

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第6911102号  
(P6911102)

(45) 発行日 令和3年7月28日(2021.7.28)

(24) 登録日 令和3年7月9日(2021.7.9)

(51) Int. Cl.	F I		
<b>G06F 9/38 (2006.01)</b>	G06F	9/38	350A
<b>G06F 9/312 (2006.01)</b>	G06F	9/38	310F
<b>G06F 9/34 (2006.01)</b>	G06F	9/312	L
	G06F	9/312	W
	G06F	9/34	350A

請求項の数 20 (全 15 頁)

(21) 出願番号 特願2019-508265 (P2019-508265)  
 (86) (22) 出願日 平成28年9月19日(2016.9.19)  
 (65) 公表番号 特表2019-525351 (P2019-525351A)  
 (43) 公表日 令和1年9月5日(2019.9.5)  
 (86) 国際出願番号 PCT/US2016/052494  
 (87) 国際公開番号 W02018/034678  
 (87) 国際公開日 平成30年2月22日(2018.2.22)  
 審査請求日 令和1年9月18日(2019.9.18)  
 (31) 優先権主張番号 15/236,882  
 (32) 優先日 平成28年8月15日(2016.8.15)  
 (33) 優先権主張国・地域又は機関  
 米国 (US)

(73) 特許権者 591016172  
 アドバンスド・マイクロ・デバイス・  
 インコーポレイテッド  
 ADVANCED MICRO DEVI  
 CES INCORPORATED  
 アメリカ合衆国 95054 カリフォル  
 ニア州、 サンタ クララ、 オーガステ  
 ンドライブ 2485  
 (74) 代理人 100108833  
 弁理士 早川 裕司  
 (74) 代理人 100111615  
 弁理士 佐野 良太  
 (74) 代理人 100162156  
 弁理士 村雨 圭介

最終頁に続く

(54) 【発明の名称】 アドレス生成時のロード及びストアキューの割り当てのためのシステム及び方法

(57) 【特許請求の範囲】

【請求項1】

マイクロ動作を処理するための方法であって、  
第1マイクロ動作を含む複数のマイクロ動作をフェッチすることと、  
ロード/ストアキュー内のキューエントリが前記第1マイクロ動作に利用できないこと  
に応じて、前記第1マイクロ動作を、前記ロード/ストアキューにディスパッチすること  
なくエイジオーダースケジューラキューにディスパッチすることであって、前記エイジオ  
ーダースケジューラキューは、前記第1マイクロ動作に関連するディスパッチペイロード  
を保持する、ことと、

前記第1マイクロ動作によって識別される前記ロード/ストアキュー内のキューエント  
リが利用可能になり、前記第1マイクロ動作に必要なソース情報が準備されたことに応じ  
て、

前記第1マイクロ動作のためのアドレス生成を実行することと、  
前記第1マイクロ動作を含むディスパッチペイロードを読み出すことと、  
前記第1マイクロ動作によって指定されるようにストアされるデータを含むディスパ  
ッチペイロードを、前記第1マイクロ動作によって識別される前記ロード/ストアキュー内  
の利用可能なキューエントリに送信することと、を含む、  
 方法。

【請求項2】

ディスパッチ時に、プログラム順で各マイクロ動作を前記ロード/ストアキュー内のキ

キューエントリに関連付けて、エイジオーダーを維持することをさらに含む、  
請求項 1 の方法。

【請求項 3】

前記ロード/ストアキューからの入力に基づいて、前記エイジオーダースケジューラキュー内のコミットされていない最古のストアマイクロ動作ストアキューエントリ番号を更新することをさらに含む、

請求項 1 の方法。

【請求項 4】

前記エイジオーダースケジューラキューにおいて、前記エイジオーダースケジューラキュー内の前記コミットされていない最古のマイクロ動作キューエントリのエントリ番号を各マイクロ動作のキューエントリと比較して、前記マイクロ動作のキューエントリが利用可能かどうかを判別することをさらに含む、

請求項 3 の方法

【請求項 5】

前記マイクロ動作は、ストアマイクロ動作である、

請求項 3 の方法。

【請求項 6】

ディスパッチウィンドウサイズは、前記ロード/ストアキューの深さ及び前記エイジオーダースケジューラキューの深さをあわせたサイズである、

請求項 1 の方法。

【請求項 7】

前記ロード/ストアキューの深さを超えるディスパッチを可能にするために、前記エイジオーダースケジューラキュー内のエントリのラップビットが追跡され、ラップビットは、前記ロード/ストアキューのサイクルを表す、

請求項 1 の方法。

【請求項 8】

マイクロ動作を処理するためのプロセッサであって、

ロード/ストアキューと、

エイジオーダースケジューラキューと、

アドレス生成スケジューラと、

デコーダと、を備え、

前記デコーダは、第 1 マイクロ動作を含む複数のマイクロ動作をフェッチするように構成されており、

前記デコーダは、前記ロード/ストアキュー内のキューエントリが前記第 1 マイクロ動作に利用できないことに応じて、前記第 1 マイクロ動作を、前記ロード/ストアキューにディスパッチすることなく前記エイジオーダースケジューラキューにディスパッチするように構成されており、

前記エイジオーダースケジューラキューは、前記第 1 マイクロ動作に関連するディスパッチペイロードを保持するように構成されており、

前記アドレス生成スケジューラは、前記第 1 マイクロ動作が利用可能になることによって識別される前記ロード/ストアキュー内のキューエントリと、前記第 1 マイクロ動作が準備されることによって必要となるソース情報と、を決定するように構成されており、

前記アドレス生成スケジューラは、前記第 1 マイクロ動作が利用可能になることによって識別される前記ロード/ストアキュー内のキューエントリと、前記第 1 マイクロ動作が準備されることによって必要となるソース情報と、を決定したことに応じて、

前記第 1 マイクロ動作のためのアドレス生成を実行することと、

前記第 1 マイクロ動作の前記ディスパッチペイロードを読み出すことと、

前記第 1 マイクロ動作によって指定されるようにストアされるデータを含む前記ディスパッチペイロードを、前記第 1 マイクロ動作によって識別される前記ロード/ストアキュー内の利用可能なキューエントリに送信することと、

10

20

30

40

50

を行うように構成されている、  
プロセッサ。

【請求項 9】

前記デコーダは、プログラム順で各マイクロ動作を前記ロード/ストアキュー内のキューエントリに関連付けて、前記マイクロ動作のエイジオーダーを維持するように構成されている、

請求項 8 のプロセッサ。

【請求項 10】

前記ロード/ストアキューは、前記関連するキューエントリが利用可能であることを前記エイジオーダースケジューラキューに通知するように構成されている、

請求項 9 のプロセッサ。

【請求項 11】

前記アドレス生成スケジューラは、

前記ロード/ストアキューからの入力に基づいて、前記エイジオーダースケジューラキュー内のコミットされていない最古のストアマイクロ動作ストアキューエントリ番号を更新し、

コミットされていない最古のマイクロ動作キューエントリを各前記マイクロ動作のキューエントリと比較して、前記マイクロ動作のキューエントリが利用可能かどうかを判別するように構成されている、

請求項 8 のプロセッサ。

【請求項 12】

前記マイクロ動作は、ストアマイクロ動作である、

請求項 8 のプロセッサ。

【請求項 13】

ディスパッチウィンドウサイズは、前記ロード/ストアキューの深さ及び前記エイジオーダースケジューラキューの深さをあわせたサイズである、

請求項 8 のプロセッサ。

【請求項 14】

前記ロード/ストアキューの深さを超えるディスパッチを可能にするために、前記エイジオーダースケジューラキュー内のエントリのラップビットが追跡され、ラップビットは

、前記ロード/ストアキューのサイクルを表す、

請求項 8 のプロセッサ。

【請求項 15】

マイクロ動作を処理するための方法であって、

ロード/ストアキュー内のキューエントリが第 1 マイクロ動作に利用できないことに応じて、前記第 1 マイクロ動作を含む複数のマイクロ動作に関連するディスパッチペイロードを保持するために、前記第 1 マイクロ動作を、前記ロード/ストアキューにディスパッチすることなくスケジューラキューにディスパッチすることと、

前記ディスパッチペイロードを前記スケジューラキューにストアすることと、

第 1 マイクロ動作によって識別される前記ロード/ストアキュー内のキューエントリが利用可能であるかどうか、及び、前記第 1 マイクロ動作によって必要とされるソース情報が準備されているかどうかを判別することと、

前記ロード/ストアキュー内の前記キューエントリが利用可能になり、前記第 1 マイクロ動作によって必要とされるソース情報が準備されたことに応じて、

前記第 1 マイクロ動作のためのアドレス生成を実行することと、

前記第 1 マイクロ動作を含む前記ディスパッチペイロードを読み出すことと、

前記第 1 マイクロ動作によって指定されるようにストアされるデータを含むディスパッチペイロードを、前記第 1 マイクロ動作によって識別される前記ロード/ストアキュー内の利用可能なキューエントリに送信することと、を含む、

方法。

10

20

30

40

50

## 【請求項 16】

エイジオーダーを維持するために、プログラム順でキューエントリを各マイクロ動作に割り当てることをさらに含む、

請求項 15 の方法。

## 【請求項 17】

前記ロード/ストアキューからの入力に基づいて、コミットされていない最古のマイクロ動作キューエントリ番号を更新することをさらに含む、

請求項 15 の方法。

## 【請求項 18】

前記コミットされていない最古のマイクロ動作キューエントリを各マイクロ動作の 1 つ以上の関連するロード/ストアキューエントリと比較して、前記マイクロ動作のキューエントリが利用可能かどうかを判別することをさらに含む、

請求項 17 の方法。

## 【請求項 19】

ディスパッチウィンドウサイズは、前記ロード/ストアキューの深さ及び前記スケジューラキューの深さを あわせたサイズ である、

請求項 15 の方法。

## 【請求項 20】

前記ロード/ストアキューの深さを超えるディスパッチを可能にするために、前記ロード/ストアキューの異なるエポックが追跡され、各エポックは、前記ロード/ストアキューのサイクルを表す、

請求項 15 の方法。

## 【発明の詳細な説明】

## 【技術分野】

## 【0001】

(関連出願の相互参照)

本願は、2016年8月15日に出願された米国特許出願第15/236,882号の利益を主張し、その内容は、完全に記載されているかのように、参照により本明細書に組み込まれる。

## 【背景技術】

## 【0002】

プロセッサには、一般に、フェッチ、デコード(又は、ディスパッチ)及び実行ステージを含む命令パイプラインが関連付けられている。デコードステージは、フェッチキューから命令を取得し、エントリをロードキュー(LDQ)又はストアキュー(STQ)に割り当てる。LDQ及びSTQリソースは、その深さ(すなわち、キューに記憶することができるエントリの数)及びキューの有効期間がプロセッサの性能に影響を与える構造である。より深いLDQ及びSTQは、クロック当たりの命令(IPC)数をより多くすることができるが、ある時点で、より深いLDQ及びSTQによって、タイミング問題が発生する。LDQ及びSTQエントリは、通常、命令のディスパッチ時に割り当てられる。ロード又はストア動作がディスパッチポイントにあるが、LDQ及びSTQが満杯である場合、LDQ及びSTQエントリが割り当て解除されるまでディスパッチが停止される。これは、LDQ及びSTQの深さがプロセッサの有効ウィンドウサイズに対する制限となること、すなわち、一度に処理できる命令の数に制限があることを意味する。

## 【0003】

添付の図面と共に例として与えられる以下の説明から、より詳細な理解が得られるであろう。

## 【図面の簡単な説明】

## 【0004】

【図1】特定の実施態様による、プロセッサのコア処理ユニットのハイレベルブロック図及びフロー図である。

10

20

30

40

50

【図2】特定の実施態様による、整数スケジューラ及び/又は実行ユニットのハイレベルブロック図及びフロー図である。

【図3】特定の実施態様による、スケジューラ及び/又は実行ユニットのハイレベルブロック図及びフロー図である。

【図4】特定の実施態様による、アドレス生成時のロード及びストアキューの割り当てのための方法のフロー図である。

【図5】1つ以上の開示された実施態様を実装することができる例示的なデバイスのブロック図である。

【発明を実施するための形態】

【0005】

プロセッサは、一般に、フェッチ、デコード（又は、ディスパッチ）及び実行ステージを含む命令パイプラインに従って動作する。デコード又はディスパッチステージは、フェッチキューから命令を取得し、エントリをロードキュー（LDQ）又はストアキュー（STQ）に割り当てる。LDQ及びSTQエントリは、通常、命令のディスパッチ時に割り当てられる。ロード又はストア動作がディスパッチポイントにあるが、LDQ又はSTQが満杯である場合、LDQ又はSTQエントリが割り当て解除されるまでディスパッチが停止される。これは、LDQ及びSTQの深さが、プロセッサの有効ウィンドウサイズに対する制限となることを意味する。

【0006】

命令のエイジオーダー（age-order）を維持する、アドレス生成時のLDQ及びSTQエントリの割り当てのためのシステム及び方法が説明される。具体的には、LDQ及びSTQエントリの書き込みは、アドレス生成時まで延期される。これにより、ロード及びストア動作がディスパッチされ、若い動作（ストア及びロード動作ではない可能性がある）もディスパッチされ、これらの命令を開始又は完了させることができる。ロード又はストア動作のアドレス生成は、ロードキューエントリ又はストアキューエントリが当該動作に使用可能になるまで、アドレス生成スケジューラキュー（AGSQ）で保持される。LDQエントリ又はSTQエントリの追跡は、ディスパッチポイントではなく（すなわち、デコードエンジンではなく）AGSQで効果的に行われる。全てがAGSQにプッシュダウンされると、LDQ及びSTQの深さがデコードエンジン側からは見えず、以下に説明するように有効な深さが増加する。

【0007】

図1は、特定の実施態様による、プロセッサ100のコア処理ユニット105のハイレベルブロック図及びフロー図である。コア処理ユニット105は、スケジューラ及び実行ユニット（SC EX）115にマイクロ動作（マイクロオペレーション）を提供するデコーダユニット110を含む。デコーダユニット110は、キャッシュ122及びマイクロ動作キャッシュ124の各々に接続された分岐予測器120を含む。また、キャッシュ122は、デコーダ126に接続されている。デコーダ126及びマイクロ動作キャッシュ124の各々は、マイクロ動作キュー128に接続されている。

【0008】

SC EX 115は、整数SC EX 130と、浮動小数点SC EX 132と、を含み、これらの両方がキャッシュ134に接続されている。また、キャッシュ134は、L2キャッシュ136と、LDQ 138と、STQ 140と、に接続されている。整数SC EX 130は、算術論理ユニット（ALU）スケジューラキュー（ALSQ）152と、アドレス生成ユニット（AGU）スケジューラキュー（AGSQ）154と、を含むスケジューラ151に接続された整数リネーマ（integer renamer）150を含む。さらに、スケジューラ151（具体的には、ALSQ 152及びAGSQ 154）は、ALU 156及びAGU 158の各々に接続されている。整数SC EX 130は、整数物理ファイルレジスタ160も含む。浮動小数点SC EX 132は、スケジューラ172に接続された浮動小数点リネーマ（floating point renamer）170を含む。また、スケジューラ172は、乗算器174及び加算器176の各々に接続されている。さらに、浮動小数点SC EX 1

10

20

30

40

50

32は、浮動小数点物理ファイルレジスタ178を含む。

【0009】

パイプラインプロセッサは、命令のストリームをパイプラインに安定に供給する必要がある。分岐予測器120は、何れのマイクロ動作のセットがパイプライン化プロセッサでフェッチされ、実行されるかを予測する。これらのマイクロ動作は、フェッチされ、キャッシュ122に記憶され、次いで、キャッシュ122がデコーダ126によってデコードされる。マイクロ動作キャッシュ124は、キャッシュ122からのマイクロ動作をデコーダ126がデコードする場合に、分岐予測器120からのマイクロ動作をキャッシュする。マイクロ動作キュー128は、マイクロ動作を実行のためにディスパッチする目的で、デコーダ126及びマイクロ動作キャッシュ124からのマイクロ動作を記憶してキューに入れる。

10

【0010】

従来のパイプライン処理では、マイクロ動作キューは、マイクロ動作に関連するペイロードを記憶するために、マイクロ動作をロードキュー又はストアキューにディスパッチする。例示すると、ストアキューは、マイクロ動作キューから6つのマイクロ動作（このうち4つはストアマイクロ動作である）を受け入れ、ディスパッチ時にディスパッチペイロードをストアキューに書き込むことができる。次に、アドレス生成時に、ストアキューは、スケジューラからキューエントリ番号を受信し、何れのストアキューエントリが処理されているかを特定する。スケジューラは、ディスパッチペイロードを読み出し、セグメンテーションチェックのためにセグメンテーションロジックに送信し、マイクロ動作パイプ

20

【0011】

物理的に、これはいくつかの理由で困難である。第1に、アーキテクチャは、マイクロ動作キューの全ての幅をサポートする必要がある。例えば、ディスパッチ時に6つのマイクロ動作（4つのストアマイクロ動作を含む）を書き込むと、大量のワイヤ及びマルチプレクサが必要となり、ディスパッチペイロードを保持するストアキューの領域で輻輳が発生する。第2に、アドレス生成時にペイロードを読み出すことは、アドレス生成ユニットからストアキューにキューエントリ番号を送信することと、ペイロードを読み出すことと、ペイロードをセグメンテーションロジック及びロードキューに送信することと、を含む困難なパス（すなわち、タイミングパス）のセットを保証することを必要とする。ストアキューはディスパッチペイロードを保持しているので、アドレスが生成されるまでストアマイクロ動作では何もできないにもかかわらず、ストアキューエントリがディスパッチ時から占有されることを意味する。この結果、マイクロ動作の幅に必要なアーキテクチャが無駄になっている。すなわち、ストアキュー及びロードキューの深さによって、プロセッサのウィンドウが制限される（すなわち、一度に処理可能な動作の数に制限がある）。

30

【0012】

一実施態様によれば、マイクロ動作は、従来のパイプライン処理のようにLDQ138及びSTQ140に直接書き込むのではなく、整数SCEX130及び浮動小数点SCEX132のみにディスパッチされる。具体的には、マイクロ動作は、1)整数リネーム150を介してスケジューラ151に送られ、2)浮動小数点リネーム170を介してスケジューラ172に送られる。スケジューラ151は、AGSQ154において、マイクロ動作のディスパッチペイロード（例えば、ロード及びストアマイクロ動作のディスパッチペイロード）を保持する。AGSQ154は、適切なLDQ138又はSTQ140内の特定のキューエントリが利用可能になるまで、マイクロ動作（例えば、ロード及びストアマイクロ動作）のディスパッチペイロードを保持する。キューエントリが利用可能になり、ソース情報が準備されると、AGSQ154は、アドレスを生成し、ディスパッチペイロードを読み出し、ディスパッチペイロードをLDQ138又はSTQ140に送信する。

40

50

## 【 0 0 1 3 】

エイジオーダーの動作又は順番通りのキューを維持するために、全てのストアマイクロ動作は、特定のストアキューエントリ又はキュー識別子と関連付けられる。ストアキューエントリは、プログラム順でディスパッチ時にストアマイクロ動作に割り当てられる（例えば、ストアマイクロ動作をストアキューエントリ 0 に割り当てることができ、プログラム順で次のストアマイクロ動作をストアキューエントリ 1 に割り当てることができ、プログラム順でさらに次のストアマイクロ動作をストアキューエントリ 2 に割り当てることができる等）。これにより、ストアマイクロ動作は、キューのエントリ番号又は位置によってストアキューに順番に保持される。位置順（ordered-by-position）ストアキューは、ストアマイクロ動作を処理するための簡単な方法であり、当業者に知られている。本実施態様では、ディスパッチ時に、ストアキューエントリがマイクロ動作に利用可能であることを必要とすることなく、その順番通りのプロパティ及びディスパッチ時のキューエントリの割り当てを保持する。

10

## 【 0 0 1 4 】

ディスパッチ時にストアキューエントリを割り当てることにより、AGSQ及びALSQは、AGSQ及びALSQにおいてコミットされていない最古のストアマイクロ動作ストアキューエントリ番号を、ストアマイクロ動作のストアキューエントリと比較することによって、ストアキューエントリがいつ解放されるかを容易に判別することができる。特に、スケジューラ 151 は、AGSQ 154 がいつアドレス生成を実行してディスパッチペイロードをSTQ 140 に送信することができるか、及び、ALSQ 152 がいつストアデータをSTQ 140 に送信することができるかを知る必要がある。したがって、特定のキューエントリが利用可能になると、当該特定のキューエントリは、STQ 140 によってAGSQ 154 及びALSQ 152 に通知される。これについては、図 1 の出力 180 に関して後述する。

20

## 【 0 0 1 5 】

特に、LDQ 138 及びSTQ 140 は、コミット - 割り当て解除信号をスケジューラ 151（例えば、AGSQ 154 及びALSQ 152）に送信して、スケジューラ 151（例えば、AGSQ 154 及びALSQ 152）が、古いストアマイクロ動作が各々のストアキューエントリを割り当て解除して解放するとき、アドレス生成、ディスパッチペイロードの送信、及び、若いストアマイクロ動作のストアデータの送信を可能にするように、コミットされていない最古のストアマイクロ動作のストアキューエントリ番号を更新することができる。これは、例えば、LDQ 138 及びSTQ 140 からの出力 180 を、スケジューラ 151（例えば、AGSQ 154 及びALSQ 152）の入力に追加することによって実施される。

30

## 【 0 0 1 6 】

スケジューラ 151 は、AGSQ 154 内の各ストアマイクロ動作及びALSQ 152 内の各ストアデータマイクロ動作のストアキューエントリ番号を認識しており、コミットされていない最古のストアマイクロ動作のストアキューエントリ番号を（ラップビットと共に）認識しているので、スケジューラ 151 は、キューエントリが利用可能な適切なマイクロ動作を判別することができる。例えば、AGSQ 154 エントリ又はALSQ 152 エントリのストアキューエントリ番号が、（ラップビットを適切に考慮する）コミットされていない最古のストアマイクロ動作のキューエントリのストアキューの深さ（XC\_\_STQDEPTH）内にある場合、当該ストアマイクロ動作のストアキューエントリが（適切なソース情報又はソースオペランドも準備ができていると仮定して）当該ストアマイクロ動作に利用可能であるため、ストアマイクロ動作のためのアドレス生成を進めること又はストアマイクロ動作ディスパッチペイロードを送信することができる。最古のストアマイクロ動作よりも若いXC\_\_STQDEPTH - 1 ストアであるストアマイクロ動作よりも若いストアマイクロ動作は、これらのストアキューエントリが依然として古いストアマイクロ動作によって占められているので、アドレス生成を実行すること、ディスパッチペイロードを送信すること又はストアデータを送信することが許可されず、マイクロ動作

40

50

ピックアップ（すなわち、ストアマイクロ動作を実行すること）の準備ができていないとみなされる。

【0017】

全てのディスパッチペイロードをAGSQ154に保持し、アドレス生成時までストアキューの割り当てを遅らせることによって、ストアキューサイズ（例えば、ストアキュー140のサイズ）よりも多くのストアマイクロ動作をディスパッチすることができる。ディスパッチストールの原因を排除することによって、さらなるマイクロ動作をウィンドウ内に導入し、実行を開始するのを可能にすることができる。すなわち、いくつかのストアマイクロ動作は、これらのストアキューエントリ内の前のストアマイクロ動作が割り当て解除されるまで開始することができないが、他のマイクロ動作を進めることができる。重要なことは、キャッシュミスであるロードは、アドレス生成をディスパッチして実行し、これらのキャッシュミスを開始させることができる。

10

【0018】

ストアキューエントリよりも多くの数のストアマイクロ動作をウィンドウ内で処理するためのサポートには、これらのマイクロ動作のエイジ（age）を比較するためのメカニズムが必要である。エイジを比較するためのメカニズムは、ストアマイクロ動作に関連するストアキューエントリ番号と、ストアキューエントリ番号に付随する「ラップ」ビットと、を使用することによって提供される。ラップビットは、関連するストアマイクロ動作が、ストアキューエントリの何れの「エポック」を使用するかを決定する。各「エポック」は、ストアキューの異なるバージョン又はサイクルを表す。単一のラップビットは、ストアキューの2つの異なる「ラップ」又は「エポック」を追跡する手段を提供する。これにより、フルストアキュー（XC\_\_STQDEPTH）のディスパッチが可能になる。ストアキューエントリよりも多くのストアマイクロ動作のディスパッチが可能な場合、ウィンドウ内に同じストアキューエントリを有するマイクロ動作が存在する場合だけでなく、ストアキューの複数の異なる「ラップ」又は「エポック」からのマイクロ動作が存在する場合がある。合計で2つのラップビットに対して1つの追加のラップビットは、ストアキューの4つの異なる「ラップ」又は「エポック」を追跡するための手段を提供し、ストアキューの深さの3倍までのディスパッチを可能にする。

20

【0019】

図示した例では、実装されたアーキテクチャが44のストアキューの深さを有し、（アドレス生成時には、最大28の追加のマイクロ動作ストアに対して）2つの14エントリのAGSQがある場合、ウィンドウ内にディスパッチすることが可能なストアの合計は72になる。したがって、プロセッサは、ストアキューの深さの2倍を超えてディスパッチせず、2つのラップビットを用いて、プロセッサ内の72のストアマイクロ動作の全てのエイジを追跡し比較する。この場合、ディスパッチストールは不要になる。ラップビットは、ディスパッチ時に計算されるものであり、AGSQに保持されているディスパッチペイロードの一部である。AGSQスケジューラの深さが、ストアキューの深さの3倍を超えてストアマイクロ動作のディスパッチを可能にする場合、追加のラップビットを追加して、任意の数のストアマイクロ動作のディスパッチを可能にすることができる。

30

【0020】

ロードマイクロ動作は、必ずしもエイジオーダーではなく、当業者に知られている他の技術を使用する。一実施態様では、ロードマイクロ動作は、ストアマイクロ動作と同様に動作する。

40

【0021】

アーキテクチャの観点からは、本明細書に記載の実施態様は、上記で概説した問題を解決する。第1に、ストアキュー内のディスパッチペイロード書き込みポートの数が低減される。例えば、ディスパッチペイロード書き込みポートの数は、4つ（ディスパッチ時に1サイクル当たり4ストア）から2つ（1サイクル当たり2ストアアドレス生成）に低減する。第2に、困難なタイミングパスが排除される。例えば、キューエントリ番号をストアキューに送信することと、ペイロードを読み出すことと、ペイロードをセグメンテーシ

50

ョンロジック及びロードキューに送信することと、を含むタイミングパスが排除される。

【 0 0 2 2 】

A G S Q 1 5 4 によってアドレス生成が実行され、ディスパッチペイロードが必要に応じて L D Q 1 3 8 及び S T Q 1 4 0 に記憶されると、コア処理ユニット 1 0 5 は、これに応じてマイクロ動作を実行する。L D Q 1 3 8 及び S T Q 1 4 0 の各々は、ロードマイクロ動作のデータを返し、ストアマイクロ動作の書き込みを実行し、スケジューラ 1 5 1 及びスケジューラ 1 7 2 は、各々のソース情報が準備完了になると、マイクロ動作を整数 S C E X 1 3 0 及び浮動小数点 S C E X 1 3 2 に発行する。

【 0 0 2 3 】

図 2 は、特定の実施態様による、整数スケジューラ/実行ユニット 2 0 0 のハイレベルブロック図である。整数スケジューラ/実行ユニット 2 0 0 は、A L S Q 2 0 4 と、A G S Q 2 0 6 と、リタイアキュー 2 0 8 に接続されている整数リネーマ/マップパー (mapper) 2 0 2 と、を含む。また、A L S Q 2 0 4 及び A G S Q 2 0 6 は、転送マルチプレクサ 2 1 0 に接続されており、転送マルチプレクサ 2 1 0 は、A L U 2 1 2 及び A G U 2 1 4 の各々に接続されている。A G U 2 1 4 は、L D Q 2 1 6 及び S T Q 2 1 8 に接続されている。さらに、整数スケジューラ/実行ユニット 2 0 0 は、物理ファイルレジスタ 2 2 0 を含む。

【 0 0 2 4 】

図 1 と同様に、マイクロ動作は、A G S Q 2 0 6 のみにディスパッチされ、L D Q 2 1 6 及び S T Q 2 1 8 にはディスパッチされない。A G S Q 2 0 6 は、要求されたソース情報及び適切なロードキュー又はストアキューエントリが利用可能になるまで、ディスパッチペイロードを保持する。具体的には、A G S Q 2 0 6 は、要求されたソース情報の準備ができており、特定のマイクロ動作のロードキューエントリ又はストアキューエントリが L D Q 2 1 6 及び S T Q 2 1 8 で利用可能になった場合に、特定のロード又はストアマイクロ動作のためのアドレス生成を実行する。これにより、マイクロ動作のエイジオーダーの実行が維持される。A G S Q 2 0 6 及び A L S Q 2 0 4 の各々は、L D Q 2 1 6 又は S T Q 2 1 8 からの出力信号 2 3 0 を介して A G S Q 2 0 6 及び A L S Q 2 0 4 にシグナリングすることによって、特定のキューエントリが利用可能になった場合に L D Q 2 1 6 及び S T Q 2 1 8 によって通知される。A L S Q 2 0 4 内のストアデータマイクロ動作は、特定のストアマイクロ動作のストアキューエントリが利用可能になった場合にのみ、ストアデータを実行してストアキューエントリに提供する。

【 0 0 2 5 】

アドレス生成が A G S Q 2 0 6 によって実行され、ディスパッチペイロードが必要に応じて、L D Q 2 1 6 及び S T Q 2 1 8 に保持されると、整数 S C E X 2 0 0 は、マイクロ動作を実行する。L D Q 2 1 6 及び S T Q 2 1 8 の各々は、ロードマイクロ動作のデータを返し、ストアマイクロ動作の書き込みを実行し、スケジューラは、これらのソース情報が準備完了になると、マイクロ動作を整数 S C E X 2 0 0 に発行する。

【 0 0 2 6 】

図 3 は、特定の実施態様による、ロードストア/データキャッシュ (L S D C) ユニット 3 0 0 のハイレベルブロック図及びフロー図である。L D S C ユニット 3 0 0 は、L D Q 3 0 2 と、S T Q 3 0 4 と、ロード 0 (L 0) ピッカー (picker) 3 0 6 と、ロード 1 (L 1) ピッカー 3 0 8 と、を含む。L 0 ピッカー 3 0 6 は、変換ルックアサイドバッファ (T L B) 及びマイクロタグアクセスパイプライン 0 (T L B 0) 3 1 0 と、データキャッシュアクセスパイプライン (データパイプ 0) 3 1 2 と、に接続されている。L 1 ピッカー 3 0 8 は、変換ルックアサイドバッファ (T L B) 及びマイクロタグアクセスパイプライン 1 (T L B 1) 3 1 4 と、データキャッシュアクセスパイプライン (データパイプ 1) 3 1 6 と、に接続されている。また、T L B 0 3 1 0 及び T L B 1 3 1 4 は、L 1 / L 2 T L B 3 1 8 と、ページウォーカー 3 2 3 と、マイクロタグアレイ 3 1 9 と、に接続されており、マイクロタグアレイ 3 1 9 は、ミスアドレスバッファ (M A B) 3 2 0 に接続されており、キャッシュ 3 2 2 からのデータの読み出しをサポートする。デー

10

20

30

40

50

パイプ0 312及びデータパイプ1 316は、キャッシュ322に接続されている。STQ304は、プリフェッチャ324及びストアパイプピッカー326に接続されており、ストアパイプピッカー326は、ストアパイプライン(STP)328に接続されている。STP328は、L1/L2 TLB318及びマイクロタグアレイ319にも接続されている。また、STQ304は、ストアコミットパイプライン330に接続されており、ストアコミットパイプライン330は、書き込み結合バッファ(WCB)332及びキャッシュ322に接続されている。

#### 【0027】

アドレス生成がAGSQによって実行され、ディスパッチペイロードが必要に応じてLDQ302及びSTQ304に保持されると、LDSC300は、マイクロ動作を実行する。図示した例では、(L0ピッカー306又はL1ピッカー308等で)ロードマイクロ動作が選択(ピッキング)されると、ロードマイクロ動作は、各々のTLB0 310又はTLB1 314のパイプラインを使用して、ロードマイクロ動作の変換及び予測されたデータキャッシュウェイについて、各々のTLB0 310又はTLB1 314のパイプラインをチェックする。また、ロードマイクロ動作は、データパイプ0 312及びデータパイプ1 316を介してキャッシュ322をチェックする。特定の実施態様では、マイクロタグアレイ319によって、マイクロ動作が、フルタグと比較することによってウェイを確認する前に、予測されたデータキャッシュウェイを決定することができる。ページウォーカー323は、マイクロ動作の物理アドレスを決定するために使用される。別の図示した例では、プリフェッチャ324は、要求がキャッシュ322に送信される前に、キャッシュ322内のラインを埋めるために使用される。

#### 【0028】

図4は、いくつかの実施態様による、アドレス生成時のロード及びストアキューの割り当てのための方法のフロー図400である。マイクロ動作がフェッチされ、デコードされ、マイクロ動作キューに提供される(ステップ402)。マイクロ動作キューは、マイクロ動作ディスパッチペイロードを記憶するために、マイクロ動作をアドレス生成ユニットスケジューラにディスパッチする(ステップ404)。ディスパッチは、エイジオーダー又はプログラム順を維持するために行われる。アドレス生成ユニットスケジューラは、ロードキュー又はストアキューによって送信された情報から、特定の(すなわち、エイジオーダーの)キューエントリが利用可能かどうかを判別する(ステップ406)。特定のキューエントリが利用可能でない場合、アドレス生成ユニットスケジューラは、現状を維持し、再度チェックする(ステップ408)。

#### 【0029】

特定のキューエントリが利用可能である場合(ステップ410)、アドレス生成ユニットスケジューラは、マイクロ動作に必要なソース情報が準備できているかどうかを判別する(ステップ412)。ソース情報が準備できていない場合、アドレス生成ユニットスケジューラは、現状を維持し、再度チェックする(ステップ414)。ソース情報が準備できている場合(ステップ416)、アドレス生成ユニットスケジューラは、アドレス生成を実行し、マイクロ動作ディスパッチペイロードを読み出し、マイクロ動作ディスパッチペイロードを適切なロードキュー又はストアキューに送信する(ステップ418)。次に、マイクロ動作が実行される(ステップ420)。動作の順序は例示に過ぎず、他の順序を使用することができる。

#### 【0030】

図5は、1つ以上の開示された例のうち1つ以上の部分を実装されている例示的なデバイス500のブロック図である。デバイス500は、例えば、ヘッドマウントデバイス、サーバ、コンピュータ、ゲームデバイス、ハンドヘルドデバイス、セットトップボックス、テレビ、携帯電話又はタブレットコンピュータを含む。デバイス500は、計算ノード又はプロセッサ502と、メモリ504と、ストレージデバイス506と、1つ以上の入力デバイス508と、1つ以上の出力デバイス510と、を含む。また、デバイス500は、入力ドライバ512及び出力ドライバ514をオプションで含むことができる。デバ

10

20

30

40

50

イス500は、図5に示されていない追加のコンポーネントを含んでもよいことを理解されたい。

【0031】

計算ノード又はプロセッサ502は、中央処理装置(CPU)、グラフィックス処理装置(GPU)、同じダイ上に配置されたCPU及びGPU、又は、1つ以上のプロセッサコアを含み、各プロセッサコアはCPU又はGPUであってもよい。メモリ504は、計算ノード又はプロセッサ502と同じダイ上に配置されてもよいし、計算ノード又はプロセッサ502とは別に配置されてもよい。一実施態様では、メモリ504は、例えば、ランダムアクセスメモリ(RAM)、ダイナミックRAM若しくはキャッシュ等の揮発性又は不揮発性メモリを含む。

10

【0032】

ストレージデバイス506は、例えば、ハードディスクドライブ、ソリッドステートドライブ、光ディスク若しくはフラッシュドライブ等の固定又は取り外し可能なストレージデバイスを含む。入力デバイス508は、キーボード、キーパッド、タッチスクリーン、タッチパッド、検出器、マイクロフォン、加速度計、ジャイロスコープ、バイオメトリックスキャナ又はネットワーク接続(例えば、無線IEEE802信号を送信及び/若しくは受信するための無線ローカルエリアネットワークカード)を含む。出力デバイス510は、ディスプレイ、スピーカ、プリンタ、触覚フィードバックデバイス、1つ以上のライト、アンテナ又はネットワーク接続(例えば、無線IEEE802信号を送信及び/若しくは受信するための無線ローカルエリアネットワークカード)を含む。

20

【0033】

入力ドライバ512は、計算ノード又はプロセッサ502、及び、入力デバイス508と通信し、計算ノード又はプロセッサ502が、入力デバイス508から入力を受信することを可能にする。出力ドライバ514は、計算ノード又はプロセッサ502、及び、出力デバイス510と通信し、プロセッサ502が出力デバイス510に出力を送信することを可能にする。入力ドライバ512及び出力ドライバ514は、オプションのコンポーネントであり、デバイス500は、入力ドライバ512及び出力ドライバ514が存在しない場合であっても同様に動作することに留意されたい。

【0034】

一実施態様では、マイクロ動作を処理するための方法は、マイクロ動作をフェッチし、当該マイクロ動作をエイジオーダースケジューラキューにディスパッチする。エイジオーダースケジューラキューは、各マイクロ動作に関連するディスパッチペイロードを保持する。次に、方法は、キュー内の関連するキューエントリが利用可能であり、マイクロ動作に必要なソース情報が準備できていることを条件として、マイクロ動作のためのアドレス生成を実行する。マイクロ動作のディスパッチペイロードが読み出され、キューに送信される。一実施態様では、方法は、ディスパッチ時に、プログラム順で各マイクロ動作をキュー内のキューエントリに関連付けて、エイジオーダーを維持する。一実施態様では、方法は、キューからの入力に基づいて、コミットされていない最古のマイクロ動作キューエントリを更新する。一実施態様では、方法は、エイジオーダースケジューラキューにおいて、コミットされていない最古のマイクロ動作キューエントリを各マイクロ動作のキューエントリと比較して、マイクロ動作のキューエントリが利用可能かどうかを判別する。一実施態様では、マイクロ動作は、ストアマイクロ動作である。一実施態様では、ディスパッチウィンドウサイズは、キューの深さ及びエイジオーダースケジューラキューの深さである。一実施態様では、ラップビットは、キューの深さを超えるディスパッチを可能にするために、キューの異なるエポックを追跡するために使用される。

30

40

【0035】

一実施態様では、マイクロ動作を処理するためのプロセッサは、キューと、エイジオーダースケジューラキューと、デコーダと、を含む。デコーダは、マイクロ動作をエイジオーダースケジューラキューにディスパッチする。エイジオーダースケジューラキューは、各マイクロ動作に関連するディスパッチペイロードを保持し、キュー内の関連するキュー

50

エントリが利用可能であり、マイクロ動作に必要なソース情報が準備できていることを条件として、マイクロ動作のためのアドレス生成を実行し、マイクロ動作のディスパッチペイロードを読み出し、ディスパッチペイロードをキューに送信する。一実施態様では、デコーダは、プログラム順で各マイクロ動作をキュー内のキューエントリに関連付けて、エイジオーダーを維持する。一実施態様では、キューは、関連するキューエントリが利用可能であることをエイジオーダースケジューラキューに通知する。一実施態様では、スケジューラは、キューからの入力に基づいて、コミットされていない最古のマイクロ動作キューエントリを更新し、コミットされていない最古のマイクロ動作キューエントリを各マイクロ動作のキューエントリと比較して、マイクロ動作のキューエントリが利用可能かどうかを判別する。一実施態様では、マイクロ動作は、ストアマイクロ動作である。一実施態様では、ディスパッチウィンドウサイズは、キューの深さ及びエイジオーダースケジューラキューの深さである。一実施態様では、ラップビットは、キューの深さを超えるディスパッチを可能にするために、キューの異なるエポックを追跡するために使用される。

10

## 【 0 0 3 6 】

一実施態様では、マイクロ動作を処理するための方法は、各マイクロ動作に関連するディスパッチペイロードを保持するためにマイクロ動作をスケジューラキューにディスパッチし、利用可能な関連するキューエントリを有するマイクロ動作であって、マイクロ動作の準備ができているソース情報を有するマイクロ動作のためのアドレス生成を実行する。次に、方法は、マイクロ動作のディスパッチペイロードを読み出し、マイクロ動作のディスパッチペイロードをキューに送信する。一実施態様では、方法は、エイジオーダーを維持するために、プログラム順でキューエントリを各マイクロ動作に割り当てる。一実施態様では、方法は、コミットされていない最古のマイクロ動作キューエントリを更新する。一実施態様では、方法は、スケジューラキューにおいて、コミットされていない最古のマイクロ動作キューエントリを各マイクロ動作のキューエントリと比較して、マイクロ動作のキューエントリが利用可能かどうかを判別する。一実施態様では、ディスパッチウィンドウサイズは、キューの深さ及びスケジューラキューの深さである。一実施態様では、ラップビットは、キューの深さを超えるディスパッチを可能にするために、キューの異なるエポックを追跡するために使用される。

20

## 【 0 0 3 7 】

一実施態様では、非一時的なコンピュータ可読記憶媒体は、処理システムで実行されると、アドレス生成時のロード及びストアの割り当てのための方法を処理システムに実行させる命令を含む。方法は、マイクロ動作をフェッチし、当該マイクロ動作をエイジオーダースケジューラキューにディスパッチする。エイジオーダースケジューラキューは、各マイクロ動作に関連するディスパッチペイロードを保持する。次に、方法は、キュー内の関連するキューエントリが利用可能であり、マイクロ動作に必要なソース情報が準備できていることを条件として、マイクロ動作のためのアドレス生成を実行する。マイクロ動作のディスパッチペイロードが読み出され、キューに送信される。一実施態様では、方法は、ディスパッチ時に、プログラム順で各マイクロ動作をキュー内のキューエントリに関連付けて、エイジオーダーを維持する。一実施態様では、方法は、キューからの入力に基づいて、コミットされていない最古のマイクロ動作キューエントリを更新する。一実施態様では、方法は、エイジオーダースケジューラキューにおいて、コミットされていない最古のマイクロ動作キューエントリを各マイクロ動作のキューエントリと比較して、マイクロ動作のキューエントリが利用可能かどうかを判別する。一実施態様では、マイクロ動作は、ストアマイクロ動作である。一実施態様では、ディスパッチウィンドウサイズは、キューの深さ及びエイジオーダースケジューラキューの深さである。一実施態様では、ラップビットは、キューの深さを超えるディスパッチを可能にするために、キューの異なるエポックを追跡するために使用される。

30

40

## 【 0 0 3 8 】

本明細書における開示に基づいて多くの変形が可能であることを理解されたい。上記では、特徴及び要素が特定の組み合わせで説明されているが、各特徴又は要素は、他の特徴

50

及び要素無しに単独で使用されてもよいし、他の特徴及び要素を伴って又は伴わずに様々な組み合わせで使用されてもよい。

【0039】

提供された方法は、汎用コンピュータ、プロセッサ又はプロセッサコアで実施されてもよい。適切なプロセッサには、例として、汎用プロセッサ、専用プロセッサ、従来のプロセッサ、デジタル信号プロセッサ(DSP)、複数のマイクロプロセッサ、DSPコアに関連する1つ以上のマイクロプロセッサ、コントローラ、マイクロコントローラ、特定用途向け集積回路(ASIC)、フィールドプログラマブルゲートアレイ(FPGA)回路、他のタイプの集積回路(IC)、及び/又は、ステートマシンが含まれる。このようなプロセッサは、処理されたハードウェア記述言語(HDL)命令(このような命令は、コンピュータ可読媒体に記憶することが可能である)の結果及びネットリストを含む他の中間データを用いて製造プロセスを構成することによって製造され得る。このようなプロセスの結果は、実施形態の態様を実施するプロセッサを製造するために半導体製造プロセスにおいて使用されるマスクワークであってもよい。

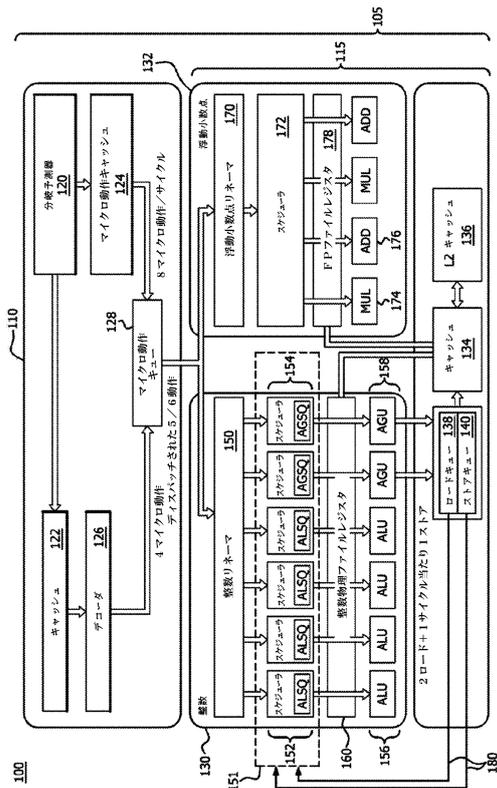
10

【0040】

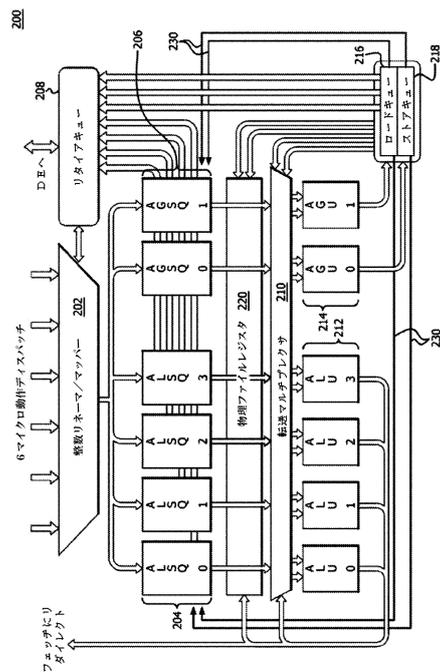
本明細書で提供された方法又はフローチャートは、汎用コンピュータ又はプロセッサによる実行のために非一時的なコンピュータ可読記憶媒体に組み込まれたコンピュータプログラム、ソフトウェア又はファームウェアで実施されてもよい。非一時的なコンピュータ可読記憶媒体の例には、例えば読み出し専用メモリ(ROM)、ランダムアクセスメモリ(RAM)、レジスタ、キャッシュメモリ、半導体メモリデバイス、内蔵ハードディスク、リムーバブルディスク等の磁気媒体、光磁気記憶媒体、例えばCD-ROMディスク及びデジタル多用途ディスク(DVD)等の光学媒体が含まれる。

20

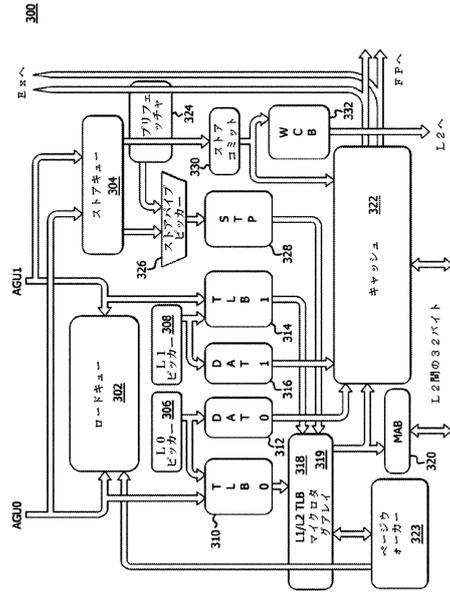
【図1】



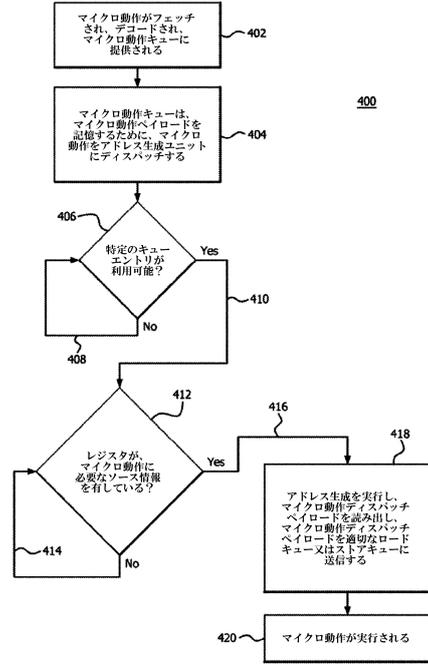
【図2】



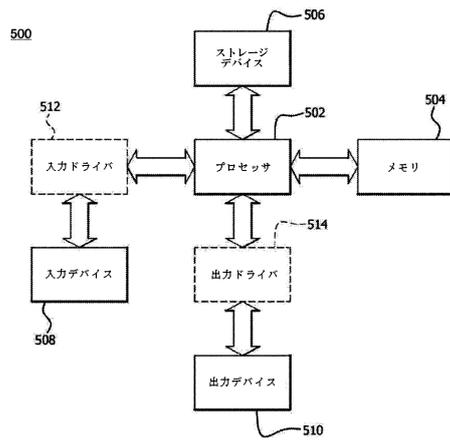
【図3】



【図4】



【図5】



---

フロントページの続き

(72)発明者 ジョン エム． キング

アメリカ合衆国 78735 テキサス州、オースティン、サウスウエスト パークウェイ 71  
71

審査官 田中 幸雄

(56)参考文献 米国特許出願公開第2014/0351562(US, A1)

米国特許出願公開第2003/0182537(US, A1)

国際公開第2016/113105(WO, A1)

米国特許第6481251(US, B1)

米国特許出願公開第2012/0278685(US, A1)

米国特許出願公開第2015/0095618(US, A1)

米国特許出願公開第2015/0205605(US, A1)

(58)調査した分野(Int.Cl., DB名)

G06F 9/38

G06F 9/312

G06F 9/34