

(19) 日本国特許庁(JP)

## (12) 特許公報(B2)

(11) 特許番号

特許第6884149号  
(P6884149)

(45) 発行日 令和3年6月9日(2021.6.9)

(24) 登録日 令和3年5月13日(2021.5.13)

(51) Int.Cl.

F 1

GO6F 12/00	(2006.01)	GO6F	12/00	5 6 0 B
GO6F 12/02	(2006.01)	GO6F	12/02	5 4 0
GO6F 9/46	(2006.01)	GO6F	12/00	5 9 4
GO6F 9/54	(2006.01)	GO6F	9/46	4 1 0
GO6F 5/10	(2006.01)	GO6F	9/54	B

請求項の数 15 (全 18 頁) 最終頁に続く

(21) 出願番号 特願2018-529209 (P2018-529209)  
 (86) (22) 出願日 平成28年12月12日 (2016.12.12)  
 (65) 公表番号 特表2019-502201 (P2019-502201A)  
 (43) 公表日 平成31年1月24日 (2019.1.24)  
 (86) 國際出願番号 PCT/US2016/066106  
 (87) 國際公開番号 WO2017/100748  
 (87) 國際公開日 平成29年6月15日 (2017.6.15)  
 審査請求日 令和1年11月21日 (2019.11.21)  
 (31) 優先権主張番号 14/966,631  
 (32) 優先日 平成27年12月11日 (2015.12.11)  
 (33) 優先権主張国・地域又は機関  
米国(US)

(73) 特許権者 507186492  
ビバンテ コーポレーション  
アメリカ合衆国、カリフォルニア州 95  
002, サンノゼ、ゴールドストリート 2150, スイート 200  
(74) 代理人 100121083  
弁理士 青木 宏義  
(74) 代理人 100138391  
弁理士 天田 昌行  
(74) 代理人 100074099  
弁理士 大菅 義之  
(74) 代理人 100106851  
弁理士 野村 泰久

最終頁に続く

(54) 【発明の名称】マルチスレッドアクセスのためのソフトウェア定義のFIFOバッファ

## (57) 【特許請求の範囲】

## 【請求項 1】

実行可能なデータを格納するメモリデバイスを提供することと、  
 前記メモリデバイスに結合され、前記実行可能なデータを抽出及び実行するように動作可能な処理デバイスを提供することと、  
 前記処理デバイスによって、バッファ初期化命令をアプリケーションから受信することと、  
 前記バッファ初期化命令に応じて、

前記処理デバイスによって、バッファとして前記メモリデバイスの一部分を割り当てるこ

前記処理デバイスによって、前記メモリデバイスの前記一部分を参照するためにハードウェアバッファコントローラのある状態を引き起こすこと、及び  
 前記処理デバイスによって、前記ハードウェアバッファコントローラへの参照をアプリケーションへ返すことと、

前記処理デバイスによって、前記アプリケーションを実行することであって、前記アプリケーションから前記バッファにアクセスするためのリクエストを実行することを含み、前記リクエストが、前記ハードウェアバッファコントローラへの前記参照のみにアドレス指定され、前記メモリデバイスの前記一部分内の何れのアドレスも含まない、前記アプリケーションを実行することと、

前記リクエストに応じて、前記処理デバイスによって前記リクエストを前記ハードウェ

10

20

アバッファコントローラに入力することと、

前記ハードウェアアバッファコントローラによって、前記リクエストを処理することであって、前記リクエストを処理することが前記リクエスト毎に、

前記ハードウェアアバッファコントローラの前記状態に基づいて前記メモリデバイスの前記一部分内のアドレスを生成すること、

前記アドレスを含むメモリアクセス命令を前記メモリデバイスへ出力すること、及び

前記ハードウェアアバッファコントローラの前記状態を更新すること

により行われる、前記リクエストを処理することと

を含む、方法。

#### **【請求項 2】**

10

前記ハードウェアアバッファコントローラは先入れ先出し( FIFO )バッファを実施し、

前記ハードウェアアバッファコントローラの前記状態は読み出しポインタ及び書き込みポインタを含み、

前記ハードウェアアバッファコントローラの前記状態を更新することは、前記リクエストの内の書き込みリクエスト毎に前記書き込みポインタをインクリメントすることを含み、

前記ハードウェアアバッファコントローラの前記状態を更新することは、前記リクエストの内の読み出しリクエスト毎に前記読み出しポインタをインクリメントすることを含む、請求項 1 に記載の方法。

#### **【請求項 3】**

20

前記書き込みポインタをインクリメントすることは、1クロック周期中に受信された複数の書き込みリクエストの数を特定することと、前記複数の書き込みリクエストの前記数だけ前記書き込みポインタをインクリメントすることとを含み、

前記読み出しポインタをインクリメントすることは、前記クロック周期中に受信された複数の読み出しリクエストの数を特定することと、前記複数の読み出しリクエストの前記数だけ前記読み出しポインタをインクリメントすることとを含む、

請求項 2 に記載の方法。

#### **【請求項 4】**

前記ハードウェアアバッファコントローラの前記状態に基づいて前記バッファ内の前記アドレスを生成することは、前記リクエストの内の書き込みリクエスト毎に前記書き込みポインタの現在値を出力することを含み、

30

前記ハードウェアアバッファコントローラの前記状態に基づいて前記バッファ内の前記アドレスを生成することは、前記リクエストの内の読み出しリクエスト毎に前記読み出しポインタの現在値を出力することを含む、

請求項 2 に記載の方法。

#### **【請求項 5】**

前記処理デバイスによって、前記書き込みリクエストの量と前記読み出しリクエストの量との差を計算することと、

前記処理デバイスによって、前記差が第 1 の閾値よりも少なくゼロではないと判定することと、

40

前記差が第 1 の閾値よりも少なくゼロではないと判定することに応じて、前記処理デバイスによって、前記バッファがほぼ空であることを示すイベントを出力することと、

前記バッファがほぼ空であることを示す前記イベントに応じて、前記処理デバイスによって、前記処理デバイスによって実行される 1 つ以上の実行スレッドが読み出しリクエストを生成することをブロックすることと

を更に含む、請求項 4 に記載の方法。

#### **【請求項 6】**

前記ハードウェアアバッファコントローラによって、前記書き込みリクエストの量と前記読み出しリクエストの量との差を計算することと、

前記ハードウェアアバッファコントローラによって、前記差が第 1 の閾値よりも大きいと

50

判定することと、

前記差が第1の閾値よりも大きいと判定することに応じて、前記ハードウェアバッファコントローラによって、前記バッファがほぼ満杯であることを示すイベントを出力することと、

前記バッファがほぼ満杯であることを示す前記イベントに応じて、前記処理デバイスによって、前記処理デバイスによって実行される1つ以上の実行スレッドが書き込みリクエストを生成することをロックすることと  
を更に含む、請求項4に記載の方法。

#### 【請求項7】

前記ハードウェアバッファコントローラによって前記リクエストを処理することは、10  
前記リクエストの内の複数の書き込みリクエストを集計することと、

前記複数の書き込みリクエストに対応する前記バッファ内の複数のアドレスを含む集計書き込み命令を前記メモリデバイスへ送信することと、

前記リクエストの内の複数の読み出しリクエストを集計することと、

前記複数の読み出しリクエストに対応する前記バッファ内の複数のアドレスを含む集計読み出し命令を前記メモリデバイスへ送信することと  
を更に含み、

前記複数の書き込みリクエストは、前記リクエストの内のその他のリクエストが前記ハードウェアバッファコントローラに入力されるよりも前に、前記書き込みリクエストの内の少なくとも1つの書き込みリクエストが受信されるような重複方式で、複数の実行スレッドによって発行される、20

請求項1に記載の方法。

#### 【請求項8】

前記処理デバイスによって、前記ハードウェアバッファコントローラの前記状態を評価することと、

前記処理デバイスによって、前記ハードウェアバッファコントローラの前記状態が閾条件を満たすと判定することと、

前記ハードウェアバッファコントローラの前記状態が前記閾条件を満たすと判定することに応じて、前記処理デバイスによって、前記ハードウェアバッファコントローラを参照する更なるリクエストを前記複数の実行スレッドが発行することをロックすることと  
を更に含む、請求項7に記載の方法。30

#### 【請求項9】

実行可能なデータを格納するためのメモリデバイスと、

処理デバイスと、

前記処理デバイスに動作可能に結合されたハードウェアバッファコントローラと  
を含み、

前記処理デバイスは、前記メモリデバイスに結合され、前記実行可能なデータを抽出及び実行するように動作可能であり、前記処理デバイスは、バッファリクエストをアプリケーションから受信すること、並びに、前記バッファリクエストに応じて、

前記メモリデバイスの一部分を前記バッファリクエストのためのバッファとして割り当てる事と、40

前記メモリデバイスの前記一部分を参照するために前記ハードウェアバッファコントローラのある状態を引き起こすこと、及び

前記ハードウェアバッファコントローラへの参照を前記アプリケーションへ返すことをするようにプログラムされ、

前記ハードウェアバッファコントローラは、前記処理デバイス及び前記メモリデバイスに結合され、前記ハードウェアバッファコントローラは、

前記メモリデバイス内の何れの格納位置に対する何れのアドレスも含まず、前記ハードウェアバッファコントローラのみにアドレス指定されたメモリアクセスリクエストを前記処理デバイスにより実行される前記アプリケーションから受信すること、50

前記ハードウェアバッファコントローラの状態に基づいて、前記メモリデバイスの前記一部分内の生成アドレスを生成すること、  
 前記生成アドレスを含むメモリアクセス命令を前記メモリデバイスへ出力すること、  
 前記ハードウェアバッファコントローラの前記状態を更新すること  
 をするようにプログラムされる、  
 システム。

**【請求項 10】**

前記ハードウェアバッファコントローラの前記状態は先入れ先出し（FIFO）バッファを実施し、

前記ハードウェアバッファコントローラは、前記リクエストの内の書き込みリクエスト毎に書き込みポインタをインクリメントすることによって、前記ハードウェアバッファコントローラの前記状態を更新するように更にプログラムされ、

前記ハードウェアバッファコントローラの前記状態を更新することは、前記リクエストの内の読み出しリクエスト毎に読み出しポインタをインクリメントすることを含み、

前記ハードウェアバッファコントローラは、1クロック周期中に受信された複数の書き込みリクエストの数を特定することと、前記複数の書き込みリクエストの前記数だけ前記書き込みポインタをインクリメントすることによって、前記書き込みポインタをインクリメントするように更にプログラムされ、

前記ハードウェアバッファコントローラは、前記クロック周期中に受信された複数の読み出しリクエストの数を特定することと、前記複数の読み出しリクエストの前記数だけ前記読み出しポインタをインクリメントすることによって、前記読み出しポインタをインクリメントするように更にプログラムされる、

請求項 9 に記載のシステム。

**【請求項 11】**

前記ハードウェアバッファコントローラは、前記リクエストの内の書き込みリクエスト毎に前記書き込みポインタの現在値を出力することによって、前記ハードウェアバッファコントローラの前記状態に基づいて、前記バッファに割り当てられた前記メモリデバイスの前記一部分内の前記生成アドレスを生成するように更にプログラムされ、

前記ハードウェアバッファコントローラは、前記リクエストの内の読み出しリクエスト毎に前記読み出しポインタの現在値を出力することによって、前記ハードウェアバッファコントローラの前記状態に基づいて、前記バッファに割り当てられた前記メモリデバイスの前記一部分内の前記生成アドレスを生成するように更にプログラムされる、  
 請求項 10 に記載のシステム。

**【請求項 12】**

前記ハードウェアバッファコントローラは、

前記書き込みリクエストの量と前記読み出しリクエストの量との差を計算すること、及び

前記差が第 1 の閾値よりも少なくゼロではない場合に、前記バッファがほぼ空であることを示すイベントを出力すること  
 をするように更にプログラムされ、

前記処理デバイスは、前記バッファがほぼ空であることを示す前記イベントに応じて、前記処理デバイスによって実行される 1 つ以上の実行スレッドが読み出しリクエストを生成することをブロックするように更にプログラムされる、

請求項 11 に記載のシステム。

**【請求項 13】**

前記ハードウェアバッファコントローラは、

前記書き込みリクエストの量と前記読み出しリクエストの量との差を計算すること、及び

前記差が第 2 の閾値よりも大きい場合に、前記処理デバイスによって、前記バッファがほぼ満杯であることを示すイベントを出力すること

10

20

30

40

50

をするように更にプログラムされ、

前記処理デバイスは、前記バッファがほぼ満杯であることを示す前記イベントに応じて、前記処理デバイスによって実行される1つ以上の実行スレッドが書き込みリクエストを生成することをロックするように更にプログラムされる、

請求項12に記載のシステム。

#### 【請求項14】

前記ハードウェアバッファコントローラは、

前記リクエストの内の複数の書き込みリクエストを集計することと、

前記複数の書き込みリクエストに対応する前記バッファ内の複数のアドレスを含む集計書き込み命令を前記メモリデバイスへ送信することと、

10

前記リクエストの内の複数の読み出しリクエストを集計することと、

前記複数の読み出しリクエストに対応する前記バッファ内の複数のアドレスを含む集計読み出し命令を前記メモリデバイスへ送信することと

によって前記リクエストを処理するように更にプログラムされる、請求項9に記載のシステム。

#### 【請求項15】

前記ハードウェアバッファコントローラは、前記リクエストの内のその他のリクエストが前記ハードウェアバッファコントローラに入力されるよりも前に、前記書き込みリクエストの内の少なくとも1つの書き込みリクエストが受信されるような重複方式で、前記処理デバイスによって実行される複数の実行スレッドから前記複数の書き込みリクエストを受信するようにプログラムされ、前記ハードウェアバッファコントローラは、

20

前記ハードウェアバッファコントローラの前記状態を評価すること、及び

前記ハードウェアバッファコントローラの前記状態が閾条件を満たす場合に、前記ハードウェアバッファコントローラを参照する更なるリクエストを前記複数の実行スレッドが発行することを、前記処理デバイスによってブロックすることを引き起こすことをするように更にプログラムされる、請求項14に記載のシステム。

### 【発明の詳細な説明】

#### 【技術分野】

#### 【0001】

30

本発明は、FIFOバッファを実施するためのシステム及び方法に関する。

#### 【背景技術】

#### 【0002】

従来のコンピュータシステムでは、先入れ先出し(FIFO)バッファは複数のスレッドによってアクセスされる場合がある。しかしながら、従来のシステムでは、同時に1つのスレッドのみがFIFOバッファにアクセスすることができる。このことにより、遅延と処理オーバヘッドとがもたらされるが、これは、FIFOバッファのデータ構造が別のスレッドにより解放されることを待ちながら、スレッドをブロックすることによって同時アクセスを阻止するためである。

#### 【0003】

40

本明細書に開示されるシステム及び方法は、複数のスレッドにより同時にアクセス可能なFIFOバッファを実施するための改善したアプローチを提供する。

#### 【図面の簡単な説明】

#### 【0004】

発明の利点が容易に理解されるように、簡単に上述した発明のより具体的な説明は、添付の図面で説明された特定の実施形態を参照することによって提供されることになる。これらの図面は、発明の典型的な実施形態のみを示し、それ故、その範囲を限定するとみされないと理解され、発明は、添付の図面の使用を通じて更なる特異性と詳細が記述及び説明される。

#### 【図1】本発明の実施形態に従った方法を実施するのに適したコンピュータシステムの概

50

略ブロック図である。

【図2A】本発明の一実施形態に従ったハードウェアFIFOバッファを実施するための構成要素の概略ブロック図である。

【図2B】本発明の一実施形態に従ったハードウェアFIFOバッファを実施するための構成要素の概略ブロック図である。

【図3A】本発明の一実施形態に従ったハードウェアFIFOバッファへの読み出し及び書き込みを実施するための方法のプロセスフロー図である。

【図3B】本発明の一実施形態に従ったハードウェアFIFOバッファへの読み出し及び書き込みを実施するための方法のプロセスフロー図である。

【図4】本発明の一実施形態に従ったハードウェアFIFOバッファを用いてバッファのオーバーフロー及びアンダーフローを防ぐための方法のプロセスフロー図である。 10

【発明を実施するための形態】

【0005】

本明細書内で包括的に記述及びの図示されるような本発明の構成要素は、幅広い様々な異なる構成で配置及び設計され得ることが容易に理解されるであろう。したがって、図面で表されるような以下の発明の実施形態のより詳細な説明は、請求項で示されるような発明の範囲を限定することを意図せず、本発明に従って現在企図される実施形態の特定の例を代表するにすぎない。現在記述された実施形態は、全体を通じて同様の部分が同様の数字で示されている図面を参照することによって最も正確に理解されるであろう。

【0006】

20

本発明に従った実施形態は、装置、方法、又はコンピュータプログラム製品として具体化されてもよい。したがって、本発明は、完全にハードウェアの実施形態、（ファームウェア、常駐ソフトウェア、マイクロコード等を含む）完全にソフトウェアの実施形態、又は“モジュール”若しくは“システム”として本明細書で概して全て称され得るソフトウェア及びハードウェアの態様の組み合わせの実施形態をとってもよい。更に、本発明は、任意の有形的表現媒体内で具体化されるコンピュータプログラム製品の形式をとってもよく、こうした有形的表現媒体内で、コンピュータ使用可能なプログラムコードが具体化される。

【0007】

30

非一時的媒体を含めた、1つ以上のコンピュータ使用可能な又はコンピュータ可読な媒体の組み合わせが利用されてもよい。例えば、コンピュータ可読媒体は、携帯可能なコンピュータディスクケット、ハードディスク、ランダムアクセスメモリ（RAM）デバイス、リードオンリーメモリ（ROM）デバイス、消去可能プログラム可能リードオンリーメモリ（EPROM又はフラッシュメモリ）デバイス、携帯可能なコンパクトディスクドライブオンリーメモリ（CD-ROM）、光学記憶デバイス、及び磁気記憶デバイス内の1つ以上を含んでもよい。選択された実施形態では、コンピュータ可読媒体は、命令実行システム、装置(apparatus)、又はデバイス(device)によってされる、又はそれに関連して使用するためのプログラムを含み得、格納し得、通信し得、伝搬し得、又は搬送し得る任意の非一時的媒体を含んでもよい。

【0008】

40

本発明の動作を遂行するためのコンピュータプログラムコードは、Java、スマートトーク、又はC++等のオブジェクト指向プログラミング言語、及び、“C”プログラミング言語若しくは同様のプログラミング言語等の同様の従来の手続型プログラミング言語を含む1つ以上のプログラミング言語の任意の組み合わせで書き込まれてもよい。プログラムコードは、スタンドアローンなソフトウェアパッケージとしてのコンピュータシステム上で全体的に実行されること、スタンドアローンなハードウェアユニット上で実行されること、コンピュータから幾らか間隔が空けられたりモートコンピュータ上で部分的にされること、又はリモートコンピュータ若しくはサーバ上で全体的に実行されることもある。後者の場合では、リモートコンピュータは、ローカルエリアネットワーク（LAN）又は広域ネットワーク（WAN）を含む任意の種類のネットワークを通じてコンピュータに 50

接続されてもよく、又は該接続は（例えば、インターネットサービスプロバイダを用いてインターネットを通じて）外部のコンピュータへなされてもよい。

【0009】

本発明は、本発明の実施形態に従った方法、装置（システム）、及びコンピュータプログラム製品のフローチャートの説明及び／又はブロック図を参照しながら以下に記述される。フローチャートの説明及び／又はブロック図中の各ブロック、並びにフローチャートの説明及び／又はブロック図中のブロックの組み合わせは、コンピュータプログラム命令又はコードによって実施され得ることが理解されるであろう。これらのコンピュータプログラム命令を、機械を生産するための汎用コンピュータ、専用コンピュータ、又はその他のプログラム可能なデータ処理装置のプロセッサに供給することもでき、こうすることで、その命令が、コンピュータ又はその他のプログラム可能なデータ処理装置のプロセッサを介して実行され、フローチャート及び／又はブロック図の1つ以上のブロックで指定される機能／作用を実施するための手段を作り出す。

【0010】

これらのコンピュータプログラム命令を、特定の機能を行うようにコンピュータ又はその他のプログラム可能なデータ処理装置に命令を出すことが可能な非一時的コンピュータ可読媒体に格納することもでき、こうすることで、コンピュータ可読媒体に格納された該命令がフローチャート及び／又はブロック図の1つ以上のブロックで指定される機能／作用を実施する命令手段を含む製品を生み出す。

【0011】

コンピュータプログラム命令はまたを、コンピュータ又はその他のプログラム可能なデータ処理装置上にロードして、一連の動作ステップをコンピュータ又はその他のプログラム可能装置上で実行させて、コンピュータ実施プロセスであって、コンピュータ又はその他のプログラム可能装置上で実行される命令が、フローチャート及び／又はブロック図の1つ以上のブロックで指定される機能／作用を実施するためのプロセスを提供するようなコンピュータ実施プロセスを生み出してもよい。

【0012】

図1は、例示的な計算デバイス100を説明するブロック図である。計算デバイス100は、本明細書で論じられるような様々な手続を実行するために使用されてもよい。計算デバイス100は、サーバ、クライアント、又は任意のその他のコンピューティングエンティティとして機能し得る。計算デバイスは、本明細書で論じられるような様々な監視機能を実行し得、本明細書で論じられるアプリケーションプログラム等、1つ以上のアプリケーションプログラムを実行し得る。計算デバイス100は、デスクトップコンピュータ、ノートブックコンピュータ、サーバコンピュータ、ハンドヘルドコンピュータ、及びタブレットコンピュータ等の任意の幅広い計算デバイスであり得る。

【0013】

計算デバイス100は、1つ以上のプロセッサ102、1つ以上のメモリデバイス104、1つ以上のインターフェース106、1つ以上の大容量記憶デバイス108、1つ以上の入出力（I/O）デバイス110、及び表示デバイス130を含み、それら全てがバス112に結合される。（1つ以上の）プロセッサ102は、（1つ以上の）メモリデバイス104及び／又は（1つ以上の）大容量記憶デバイス108に格納された命令を実行する1つ以上のプロセッサ又はコントローラを含む。（1つ以上の）プロセッサ102は、キャッシュメモリ等の様々な種類のコンピュータ可読媒体をも含む。

【0014】

（1つ以上の）メモリデバイス104は、揮発性メモリ（例えば、ランダムアクセスメモリ（RAM）114）及び／又は不揮発性メモリ（例えば、リードオンリーメモリ（ROM）116）等の様々なコンピュータ可読媒体を含む。（1つ以上の）メモリデバイス104は、フラッシュメモリ等の再書き込み可能なROMをも含んでもよい。

【0015】

（1つ以上の）大容量記憶デバイス108は、磁気テープ、磁気ディスク、光ディスク

10

20

30

40

50

、及び固体状態メモリ（例えば、フラッシュメモリ）等の様々なコンピュータ可読媒体を含む。図1に示すように、特定の大容量記憶デバイスが、ハードディスクドライブ124である。様々なコンピュータ可読媒体からの読み出し及び／又は様々なコンピュータ可読媒体への書き込みができるように、（1つ以上の）大容量記憶デバイス108には様々なドライブも含まれてもよい。（1つ以上の）大容量記憶デバイス108は、取り外し可能媒体126及び／又は不揮発性媒体を含む。

#### 【0016】

（1つ以上の）入出力デバイス110は、データ及び／又はその他の情報が計算デバイス100に入力され得るように、又は計算デバイス100から抽出され得るようにする様々なデバイスを含む。例示的な（1つ以上の）入出力デバイス110は、カーソル制御デバイス、キーボード、キーパッド、マイクロホン、モニタ若しくはその他の表示デバイス、スピーカ、プリンタ、ネットワークインターフェースカード、モデム、レンズ、及びCCD若しくはその他の画像取込デバイス等を含む。10

#### 【0017】

表示デバイス130は、計算デバイス100の1人以上のユーザに情報を表示可能な任意の種類のデバイスを含む。表示デバイス130の一例は、モニタ、表示端末、及び映像投射デバイス等を含む。

#### 【0018】

（1つ以上の）プロセッサ102及び／又は表示デバイス130にはグラフィック処理ユニット（GPU）132が結合されてもよい。GPUは、コンピュータ生成の画像をレンダリングし、その他のグラフィック処理を実行するように動作可能であってもよい。GPUは、（1つ以上の）プロセッサ102等の汎用プロセッサの幾つか又は全ての機能を含んでもよい。GPUは、画像処理に特有の付加的機能をも含んでもよい。GPUは、座標変換、シェーディング、テクスチャリング、ラスタイゼーション、及びコンピュータ生成の画像をレンダリングするのに役立つその他の機能に関連するハードコードされた及び／又はハードワイヤードのグラフィック機能を含んでもよい。20

#### 【0019】

（1つ以上の）インターフェース106は、計算デバイス100がその他のシステム、デバイス、又は計算環境と相互作用できるようにする様々なインターフェースを含む。例示的な（1つ以上の）インターフェース106は、ローカルエリアネットワーク（LAN）、広域ネットワーク（WAN）、無線ネットワーク、及びインターネットへのインターフェース等の任意の数の異なるネットワークインターフェース120を含む。その他の（1つ以上の）インターフェースは、ユーザインターフェース118及び周辺デバイスインターフェース122を含む。（1つ以上の）インターフェース106は、1つ以上のユーザインターフェース要素118をも含んでもよい。（1つ以上の）インターフェース106は、プリンタ、ポイントティングデバイス（マウス、トラックパッド等）、及びキーボード等のためのインターフェース等、1つ以上の周辺インターフェースをも含んでもよい。30

#### 【0020】

バス112は、（1つ以上の）プロセッサ102、（1つ以上の）メモリデバイス104、（1つ以上の）インターフェース106、（1つ以上の）大容量記憶デバイス108、及び（1つ以上の）入出力デバイス110が、互いに通信を行うだけでなく、バス112に結合されたその他のデバイス又は構成要素と通信できるようにする。バス112は、システムバス、PCIバス、IEEE1394バス、及びUSBバス等の幾つかの種類のバス構造体のうちの1つ以上を表す。40

#### 【0021】

説明目的で、プログラム及びその他の実行可能プログラム構成要素が別々のブロックとして本明細書では示されるが、そうしたプログラム及び構成要素は、計算デバイス100の異なる記憶構成要素中に様々な時間に存在し得、（1つ以上の）プロセッサ102により実行されることが理解される。或いは、本明細書で記述されるシステム及び手続は、ハードウェア、又はハードウェア、ソフトウェア、及び／若しくはファームウェアの組み合50

わせで実施され得る。例えば、本明細書で記述されるシステム及び手続の1つ以上を遂行するように1つ以上の特定用途向け集積回路(ASIC)がプログラムされ得る。

#### 【0022】

図2Aを参照すると、処理デバイス102は、1つ以上のスレッド200を実行してもよい。上述のように、コンピュータシステム100は、複数の処理デバイス102を含んでもよい。処理デバイス102の内の任意の1つ、又は処理デバイス102の組み合わせが、本明細書の以下の説明において“処理デバイス”102に属する機能を実行し得る。スレッドは、処理デバイス102により実行されるアプリケーションによってインスタンス化されてもよい。プロセッサは、処理資源を分配してもよく、あるスレッド200から別のスレッドへ従来の方法で実行を切り替えてよい。スレッド200は、メモリデバイス104を含み、処理デバイス102とメモリデバイス104との間に置かれたメモリコントローラを含むこともあるメモリサブシステム204に、メモリアクセスリクエスト202(例えば、読み出し及び書き込みのリクエスト)を発行してもよい。メモリアクセスリクエスト202は、<オペコード、ソースタイプ、ソースアドレス、宛先タイプ、宛先アドレス>等のフィールド(ここで、オペコードは、実行される動作を示すコードであり、ソースタイプ及び宛先タイプは、何れのメモリデバイス104へのアクセスかを示し、ソースアドレスは、データを抽出する記憶装置104内の位置であり、宛先アドレスは、データが書き込まれるメモリデバイス104内である)の内の幾つか又は全てを含む命令等、当該技術分野で周知の任意の従来型のメモリアクセスリクエストであってよい。同様に、メモリリクエスト202の処理の方法は従来型であってよい。オペコードは、読み出し動作を示してもよく、それ故、ソースアドレスからデータを抽出し、リクエストを発行したプロセッサスレッド200に返すように実行されてもよい。同様に、オペコードは、データを宛先アドレスに書き込むように実行される書き込み動作を示してもよい。

#### 【0023】

プロセッサスレッド200は、メモリアクセスリクエスト206をハードウェアバッファコントローラ208に発行してもよい。ハードウェアバッファコントローラ208は、本明細書中ではハードウェアバッファコントローラ208に属する機能を実行するよう構成された回路を含むハードウェアデバイスである。ハードウェアバッファコントローラ208は、処理デバイス102と異なる若しくは同じダイ、又はメモリサブシステム204の1つ以上の構成要素と同じダイ上に設置されてもよい。メモリアクセスリクエスト206は、従来のメモリアクセスリクエスト202と同じ又は異なるフォーマットを有してもよい。しかしながら、メモリアクセスリクエスト206は、メモリデバイス204内の何れのアドレス又はアドレス範囲も直接参照しない。代わりに、メモリアクセスリクエスト202が、特定のハードウェアバッファコントローラ208を参照する。ハードウェアバッファコントローラ208は、1つ以上のバッファを定義するために初期化されてもよく、各々のバッファの定義は、当該バッファに割り当てるメモリデバイス104のうちの1つメモリデバイス内のアドレスの割り当てを含む。該割り当ては、開始及び終了のアドレスの形式、又は開始アドレス及びバッファサイズの形式で、バッファ毎に記録されてもよい。各ハードウェアバッファコントローラ208の初期化は、最初の読み出し位置を指し示すように読み出しど点タ210を初期化することと、最初の書き込み位置を指し示すように書き込み点タ212を初期化することを含んでもよい。複数のバッファがハードウェアバッファコントローラ208内で初期化されてもよく、又は複数のハードウェアコントローラ208が同様の方法で初期化されてもよい。

#### 【0024】

バッファに対するハードウェアバッファコントローラ208の初期化は、処理デバイス102によって引き起こされてもよく、処理デバイス102は、当該技術分野で周知の任意の方法に従ってメモリデバイス104内でバッファに対してメモリを同じように割り当ててもよい。バッファの初期化は、1つ以上の処理スレッド200をインスタンス化したアプリケーションによって引き起こされよく、初期化されたバッファへの参照は、該バッファへの読み出し及び書き込みのためのスレッドによる使用のために、アプリケーション

10

20

30

40

50

に返されてもよい。

#### 【0025】

図示の実施形態では、ハードウェアバッファコントローラ208は、FIFOバッファを実施する。しかしながら、その他の種類のバッファ（例えば、後入れ先出し）も実施されてもよい。

#### 【0026】

ハードウェアバッファコントローラ208は加算器214を含んでもよい。加算器214は、メモリアクセスリクエスト206を受信し、バッファ毎にリクエスト206の数を特定する。例えば、複数のリクエスト206がプロセッサ102、バス112、又はメモリサブシステム204の1クロック周期中に受信されてもよい。具体的には、同じバッファを参照する複数のリクエストが受信されてもよい。図示の実施形態では、加算器214は、各クロック周期中にバッファ毎に受信された読み出しリクエスト206の数と、各クロック周期中にバッファ毎に受信された書き込みリクエスト206の数とを合計する。各リクエスト206は、加算器214が同じバッファ識別子を有する全ての読み出しリクエスト206と、同じバッファ識別子206を有する全ての書き込みリクエスト206を得るようにバッファ識別子を含んでもよい。10

#### 【0027】

次いで、バッファ毎の読み出しポインタ210は、各バッファを参照するクロック周期中に受信された読み出しリクエスト206の数だけ、ハードウェアバッファコントローラ208によってインクリメントされる。同様に、バッファ毎の書き込みポインタ212は、各バッファを参照するクロック周期中に受信された書き込みリクエスト206の数だけ、ハードウェアバッファコントローラ208によってインクリメントされる。20

#### 【0028】

FIFOアドレス生成器216は、加算器214の出力又は読み出しポインタ210の変化のうちの一方又は両方を検出し、あるいは、1クロック周期中にバッファ毎に受信された任意の読み出しリクエスト206及び書き込みリクエストを通知される。次いで、FIFOアドレス生成器216は、メモリアクセスリクエスト218をメモリサブシステム204へ出力する。具体的には、1クロック周期中にN個の読み出しリクエストが受信された場合、FIFOアドレス生成器216は、N個の読み出しリクエスト218又はN個のアドレスの範囲を含む読み出しリクエスト218の何れかを出力する。同様に、1クロック周期中にN個の読み出しリクエストが受信された場合、FIFOアドレス生成器216は、N個の書き込みリクエスト218、又は書き込まれるN個の値とN個のアドレスの範囲とを含む書き込みリクエスト218を生成する。30

#### 【0029】

FIFOアドレス生成器216は、読み出しポインタ210の状態に基づいたアドレスを各リクエスト218中に含む。例えば、単一の読み出し又は書き込みリクエストに対して、FIFO生成器216は、読み出し又は書き込みポインタ210、212の現在値(C)又はC+1を含む読み出し又は書き込みリクエストを出力してもよいが、これは、Cがバッファ中の次に利用可能なアドレスを指し示すか、バッファ中の以前アクセスされたアドレスの何れを指し示すかに応じて行われてもよい。N個の読み出し又は書き込みリクエストに対して、FIFO生成器216は、アドレスC、C+1、C+2、...、C+N-1を含むN個の読み出し又は書き込みリクエスト、又はCからC+N-1までのアドレスを参照する単一の読み出し又は書き込みリクエストを出力する。幾つかの実施形態では、ポインタ210、212の値は、リクエスト218に対するアドレスを特定するより前にインクリメントされる。したがって、リクエスト218のアドレスは、単一のリクエスト218について、C+1、C+2、...、C+N、又は、C+1からC+Nの範囲となる。40

#### 【0030】

前述したように、ハードウェアバッファコントローラ208は、複数のバッファを含んでもよい。したがって、MUX220は、FIFOアドレス生成器216へポインタ2150

0、212を選択的に結合してもよく、次いで、FIFOアドレス生成器216は、前述の段落で記述された方法でバッファ毎にリクエスト218を生成する。同様に、メモリアクセスリクエスト202はまた、メモリサブシステム204に入力されているため、リクエスト202、218をMUX222に入力することもでき、MUX222は、ラウンドロビン選択スキーム又はその他の負荷平衡アプローチに従ってリクエスト202、218をメモリサブシステム204中に選択的に入力する。

#### 【0031】

図2Bを参照すると、ハードウェアバッファコントローラ208は、その状態に基づいてイベント224を更に生成してもよい。イベント224は、処理デバイス102により実行されるスレッドスケジューラ226に入力されてもよい。スレッドスケジューラ226は、処理デバイス102上で実行するのはいずれのプロセッサスレッド200とするのかの確定、スレッドのロック、他のスレッドへの分岐を行い、また、当該技術分野で周知のその他のスレッド管理機能を実行する。各イベント224は、該イベントが生成されたバッファを示してもよい。10

#### 【0032】

幾つかの実施形態では、スレッドスケジューラ226は、イベント224を受信し、それについてのスレッド管理タスクを実行する。例えば、イベント224は、バッファが満杯又はほぼ満杯であることを示してもよい。したがって、スレッド200は、該バッファへの書き込みがロックされてもよい。例えば、アプリケーションが、“ほぼ満杯”のイベントに応答して、ロックを行うために、1つ以上のスレッド200をスレッドスケジューラに登録してもよい。同様に、スケジューラ226は、バッファが空又はほぼ空であることをイベント224が示すことに応答して、特定のスレッドがバッファから読み出されることをロックするようにプログラムされてもよい。20

#### 【0033】

図示の実施形態では、イベントは、読み出しカウンタ226又は書き込みカウンタ228に基づいて生成される。バッファ毎に、読み出しカウンタ226の値は、減算器230によって書き込みカウンタの値から減算される。カウンタ226、228の値の差は、1つ以上の閾値234と共に比較器232にその後入力される。

#### 【0034】

差がゼロであることは、バッファが空であること、ちょうど書き込みと同数の読み出しがあったことを示す。幾つかの実施形態では、差がゼロである場合、バッファを特定する“バッファ空”イベント224が生成されてもよい。閾値234は低閾値を含んでもよく、これにより、参照値がこの低閾値よりも少ない場合に、バッファを参照する“ほぼ空”イベント224が生成され得る。30

#### 【0035】

差がバッファのサイズと等しいことは、バッファ中のエントリの総数だけ、書き込み動作が読み出し動作よりも多いためバッファが満杯であることを示す。幾つかの実施形態では、差がバッファのサイズと等しい場合、バッファが参照される“バッファ満杯”イベントが生成されてもよい。閾値234は、バッファのサイズよりも小さい、例えば、バッファのサイズの80から90%の間に等しい高閾値を含んでもよい。差が高閾値よりも大きい場合、バッファを参照する“ほぼ満杯”イベント224が生成されてもよい。40

#### 【0036】

幾つかの実施形態では、カウンタ226、228のオーバーフローを避けるために、例えば、定期的に、又はカウンタ226、228によって表すことが可能な最大値にカウンタ226、228の内の1つが近づくことに応じて、カウンタ226、228の内の小さい方の値がカウンタ226、228から減算されてもよい。

#### 【0037】

幾つかの実施形態では、比較器232は、空、ほぼ空、満杯、又はほぼ満杯の状態がなくなった場合にイベントを生成してもよい。例えば、バッファが満杯又はほぼ満杯であることが検出され、その後、満杯又はほぼ満杯ではないことが検出された場合には、“バッ50

ファレディ ( ready ) ” 又は “ バッファ満杯解消 ” イベントが比較器 232 によって生成されてもよい。同様に、バッファが空又はほぼ空であることが検出され、その後、空又はほぼ空ではないと検出された場合には、“ バッファレディ ” 又は “ バッファ空解消 ” イベントが比較器 232 によって生成されてもよい。

#### 【 0038 】

上述したように、ハードウェアバッファコントローラ 208 は、複数のバッファを実施してもよい。したがって、こうした実施形態では、バッファ毎のカウンタ 226 、 228 間の差を評価し、必要に応じて上述したアプローチに従ってバッファ毎にイベントを生成するために、マルチプレクサ 236 、 238 は、バッファ毎のカウンタ 226 、 228 を減算器 230 及び比較器 232 に選択的に結合してもよい。

10

#### 【 0039 】

イベント 224 を受信すると、スレッドスケジューラ 224 は、プロセッサスレッド 200 に関する 1 つ以上の動作を呼び出してもよい。例えば、アプリケーションが、“ バッファ空 ” 又は “ ほぼ空 ” イベントを受信するとスレッド 200 をブロックするようにスケジューラ 226 に命じてもよい。したがって、こうしたイベントを受信すると、該スレッド 200 は、実行をブロックされるであろう。“ バッファレディ ” 又は “ バッファ空解消 ” イベントを受信すること等に応じて、“ バッファ空 ” 又は “ バッファ空 ” イベントが解消されるまで、スレッドは、スレッドスケジューラ 224 によってブロックされてもよい。

#### 【 0040 】

20

同様に、アプリケーションは、“ バッファ満杯 ” 又は “ ほぼ満杯 ” イベントを受信するとスレッド 200 をブロックするようにスケジューラ 226 に命じてもよい。したがって、こうしたイベントを受信すると、スレッド 200 は、実行をブロックされるであろう。“ バッファレディ ” 又は “ バッファ空解消 ” イベントを受信すること等に応じて、“ バッファ満杯 ” 又は “ ほぼ満杯 ” イベントが解消されるまで、スレッドは、スレッドスケジューラ 224 によってブロックされてもよい。

#### 【 0041 】

アプリケーションが、それ故、バッファに対する書き込みリクエストを実行するようにプログラムされた幾つかのスレッド 200 を有してもよく、該バッファを参照する “ バッファ満杯 ” 又は “ ほぼ満杯 ” イベントを受信すると、こうしたスレッド 200 をブロックするようにスレッドスケジューラ 226 に命じてもよい。アプリケーションは、該バッファに対する読み出しリクエストを実行するようにプログラムされた 1 つ以上の異なるスレッド 200 を有してもよく、該バッファに対する “ バッファ空 ” 又は “ ほぼ空 ” イベントを受信すると、こうしたスレッド 200 をブロックするようにスレッドスケジューラ 226 に命じてもよい。

30

#### 【 0042 】

スレッド 200 は、2 つ以上のバッファに対してリクエストを発行してもよく、アプリケーションは、それ故、ハードウェアバッファコントローラ 208 によって実施される 1 つ以上のバッファを参照するイベントに基づいてスレッドをブロックするようにスレッドスケジューラ 226 に命じてもよい。

40

#### 【 0043 】

図 3A を参照すると、図示の方法 300 は、バッファを参照する書き込みリクエストの受信に応じて、ハードウェアバッファコントローラ 208 によって実行されてもよく、該書き込みリクエストは、バッファのエントリが実際に格納されるメモリデバイス 104 内のアドレスを含まないが、バッファの識別子を含む。

#### 【 0044 】

方法 300 は、バッファに対する 1 つ以上の書き込みリクエストを受信すること ( 302 ) と、ある期間中にバッファに対して受信された書き込みリクエストの数を特定するために書き込みリクエストを合計すること ( 304 ) とを含む。例えば、処理デバイス 102 、バス 112 、メモリデバイス 104 、又はハードウェアバッファ 208 の 1 クロック

50

周期中に受信された書き込みリクエストの数がステップ 304 でカウントされてもよい。1 クロック周期中に書き込みリクエストが受信されたバッファ毎に方法 300 のステップが実行されるように、上述したように、複数のバッファがハードウェアバッファコントローラ 208 によって実施されてもよい。図 3 A 及び図 3 B の目的で、“バッファ (the buffer)” は、図示の方法が実行されているバッファを指す。

#### 【0045】

方法 300 は、書き込みポインタ 212 及び書き込みカウンタ 228 の一方又は両方を更新すること (306) を含んでもよい。幾つかの実施形態では、これらの値の 1 つのみがバッファの状態を表すために使用される。例えば、ハードウェアバッファコントローラ 208 は、カウンタではなく、バッファの状態を特定するために書き込みポインタ 212 及び読み出しポインタ 210 のみを格納してもよい。その他では、カウンタ 226、228 のみが使用され、アドレスは、該カウンタとバッファの開始アドレスとから判定される。具体的には、ポインタ 212 及びカウンタ 228 を更新することは、クロック周期中に受信された書き込みリクエストの数ずつ、それら各々をインクリメントすることを含んでもよい。

#### 【0046】

方法 300 は、書き込みポインタ 212 の状態と 302 で受信された書き込みリクエストの数に基づいてアドレスを生成すること (308) を更に含む。上述したように、ポインタの現在値が書き込みリクエストを受信すること (302) 及び書き込みポインタ 212 を更新すること (306) よりも前に C であり、N 個の書き込みリクエストが受信された場合、生成されるアドレスは、C、C+1、...、C+N-1 であってもよく、ただし、C は、次に書き込まれるアドレスを示す。C が最も直近に書き込まれたアドレスを指示する場合には、生成されるアドレスは、C+1、C+2...、C+N である。アドレスは、範囲の形式、例えば、データ対 < C, N-1 > 又は < C+1, N > の形式で生成されてもよく、ただし、一方の値が開始アドレスであり、他方の値が開始アドレスからのオフセットである。

#### 【0047】

方法 300 は、ステップ 308 で生成されたアドレスを含む 1 つ以上の書き込みコマンドであって、メモリデバイス 104 のうちの 1 つのメモリデバイス中の、1 つ以上の書き込みコマンド内のアドレスにて、書き込みリクエストからデータの書き込みを引き起こす効果のある該書き込みコマンドをメモリサブシステム 204 へ出力すること (310) を更に含んでもよい。上述したように、N 個の書き込みコマンドが、302 で受信された書き込みリクエスト毎に、ステップ 308 で生成されたアドレスの内の 1 つを夫々含んで出力されてもよい (310)。或いは、ステップ 308 で生成された N 個全てのアドレス、又は N 個全てのアドレスに渡るアドレスの範囲 (例えば、開始アドレス及びオフセット) を指定する、ただ 1 つの書き込みコマンドが出力されてもよい (310)。書き込みコマンドは、それらがアドレス指定されるメモリデバイス 104 によってその後実行され、方法は終了する。

#### 【0048】

図 3 B を参照すると、図示の方法 312 は、バッファを参照する読み出しリクエストであって、バッファの識別子を含み、バッファのエントリが実際に格納されるメモリデバイス 104 内のアドレスを含まない該読み出しリクエストの受信に応じてハードウェアバッファコントローラ 208 により実行されてもよい。

#### 【0049】

方法 312 は、バッファに対する 1 つ以上の読み出しリクエストを受信すること (314) と、ある期間中にバッファに対して受信された読み出しリクエストの数を特定するために読み出しリクエストを合計すること 316 とを含む。例えば、処理デバイス 102、バス 112、メモリデバイス 104、又はハードウェアバッファ 208 の 1 クロック周期中に受信された読み出しリクエストの数がステップ 316 でカウントされてもよい。上述したように、1 クロック周期中に書き込みリクエストが受信されたバッファ毎に方法 31

10

20

30

40

50

2 のステップが実行されるように、複数のバッファがハードウェアバッファコントローラ 208 によって実施されてもよい。

#### 【0050】

方法 312 は、読み出しポインタ 210 及び読み出しカウンタ 226 の一方又は両方を更新すること(318)を含んでもよい。幾つかの実施形態では、これらの値の 1 つのみがバッファの状態を表すために使用される。例えば、ハードウェアバッファコントローラ 208 は、カウンタではなく、バッファの状態を特定するために書き込みポインタ 212 及び読み出しポインタ 210 のみを格納してもよい。その他では、カウンタ 226、228 のみが使用され、アドレスは、該カウンタとバッファの開始アドレスとから特定される。ポインタ 210 及びカウンタ 226 を更新すること(318)は、クロック周期中に受信された書き込みリクエストの数ずつ、それら各自をインクリメントすることを含んでもよい。

10

#### 【0051】

方法 312 は、読み出しポインタ 210 の状態と 314 で受信された読み出しリクエストの数とに基づいてアドレスを生成すること(320)を更に含む。上述したように、ポインタの現在値が読み出しリクエストを受信すること(314)及び読み出しポインタ 210 を更新すること(318)よりも前に C であり、N 個の読み出しリクエストが受信された場合、生成されるアドレスは、C、C+1、...、C+N-1 であってもよく、ただし、C は、次に書き込まれるアドレスを指し示す。C が最も直近に読み出されたアドレスを指し示す場合には、生成されるアドレスは、C+1、C+2...、C+N である。アドレスは、範囲の形式、例えば、データ対 < C, N-1 > 又は < C+1, N > の形式で生成されてもよく、ただし、一方の値が開始アドレスであり、他方の値が開始アドレスからのオフセットである。

20

#### 【0052】

方法 312 は、ステップ 308 で生成されたアドレスを含む 1 つ以上の読み出しコマンドであって、メモリデバイス 104 のうちの 1 つのメモリデバイスから、1 つ以上の書き込みコマンド内のアドレスにてデータを抽出し、このデータを処理デバイス 102 へ返すことを引き起こす効果のある該読み出しコマンドをメモリサブシステム 204 へ出力すること(322)を更に含んでもよい。上述したように、N 個の読み出しコマンドが、302 で受信された読み出しリクエスト毎に、ステップ 308 で生成されたアドレスの内の 1 つを夫々含んで出力されてもよい(322)。或いは、ステップ 308 で生成された N 個全てのアドレス、又は N 個全てのアドレスに渡るアドレスの範囲(例えば、開始アドレス及びオフセット)を指定するただ 1 つの読み出しコマンドが出力されてもよい(322)。読み出しコマンドは、それらがアドレス指定されるメモリデバイス 104 によってその後実行され、方法は終了する。

30

#### 【0053】

方法 300 及び 312 は、実質的に同時に実行されてもよい。例えば、別々の読み出し及び書き込みのパイプラインがハードウェアバッファコントローラ 208 によって実施されてもよく、そうしたメモリアクセスコマンドは、読み出し及び書き込みのリクエスト両方に対して同時に生成され得る。ステップ 310 及び 322 からのメモリアクセスコマンドは、当該技術分野で周知の任意のコマンド実行スキームに従ってメモリサブシステム 204 によって順次バッファリングされ実行してもよい。

40

#### 【0054】

図 4 を参照すると、図示の方法 400 は、バッファの満杯、ほぼ満杯、空、及びほぼ空の状態を検出するためにハードウェアバッファコントローラ 208 によって実行されてもよい。方法 400 は、ハードウェアバッファコントローラ 208 によって制御されるバッファ(以後“バッファ(the buffer)”と称する)毎に実行されてもよい。

#### 【0055】

方法 400 は、図 3A に関して上述したようなクロック周期中に書き込み命令を受信すこと(302)と、クロック周期中に受信された書き込み命令を合計すること(304

50

)とのステップを含んでもよく、又は方法400にそれらのステップが先行してもよい。合計するステップ304は、方法300及び方法400の両方に対して一回実行されてもよく、又は方法毎に別々に実行されてもよい。

#### 【0056】

方法400は更に、図3Bに関して上述したようなクロック周期中に書き込み命令を受信すること(314)と、クロック周期中に受信された書き込み命令を合計すること(316)とのステップを含んでもよく、又は方法400にそれらのステップが先行してもよい。合計するステップ316は、方法300及び方法400の両方に対して一回実行されてもよく、又は方法毎に別々に実行されてもよい。

#### 【0057】

方法400は、書き込みカウンタ238を更新すること(402)と、読み出しカウンタ236を更新すること(404)とを更に含んでもよい。上述したように、これは、クロック周期中に受信された書き込みリクエストの数を書き込みカウンタ238に加算することと、クロック周期中に受信された読み出しリクエストの数を読み出しカウンタ236に加算することとを含んでもよい。

10

#### 【0058】

方法400は、書き込みカウンタと読み出しカウンタの差を計算すること(406)を更に含んでもよい。読み出しが書き込みを超えるアンダーフロー状態ではないときは、書き込みカウンタと読み出しカウンタの差は正又はゼロであろう。幾つかの実施形態では、ステップ406の差分機能を回路により実行し、カウンタ236、238をレジスタにより格納することによって処理可能な最大数を超えることを回避するために、方法400の繰り返し毎に、書き込みカウンタは該差の値に等しく設定されてもよく、読み出しポインタはゼロに設定されてもよい。

20

#### 【0059】

方法400は、1つ以上のスレッドに関してステップ406で計算された差を評価することを含んでもよい。例えば、差が“ほぼ満杯”の閾値よりも大きいと408で検出された場合、“ほぼ満杯”的イベントが410で生成されてもよい。差が“ほぼ空”的閾値よりも小さいと412で検出された場合、“ほぼ空”的イベントが414で生成されてもよい。差がバッファのサイズよりも大きいと416で検出された場合、“バッファ満杯”的イベントが418で生成されてもよい。差がゼロであると420で検出された場合、“バッファ空”的イベントが生成されてもよい。ステップ408～420の閾値は、並列バイブライインを使用して任意の順序で又は同時に評価されてもよい。

30

#### 【0060】

“ほぼ満杯”及び“ほぼ空”的閾値は、方法400の実行よりも前にハードワイヤード化またはプログラムされた所定の値でもよい。幾つかの実施形態では、“ほぼ満杯”及び“ほぼ空”的閾値は、実行するスレッドの数と、ほぼ満杯又はほぼ空の状態が存在すると検出される時とそのイベントに応じてスレッドがロックされる時との間の遅延との関数であってもよい。例えば、書き込みリクエストを発行するようにプログラムされたM個のスレッドが実行され、イベントに応答してスレッドをロックするために少なくともQのクロック周期かかる場合、“ほぼ満杯”的閾値は、 $S - Q * M - R$ に等しく設定されることもあり、ただし、Sは、バッファのサイズであり、Rは、オーバーフローが生じないようにするためにゼロよりも大きい値である。このように、ほぼ満杯の状態が検出された場合、書き込みリクエストを発行し得るM個のスレッドのブロッキングが“ほぼ満杯”的イベントによりもたらされる前にバッファのオーバーフローは生じない。

40

#### 【0061】

同様に、読み出しリクエストを発行するようにプログラムされたM個のスレッドが実行されている場合、“ほぼ空”的閾値は、 $Q * M + R$ に等しく設定されることもあり、ただし、Sは、バッファのサイズであり、Rは、アンダーフローが生じないようにするためにゼロよりも大きい値である。このように、ほぼ空の状態が検出された場合、読み出しリクエストを発行し得るM個のスレッドのブロッキングが“ほぼ空”的イベントによりもたらされ

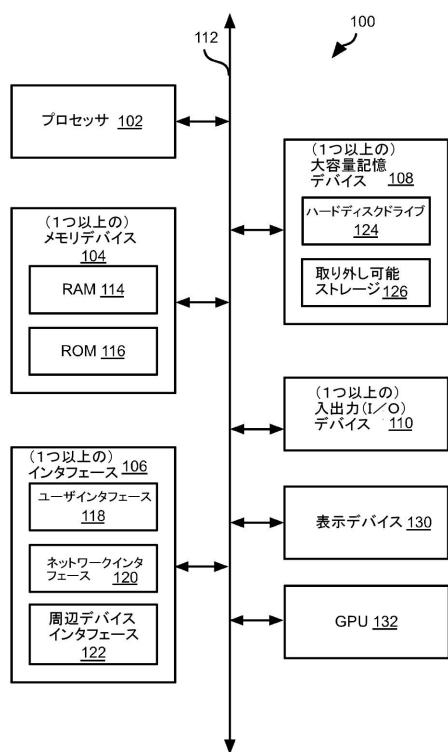
50

される前にバッファのアンダーフローは生じない。

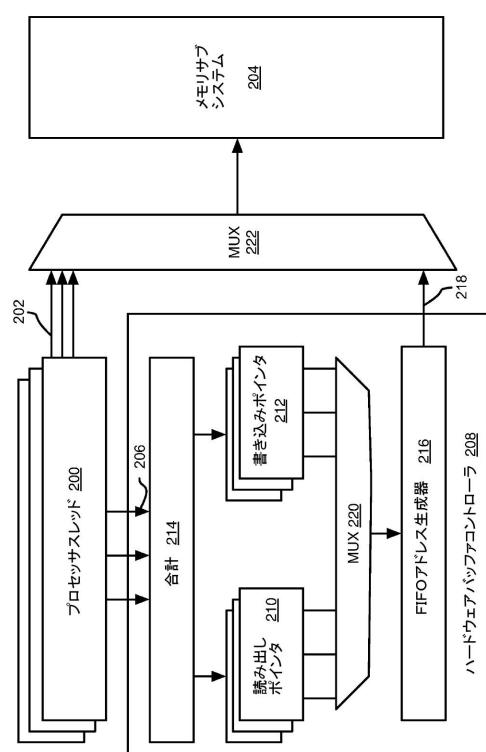
【0062】

本発明は、その精神又は本質的特徴から離れることなくその他の特定の形態で具体化されてもよい。記述された実施形態は、全ての点で、例示あって限定的ではないと考慮されるべきである。発明の範囲は、それ故、前述の説明よりもむしろ添付の請求項の範囲によつて示される。請求項の意図及び均等範囲内に入る全ての変更は、それらの範囲内に包含されるべきである。

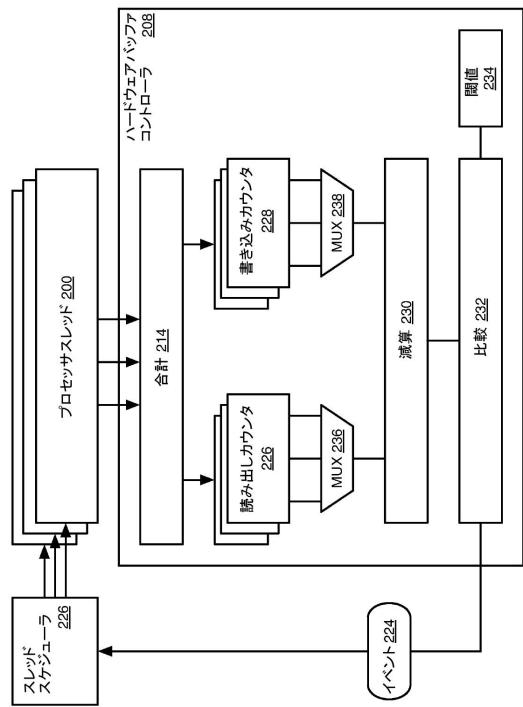
【図1】



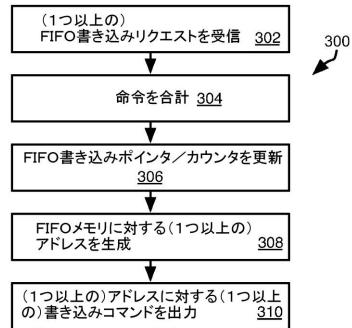
【図2A】



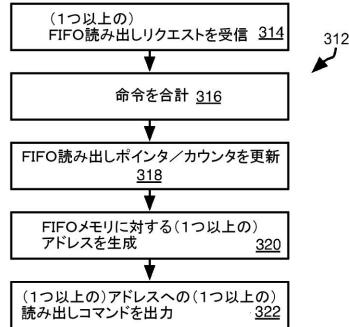
【図2B】



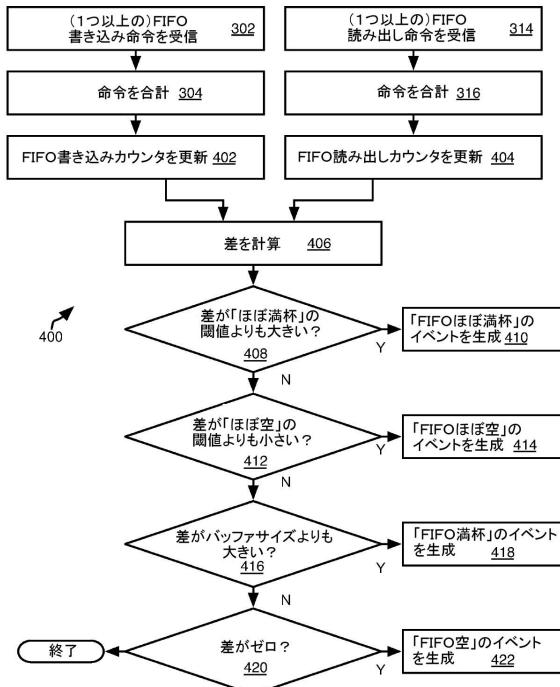
【図3A】



【図3B】



【図4】



---

フロントページの続き

(51)Int.Cl. F I  
G 06 F 5/10

(72)発明者 ロー, マンキット  
アメリカ合衆国, カリフォルニア州 95002, サンノゼ, ゴールドストリート 2150  
, スイート 200

審査官 後藤 彰

(56)参考文献 特開2013-156783 (JP, A)  
特開2008-293484 (JP, A)  
米国特許出願公開第2008/0005401 (US, A1)

(58)調査した分野(Int.Cl., DB名)

G 06 F 12/00  
G 06 F 12/02  
G 06 F 5/10  
G 06 F 9/46  
G 06 F 9/54