プロセッサは、B M G R 4 0 に対して、バッファプールのアドレス範囲内のポインタアドレスを含む書き込みコマンドを送ることができる、B M G R 4 0 は、バッファプールに対応するプールを決定する。

10

【 0 0 2 0 】

以下の例 1 は、プロセス 1 0 0 の機能に対応するシステム 1 0 の動作の例を表す。

【 0 0 2 1 】

例 1 ；

1) プロセッサ 2 1 は、p o o l _ 1 0 _ a l l o c / d e a l l o c レジスタ (@ B M G R アドレス = 0 x 0 0 0 0 _ 0 5 4 0) を読み込み、バッファインデックス # 5 は、フリーリスト先頭ポインタが示している場所にある (そのため、バッファインデックス # 5 は、局部メモリ 4 1 事前取出バッファになければならない)。

【 0 0 2 2 】

2) B M G R 4 0 は、バッファインデックス # 5、例えば、ポインタ (p o o l = 1 0、i n d i c e = 5) = p o o l _ 1 0 _ b a s e + (p o o l _ 1 0 _ b u f f e r _ s i z e * 5) に対応するポインタアドレスを決定する。

20

【 0 0 2 3 】

3) B M G R は、決定されたポインタアドレス (1 0、5) をプロセッサ 2 1 に送り、また、巡回待ち行列フリーリストの先頭ポインタをインクリメント させ、他の割り当てコマンドに備えて 巡回待ち行列フリーリストの次の場所 に指定する。

【 0 0 2 4 】

4) プロセッサ 2 1 は、ポインタ (1 0、5) をプロセッサ 2 2 に送る (場合によっては、また、プロセッサ 2 1 は、例えば、プロセッサ 2 2 のコマンドを送ってプロセッサ 2 1 に割り当てられたバッファ中に見出される データ を処理する。)

30

5) プロセッサ 2 1 または 2 2 は、インクリメント コマンドを B M G R 4 0 に送り、例えばポインタ (1 0、5) 中のアドレスに書き込みコマンドを送ることにより、ポインタ (1 0、5) に対応する R_C N T を インクリメント する (R_C N T を インクリメント する書き込みコマンドは、特定のバッファのアドレス範囲内のいかなるアドレスも含むことができ、B M G R 4 0 は、アドレスからの適切なバッファを決定し、対応する b u f f R_C N T (1 0、5) を インクリメント することを認識していただきたい。)

6) プロセッサ 2 1 および 2 2 は、割り当てられたバッファ (1 0、5) に継続的にアクセスすることができる。それとほぼ同時に、B M G R は、メモリ 3 0 から R_C N T (1 0、5) を読み込み、R_C N T を インクリメント し、メモリ 3 0 に更新した R_C N T 値を書き戻すことにより、対応する バッファ R_C N T (1 0、5) をインクリメント する。

40

【 0 0 2 5 】

7) プロセッサ 2 1 は、バッファ (1 0、5) にアクセスする際、バッファ (1 0、5) 内のいずれのアドレスをもいずれの p o o l _ x 割り当て解除レジスタ 3 4 に書き込むことにより、割り当て解除コマンドを B M G R 4 0 に送る。

【 0 0 2 6 】

8) B M G R 4 0 は、割り当て解除コマンドを受信し、バッファ (1 0、5) の 対応する R_C N T 値 を決定し、メモリ 3 0 から R_C N T (1 0、5) 値を読み込み、R_C N T (1 0、5) を デクリメント し、デクリメント された R_C N T (1 0、5) = 1 であることを決定し、デクリメント された R_C N T (1 0、5) 値 をメモリ 3 0 に書き戻す

50

。

【 0 0 2 7 】

9) プロセッサ 22 は、バッファ (10、5) へのアクセスを行うと、バッファ (10、5) 内のいずれかのアドレスをいずれかの pool_x 割り当て解除レジスタ 34 に書き込むことにより、割り当て解除コマンドを BMGR 40 に送る。

【 0 0 2 8 】

10) BMGR 40 は、割り当て解除コマンドを受信し、バッファ (10、5) の対応する R_CNT (10、5) 値を決定し、メモリ 30 から R_CNT (10、5) 値を読み込み、R_CNT (10、5) = 0 にデクリメントし、デクリメントされた R_CNT (10、5) = 0 であることを決定し、この場合、メモリに R_CNT (10、5) = 0 を書き戻す代わりに、R_CNT (10、5) = 1 をメモリに残す。2 個のバスコマンドトランザクション、例えば、1_0 と移行する R_CNT 値を読み込み、更新し、書き戻す第 1 バスコマンドおよび R_CNT を 0_1 と移行する第 2 バスコマンドの実行を省くことができるため、R_CNT (pool, index) = 1 を残すことは重要である。

10

【 0 0 2 9 】

11) また、BMGR 40 は、局部メモリ 41 におけるこのバッファプール巡回待ち行列のフリーリスト最後尾ポインタをインクリメントする。

【 0 0 3 0 】

例 1 において、工程 7) で、BMGR 40 が R_CNT (10、5) をインクリメントした後、工程 9) で、BMGR 40 は、R_CNT (10、5) をデクリメントすることを認識していただきたい。一実施形態において、工程 7) および 9) の操作を除くことが可能である。より詳細には、第 1 プロセッサ (例えば、プロセッサ 21) を「割り当て装置」として指定し、第 2 プロセッサ (例えば、プロセッサ 22) を「割り当て解除装置」として指定した場合。この場合、インクリメントおよび割り当て解除をそれぞれ行う複数のプロセッサを有する代わりに、1 つのプロセッサのみがバッファ割り当てコマンドのみを実行し、第 2 プロセッサは、特定のバッファの割り当て解除コマンドのみを実行する。システム 10 の操作および / またはプロセス 100 でこの方法を行うことにより、バッファの割り当て、割り当て解除および / または R_CNT のインクリメントに要するバスコマンド周期の数を減少させることができる。

20

【 0 0 3 1 】

一実施形態において、割り当てまたは割り当て解除コマンドには、割り当てられるか割り当て解除されるバッファのアドレス境界内にあるいずれかのアドレスを含むことができる。より詳細には、BMGR 40 は、プロセッサにより送られたポインタアドレス上でアドレス境界チェックを行い、どのバッファの R_CNT が更新を必要とするかを判定することができる (例えば、図 2 に説明した動作 (155) および (135) を参照)。BMGR 40 がアドレス境界チェックを行うこの方法は、バッファ割り当てまたは割り当て解除を要求するために、プロセッサ 21 - 29 が割り当てられたバッファのベースプールアドレスを記憶する必要はないことを示している。従って、バッファ内の位置のポインタアドレスはどれも、プロセッサから BMGR 40 に送られた割り当てまたは割り当て解除コマンドの一部として使用することができる。

30

40

【 0 0 3 2 】

一実施形態において、割り当てられた (または、以前割り当てられた) バッファに対応する R_CNT をインクリメントまたはデクリメントするために、BMGR 40 は、読み込みコマンドをメモリ 30 に送り、該バッファに対応する R_CNT アレイ 98 から R_CNT 値を読み出させる。BMGR 40 は、メモリ 30 から R_CNT を受信すると、R_CNT 値をインクリメントまたはデクリメントし、(R_CNT の更新値により) 更新済み R_CNT 値をメモリ 30 に書き戻すことができる。

【 0 0 3 3 】

一実施形態において、復号論理ブロック 44 を、他のコマンドの処理前にバッファ割り当てコマンドを処理するように構成する。より詳細には、復号論理 44 には、復号論理 4

50

4 a、マスターインターフェース論理 4 4 b および プール状態制御論理 4 4 d が含まれる。マスターインターフェース論理ブロック 4 4 b には、復号論理 4 4 a からの割り当て解除コマンドおよびインクリメントコマンドを保持するために使用される F I F O 待ち行列 4 4 c が含まれる。バッファ割り当て解除コマンドおよび / または インクリメントコマンドが後にマスターインターフェース制御論理 4 4 c により処理されるために F I F O 4 4 c に保存されるのに対し、復号論理 4 4 a により受信されたバッファ割り当てコマンドは、受信されたときに処理される。そのため、バッファ割り当てコマンドが B M G R 4 0 によって処理されるのを待機しているプロセッサは、バッファ処理割り当て解除および / または R _ C N T 更新の処理を待機する必要はない。

【 0 0 3 4 】

10

一実施形態において、復号論理 4 4 には、プールコンテキスト情報を記憶して局部メモリ 4 1 にアクセスするプール状態制御 4 4 d が含まれる。本例において、プール状態制御 4 4 d は、接続されてコマンド復号 4 4 a およびマスターインターフェース制御 4 4 b 双方からアクセス要求を受信する。プール状態制御 4 4 b は、コマンド復号 4 4 a およびマスターインターフェース論理 4 4 d からのアクセス要求の調停を行い、データの読み込みおよび書き出しを局部メモリ 4 1 に対して行う。

【 0 0 3 5 】

各プロセッサ 2 1 - 2 9 および B M G R 4 0 は、オペレーティングシステムを装備することができ、オペレーティングシステムとは、プロセッサの動作および資源の割り当てを制御するソフトウェアである。用語「プロセス」または「プログラム」とは、ソフトウェア、例えば、プロセッサまたはコンピュータシステム上で実行することのできるアプリケーションプログラムのことをいう。アプリケーションプログラムは、オペレーティングシステムを介して利用可能なコンピュータ資源を使用して、利用者が望むタスクを行う実行可能な命令のセットである。

20

【 0 0 3 6 】

プロセッサ 2 1 - 2 9 および B M G R 4 0 は、ハードウェア、ソフトウェアまたはこれら 2 つの組み合わせの中に実施することができる。プロセッサ、プロセッサにより読み取り可能な記憶媒体（揮発性メモリおよび不揮発性メモリおよび / または記憶部品を含む）、少なくとも 1 個の入力装置および 1 個以上の出力装置をそれぞれ含むプログラム可能なコンピュータまたは他の機械で実行されるコンピュータプログラムにおいて、それらを実施することができる。プログラムコードは、入力装置（例えば、マウスまたはキーボード）を用いて入力されるデータに適用して、アプリケーションを実行し、出力情報を生成することができる。

30

【 0 0 3 7 】

アプリケーションを実行するコンピュータが記憶媒体または記憶装置を読み込んだとき、コンピュータを構成および操作する一般または特別目的プログラム可能なコンピュータにより読み取ることが可能な記憶媒体 / 製品（例えば、C D - R O M、ハードディスクまたは磁気ディスク）上で、各コンピュータプログラムを記憶することができる。また、実行されるとコンピュータプログラム内の命令により、アプリケーションに従って機械を作動させるコンピュータプログラムにより構成された機械読み取り可能な記憶媒体として、それらを実施することができる。

40

【 0 0 3 8 】

本発明は、上述した特定の実施形態に限定されるものではない。例えば、上述の例は、マルチプロセッサシステムのプロセッサを用いて説明した。しかしながら、例えば、共有メモリにアクセスする機能を含む機能装置として 1 個以上のプロセッサを実施することができ、例えば、アプリケーション特化集積回路（“ A S I C S ”）として機能装置を実施することができる。例えば、B M G R 4 0 の局部メモリを用いて上記装置を説明した。しかしながら、局部メモリは、レジスタとして部分的に装備され、例えば、バッファインデックスおよびプールコンテキスト情報を記憶するために使用することができる。他の例として、割り当て / 割り当て解除レジスタを特定する B M G R 4 0 に読み取りコマンドを送るこ

50

とにより、バッファ割り当てを要求するプロセッサを説明した。しかしながら、レジスタの特定を必要としない他の方法で、割り当て / 割り当て解除コマンドを実行することが可能である。

【 0 0 3 9 】

また、本明細書に記載のない他の実施形態についても、以下の請求項の範囲である。

【図面の簡単な説明】

【 0 0 4 0 】

【図 1】図 1 は、マルチプロセッサシステムのブロック図である。

【図 2】図 2 は、バッファ管理のための工程の流れ図である。

【図 1】

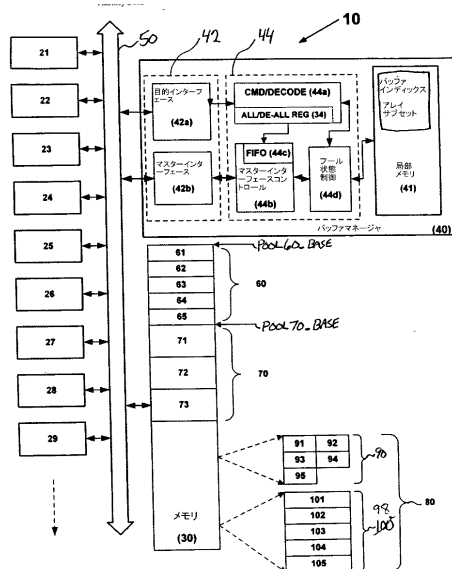
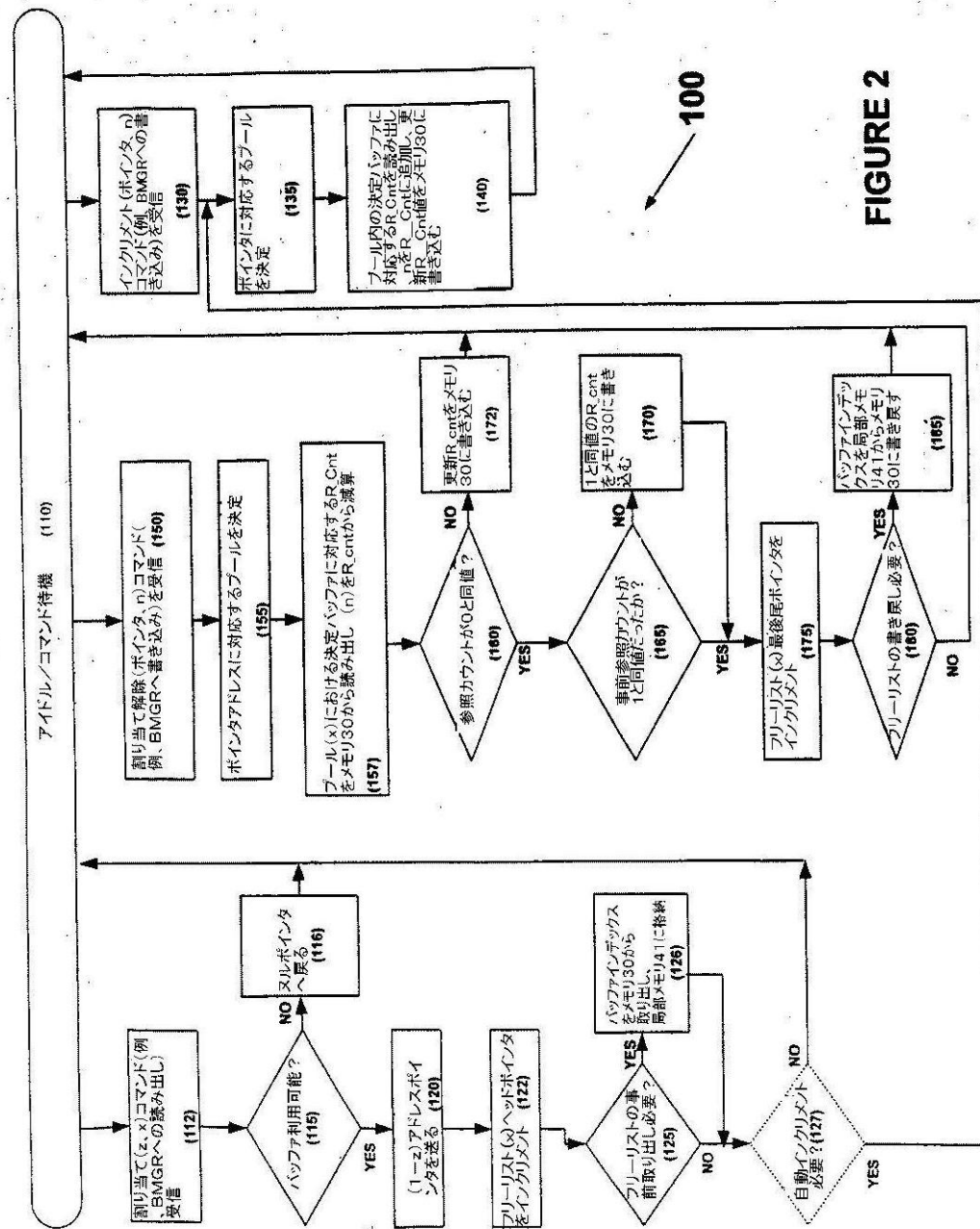


FIGURE 1

【図2】



フロントページの続き

- (72)発明者 クレイトン, ショーン アダム
アメリカ合衆国 カリフォルニア 9 2 6 2 6 , コスタ メサ, スーザン ストリート 3 3
3 3
- (72)発明者 マクマスター, ショーン
アメリカ合衆国 カリフォルニア 9 2 6 2 6 , コスタ メサ, スーザン ストリート 3 3
3 3
- (72)発明者 スペンサー, トーマス ブイ.
アメリカ合衆国 カリフォルニア 9 2 6 2 6 , コスタ メサ, スーザン ストリート 3 3
3 3

審査官 多賀 実

- (56)参考文献 特開昭 6 2 - 0 2 4 3 5 5 (J P , A)
特開昭 6 0 - 1 1 8 9 3 7 (J P , A)
特開昭 6 1 - 2 9 0 5 4 8 (J P , A)
特開昭 5 8 - 0 3 7 8 8 7 (J P , A)
特開平 0 8 - 2 9 7 6 2 6 (J P , A)
特開平 0 6 - 1 9 5 2 5 6 (J P , A)

(58)調査した分野(Int.Cl. , D B 名)

G06F12/02
G06F 9/50
G06F15/167