

(19) 日本国特許庁(JP)

## (12) 特許公報(B2)

(11) 特許番号

特許第4932904号  
(P4932904)

(45) 発行日 平成24年5月16日(2012.5.16)

(24) 登録日 平成24年2月24日(2012.2.24)

(51) Int.Cl.

F 1

G06F 9/455 (2006.01)  
G06F 11/28 (2006.01)G06F 9/44 310A  
G06F 11/28 340A

請求項の数 20 (全 15 頁)

(21) 出願番号 特願2009-509939 (P2009-509939)  
 (86) (22) 出願日 平成19年4月20日 (2007.4.20)  
 (65) 公表番号 特表2009-535746 (P2009-535746A)  
 (43) 公表日 平成21年10月1日 (2009.10.1)  
 (86) 國際出願番号 PCT/US2007/067146  
 (87) 國際公開番号 WO2007/130807  
 (87) 國際公開日 平成19年11月15日 (2007.11.15)  
 審査請求日 平成21年2月12日 (2009.2.12)  
 (31) 優先権主張番号 60/746,267  
 (32) 優先日 平成18年5月3日 (2006.5.3)  
 (33) 優先権主張国 米国(US)  
 (31) 優先権主張番号 60/746,268  
 (32) 優先日 平成18年5月3日 (2006.5.3)  
 (33) 優先権主張国 米国(US)

(73) 特許権者 310021766  
 株式会社ソニー・コンピュータエンタテインメント  
 東京都港区港南1丁目7番1号  
 (74) 代理人 100105924  
 弁理士 森下 賢樹  
 (72) 発明者 スチュアート サーゲイソン  
 アメリカ合衆国、カリフォルニア州 94404、フォスター・シティー、セカンド・フロア、イースト・ヒルズデイル・ブルバード 919

最終頁に続く

(54) 【発明の名称】 インタープリトおよびトランスレートされたインストラクションの両方に関わるエミュレーションにおけるクロック管理の問題を解決するための方法と装置

## (57) 【特許請求の範囲】

## 【請求項 1】

ターゲットシステムをエミュレーションするためのホストシステムであって、  
1以上のプロセッサと、

前記1以上のホストシステムプロセッサに接続されたメモリと、

前記メモリに具体化されたプロセッサで実行可能なインストラクションセットとを含み

当該プロセッサで実行可能なインストラクションは、ホストシステム上でのターゲットシステムのエミュレーションにおけるクロック管理の問題を解決するための方法を実装するためのインストラクションを含み、

当該方法は、

ターゲットプログラムの第1コードインストラクションセットをインタープリトして前記ホストシステム上で前記ターゲットシステムの第1コンポーネントをエミュレートするインタープリトされたコードインストラクションを生成するステップと、

第2コードインストラクションセットをトランスレートして前記ホストシステム上で前記ターゲットシステムの第2コンポーネントをエミュレートするトランスレートされたコードインストラクションを生成するステップと、

インタープリトされたインストラクションを第1クロックにもとづいて実行するステップと、

トランスレートされたインストラクションを第2クロックにもとづいて実行するステップと、

10

トランスレートされたインストラクションを第2クロックにもとづいて実行するステップと、

20

と、

トランスレートされたインストラクションとインタープリトされたインストラクションの間で望ましい同期を維持するために、前記第1または第2クロック、前記トランスレートまたはインターパリトされたインストラクションの実行、またはメモリアクセスを調整するステップとを含むことを特徴とするホストシステム。

【請求項2】

前記第1クロックは固定クロック、前記第2クロックは可変クロックである請求項1のシステム。

【請求項3】

前記第1または第2クロック、前記トランスレートまたはインターパリトされたインストラクションの実行レート、またはメモリアクセスを調整するステップは、前記第2クロックが前記第1クロックよりも速く進んでいるとき、トランスレートされたインストラクションの実行またはメモリアクセスを停止するステップを含む請求項2のシステム。

10

【請求項4】

前記トランスレートされたインストラクションは、オペレーションにかかる時間を推定するためにエミュレートされたサイクルカウンタを用いる請求項2のシステム。

【請求項5】

前記トランスレートされたインストラクションの実行タイミングは前記可変クロックに合わせられ、前記インターパリトされたインストラクションの実行タイミングは前記固定クロックに合わせられる請求項4のシステム。

20

【請求項6】

前記第2クロックが前記第1クロックよりも速く進んでいるとき、トランスレートされたインストラクションの実行またはメモリアクセスを停止するステップは、あらかじめ定められた数のエミュレートされたサイクルを実行するステップと、前記可変クロックが前記固定クロックよりも速く進んでいるかどうかをチェックするステップと、前記可変クロックが前記固定クロックよりも速く進んでいる場合、前記可変クロックを停止するステップとを含む請求項4のシステム。

【請求項7】

前記ホストシステムは、パワー・プロセッサ・エレメントと前記パワー・プロセッサ・エレメントに接続された複数のシナジスティック・プロセッサ・エレメントとを有するセルプロセッサを含む請求項1のシステム。

30

【請求項8】

前記第1コンポーネントはエモーション・エンジンであり、前記第2コンポーネントは、前記エモーション・エンジンに接続された入出力プロセッサである請求項1のシステム。

【請求項9】

フレームをディスプレイデバイスに提示する前に、ホストデバイスがトランスレートされたインストラクションのオペレーションを完了するのに十分な時間がない状況において、トランスレートされたインストラクションとインターパリトされたインストラクションの間で望ましい同期を維持するために、前記第1または第2クロック、前記トランスレートまたはインターパリトされたインストラクションの実行、またはメモリアクセスを調整するステップは、前記トランスレートされたインストラクションのオペレーションを完了するためのさらなる時間を与えるために、現在のフレームを繰り返すステップを含む請求項1のシステム。

40

【請求項10】

トランスレートされたインストラクションとインターパリトされたインストラクションの間で望ましい同期を維持するために、前記第1または第2クロック、前記トランスレートまたはインターパリトされたインストラクションの実行、またはメモリアクセスを調整するステップは、

前記ターゲットプログラムの1以上の部分による前記第2コンポーネントの使用効率を

50

調べるステップと、その使用効率にもとづいて前記第2クロックのレートを調整するステップとを含む請求項1のシステム。

【請求項11】

トランスレートされたインストラクションとインタープリトされたインストラクションの間で望ましい同期を維持するために、前記第1または第2クロック、前記トランスレートまたはインターパリトされたインストラクションの実行、またはメモリアクセスを調整するステップは、

前記第1クロックのレートと前記第2クロックのレートの間の違いのために、エミュレートされたイベントがタイムアウトするかどうかを決定するステップと、

そのエミュレートされたイベントのタイムアウトを避けるために、前記第1クロックのレートまたは前記トランスレートされたインストラクションの実行を調整するステップとを含む請求項1のシステム。 10

【請求項12】

前記第1クロックのレートと前記第2クロックのレートの間の違いのために、エミュレートされたイベントがタイムアウトするかどうかを決定するステップは、前記ターゲットプログラムが制御レジスタに値を設定し、その後、結果を待つために繰り返しその制御レジスタをポーリングしたかどうかを検出するステップを含む請求項11のシステム。

【請求項13】

前記タイムアウトを避けるために、前記第1クロックのレートまたは前記トランスレートされたインストラクションの実行を調整するステップは、

前記ターゲットプログラムがあらかじめ定められた回数よりも多く前記レジスタをポーリングするならば、前記ターゲットプログラムのエミュレーションの実行を、タイミングアウトを避けるやり方に変更するステップを含む請求項12のシステム。 20

【請求項14】

前記ターゲットプログラムのエミュレーションの実行を変更するステップは、前記タイムアウトを防ぐために十分な余裕をもって前記エミュレーションの速度を落とすステップを含む請求項13のシステム。

【請求項15】

前記ターゲットプログラムのエミュレーションの実行を変更するステップは、ハードウェアステートが変化するまで前記エミュレートされたターゲットプログラムにどんなレスポンスも返さないステップを含む請求項13のシステム。 30

【請求項16】

前記ターゲットプログラムのエミュレーションの実行を変更するステップは、ポーリングオペレーションが起きているかどうかを判定するために前記ターゲットプログラムの関連する部分を分析するステップを含む請求項13のシステム。

【請求項17】

エミュレートされたハードウェアで前記タイムアウトに関係しない機能を実行する一方で、前記エミュレートされたイベントのタイムアウトを避けるために、前記第1クロックのレートまたは前記トランスレートされたインストラクションの実行を調整するステップをさらに含む請求項12のシステム。 40

【請求項18】

前記トランスレートされたインストラクションおよび/またはインターパリトされたインストラクションを実行するステップは、第1のエミュレートされたデバイスから1以上のライトインストラクションを書き出すステップと、第2のエミュレートされたデバイスで前記1以上のライトインストラクションを読み出すステップとを含み、

トランスレートされたインストラクションとインターパリトされたインストラクションの間で望ましい同期を維持するために、前記第1または第2クロック、前記トランスレートまたはインターパリトされたインストラクションの実行、またはメモリアクセスを調整するステップは、前記1以上のライトインストラクションを書き出された順番でジャーナルの形で格納するステップと、書き出された順番で前記第2のエミュレートされたデバイ 50

上で前記 1 以上のライトインストラクションを読み出すステップとを含む請求項 1 のシステム。

【請求項 19】

ホストシステム上でのターゲットシステムのエミュレーションにおけるクロック管理の問題を解決するための方法であって、

前記ホストシステム上で前記ターゲットシステムの第 1 コンポーネントをエミュレートするインタープリトされたコードインストラクションを生成するためにターゲットプログラムの第 1 コードインストラクションセットをインターパリトするステップと、

前記ホストシステム上で前記ターゲットシステムの第 2 コンポーネントをエミュレートするトランスレートされたコードインストラクションを生成するために第 2 コードインストラクションセットをトランスレートするステップと、  
10

インターパリトされたインストラクションを第 1 クロックにもとづいて実行するステップと、

トランスレートされたインストラクションを第 2 クロックにもとづいて実行するステップと、

トランスレートされたインストラクションとインターパリトされたインストラクションの間で望ましい同期を維持するために、前記第 1 または第 2 クロック、前記トランスレートまたはインターパリトされたインストラクションの実行、またはメモリアクセスを調整するステップとを含むことを特徴とする方法。  
20

【請求項 20】

前記ホストシステムは、パワー・プロセッサ・エレメントと前記パワー・プロセッサ・エレメントに接続された複数のシナジスティック・プロセッサ・エレメントとを有するセルプロセッサを含む請求項 19 の方法。

【発明の詳細な説明】

【技術分野】

【0001】

【優先権の主張】

本出願は、2007年4月4日に出願された、Stewart Sargaisonらの「インターパリトおよびトランスレートされたインストラクションの両方に関わるエミュレーションにおけるクロック管理の問題を解決するための方法と装置」と題する同一出願人による同時係属出願第11/696,699号（アトニー・ドケット番号SCEA 05055US01）の優先権の利益を主張し、その開示内容全体を参照によりここに援用する。  
30

【0002】

本出願は、2006年5月3日に出願された、「インターパリトされたコードとトランスレートされたコードの両方に関するエミュレーションにおけるクロック管理の問題を解決するための方法と装置」と題する米国仮出願第60/746,273号の優先権の利益を主張し、その開示内容全体を参照によりここに援用する。本出願は、2006年5月3日に出願された、Stewart Sargaisonらの「ホストシステム上でターゲットシステムをエミュレーションする際のトランスレーションブロックの無効の事前ヒント」と題する米国仮出願第60/746,267号の優先権の利益を主張し、その開示内容全体を参照によりここに援用する。本出願は、2006年5月3日に出願された、Stewart Sargaisonらの「ホストシステム上でターゲットシステムをエミュレーションする際のレジスタマッピング」と題する米国仮出願第60/746,268号の優先権の利益を主張し、その開示内容全体を参照によりここに援用する。本出願は、2006年5月3日に出願された、Victor Subaの「ストール予測スレッド管理」と題する米国仮出願第60/797,762号の優先権の利益を主張し、その開示内容全体を参照によりここに援用する。本出願は、2006年5月3日に出願された、Stewart Sargaisonらの「DMAおよびグラフィックスインターフェースエミュレーション」と題する米国仮出願第60/797,435号の優先権の利益を主張し、そ  
40  
50

の開示内容全体を参照によりここに援用する。本出願は、2006年5月3日に出願された、Stewart Sargaisonらの「コードトランスレーションおよびパイプライン最適化」と題する米国仮出願第60/797,761号の優先権の利益を主張し、その開示内容全体を参照によりここに援用する。

#### 【0003】

本出願は、2007年1月30日に出願された、米国特許出願第11/700,448号の優先権の利益を主張し、その開示内容全体を参照によりここに援用する。本出願は、2007年4月4日に出願された、Stewart Sargaisonらの「ホストシステム上でターゲットシステムをエミュレーションする際のトランスレーションブロックの無効の事前ヒント」と題する同一出願人による同時係属出願第11/696,684号(アトニー・ドケット番号SCEA05053US01)の優先権の利益を主張し、その開示内容全体を参照によりここに援用する。本出願は、2007年4月4日に出願された、Stewart Sargaisonらの「ホストシステム上のターゲットシステムのエミュレーションにおけるレジスタマッピング」と題する同一出願人による同時係属出願第11/696,691号(アトニー・ドケット番号SCEA05054US01)の優先権の利益を主張し、その開示内容全体を参照によりここに援用する。

10

#### 【0004】

##### [関連出願との相互参照]

本出願は、2006年1月30日に出願された米国仮特許出願第60/763,568号に関連し、その開示内容全体を参照によりここに援用する。

20

#### 【0005】

##### [技術分野]

本発明の実施の形態は、ホストコンピュータプラットフォームでのターゲットコンピュータプラットフォームのエミュレーションに関し、特にターゲットプラットフォームのあるコンポーネントがインタープリテーションによってエミュレートされ、他のコンポーネントがトランスレーションによってエミュレートされる場合のクロック管理に関する。

#### 【背景技術】

#### 【0006】

第1コンピュータプラットフォーム(「ターゲットシステム」)の機能を第2コンピュータプラットフォーム(「ホストシステム」)上でエミュレートすることにより、ホストシステムがターゲットシステム向けに設計されたプログラムを実行できるようにする処理は、「エミュレーション」として知られている。エミュレーションは一般に、ターゲットプラットフォーム用に設計されたプログラムインストラクション(ターゲットコードインストラクション)をホストプラットフォームのネイティブ言語(ホストインストラクション)に変換することによって互換性を達成するソフトウェアを生成することによって実現してきた。最近では、エミュレーションは「仮想マシン」を生成することによっても実現されている。仮想マシンでは、ターゲットプラットフォームの物理アーキテクチャ、すなわちハードウェア自身の設計が仮想モデルを介してソフトウェアで再現される。

30

#### 【0007】

現在、二つのエミュレーションストラテジーのタイプがエミュレーション分野で利用可能である。第1ストラテジーは「インターパリテーション」として知られるものである。インターパリテーションでは、各ターゲットコードインストラクションが呼び出された順にデコードされ、ターゲットコードインストラクションと意味的に等価であるホストインストラクションの小さなシーケンスが生成され、実行に移される。そのようなエミュレータの主要なコンポーネントは典型的にはソフトウェア・インターパリタである。ソフトウェア・インターパリタはターゲットマシン言語で書かれた任意のプログラムの各インストラクションをホストマシン言語のインストラクションセットに変換する。ここで、ホストマシン言語は、エミュレータが動作するホストコンピュータのコード言語である。インターパリタはコンピュータハードウェアまたはファームウェアで実装された例もある。これによれば、エミュレートされたプログラムを比較的速く実行することができる。

40

50

**【0008】**

他の主要なエミュレーションストラテジーは「トランスレーション」として知られる。トランスレーションでは、ターゲットインストラクションが分析され、デコードされる。これは「リコンパイル」または「クロスコンパイル」とも呼ばれる。コンピュータプログラムの実行速度はしばしばインタプリタによって劇的に減少してしまうことは周知のことである。コンピュータプログラムは、エミュレーションによって実行された場合、等価なプログラムがターゲットマシンコードにリコンパイルされ、ターゲットコードバージョンが実行される場合に比べて、10から20倍遅くなるのはよくあることである。ソフトウェアエミュレーションが遅いことはよく知られているため、数多くの製品は、ターゲットプログラムの一部をランタイムでホストマシンコードにトランスレートし、リコンパイルされたプログラム部分を実行することによって、ソースアプリケーションの実行速度を改善するのに成功してきた。<sup>10</sup>トランスレーション処理は、たとえば、ターゲットコードの1インストラクション毎に50～100マシン／クロックサイクルかかるが、その結果得られるホストマシンコードは、平均的には、たいていのソースアプリケーションの全体的な実行速度を改善するのに十分なくらい、ずっと速くなる。

**【発明の開示】****【発明が解決しようとする課題】****【0009】**

ターゲットコードがインタープリトされるか、トランスレートされるかを問わず、ターゲットマシンが元のターゲットインストラクションを実行するときとは異なるレートで、<sup>20</sup>ホストマシンは結果として得られるインタープリトまたはトランスレートされたインストラクションを実行する。その結果、ホストマシンは、エミュレートされるターゲットマシンよりも早くまたは遅く実行される。実行速度のそのような違いは、ワードプロセッサやスプレッドシートのようなプログラムでは、許容できるか、もしくは望ましいことさえある。しかし、実行速度のこののような違いは、次のようなタイムクリティカルなオペレーションにとっては重要な問題である。すなわち、(1)音やビデオの再生、(2)データが一定レートでプロセッサに伝達される「ストリーミング」情報の処理、および(3)動きを正確に表示するために画面の更新を要求するゲームやアニメーションである。

**【0010】**

ターゲットマシンとホストマシンの異なる実行レートは、たとえばホストマシンの実効レートを調整することにより、ソフトウェアで解決してもよい。たとえば、Lindenの米国特許6,882,962号は、ホストプラットフォーム内でタイミングの変化を測定、予測、および動的に調整することにより、一貫性のあるインストラクションの実行速度に対して設計されたターゲットプラットフォームのタイミング特性をシミュレートするための方法を記述している。この技術は、任意の「時間量子」を一つの指示対象として用いる。この指示対象は、指定された時間にターゲットシステムが実行するインストラクションの数を決定するためにターゲットシステムのインストラクションサイクルの実行速度がかけ算される。<sup>30</sup>

**【0011】**

ネイティブでないコードがホストシステム上で実行されるとき、カウンタを用いて実行されたインストラクションの数を追跡し、目標数に達したら割り込みをかける。プロセッサアクティビティから独立したタイミングソースに問い合わせて、経過時間を決定する。そして、経過時間は元の「時間量子」と比較される。その結果得られる比率は、任意の特定のホストシステムの動作速度特性から独立したタイミングリファレンスである。このリファレンスは、ホストシステムの動作速度を予測し、次のインストラクションブロックを実行し、処理を繰り返す前に、より正確にターゲットシステムに適合するように、ホストコンピュータとエミュレーションプロセスのファクターを調整するために用いられる。<sup>40</sup>

**【0012】**

このシステムは、ターゲットインストラクションの実行速度が一貫している場合に作用するけれども、ターゲットシステムの異なる部分が異なるやり方でエミュレートされ、異<sup>50</sup>

なるクロックレートをもつ状況に取り組むことはしない。たとえば、ターゲットシステムのあるコンポーネントはインタープリテーションによってエミュレートされ、その場合、インタープリトされたターゲットインストラクションは固定クロックでホストシステム上で実行される。しかし、ターゲットマシンの別のコンポーネントは、トランスレーションによってエミュレートされ、その場合、トランスレートされたターゲットインストラクションは可変クロックを用いてホストシステム上で実行される。ホストシステム上のこのような二つの異なるクロックを管理することは、全く異なる問題を提示する。

#### 【0013】

このように、これらの問題を解決するエミュレーションシステムと方法が当該分野で求められている。

10

#### 【課題を解決するための手段】

#### 【0014】

上記の不利益を克服するために、本発明の実施の形態は、ホストシステム上のターゲットシステムのエミュレーションにおけるクロック管理の問題を解決するための方法およびシステムに向けられる。ターゲットプログラムの第1コードインストラクションセットをインターパリトして前記ホストシステム上で前記ターゲットシステムの第1コンポーネントをエミュレートするインターパリトされたコードインストラクションを生成する。第2コードインストラクションセットをトランスレートして前記ホストシステム上で前記ターゲットシステムの第2コンポーネントをエミュレートするトランスレートされたコードインストラクションを生成する。インターパリトされたインストラクションを第1クロックにもとづいて実行する。トランスレートされたインストラクションを第2クロックにもとづいて実行する。ホストシステムは、トランスレートされたインストラクションとインターパリトされたインストラクションの間で望ましい同期を維持するために、前記第1または第2クロック、前記トランスレートまたはインターパリトされたインストラクションの実行、またはメモリアクセスを調整する。

20

#### 【0015】

たとえば、前記第1クロックは固定クロック、前記第2クロックは可変クロックである場合、前記第2クロックが前記第1クロックよりも速く進んでいるとき、トランスレートされたインストラクションの実行またはメモリアクセスを停止してもよい。あるいは、ターゲットプログラムの1以上の部分によるターゲットシステムのあるコンポーネントの使用効率にもとづいて1以上のクロックレートを調整してもよい。フレームをディスプレイデバイスに提示する前に、ホストデバイスがトランスレートされたインストラクションのオペレーションを完了するのに十分な時間がない状況において、前記トランスレートされたインストラクションのオペレーションを完了するためのさらなる時間を与えるために、現在のフレームを繰り返してもよい。ある実施の形態において、前記ターゲットプログラムがあらかじめ定められた回数よりも多くハードウェアレジスタをポーリングするならば、前記ターゲットプログラムのエミュレーションの実行をタイミングアウトを避けるやり方に変更してもよい。他の状況において、ターゲットシステムのあるエミュレートされたコンポーネントが別のコンポーネントによって読まれるインストラクションを書き込んでよい。読み出しコンポーネントが書き込みコンポーネントの先を行くことを防ぐために、これらのインストラクションはジャーナルに格納され、書き込まれた順番でジャーナルから読み出されてもよい。

30

#### 【発明を実施するための最良の形態】

#### 【0016】

本発明の教示内容は、以下の詳細な説明を添付図面とともに参酌することによって容易に理解することができる。

#### 【0017】

これから述べる詳細な説明には、例示のために特定の詳細な内容が多く含まれるが、当業者であれば、これらの詳細な内容にいろいろなバリエーションや変更を加えても、本発明の範囲を超えないことは理解できよう。したがって、以下で説明する本発明の例示的な

40

50

実施の形態は、権利請求された発明に対して、一般性を失わせることなく、また、何ら限定をすることもなく、述べられたものである。

#### 【0018】

上述のように、たいていのエミュレーション技術は、固定クロックを用いるインターブリテーション、または可変クロックを用いるトランスレーションのいずれかに関する。ソニー・プレイステーション（登録商標）2（P S 2）のようなゲームコンソールデバイスのエミュレーションにおいて、あるターゲットコンポーネントはインターブリテーションによってエミュレートされ、他はトランスレーションによってエミュレートされる。プレイステーション（登録商標）は日本のソニー・コンピュータ・エンタテインメント社の登録商標である。本発明の実施の形態は、ターゲットデバイスの異なるコンポーネントをエミュレートするために使われる、異なるタイプのエミュレーションから生じるクロック管理の問題を解決する。10

#### 【0019】

一例として、図1Aはゲームコンソールデバイスの形のターゲットシステム100のブロック図である。ターゲットシステムは、エモーション・エンジンと呼ばれるメインプロセッサモジュール102、グラフィックシンセサイザ104、入出力（I/O）プロセッサ（IOP）106、およびサウンドプロセッサユニット108を中心に構築される。エモーション・エンジン102は典型的にはCPUコア、コプロセッサ、システムクロックを含み、ランダムアクセスメモリ（RAM）110がつなげられている。エモーション・エンジン102はアニメーション計算を実行し、シーンを解析し、それをラスタライゼーションのためにグラフィックシンセサイザ（GS）104に送られる2次元画像に変換する。20

#### 【0020】

図1Bに示すように、EE102は、CPUコア122とそれにつながれた浮動小数点ユニット（FPU）コプロセッサ124、第1および第2ベクトルコプロセッサ126、128、グラフィックス・インターフェース・コントローラ130、およびインタラプト・コントローラ（INTC）132を含む。CPU122、ベクトルコプロセッサ126、128、GIF130、およびINTC132は、128ビットメインバス134に接続されている。FPU124はCPU122に直接つながれている。CPU122は第1ベクトルコプロセッサ（VU0）126に接続されており、VU0は第2ベクトルコプロセッサ（VU1）128に接続されている。第2ベクトルコプロセッサ（VU1）128はグラフィックスインターフェース（GIF）130に接続されている。EE102は、さらにタイマ136、ダイレクト・メモリ・アクセス・コントローラ（DMAC）138、イメージデータ伸張プロセッサ（IPU）140、DRAMコントローラ142、およびEE102とIOP106間の通信を容易にするサブバス・インターフェース（SIF）144を含む。30

#### 【0021】

CPUコア122は、300メガヘルツのクロック周波数で動作する128ビットプロセッサであってもよい。このプロセッサは、64ビットのインストラクションをもつMIPSインストラクションセットを用いて、128ビットマルチメディアインストラクションをもつ2ウェイのスーパースカラとして動作する。CPU122は、データキャッシュ、インストラクションキャッシュ、およびスクラッチパッドと呼ばれることがあるオンチップメモリの領域を含む。スクラッチパッドが利用可能な小さなローカルメモリとして機能するため、メインバス134がコードおよび/またはデータの転送でビジーである間でも、CPU122は演算を実行することができる。第1ベクトルユニット126はアニメーションと物理計算のために用いることができる。第2ベクトルユニット128はジオメトリ変換のために用いることができる。GIF130はEE102とGS104の間のメインインターフェースとして機能する。40

#### 【0022】

IOP106はターゲットシステム100の以前のバージョンと下位互換性のあるプロ50

セッサとそれにつなげられたRAM112を含む。IOP106は、コントローラ、USBデバイス、ハードディスク、イーサネット（商標または登録商標）カードまたはモデムのような外部デバイスや、サウンドプロセッサユニット108、ROM114およびCD/DVDユニット116のようなシステムの他のコンポーネントからの入出力を処理する。ターゲットプログラム118は、CD/DVDユニット116にロードされたCD/R ROMディスク上に格納されてもよい。ターゲットプログラム118からのインストラクションはEEのRAM108またはIOPのRAM112に格納され、ターゲットシステム100のいろいろなプロセッサによって、これらのプロセッサが読むことのできるネイティブマシンコードで実行されてもよい。

## 【0023】

10

本発明の実施の形態において、ターゲットシステム100が並列プロセッシングホストシステム200を用いてエミュレートされるため、その結果、ホストシステム200は、ターゲットプログラム118のようなターゲットシステム100にとってネイティブなコードで書かれたプログラムを実行することができる。図2Aは、ターゲットシステム100をエミュレートするように構成されたセルプロセッサ201にもとづくホストシステム200の例を示す。セルプロセッサ201は、メインメモリ202、一つのパワー・プロセッサ・エレメント（PPE）204、および8個のシナジスティック・プロセッサ・エレメント（SPE）206を含む。もっとも、セルプロセッサ201は一つよりも多いPPEと任意の数のSPEで構成されてもよい。各SPE206はシナジスティック・プロセッサ・ユニット（SPU）とローカルストア（LS）を含む。メモリ202、PPE204、および複数のSPE226は、リング型のエレメント・インターフェクト・バス（EIB）210上でお互いに通信可能であり、また、I/Oデバイス208とも通信可能である。PPE204と複数のSPE206は、バスインターフェースユニット（BIU）を介してEIB210にアクセスすることができる。PPE204と複数のSPE206はメモリ・フロー・コントローラ（MFC）を用いてEIB210経由でメインメモリ202にアクセスできる。メモリ202は、ターゲットシステム100用に書かれたコード化されたインストラクションのインターブリテーションおよびトランスレーションが実装されたエミュレーションプログラム209を含む。これらのコード化されたインストラクションはI/Oデバイス208につなげられたCD/DVDリーダ211にあるCD/R ROMディスクから読み出されてもよい。ターゲットプログラム118を含むCD/R ROMディスクはCD/DVDリーダ211にロードされる。少なくとも一つのSPE206は、図1A、図1Bを参照して上述されたIOP106をエミュレートするインストラクションをもったエミュレートされたIOPコード205をローカルストア内に受け取る。

20

## 【0024】

30

一例として、本発明の実施の形態において、PPE204上で実行されるトランスレータ212は、ターゲットプログラム118をPPE204上で実行可能なマシンコード213にトランスレートすることにより、ターゲットシステム100のEE102をエミュレートする。トランスレートされたコード213は可変クロックを用いる。本発明のこの実施の形態において、ターゲットプログラム118のIOPインストラクションをインターブリトすることにより、IOP106をエミュレートするインターブリタがPPE204に実装される。インターブリタ214は固定クロックを用いて実行される。ターゲットシステム100上で、IOPクロックレートは、セルプロセッサ201に対するクロックレートよりもずっと小さい（たとえば、約100分の1である）。結果的に、あるトランスレートされたEEコードは非常に速く実行され、あるものは非常に遅く実行されることがある。さらに、ターゲットシステム100用に書かれたプログラムは、EE102を用いる効率が異なることがある。結果的に、エミュレートされたプログラムの異なる部分が非常に速くあるいは非常に遅く実行されることがある。

40

## 【0025】

このプログラム209は、EEクロックが非常に速く（すなわち、IOPクロックよりも速く）進むとき、IOPクロックとEEクロックの同期を取るために、実行またはダイ

50

レクト・メモリ・アクセス (DMA) を止めることにより、異なるクロックレートを扱うことができる。たとえば、図2Bに示すように、トランスレートされたEEコード213は、オペレーションがどれくらいの時間がかかるかを推定するためにエミュレートされたサイクルカウンタ215を使ってもよい。トランスレートされたEEコードインストラクションの実行タイミングは可変EEクロック217に合わせる。同様に、インタープリトされたIOPコードの実行タイミングは固定IOPクロック207に合わせる。プログラム209はステップ220に示されるようにN回のエミュレートされたサイクルを実行する。Nサイクルの実行の後、プログラム209はIOPクロック207とEEクロック217が同期しているかどうか調べることができる。たとえば、ステップ222において、プログラム209は、EEクロック217がIOPクロック207よりも速く進んでいるかどうかを調べる。もしそうでないなら、通常の実行がもうNサイクルの間、進められる。もしそうであるなら、プログラム209は、ステップ224で示すように、トランスレートされたコードに所定数の「ノーオペレーション」(NOP)を挿入するなどにより、EEクロックを停止することができ、その後、ステップ226に示すように、次のEEインストラクションを実行するか、DMAインストラクションを実行する。  
10

#### 【0026】

ある状況では、フレームがCRTモニタやテレビなどの表示装置に提供される前に、トランスレートされたコード213がEEオペレーションを完了するには十分な時間がないこともある。図2Cに示すように、ホストシステム200上で実行されるプログラム209は、ステップ232において、EEクロックが遅すぎるかどうかを決定する。もしそうであり、かつ、ステップ234において、処理中のフレームを完了するのに十分な時間がないと判定される場合、ステップ236においてフレーム、たとえば現在のフレーム、すなわち、画面に現在提供されているフレームまたはもっとも最近に提供されたフレームを繰り返すことで、トランスレートされたEEコード213のより遅い実行に対して埋め合わせをするためにさらなる時間を提供する。  
20

#### 【0027】

本発明のある実施の形態において、与えられたターゲットプログラムタイトルに対するコードは、EEの使用効率を分析され、EEクロックレートが、コードの異なる部分に対する当該効率にもとづいて調整される。たとえば、図3に示されるように、メモリ202は、ターゲットシステム100上で実行可能な異なるプログラムに対するEE効率のデータが格納されたルックアップテーブル240を含む。プログラムの異なる部分は異なる効率でEEを使用するため、与えられたプログラムに対するEE効率のデータは、当該プログラムの異なる部分に関する各エントリをともなって複数のエントリを含む。ルックアップテーブル240に対するデータは、既知のターゲットシステムのプログラムを効率について分析することによって生成することができる。ソニープレイステーション(登録商標)2のようなゲーム機用のプログラムの場合は、EEアライザを用いて、ターゲットシステムプログラム118のいろいろな部分に対してEEの利用効率を判定することができる。  
30

#### 【0028】

プログラム209は図3の右側に示されるように、ルックアップテーブルのデータを利用することができる。具体的には、ステップ302に示すように、プログラム209はターゲットシステムプログラムのタイトル(たとえば、CD/DVDリーダ211にあるゲームCDからのゲームタイトル)を読み取る。タイトルにもとづいて、プログラム209は、そのターゲットシステムプログラムの1以上の部分に対するEE効率を検索することができる。プログラム209は、上述のインターパリトされたIOPコード205、固定されたIOPクロック207、トランスレートされたEEコード213、および可変EEクロック217を用いて、ステップ306で示すようにホストマシンプログラムのエミュレーションを実行する。ステップ308で示すように、エミュレーションプログラム209は、ターゲットシステムプログラムの一部に対するEE効率にもとづいてEEクロック217に対するクロックレートを調整する。たとえば、ターゲットシステムプログラムの  
40  
50

ある部分が比較的高い効率をもつならば、EEクロックレートをその部分については増加させる。同様に、低いEE効率をもつターゲットシステムプログラムの部分については、EEクロックレートを減少させる。

#### 【0029】

あるハードウェアのエミュレーションの過程で、ターゲットシステム100とホストシステム200の間でクロックレートの違いがあるために、あるイベントがタイムアウトする可能性がある。たとえば、あるターゲットシステムプログラムは、制御レジスタに値を設定し、その後、結果を待つためにそれを繰り返しポーリングすることがよくある。そのようなプログラムは、ポーリングが所定回数の試行の後、不成功であった場合、ハードウェアエラーを示すタイムアウトコードを含むことがある。しかしながら、ターゲットプログラム118がホストシステム200上でずっと速く実行されるなら、ハードウェアエラーがなくてもエミュレーションの間、このカウントは超えてしまうかもしれない。10

#### 【0030】

エミュレーションプログラム209はハードウェアエミュレーションコード、すなわち、ターゲットシステム100上のあるハードウェアをエミュレートするソフトウェアコードを含んでもよい。タイムアウトハザードが存在するかどうかを判定し、ターゲットプログラム118のエミュレーションの実行をタイムアウトを避けるやり方に変更することにより、当該ハザードに対する補償をするタイムアウトトラッキング制御をハードウェアエミュレーションコードに含めることもできる。タイムアウトトラッキング制御方法400の一例を図4で説明する。ステップ402において、ターゲットプログラム118がハードウェア制御レジスタに書き込みをしたかどうかを判定する。たとえば、メモリマップされたI/O(MMIO)を用いた、ハードウェアに対するメモリマップされたリードは、ターゲットプログラム118がハードウェア制御レジスタに書き込みをしたことを示すものとして用いることができる。ステップ404に示されるように、同じレジスタに対して連続的にリードがなされた回数をカウントする。たとえば、ベクトルコプロセッサユニット126、128やIPU140のような、EEコンポーネント内のあるレジスタはタイムアウトトラッキングを利用することができる。ステップ406において、連続するリードの回数がN回リード(たとえば、一つのライトの後、50回のリード)の閾値を超えるなら、ハードウェアエミュレーションコードは、ホストシステムソフトウェアがこのレジスタのポーリングを試みていると判定する。それに応じて、ハードウェアエミュレーションコードは、ステップ408で示すように、タイムアウトが起きないように補償する。たとえば、ハードウェアエミュレーションコードは、ホストシステムソフトウェアがタイムアウトに到達するのを防ぐために十分な余裕をもたせて速度を落としてもよい。あるいは、ハードウェアステートが変化する(たとえば、エミュレートされたハードウェアが要求されたオペレーションを終了する)まで、ハードウェアエミュレーションコードはエミュレートされたホストシステムソフトウェアに何らレスポンスを返さないようにしてもよい。20  
。

#### 【0031】

別の方法として、タイムアウトハザードを提示しない、ポーリングオペレーションが起きているかどうかを、何もしないタイミンググループやメモリコピー操作とははつきり区別できるものとして判定するために、ターゲットプログラム118の関連する部分を解析することができる。タイムアウトトラッキングをやっている一方で、エミュレートされたハードウェアが割り込みハンドリングのような無関係のファンクションを実行してもよい。30

#### 【0032】

ターゲットシステム100上で、EE102があるデバイスから読み出し、別のデバイスに書き込んでもよい。たとえば、EE102はIOP106からデータを読み出し、そのデータを処理してGS104に処理結果を書き出してもよい。ホストシステム200上のそのようなオペレーションのエミュレーションにおいて、読み出し側が書き出し側の先を行くことがないようにすることは重要である。もしリードオペレーションをエミュレー40  
50

トしているエミュレーションプログラム209の部分がストールしたなら、ライトオペレーションをエミュレートしている当該プログラム209の部分はどこでストールしたかを知っている必要がある。ターゲットシステム100上で、EEのRAM108、スクラッチパッド123、IPU140およびSIF144に対してリードまたはライトするデバイスは、他のデバイスにストールアドレスがどこであるかを伝えることができる。

#### 【0033】

たとえば、図5に例示されるように、IOPエミュレータ502がサウンドプロセッサエミュレータ504にインストラクションを書き込むとき、ライトインストラクションのタイミングと順番はサウンドプロセッサエミュレータ504にとって重要である。IOPエミュレータ502がサウンドプロセッサエミュレータ504に書き出すインストラクションは、すべてのサウンドプロセッサライトインストラクション508とサウンドプロセッサレジスタ値510の順序を記録するライト「ジャーナル」506に格納される。サウンドプロセッサエミュレータ504はジャーナル506から順番にライトインストラクション508を対応するレジスタ値510とともに引き出す。書かれた順番にジャーナル506からライトインストラクション508を引き出すことにより、読み出し側（この場合サウンドプロセッサ）が書き出し側（IOP）の先を進むことはない。図5で説明した概念は、ターゲットシステム100の任意のエミュレートされたコンポーネントのペアであって、一方がデータまたはインストラクションを書き出し、他方がそれを読み出すコンポーネントのペアに一般的に適用できる。一例として、IOPエミュレータ502とサウンドプロセッサエミュレータ504は、図2Aに関して上述したセルプロセッサ201上のSPE206によって実装してもよい。

#### 【0034】

本発明の好ましい実施の形態を完全な形で説明してきたが、いろいろな代替物、変形、等価物を用いることができる。したがって、本発明の範囲は、上記の説明を参照して決められるものではなく、請求項により決められるべきであり、均等物の全範囲も含まれる。ここで述べた特徴はいずれも、好ましいかどうかを問わず、他の特徴と組み合わせてもよい。請求項において、明示的に断らない限り、各項目は1またはそれ以上の数量である。請求項において「～のための手段」のような語句を用いて明示的に記載する場合を除いて、請求項がミーンズ・プラス・ファンクションの限定を含むものと解してはならない。

#### 【図面の簡単な説明】

【図1A】本発明の実施の形態に係るエミュレートされるべきターゲットデバイスのプロック図である。

【図1B】図1Aのターゲットデバイスのエモーション・エンジンのブロック図である。

【図2A】図1A、図1Bのターゲットデバイスをエミュレートするホストデバイスの概略図である。

【図2B】本発明の実施の形態に係るターゲットデバイスの異なる部分のエミュレーションのための異なるクロックレートの管理を説明するフローダイアグラムである。

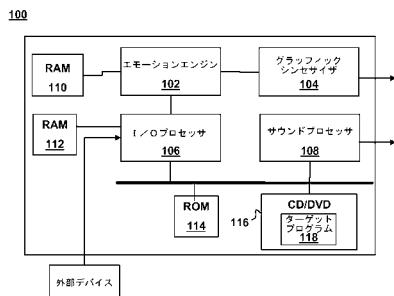
【図2C】本発明の別の実施の形態に係るターゲットデバイスの異なる部分のエミュレーションのための異なるクロックレートの管理を説明するフローダイアグラムである。

【図3】本発明の別の実施の形態に係るターゲットデバイスの異なる部分のエミュレーションのための異なるクロックレートの管理を説明するフローダイアグラムである。

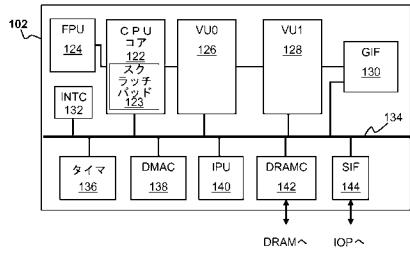
【図4】本発明の別の実施の形態に係るターゲットデバイスのエミュレーションのためのタイムアウトキャッシング制御を説明するフローダイアグラムである。

【図5】本発明の実施の形態に係るエミュレーションにおけるリード／ライトジャーナルの使用を説明するフローダイアグラムである。

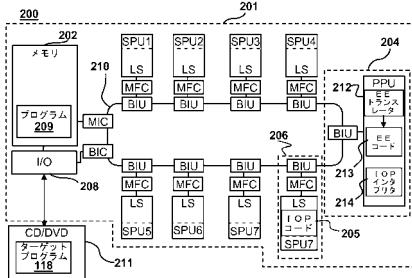
【図1A】



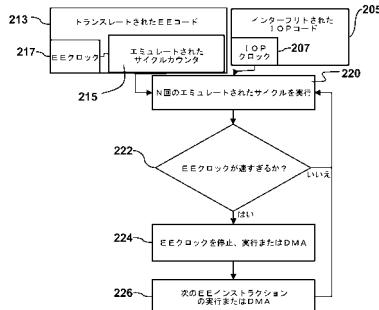
【図1B】



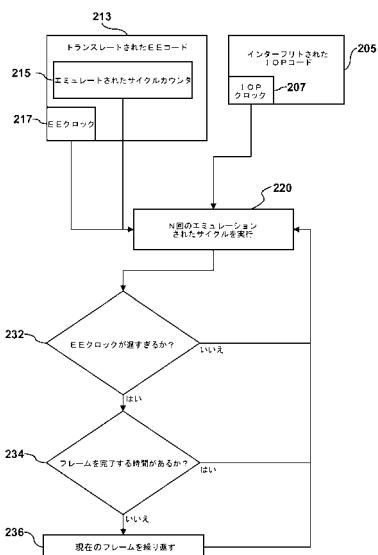
【図2A】



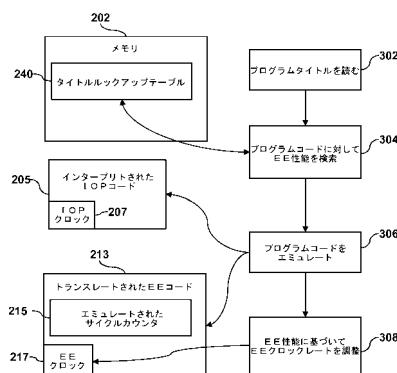
【図2B】



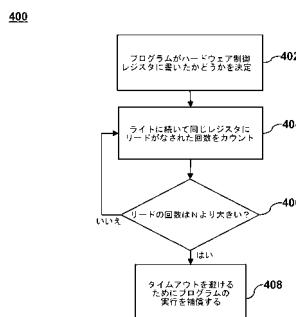
【図2C】



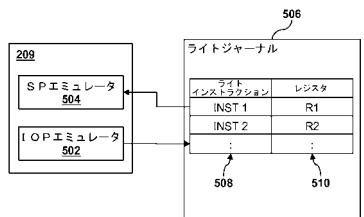
【図3】



【図4】



【図5】



---

フロントページの続き

(31)優先権主張番号 60/746,273  
 (32)優先日 平成18年5月3日(2006.5.3)  
 (33)優先権主張国 米国(US)  
 (31)優先権主張番号 60/797,435  
 (32)優先日 平成18年5月3日(2006.5.3)  
 (33)優先権主張国 米国(US)  
 (31)優先権主張番号 60/797,761  
 (32)優先日 平成18年5月3日(2006.5.3)  
 (33)優先権主張国 米国(US)  
 (31)優先権主張番号 60/797,762  
 (32)優先日 平成18年5月3日(2006.5.3)  
 (33)優先権主張国 米国(US)  
 (31)優先権主張番号 11/700,448  
 (32)優先日 平成19年1月30日(2007.1.30)  
 (33)優先権主張国 米国(US)  
 (31)優先権主張番号 11/696,691  
 (32)優先日 平成19年4月4日(2007.4.4)  
 (33)優先権主張国 米国(US)  
 (31)優先権主張番号 11/696,699  
 (32)優先日 平成19年4月4日(2007.4.4)  
 (33)優先権主張国 米国(US)  
 (31)優先権主張番号 11/696,684  
 (32)優先日 平成19年4月4日(2007.4.4)  
 (33)優先権主張国 米国(US)

(72)発明者 ピクター スバ ミウラ  
 アメリカ合衆国、カリフォルニア州 94404、フォスター・シティー、セカンド・フロアー、  
 イースト・ヒルスデイル・ブルバード 919  
 (72)発明者 ブライアン ワトソン  
 アメリカ合衆国、カリフォルニア州 94404、フォスター・シティー、セカンド・フロアー、  
 イースト・ヒルスデイル・ブルバード 919

審査官 坂庭 剛史

(56)参考文献 特開2006-092542(JP,A)  
 特開2002-132496(JP,A)  
 特開2001-314644(JP,A)  
 特開2001-340641(JP,A)  
 特開平07-160537(JP,A)  
 特表2002-533837(JP,A)  
 特開2004-318449(JP,A)  
 特開2001-216177(JP,A)  
 米国特許第6882968(US,B1)

(58)調査した分野(Int.Cl., DB名)

G06F 9/455  
 G06F 11/28