(19)日本国特許庁(JP)　(12)公開特許公報(A)　(11)特許出願公開番号
特開2007-200288
(P2007-200288A)
(43)公開日　平成19年8月9日(2007.8.9)

(51)Int.Cl.	F I		テーマコード（参考）
G06F 9/38 (2006.01)	G06F	9/38　370X	5B013

審査請求 有　請求項の数 10　ＯＬ　外国語出願　（全 12 頁）

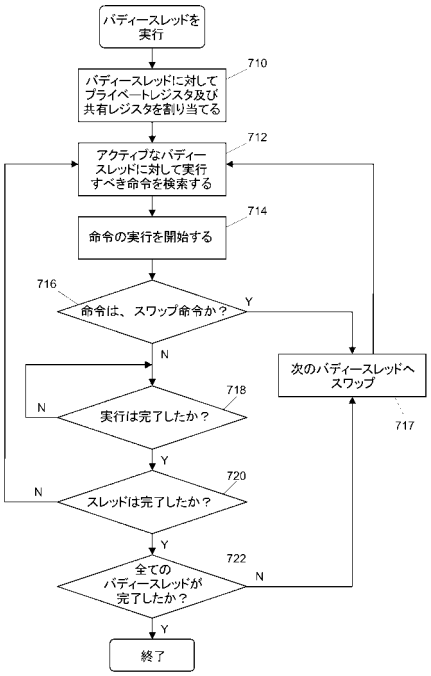
(21)出願番号　特願2006-338917 (P2006-338917)
(22)出願日　平成18年12月15日 (2006.12.15)
(31)優先権主張番号　11/305558
(32)優先日　平成17年12月16日 (2005.12.16)
(33)優先権主張国　米国 (US)

(71)出願人　501261300
エヌヴィディア　コーポレイション
アメリカ合衆国，カリフォルニア　９５０５０，サンタ　クララ，サン　トーマス　エクスプレスウェイ　２７０１
(74)代理人　100094318
弁理士　山田　行一
(74)代理人　100123995
弁理士　野田　雅一
(72)発明者　ブレット　ダブリュー.　クーン
アメリカ合衆国，　カリフォルニア州，サン　ホゼ，　ニューゲート　コート　５８０３

最終頁に続く

(54)【発明の名称】実行スレッドをグループ化するためのシステム及び方法

(57)【要約】　（修正有）
【課題】実行ハードウェアがより効率的に利用されるように実行スレッドをグループ化する方法を提供する。
【解決手段】　複数のスレッドが、二つ以上のスレッドをもつバディーグループに分割され、各スレッドには一以上のバディースレッドが割り当てられる。各バディーグループのうちの一つのスレッドだけが命令をアクティブに実行し、これによって、レジスタのようなハードウェアリソースをバディースレッドが共有することを可能にする。アクティブなスレッドがスワップ命令のようなスワップイベントに遭遇すると、アクティブなスレッドが実行を保留し、そのバディースレッドのうちの一つのスレッドが、当該スレッドのプライベートハードウェアリソース及びバディーグループの共有ハードウェアリソースを使用して実行を開始する。その結果、スレッドごとのハードウェアリソースの全てを複製せずにスレッド数を増加することができる。
【選択図】図７



【特許請求の範囲】
【請求項１】
処理ユニットで複数のスレッドの命令を実行する方法であって、
前記処理ユニットのハードウェアリソースの第１のセット、第２のセット、及び共有のセットを、第１のスレッドの命令及び第２のスレッドの命令に割り当てるステップと、
前記ハードウェアリソースの第１のセット及び共有セットを使用して、所定のイベントが発生するまで、前記第１のスレッドの命令を実行するステップと、
前記所定のイベントの発生に応答して、前記第１のスレッドの命令の実行を保留すると共に、前記ハードウェアリソースの第２のセット及び共有セットを使用して、前記第２のスレッドの命令を実行するステップと、
を備えた方法。
【請求項２】
前記第２のスレッドの命令は、別の所定のイベントが発生するまで実行され、該別の所定のイベントの発生に応答して、前記第２のスレッドの実行を保留し、前記命令の第１のスレッドの実行を再開する、請求項１に記載の方法。
【請求項３】
前記第１のスレッドの命令がスワップ命令を含み、前記第１のスレッドにおける該スワップ命令が実行されたときに前記所定のイベントが発生し、前記第２のスレッドの命令がスワップ命令を含み、前記第２のスレッドの該スワップ命令が実行されたときに前記別の所定のイベントが発生する、請求項２に記載の方法。
【請求項４】
ハードウェアリソースの第３のセット及びハードウェアリソースの前記共有セットを第３のスレッドの命令に割り当てるステップを更に備え、前記第２スレッドの命令が、別の所定のイベントが発生するまで実行され、該別の所定のイベントの発生に応答して、前記第２のスレッドの命令の実行を保留すると共に、前記第３のスレッドの命令を実行する、請求項１に記載の方法。
【請求項５】
前記所定のイベントは、前記第１のスレッドの命令のうち待ち時間の長い命令が実行されたときに発生する、請求項１に記載の方法。
【請求項６】
前記待ち時間の長い命令はメモリアクセス命令を含む、請求項５に記載の方法。
【請求項７】
前記ハードウェアリソースはレジスタを含む、請求項１に記載の方法。
【請求項８】
前記ハードウェアリソースは命令バッファを更に含む、請求項７に記載の方法。
【請求項９】
前記処理ユニットのハードウェアリソースの第３のセット、第４のセット、及び第５のセットを、第３のスレッドの命令及び第４のスレッドの命令に割り当てるステップと、
前記ハードウェアリソースの第３のセット及び第５のセットを使用して、前記第３のスレッドに対するスワップイベントが発生するまで、前記第３のスレッドの命令を実行するステップと、
前記３のスレッドに対する前記スワップイベントの発生に応答して、前記第３のスレッドの命令の実行を保留し、前記ハードウェアリソースの前記第４のセット及び前記第５のセットを使用して、前記第４のスレッドの命令を実行するステップと、
を更に備える請求項１に記載の方法。
【請求項１０】
前記第４のスレッドの命令は、該第４のスレッドに対するスワップイベントが発生するまで実行され、該第４のスレッドに対する前記スワップイベントの発生に応答して、前記第４のスレッドの命令の実行を保留し、前記第３のスレッドの命令の実行を再開する、請求項９に記載の方法。

【発明の詳細な説明】
【発明の分野】
【０００１】
[0001]本発明の実施の形態は、広くマルチスレッド処理に係り、より詳細には、改良されたハードウェアの利用を達成するために実行スレッドをグループ化するシステム及び方法に係る。
【関連技術の説明】
【０００２】
[0002]一般に、コンピュータの命令は、実行のために複数のクロックサイクルを必要とする。このために、マルチスレッドプロセッサは、命令の並列スレッドを連続的に実行して、命令を実行するためのハードウェアをできるだけビジー状態に保持することを可能にする。例えば、以下に示す特性を有する命令のスレッドを実行する場合には、マルチスレッドプロセッサは、四つの並列のスレッドを連続的にスケジュールすることができる。このようにスレッドをスケジュールすることによって、マルチスレッドプロセッサは、四つのスレッドの実行を、２３個のクロックサイクルの後に完了することができる。ここで、第１スレッドは、クロックサイクル１－２０の間に実行され、第２スレッドは、クロックサイクル２－２１の間に実行され、第３スレッドは、クロックサイクル３－２２の間に実行され、更に、第４スレッドは、クロックサイクル４－２３の間に実行される。これに比して、プロセッサが、プロセス中のスレッドが実行を完了するまでスレッドをスケジュールしない場合には、四つのスレッドの実行を完了するのに８０個のクロックサイクルを行うことになる。ここでは、第１スレッドがクロックサイクル１－２０の間に実行され、第２スレッドがクロックサイクル２１－４０の間に実行され、第３スレッドがクロックサイクル４１－６０の間に実行され、第４スレッドがクロックサイクル６１－８０の間に実行される。

命令	待ち時間	必要なリソース
1	４クロックサイクル	３個のレジスタ
2	４クロックサイクル	４個のレジスタ
3	４クロックサイクル	３個のレジスタ
4	４クロックサイクル	５個のレジスタ
5	４クロックサイクル	３個のレジスタ

【０００３】
[0003]しかしながら、上述した並列処理は、多量のハードウェアリソース、例えば、多数のレジスタを必要とする。上述した例では、並列処理に要するレジスタの数が、非並列処理の場合の５に比して、２０となる。
【０００４】
[0004]多くの場合には、実行の待ち時間(latency)が均一でない。例えば、グラフィック処理のケースでは、命令のスレッドが、通常、１０クロックサイクル未満の待ち時間をしばしば有する数学(math)オペレーションと、１００クロックサイクル以上の待ち時間を有するメモリアクセスオペレーションとを含む。このようなケースでは、並列スレッドの実行を連続的にスケジュールしても、あまりうまく機能しない。連続的に実行される並列スレッドの数が少な過ぎる場合には、メモリアクセスオペレーションの待ち時間が長くなる結果として、実行ハードウェアの多くが過少利用となる。他方、連続的に実行される並列スレッドの数が、メモリアクセスオペレーションの長い待ち時間をカバーするに充分なほど多くされた場合には、実行中のスレッド（live thread）をサポートするに要するレジスタの数が著しく増加する。
【発明の概要】
【０００５】
[0005]本発明は、実行ハードウェアがより効率的に利用されるように実行スレッドをグ

ループ化する方法を提供する。また、本発明は、実行ハードウェアがより効率的に利用されるように実行スレッドをグループ化するよう構成されたメモリユニットを備えるコンピュータシステムも提供する。

【０００６】

[0006]本発明の一実施の形態によれば、複数のスレッドが、二つ以上のスレッドのバディー(buddy：仲間)グループに分割され、各スレッドには一以上のバディースレッドが割り当てられる。各バディーグループの一つのスレッドだけが命令をアクティブに実行する。アクティブなスレッドが、スワップ命令のようなスワップイベントに遭遇すると、アクティブなスレッドは、実行を保留し、そのバディースレッドのうちの一つが実行を開始する。

【０００７】

[0007]スワップ命令は、通常、待ち時間の長い命令の後に現われ、現在アクティブなスレッドを、アクティブな実行リストにおけるそのバディースレッドのうちの一つとスワップさせる。バディースレッドの実行は、当該バディースレッドがスワップ命令に遭遇するまで続き、この遭遇がバディースレッドをアクティブな実行リストにおけるそのバディースレッドのうちの一つとスワップさせる。グループに二つのバディースレッドしかない場合には、そのバディースレッドがアクティブな実行リストにおけるオリジナルスレッドとスワップされ、オリジナルスレッドの実行が再開する。グループにバディースレッドが三つ以上ある場合には、そのバディースレッドは、ある所定の順序に基づきグループにける次のバディースレッドとスワップされる。

【０００８】

[0008]レジスタファイルの使用を節約するために、各バディースレッドは、そのレジスタ割り当てを、プライベート及び共有の二つのグループに分割している。プライベートグループに属するレジスタだけがスワップが生じた場合でも値を保持する。共有レジスタは、常に、バディーグループの現在のアクティブなスレッドにより所有される。

【０００９】

[0009]バディーグループは、プログラムが実行のためにロードされるときにスレッドが設定されるテーブルを使用して編成される。このテーブルは、オンチップレジスタに維持されてもよい。このテーブルは、複数の行を有し、各バディーグループ内のスレッドの数に基づいて構成される。例えば、各バディーグループに二つのスレッドがある場合には、テーブルが二つの列で構成される。各バディーグループに三つのスレッドがある場合には、テーブルが三つの列で構成される。

【００１０】

[0010]コンピュータシステムは、本発明の一実施の形態によれば、上述したテーブルをメモリに記憶し、更に、第１及び第２の実行パイプラインを用いて構成された処理ユニットを備えている。第１の実行パイプラインは数学オペレーションを実行するために使用され、第２の実行パイプラインはメモリオペレーションを実行するために使用される。

【００１１】

[0011]本発明の上述の特徴を詳細に理解できるように、上に要約した本発明を、実施の形態を参照して詳細に説明する。実施の形態のうち幾つかについては、添付図面に示す。添付図面は、本発明の典型的な実施形態を示すに過ぎず、それ故、本発明の範囲を限定するものではない。これは、本発明が、他の同様に有効な実施の形態にも通じるものであるからである。

【詳細な説明】

【００１２】

[0019]図１は、本発明を実施し得る複数の処理ユニットを有するグラフィック処理ユニット（ＧＰＵ）１２０を実装したコンピュータシステム１００の簡単なブロック図である。ＧＰＵ１２０は、複数の処理ユニット１２４－１、１２４－２、・・・１２４－Ｎに結合されたインタフェイスユニット１２２を備えている。ここで、Ｎは、１より大きな整数である。処理ユニット１２４は、メモリコントローラ１２６を介してローカルグラフィッ

クメモリ１３０へアクセスすることができる。ＧＰＵ１２０及びローカルグラフィックメモリ１３０は、システムメモリ１１２に記憶されたドライバを使用してコンピュータシステム１００の中央処理ユニット（ＣＰＵ）１１０によりアクセスされるグラフィックサブシステムである。

【００１３】

[0020]図２は、処理ユニット１２４の一つを更に詳細に示す。図２に示す処理ユニットは、本明細書では参照符号２００によって参照されており、図１に示す処理ユニット１２４のうち任意の一つを表わしている。処理ユニット２００は、処理ユニット２００によって実行されるべき命令を発行するための命令ディスパッチユニット２１２と、命令の実行に使用されるオペランドを記憶するレジスタファイル２１４と、一対の実行パイプライン２２２及び２２４と、を備えている。第１の実行パイプライン２２２は、数学オペレーションを実行するように構成されており、第２の実行パイプライン２２４は、メモリアクセスオペレーションを実行するように構成されている。一般的に、第２の実行パイプライン２２４で実行される命令の待ち時間は、第１の実行パイプライン２２２で実行される命令の待ち時間よりかなり長い。命令ディスパッチユニット２１２が命令を発行するときには、命令ディスパッチユニット２１２は、二つの実行パイプライン２２２及び２２４の一方にパイプラインコンフィギュレーション信号を送信する。命令が数学形式である場合には、パイプラインコンフィギュレーション信号は、第１の実行パイプライン２２２へ送信される。命令がメモリアクセス形式である場合には、パイプラインコンフィギュレーション信号は、第２の実行パイプライン２２４へ送信される。二つの実行パイプライン２２２及び２２４の実行結果は、レジスタファイル２１４へ書き戻される。

【００１４】

[0021]図３は、命令ディスパッチユニット２１２の機能ブロック図である。命令ディスパッチユニット２１２は、複数のスロットを有する命令バッファ３１０を備えている。この実施の形態におけるスロットの数は、１２であり、各スロットは、２個までの命令を保持することができる。スロットのうち何れか一つが別の命令のためのスペースを有する場合には、フェッチ３１２が、スレッドプール３０５から命令キャッシュ３１４へなされる。スレッドプール３０５には、プログラムが実行のためにロードされるときに、スレッドが設定される。命令キャッシュ３１４に記憶された命令が、現在実行中の命令、即ち発行されたが完了されておらず且つ命令バッファ３１０の空きスペースに置かれている命令を追跡するスコアボード３２２に追加される前に、命令はデコード３１６される。

【００１５】

[0022]命令ディスパッチユニット２１２は、更に、発行(issue)ロジック３２０も備えている。この発行ロジック３２０は、スコアボード３２２を検査し、そして実行中の何れの命令にも依存しない命令を、命令バッファ３１０から発行する。命令バッファ３１０からの発行と共に、発行ロジック３２０は、パイプラインコンフィギュレーション信号を適切な実行パイプラインへ送信する。

【００１６】

[0023]図４は、本発明の第１の実施の形態に係るスレッドプール３０５の構成を示す。スレッドプール３０５は、１２行２列のテーブルとして構成される。テーブルの各セルは、スレッドを記憶するメモリスロットを表わす。テーブルの各行は、バディーグループを表わす。従って、テーブルのセル０Ａのスレッドは、テーブルのセル０Ｂのスレッドのバディースレッドである。本発明の実施の形態によれば、バディーグループのうちの一つのスレッドのみが、一度にアクティブとなる。命令フェッチの間に、アクティブなスレッドからの命令がフェッチされる。フェッチされた命令は、その後、デコードされ、命令バッファ３１０の対応スロットに記憶される。本明細書に示す本発明の実施の形態では、スレッドプール３０５のセル０Ａ又はセル０Ｂの何れかからフェッチされた命令は、命令バッファ３１０のスロット０に記憶され、スレッドプール３０５のセル１Ａ又はセル１Ｂの何れかからフェッチされた命令は、命令バッファ３１０のスロット１に記憶され、等々となる。また、命令バッファ３１０に記憶された命令は、発行ロジック３２０に従って連続す

るクロックサイクルで発行される。図６に示す簡単な例では、命令バッファ３１０に記憶された命令は、行０の命令、次いで、行１の命令、等々で始まる連続するクロックサイクルで発行される。

【００１７】

[0024]図５は、本発明の第２の実施の形態に係るスレッドプール３０５の構成を示す。スレッドプール３０５は、８行３列のテーブルとして構成される。テーブルの各セルは、スレッドを記憶するメモリスロットを表わす。テーブルの各行は、バディーグループを表わす。従って、テーブルのセル０Ａ、０Ｂ及び０Ｃのスレッドは、バディーススレッドと考えられる。本発明の実施の形態によれば、バディーグループのうちの一つのスレッドのみが、一度にアクティブとなる。命令フェッチの間に、アクティブなスレッドからの命令がフェッチされる。フェッチされた命令は、その後、デコードされ、命令バッファ３１０の対応のスロットに記憶される。本明細書に示す本発明の実施の形態では、スレッドプール３０５のセル０Ａ、セル０Ｂ又はセル０Ｃからフェッチされた命令が命令バッファ３１０のスロット０に記憶され、スレッドプール３０５のセル１Ａ、セル１Ｂ又はセル１Ｃの何れかからフェッチされた命令が命令バッファ３１０のスロット１に記憶され、等々となる。また、命令バッファ３１０に記憶された命令は、発行ロジック３２０に従って連続するクロックサイクルで発行される。

【００１８】

[0025]スレッドプール３０５にスレッドが設定されるときには、当該スレッドプール３０５は列順(column major order)にロードされる。セル０Ａが最初にロードされ、その後、セル１Ａ、セル２Ａ等々と続き、セルＡが満たされるまでロードされる。次いで、セル０Ｂがロードされ、その後、セル１Ｂ、セル２Ｂ等々と続き、セルＢが満たされるまでロードされる。スレッドプール３０５が追加の列をもって構成される場合には、このスレッドロードプロセスは、全ての列が満たされるまで同様に続けられる。スレッドプール３０５を列順にロードすることにより、バディースレッドを、一時的に、互いに可能な限り分離することができる。また、バディースレッドの各行は、他の行とは全く独立しており、命令バッファ３１０から命令が発行されるときに、行間の順序は発行ロジック３２０によって最小限に強制される。

【００１９】

[0026]図６は、グループ当たり二つのバディースレッドがある場合のアクティブな実行スレッドのスワップを示すタイミングチャートである。実線の矢印は、アクティブなスレッドに対して実行される命令のシーケンスに対応する。このタイミング図は、スレッドプール３０５におけるセル０Ａのスレッドが最初に開始され、そのスレッドからスワップ命令が発行されるまで当該スレッドからの命令のシーケンスが実行されることを示している。スワップ命令が発行されると、スレッドプール３０５のセル０Ａのスレッドがスリープ状態に入り（即ち、インアクティブにされ）、そのバディースレッド、即ちスレッドプール３０５のセル０Ｂのスレッドがアクティブにされる。その後、スレッドプール３０５のセル０Ｂのスレッドからの命令のシーケンスが、そのスレッドからスワップ命令が発行されるまで、実行される。このスワップ命令が発行されると、スレッドプール３０５のセル０Ｂのスレッドがスリープ状態に入り、そのバディースレッド、即ちスレッドプール３０５のセル０Ａのスレッドがアクティブにされる。これは、両スレッドがそれらの実行を完了するまで続けられる。バディースレッドへのスワップは、スレッドが実行を完了したがそのスレッドのバディースレッドが完了しないときにも行われる。

【００２０】

[0027]図６に示すように、スレッドプール３０５の他のアクティブなスレッドは、セル０Ａのスレッドの後に連続して開始される。セル０Ａのスレッドと同様に、他のアクティブなスレッドの各々も、そのスレッドからスワップ命令が発行されるまで実行され、スワップ命令が発行されたときに、当該スレッドはスリープ状態に入り、そのスレッドのバディースレッドがアクティブにされる。次いで、アクティブな実行が、バディースレッド間で、両スレッドがそれらの実行を完了するまで、交互に行われる。

【００２１】

[0028]図７は、バディーグループのスレッド（又は手短に言えば、バディースレッド）を実行するときに処理ユニットにより実行されるプロセスの各ステップを示すフローチャートである。ステップ７１０において、バディースレッドに対するハードウェアリソース、特に、レジスタが割り当てられる。割り当てられるレジスタは、バディースレッドの各々に対するプライベートレジスタ、及びバディースレッドにより共有されるべき共有レジスタを含む。共有レジスタの割り当ては、レジスタの使用を節約する。例えば、二つのバディースレッドがあり、且つ、バディースレッドの各々により２４個のレジスタが必要とされる場合には、従来のマルチ処理方法を実行するには、合計４８個のレジスタが必要になる。しかしながら、本発明の実施の形態では、共有レジスタが割り当てられる。これらのレジスタは、スレッドがアクティブであるときには必要であるが、スレッドが非アクティブであるとき、例えば、スレッドが待ち時間の長いオペレーションの完了を待機しているときには必要とされないレジスタに対応する。プライベートレジスタは、スワップとスワップとの間に保存する必要のある情報を記憶するために割り当てられる。二つのバディースレッドの各々により２４個のレジスタが必要とされる実施例では、これらレジスタのうち１６個を共有レジスタとして割り当てることができる場合に、両バディースレッドを実行するのに合計３２個のレジスタしか必要とされない。バディーグループ当たり三つのバディースレッドがある場合には、節減が更に大きくなる。この実施例において、本発明では合計４０個のレジスタが必要となるのに比して、従来のマルチ処理方法では合計７２個のレジスタが必要になる。

【００２２】

[0029]バディースレッドのうち一つが、アクティブなスレッドとしてスタートし、このスレッドからの命令が実行のために取り出される（ステップ７１２）。ステップ７１４では、ステップ７１２で取り出された命令の実行が開始される。次いで、ステップ７１６において、その取り出された命令を検査して、スワップ命令であるかどうか調べる。スワップ命令である場合には、現在アクティブなスレッドが非アクティブにされ、バディーグループにおける他のスレッドのうちの一つがアクティブにされる（ステップ７１７）。スワップ命令でない場合には、ステップ７１４で開始された実行が完了しているか否かについて調べられる（ステップ７１８）。この実行が完了すると、現在アクティブなスレッドを検査して、実行されるべき命令が残っているかどうか調べる（ステップ７２０）。もし残っていれば、プロセスの流れがステップ７１２へ戻り、実行されるべき次の命令が現在アクティブなスレッドから取り出される。そうでなければ、全てのバディースレッドが実行を完了したか否かを調べるためにチェックがなされる（ステップ７２２）。完了した場合には、プロセスは終了となる。完了しない場合には、プロセスの流れがステップ７１７へ戻り、完了していないバディースレッドへのスワップが行われる。

【００２３】

[0030]上述した本発明の実施の形態では、プログラムがコンパイルされるときにスワップ命令が挿入される。スワップ命令は、通常、待ち時間の長い命令の直後に挿入され、好ましくは、プライベートレジスタの数に比して多数の共有レジスタを割り当てできるプログラム内の各ポイントにおいて挿入される。例えば、グラフィック処理では、スワップ命令がテクスチャ命令の直後に挿入される。本発明の別の実施の形態では、スワップイベントがスワップ命令でなく、ハードウェアが認識する何らかのイベントであってもよい。例えば、ハードウェアは、命令の実行において長い待ち時間を認識するように構成されていることがある。これを認識すると、長い待ち時間を生じさせる命令を発行したスレッドをインアクティブ状態に至らせ、同じバディーグループの別のスレッドをアクティブにさせることができる。また、スワップイベントは、長い待ち時間のオペレーション中の何らかの認識可能なイベント、例えば、長い待ち時間のオペレーション中に生じる第１のスコアボードの停止（ストール）であってもよい。

【００２４】

[0031]以下の命令シーケンスは、スワップ命令がコンパイラーにより挿入され得るシェ

ーダープログラムの箇所を例示するものである。

```
Inst_00:      Interpolate    iw
Inst_01:      Reciprocal      w
Inst_02:      Interpolate    s, w
Inst_03:      Interpolate    t, w
Inst_04:      Texture        s, t        //Texturereturns r, g, b, a values
Inst_05:      Swap
Inst_06:      Multiply      r, r, w
Inst_07:      Multiply      g, g, w
```

スワップ命令（ｉｎｓｔ＿０５）は、コンパイラーにより待ち時間の長いテクスチャ命令（ｉｎｓｔ＿０４）の直後に挿入される。このように、バディースレッドへのスワップは、待ち時間の長いテクスチャ命令（ｉｎｓｔ＿０４）が実行される間に行うことができる。スワップ命令を乗算命令（ｉｎｓｔ＿０６）の後に挿入するのは、あまり望ましくない。これは、乗算命令（ｉｎｓｔ＿０６）が、テクスチャ命令（Ｉｎｓｔ＿０４）の結果に依存し、バディースレッドへのスワップを、待ち時間の長いテクスチャ命令（Ｉｎｓｔ＿０４）がその実行を完了する後まで行えないからである。

【００２５】

[0032]例示を簡単化するために、本発明の実施の形態の上述の説明で使用したスレッドは、単一スレッドの命令としている。しかしながら、本発明は、同様のスレッドが共にグループ化され、コンボイ(convoy)とも称されるこのグループからの同じ命令が、単一命令マルチデータ（ＳＩＭＤ）プロセッサを使用して複数の並列データパスを介して処理されるような実施形態にも適用することができる。

【００２６】

[0033]以上、本発明の実施の形態を説明したが、本発明の基本的な範囲から逸脱せずに、他の及び更に別の実施形態を案出することも可能である。本発明の範囲は、特許請求の範囲により決定される。

【図面の簡単な説明】

【００２７】

【図１】本発明を実施し得る複数の処理ユニットを有するＧＰＵを実装したコンピュータシステムの簡単なブロック図である。

【図２】図１の処理ユニットを更に詳細に示す図である。

【図３】図２に示す命令ディスパッチユニットの機能ブロック図である。

【図４】本発明の第１の実施の形態によるスレッドプール及び命令バッファを示す概念図である。

【図５】本発明の第２の実施の形態によるスレッドプール及び命令バッファを示す概念図である。

【図６】バディースレッド間でのアクティブな実行スレッドのスワップを示すタイミングチャートである。

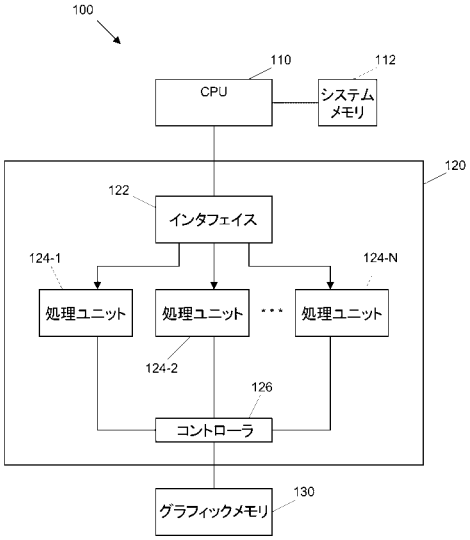
【図７】バディースレッドを実行するときに処理ユニットによって実行されるプロセスステップを示すフローチャートである。

【符号の説明】

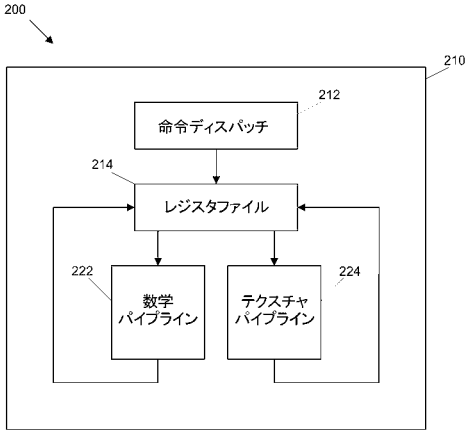
【００２８】

１００…コンピュータシステム、１１０…中央処理ユニット(ＣＰＵ)、１１２…システムメモリ、１２０…グラフィック処理ユニット(ＧＰＵ)、１２２…インタフェイスユニット、１２４…処理ユニット、１２６…メモリコントローラ、１３０…ローカルグラフィックメモリ、２００…処理ユニット、２１２…命令ディスパッチユニット、２１４…レジスタファイル、２２２，２２４…実行パイプライン、３０５…スレッドプール、３１０…命令バッファ、３１４…命令キャッシュ、３２０…発行ロジック、３２２…スコアボード。

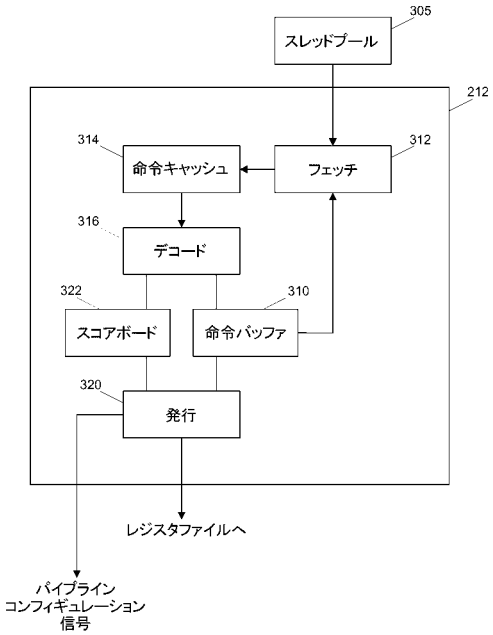
【図１】


100
110
CPU
112
システム
メモリ
120
122
インタフェイス
124-1
124-N
処理ユニット
処理ユニット
処理ユニット
124-2
126
コントローラ
グラフィックメモリ
130

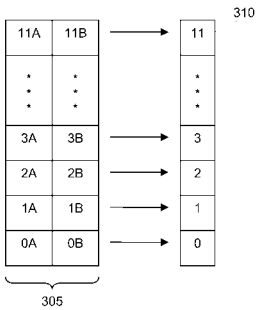
【図２】


200
210
212
命令ディスパッチ
214
レジスタファイル
222
数学
パイプライン
224
テクスチャ
パイプライン

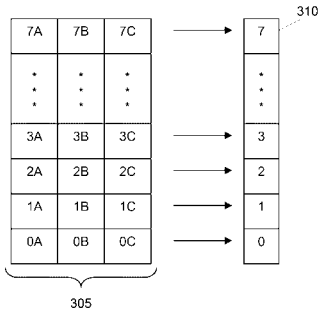
【図３】


305
スレッドプール
212
314
命令キャッシュ
312
フェッチ
316
デコード
322
スコアボード
310
命令バッファ
320
発行
レジスタファイルへ
パイプライン
コンフィギュレーション
信号

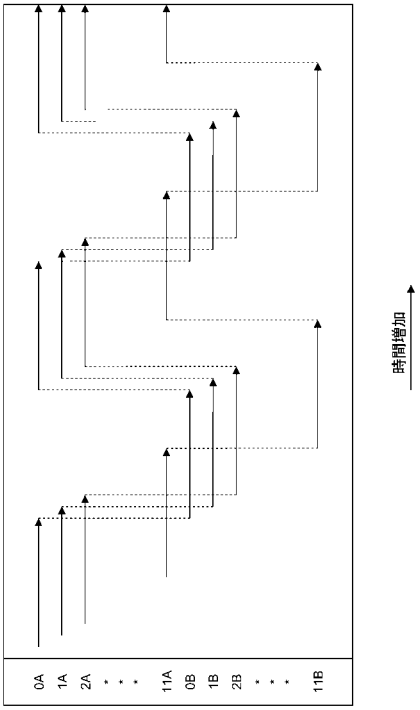
【図４】


310
11A
11B
11
3A
3B
3
2A
2B
2
1A
1B
1
0A
0B
0
305

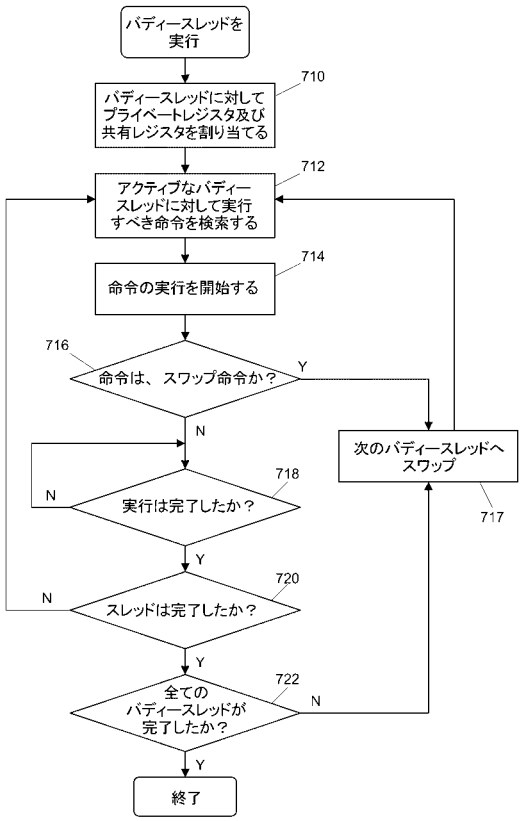
【図５】


7A
7B
7C
3A
3B
3C
2A
2B
2C
1A
1B
1C
0A
0B
0C
305
310
7
3
2
1
0

【図６】


0A
1A
2A
11A
0B
1B
2B
11B
時間増加

【図７】


バディースレッドを実行
バディースレッドに対してプライベートレジスタ及び共有レジスタを割り当てる
710
アクティブなバディースレッドに対して実行すべき命令を検索する
712
命令の実行を開始する
714
716
命令は、スワップ命令か？
Y
N
次のバディースレッドへスワップ
717
718
実行は完了したか？
N
Y
720
スレッドは完了したか？
N
Y
722
全てのバディースレッドが完了したか？
N
Y
終了

フロントページの続き

(72)発明者　ジョン　エリック　リンドホルム
アメリカ合衆国,　カリフォルニア州,　サラトガ,　ライス　コート　２０６８２

Ｆターム(参考) 5B013 DD04

【外国語明細書】
2007200288000001.pdf