

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第6321167号
(P6321167)

(45) 発行日 平成30年5月9日(2018.5.9)

(24) 登録日 平成30年4月13日(2018.4.13)

(51) Int.Cl.

G06F 1/04 (2006.01)
G06T 15/00 (2011.01)

F 1

G06F 1/04
G06T 15/00570
501

請求項の数 30 (全 23 頁)

(21) 出願番号 特願2016-529907 (P2016-529907)
 (86) (22) 出願日 平成26年11月11日 (2014.11.11)
 (65) 公表番号 特表2017-505931 (P2017-505931A)
 (43) 公表日 平成29年2月23日 (2017.2.23)
 (86) 國際出願番号 PCT/US2014/065045
 (87) 國際公開番号 WO2015/073445
 (87) 國際公開日 平成27年5月21日 (2015.5.21)
 審査請求日 平成29年10月12日 (2017.10.12)
 (31) 優先権主張番号 61/903,843
 (32) 優先日 平成25年11月13日 (2013.11.13)
 (33) 優先権主張国 米国(US)
 (31) 優先権主張番号 14/448,556
 (32) 優先日 平成26年7月31日 (2014.7.31)
 (33) 優先権主張国 米国(US)

早期審査対象出願

(73) 特許権者 595020643
 クアアルコム・インコーポレイテッド
 QUALCOMM INCORPORATED
 アメリカ合衆国、カリフォルニア州 92
 121-1714、サン・ディエゴ、モア
 ハウス・ドライブ 5775
 (74) 代理人 100108855
 弁理士 蔵田 昌俊
 (74) 代理人 100109830
 弁理士 福原 淑弘
 (74) 代理人 100158805
 弁理士 井関 守三
 (74) 代理人 100112807
 弁理士 岡田 貴志

最終頁に続く

(54) 【発明の名称】 ゲーミングの作業負荷のための C P U 周波数を動的にスロットリングするシステムおよび方法

(57) 【特許請求の範囲】

【請求項 1】

少なくとも 1 つの中央処理装置 (C P U) の C P U 周波数をスケーリングする方法であつて、

レンダリングのフレームごとのグラフィックスライブラリドローコールごとになされたグラフィックスライブラリコールの平均の量をトラッキングすることと、

C P U を含むコンピューティングデバイス上のゲーミングの作業負荷を、レンダリングのフレームごとのグラフィックスライブラリドローコールごとになされたグラフィックスライブラリコールの前記平均の量を前記トラッキングすることに基づいて、検出することと、

前記ゲーミングの作業負荷を検出することに応答して、前記コンピューティングデバイスをゲーミングモードに切り替えることと、ここにおいて、前記コンピューティングデバイスを前記ゲーミングモードに前記切り替えることは、前記コンピューティングデバイス上で実行している前記 C P U の C P U 周波数最大値 (F M a x) を低減することを含む、を備える、方法。

【請求項 2】

N 個の連続するフレームのスライディングウィンドウは、定義され、前記トラッキングすることは、レンダリングの前記 N 個の連続するフレームの各々ごとのグラフィックスライブラリドローコールごとになされたグラフィックスライブラリコールの平均の量をトラッキングすることを含む、請求項 1 に記載の方法。

【請求項 3】

前記トラッキングすることの結果が閾値を満たさないとき、前記コンピューティングデバイスが、ゲーミングモードを処理していないことを検出することと、

前記コンピューティングデバイスをノンゲーミングモードに切り替えることと、ここにおいて、前記コンピューティングデバイスをノンゲーミングモードに前記切り替えることは、前記CPU FMaxを最大の許可された周波数にリセットすることを含む、

をさらに含む、請求項1に記載の方法。

【請求項 4】

前記結果が25未満であるとき、前記結果は、前記閾値を満たさない、請求項3に記載の方法。10

【請求項 5】

レンダリングのフレームごとのグラフィックスライブラリドローコールごとになされたテクスチャコールの平均の量をトラッキングすることと、

前記CPUを含む前記コンピューティングデバイス上の前記ゲーミングの作業負荷を、レンダリングのフレームごとのグラフィックスライブラリドローコールごとになされたテクスチャコールの平均の量を前記トラッキングすることに基づいて、検出することと、

をさらに含む、請求項3に記載の方法。

【請求項 6】

M個の連続するフレームの第2のスライディングウィンドウは、定義され、テクスチャコールの平均の量を前記トラッキングすることは、レンダリングの前記M個の連続するフレームの各々のためのグラフィックスライブラリドローコールごとになされたテクスチャコールの前記平均の量をトラッキングすることを含む、請求項5に記載の方法。20

【請求項 7】

グラフィックスライブラリコールの平均の量を前記トラッキングすることの第1の結果が第2の閾値を満たし、かつテクスチャコールの平均の量を前記トラッキングすることの第2の結果が第3の閾値を満たすとき、前記ゲーミングの作業負荷は、前記コンピューティングデバイス上で検出される、請求項6に記載の方法。

【請求項 8】

グラフィックスライブラリコールの平均の量を前記トラッキングすることの前記第1の結果が前記第2の閾値を満たさないとき、またはテクスチャコールの平均の量を前記トラッキングすることの前記第2の結果が前記第3の閾値を満たさないとき、前記コンピューティングデバイスが、ゲーミングの作業負荷を処理していないことを検出する、をさらに含む、請求項7に記載の方法。30

【請求項 9】

連続するフレームの第1のスライディングウィンドウは、5であり、かつ連続するフレームの前記第2のスライディングウィンドウは、5である、請求項8に記載の方法。

【請求項 10】

前記第1の結果が15以上であるとき、前記第1の結果は、前記第2の閾値を満たし、前記第2の結果が1より大きいとき、前記第2の結果は、前記第3の閾値を満たす、請求項8に記載の方法。40

【請求項 11】

グラフィックスライブラリコールの平均の量を前記トラッキングすることは、N個の連続するフレームのあらゆるスライディングウィンドウのために、レンダリングの前記N個の連続するフレームの各々のためのグラフィックスライブラリドローコールごとになされたグラフィックスライブラリコールの前記平均の量をトラッキングすることを含む、請求項1に記載の方法。

【請求項 12】

レンダリングのM個の連続するフレームの各々のためのグラフィックスライブラリドローコールごとになされたテクスチャコールの平均の量をトラッキングすること、

をさらに含む、請求項11に記載の方法。50

【請求項 1 3】

グラフィックスライプラリコールの平均の量を前記トラッキングすることの結果が第1の閾値を満たすとき、前記検出することは、前記コンピューティングデバイス上の前記ゲーミングの作業負荷を検出することを含む、請求項12に記載の方法。

【請求項 1 4】

グラフィックスライプラリコールの平均の量を前記トラッキングすることの第1の結果が第2の閾値を満たし、かつテクスチャコールの平均の量を前記トラッキングすることの第2の結果が第3の閾値を満たすとき、前記検出することは、前記コンピューティングデバイス上の前記ゲーミングの作業負荷を検出することを含む、請求項13に記載の方法。

【請求項 1 5】

前記第1の結果が25以上であるとき、前記第1の結果は、前記第1の閾値を満たし、前記第1の結果が15以上であるとき、前記第1の結果は、前記第2の閾値を満たし、前記第2の結果が1より大きいとき、前記第2の結果は、前記第3の閾値を満たす、請求項14に記載の方法。

【請求項 1 6】

CPU FMax を前記低減することは、前記CPUの前記CPU FMax を1.49 GHz と 1.72 GHz 間の範囲に制限することを含む、請求項1に記載の方法。

【請求項 1 7】

前記コンピューティングデバイスは、カーネルを含み、CPU FMax を前記低減することは、前記カーネルの中でAPIコールを呼び出すことを含む、請求項1に記載の方法。

【請求項 1 8】

前記コンピューティングデバイスは、モバイルデバイスである、請求項1に記載の方法。

【請求項 1 9】

前記モバイルデバイスは、スマートフォン、タブレット、ラップトップ、または携帯情報端末である、請求項18に記載の方法。

【請求項 2 0】

少なくとも1つの中央処理装置(CPU)のCPU周波数をスケーリングするためのシステムであって、

CPUと、

レンダリングのフレームごとのグラフィックスライプラリドローコールごとにされたグラフィックスライプラリコールの平均の量をトラッキングし、前記CPUを含むコンピューティングデバイス上のゲーミングの作業負荷を、グラフィックスライプラリコールの前記トラッキングされた平均の量の結果に基づいて、検出するゲーミングモード検出器と、ここにおいて、前記ゲーミングの作業負荷を検出することに応答して、前記ゲーミングモード検出器は、前記コンピューティングデバイスをゲーミングモードに切り替え、前記コンピューティングデバイス上で実行している前記CPUのCPU周波数最大値(FMax)を低減する、

を備えるシステム。

【請求項 2 1】

前記ゲーミングモード検出器は、N個のフレームのスライディングウィンドウを定義し、レンダリングの前記N個の連続するフレームの各々のためのグラフィックスライプラリドローコールごとにされたグラフィックスライプラリコールの前記平均の量をトラッキングし、ここにおいて、前記結果が第1の閾値を満たすとき、前記ゲーミングモード検出器は、前記コンピューティングデバイス上の前記ゲーミングの作業負荷を検出する、請求項20に記載のシステム。

【請求項 2 2】

前記コンピューティングデバイスが前記ゲーミングモードにあることを決定され、かつ前記結果が前記第1の閾値を満たさないとき、前記ゲーミングモード検出器は、前記コン

10

20

30

40

50

ピューティングデバイスをノンゲーミングモードに切り替える、請求項 2 1 に記載のシステム。

【請求項 2 3】

前記結果が 25 以上であるとき、前記結果は、前記第 1 の閾値を満たす、請求項 2 1 に記載のシステム。

【請求項 2 4】

前記ゲーミングモード検出器は、レンダリングの M 個の連続するフレームの各々のためのグラフィックスライブラリドローコールごとになされたテクスチャコールの平均の量をトラッキングし、ここにおいて、グラフィックスライブラリコールの前記平均の量をトラッキングすることの第 1 の結果が第 2 の閾値を満たし、かつテクスチャコールの前記平均の量をトラッキングすることの第 2 の結果が第 3 の閾値を満たすとき、前記ゲーミングモード検出器は、前記コンピューティングデバイス上の前記ゲーミングの作業負荷を検出する、請求項 2 1 に記載のシステム。10

【請求項 2 5】

前記第 1 の結果が 25 以上であるとき、前記第 1 の結果は、前記第 1 の閾値を満たし、前記第 1 の結果が 15 以上であるとき、前記第 1 の結果は、前記第 2 の閾値を満たし、前記第 2 の結果が 1 より大きいとき、前記第 2 の結果は、前記第 3 の閾値を満たす、請求項 2 4 に記載のシステム。

【請求項 2 6】

前記ゲーミングモード検出器は、前記 CPU の前記 CPU_FMax を前記 CPU の最大の許可された周波数レベルにリセットし、前記 CPU の前記 CPU_FMax を増加させる、請求項 2 0 に記載のシステム。20

【請求項 2 7】

オペレーションを行うためのコンピュータ実行可能な命令を記憶した非一時的なコンピュータ可読媒体であって、

レンダリングのフレームごとのグラフィックスライブラリドローコールごとになされたグラフィックスライブラリコールの平均の量をトラッキングすることと、

CPU を含むコンピューティングデバイス上のゲーミングの作業負荷を、レンダリングのフレームごとのグラフィックスライブラリドローコールごとになされたグラフィックスライブラリコールの平均の量を前記トラッキングすることに基づいて、検出することと、30

前記ゲーミングの作業負荷を検出することに応答して、前記コンピューティングデバイスをゲーミングモードに切り替えることと、ここにおいて、前記コンピューティングデバイスをゲーミングモードに前記切り替えることは、前記コンピューティングデバイス上で実行している前記 CPU の CPU 周波数最大値 (FMax) を低減することを含む、

を備える、非一時的なコンピュータ可読媒体。

【請求項 2 8】

前記オペレーションは、

N 個の連続するフレームのスライディングウィンドウを定義することと、前記トラッキングすることは、レンダリングの前記 N 個の連続するフレームの各々のためのグラフィックスライブラリドローコールごとになされたグラフィックスライブラリコールの平均の量をトラッキングすることを含む、40

グラフィックスライブラリドローコールごとになされたグラフィックスライブラリコールの前記平均の量の結果が第 1 の閾値を満たすとき、前記コンピューティングデバイス上の前記ゲーミングの作業負荷を検出することと、

をさらに含む、請求項 2 7 に記載のコンピュータ可読媒体。

【請求項 2 9】

前記オペレーションは、

レンダリングの M 個の連続するフレームの各々のためのグラフィックスライブラリドローコールごとになされたテクスチャコールの平均の量をトラッキングすることと、

グラフィックスライブラリドローコールごとになされたグラフィックスライブラリコー50

ルの前記平均の量の結果が第2の閾値を満たし、かつテクスチャコールの前記平均の量の結果が第3の閾値を満たすとき、前記コンピューティングデバイス上で前記ゲーミングの作業負荷を検出することと、前記第1の閾値が、前記第2の閾値とは異なる、

をさらに含む、請求項28に記載のコンピュータ可読媒体。

【請求項30】

少なくとも1つの中央処理装置(CPU)のCPU周波数をスケーリングするための装置であって、

レンダリングのフレームごとのグラフィックスライブラリドローコールごとになされたグラフィックスライブラリコールの平均の量をトラッキングするための手段と、

CPUを含むコンピューティングデバイス上のゲーミングの作業負荷を、レンダリングのフレームごとのグラフィックスライブラリドローコールごとになされたグラフィックスライブラリコールの前記平均の量を前記トラッキングすることに基づいて、検出するための手段と、

前記ゲーミングの作業負荷を検出することに応答して、前記コンピューティングデバイスをゲーミングモードに切り替えるための手段と、ここにおいて、前記切り替えることは、前記コンピューティングデバイス上で実行している前記CPUのCPU周波数最大値(FMax)を低減することを含む、

を備える、装置。

【発明の詳細な説明】

【関連出願の相互参照】

20

【0001】

[0001] 本願は、2013年11月13日に出願された米国仮特許出願第61/903,843号に対する優先権を主張する、2014年7月31日に出願された米国非仮特許出願第14/448,556号に対する優先権を主張し、これら両方は、それらの全文を参照により本明細書に組み込まれる。

【技術分野】

【0002】

[0002] 本開示は、中央処理装置(CPU)およびグラフィックス処理ユニット(GPU)に一般的に関し、より具体的にはGPUの重い作業負荷のためのCPU周波数をコントロールすることに関する。

30

【背景技術】

【0003】

[0003] 動的なCPU周波数のスケーリングは、CPU周波数が、例えば、電力を保持するためにCPUのパーセンテージ負荷(CPU percentage load)に基づいて調整されるコンピューティングシステムにおける技法である。動的なCPU周波数のスケーリングは、電力を保持するためにコンピューティングシステムで使用されることができ、限られた電力供給を有するモバイルデバイスで使用するために特に有益であり得る。これらのモバイルデバイスは、モバイルデバイスの中のバッテリ以外の一貫した電力供給(consistent power supply)を典型的に有さない。動的な周波数スケーリングはまた、軽く負荷をかけられたマシン lightly loaded machines のためのエネルギーおよび冷却費用(cooling costs)を減らすために使用され得る。

【0004】

[0004] モバイルデバイスは、ユビキタス(ubiquitous)であり、スマートフォン、タブレット、携帯デジタルアシスタント(portable digital assistant)(PDA)、携帯ゲームコンソール、パームトップコンピュータ、および他の携帯電子デバイスを含み得る。これらのデバイスの主要な機能に加えて、多くは、周辺機能(peripheral functions)を含む。例えば、スマートフォンは、通話する主要な機能、ならびにゲームをプレーすること、スチルカメラ(still camera)、ビデオカメラ、全世界測位システム(GPS)ナビゲーション、ウェブブラウジング、eメールの送受信、およびテキストメッセージの送受信の周辺機能を含み得る。そのようなデバイスの機能性が増加するにつれて、そのよう

40

50

な機能性をサポートするために要求される処理電力もまた、増加する。さらに、コンピューティング電力が増加するにつれて、コンピューティング電力を提供するプロセッサを効率的に管理するより大きな必要性が存在する。

【発明の概要】

【0005】

[0005] 本開示は、C P U sに関する。C P U周波数をコントロールするための方法、システム、および技法が提供される。

【0006】

[0006] いくつかの実施形態にしたがって、少なくとも1つの中央処理装置(C P U)のC P U周波数をスロットリングする(throttling)方法は、レンダリングのフレームごとのグラフィックスライブラリドローコール(graphics library draw call)ごとになされたグラフィックスライブラリコールの平均の量をトラッキングすることを含む。方法は、C P Uを含むコンピューティングデバイス上のゲーミングの作業負荷を、レンダリングのフレームごとのグラフィックスライブラリドローコールごとになされたグラフィックスライブラリコールの平均の量をトラッキングすることに基づいて、検出することをさらに含む。方法はまた、コンピューティングデバイスをゲーミングモード(a gaming mode)に切り替えることも含む。コンピューティングデバイスをゲーミングモードに切り替えることは、コンピューティングデバイス上で実行しているC P UのC P U周波数最大値(CPU frequency maximum)を低減することを含む。10

【0007】

[0007] いくつかの実施形態にしたがって、少なくとも1つの中央処理装置(C P U)のC P U周波数をスロットリングするためのシステムは、C P Uを含む。システムはまた、レンダリングのフレームごとのグラフィックスライブラリドローコールごとになされたグラフィックスライブラリコールの平均の量をトラッキングし、前記C P Uを含むコンピューティングデバイス上のゲーミングの作業負荷を、レンダリングのフレームごとのグラフィックスライブラリドローコールごとになされたグラフィックスライブラリコールのトラッキングされた平均の量に基づいて、検出するゲーミングモード検出器を含む。ゲーミングモード検出器は、コンピューティングデバイスをゲーミングモードに切り替え、コンピューティングデバイス上で実行しているC P UのC P U周波数最大値を低減する。20

【0008】

[0008] いくつかの実施形態にしたがって、オペレーションを行うためのコンピュータ実行可能な命令を記憶したコンピュータ可読媒体は、レンダリングのフレームごとのグラフィックスライブラリドローコールごとになされたグラフィックスライブラリコールの平均の量をトラッキングすることと、C P Uを含むコンピューティングデバイス上のゲーミングの作業負荷を、レンダリングのフレームごとのグラフィックスライブラリドローコールごとになされたグラフィックスライブラリコールの平均の量をトラッキングすることに基づいて、検出することと、コンピューティングデバイスをゲーミングモードに切り替えることと、ここで、コンピューティングデバイスをゲーミングモードに切り替えることは、コンピューティングデバイス上で実行しているC P UのC P U周波数最大値を低減することを含む、を含む。30

【0009】

[0009] いくつかの実施形態にしたがって、少なくとも1つの中央処理装置(C P U)のC P U周波数をスロットリングするための装置は、レンダリングのフレームごとのグラフィックスライブラリドローコールごとになされたグラフィックスライブラリコールの平均の量をトラッキングするための手段を含む。装置はまた、C P Uを含むコンピューティングデバイス上のゲーミングの作業負荷を検出するための手段を含む。装置は、コンピューティングデバイス上で実行しているC P UのC P U周波数最大値を低減するための手段をさらに含む。40

【図面の簡単な説明】

【0010】

50

[0010] 本明細書の一部を形成する、添付図面は、発明の実施形態および本明細書とともに例示し、実施形態の原理を説明するためにさらに役立つ。図面において、同様の参照番号は、同一の、または機能的に同様の要素を示し得る。要素がまず現れる図面は、対応する参照番号の中の最も左の数字によって一般的に示される。

【0011】

【図1】[0011] 図1は、特定のオンデマンドCPUガバナ(governor)およびゲーム、Asphalt7(商標)に関する時間にわたる(over time)FPS対CPU周波数最大値の例示的なグラフである。

【図2】[0012] 図2は、いくつかの実施形態にしたがって、少なくとも1つのCPUのCPU周波数をスロットリングするためのシステムを例示するブロック図である。 10

【図3】[0013] 図3は、いくつかの実施形態にしたがって、少なくとも1つのCPUのCPU周波数をスケーリングする方法を例示する流れ図である。

【図4】[0014] 図4は、本開示の1つ以上の実施形態をインプリメントするために適しているコンピュータシステムのブロック図である。

【発明を実施するための形態】

【0012】

I. 概要

II. 例示的なシステムアーキテクチャ

A. ゲーミングの作業負荷を検出する

1. ステートアップデート/ドロー(State-Update/Draw) (SUPD) メトリック 20

2. SUPDおよびバインドテクスチャドロー(Bind-Texture Draw) (BTPD) メトリック

B. ゲーミングモードに切り替える

III. 例示的な方法

IV. 例示的なコンピューティングデバイス

【0013】

I. 概要

[0015] 続く開示は、本開示の異なる特徴をインプリメントするための、多くの異なる実施形態、または例を提供することが理解されることとなる。いくつかの実施形態は、これらの特定の詳細のいくつか、またはすべてなしで実施され得る。コンポーネント、モジュール、および配列の特定の例は、本開示を簡略化するために以下に説明される。これらは、もちろん、単に例にすぎず、限定していると意図されない。 30

【0014】

[0016] モバイルゲーミングは、人気を得ており、ユーザが、彼または彼女のモバイルデバイス上でゲームをプレーすることを可能にする。かなりの数の人気のゲーム(例えば、Need For Speed - Most Wanted(商標)、Asphalt - 7(商標)、およびAsphalt - 8(商標))は、「CPUバウンド(CPU bound)」というよりはむしろ高度なグラフィックス処理ユニット(heavily graphics processing unit)(GPU)バウンドである。商標は、それらそれぞれのオーナーの財産である。CPUは、ゲームのためのドローコールに基づくドローコマンドを構築し、ドローコマンドに基づくコマンドストリームを生成し、コマンドストリームをGPUに提出する。GPUは次に、コマンドストリームの中のドローコマンドを処理することの重作業(heavy work)を行うために進み(proceed to)得る。CPU側の作業負荷は、GPUがゲーミングの作業負荷の重い処理を行うため重くなり得ないが、CPUはまだ、最大のサポートされたCPU周波数(maximum supported CPU frequency)で不必要に動作している可能性がある。 40

【0015】

[0017] CPU周波数最小値(CPU frequency minimum)(FMin)は、コンピューティングデバイス上で作業負荷を実行しているときに使用されることができる最小の許可されたCPU周波数(minimum allowed CPU frequency)に対応する。CPU周波数最大 50

値 (F M a x) は、コンピューティングデバイス上で作業負荷を実行しているときに使用されることができる最大の許可された C P U 周波数に対応する。動的な C P U 周波数スケーリングルーチンは、コンピューティングデバイス上の C P U F M i n と C P U F M a x 間で稼働し得る。C P U F M a x を動的に低減することは、動的な C P U 周波数スケーリングルーチンに、コンピューティングデバイス上の C P U 周波数のより小さい範囲内で動作させる。

【 0 0 1 6 】

[0018] C P U 周波数は、C P U と相互に作用するオンデマンド C P U ガバナによってコントロールされ、それは、C P U に、デフォルト C P U 周波数最大値（例えば、2.15 ギガヘルツ (G H z)）で動作させ得る。説明は、C P U と相互に作用するようにオンデマンド C P U ガバナを説明し得るが、他のガバナがC P U と相互に作用し得る。他の C P U ガバナの例は、とりわけ、パフォーマンスガバナ (performance governor)、パワーセーブガバナ (powersave governor)、インタラクティブガバナ (interactive governor) である。例において、C P U ガバナは、L i n u x (登録商標) オペレーティングシステムで利用可能であり得る。

【 0 0 1 7 】

[0019] 図 1 は、ゲーム、A s p h a l t 7 (商標) を動作している間、特定のオンデマンド C P U ガバナに関する時間にわたる 1 秒あたりのフレーム数 (frames per second) (F P S) 対 C P U 周波数最大値の例示的なグラフである。図 1 において、デフォルト C P U 周波数最大値は、例えば、2.15 G H z であり得る。C P U ガバナは、さらに以下に説明されることとなる C P U 周波数最大値をコントロールし得る。

【 0 0 1 8 】

[0020] 例において、C P U 時間の 3 分の 1 が、最大 C P U クロック (maximum CPU clock) で動作することに費やされ、それは、電力を不必要に消費する。図 1 に例示されるように、ディスプレイパネルリフレッシュレート（例えば、60 F P S）に関連して閾値量（例えば、1.95 G H z - 1.72 G H z）まで C P U F M a x を徐々に低減することは、少量の F P S ロス、または少しの F P S ロスもない結果になり得る。図 1 に示されるように、C P U F M a x が、約 1.49 G H z、約 59 F P S の結果であるとき、C P U F M a x を 2.15 G H z のデフォルトから 1.49 G H z に低減することは、A s p h a l t - 7 (商標) のための同様のパフォーマンスをもたらす。例えば、おおよそ同じパフォーマンスが、低減された最大周波数で C P U を動作させることによって結果として生じることができると、そのデフォルト / より高い最大周波数で C P U を動作させることは望ましくない可能性があり、C P U ガバナが現在稼働するための C P U 周波数のより低い範囲を有するとき、より低い周波数最大値で C P U を稼働することは、C P U の電力節約 (power savings) をさらに生じ得る。

【 0 0 1 9 】

[0021] コンピューティングデバイス上でゲーミングの作業負荷を識別し、コンピューティングデバイス上で C P U F M a x を選択的に、および動的に低減することは望ましくあり得る。このことは、G P U が比較的重いゲーミングの作業負荷に対処する間、C P U に対する電力消費を低減し得る。さらに、より高い C P U 周波数レベルで、コンピューティングデバイスからリリースされるサーマルヒート (thermal heat) は、重要なファクタとなる。C P U F M a x を決定的にコントロールすることは、コンピューティングデバイスが容易に、および不必要に熱くなり、C P U F M a x のさらなるサーマルスロットリングをもたらし、概して望ましくないユーザエクスペリエンスをもたらすのを防ぐのに役立つ。

【 0 0 2 0 】

[0022] 本開示は、C P U 周波数をプロアクティブにコントロールする (proactively control) ためのゲーミングの作業負荷の検出に基づく技法を提供する。したがって、C P U F M a x は、ゲーミングの間モバイルプラットフォームによって使用される全体の電力を低減するために低減され得る。ある実施形態は、電力消費およびサーマルヒートを

10

20

30

40

50

低減させながら、パフォーマンスを維持することの利点を有し得る。

【0021】

I I . 例示的なシステムアーキテクチャ

[0023] 図2は、いくつかの実施形態にしたがって、少なくとも1つのCPUのCPU周波数をスロットリングするためのシステムを例示するブロック図400である。図400は、ゲーミングモード検出器412、オペレーティングシステム414、CPU416、およびGPU418を含むコンピューティングデバイス410を含む。ゲーミングモード検出器412、オペレーティングシステム414、CPU416、およびGPU418は、コンピューティングデバイス410においてオペレーションを実行することができ、オペレーションを行い得る。10

【0022】

[0024] コンピューティングデバイス410は、固定のデバイス、またはモバイルデバイスであり得る。モバイルデバイスは、例えば、スマートフォン、タブレット、ラップトップ、または携帯情報端末であり得る。ゲーミングモード検出器412は、ゲーミングの間モバイルプラットフォームによって使用される全体の電力を低減する間、CPU416のパフォーマンスを維持する。

【0023】

[0025] ゲーミングモード検出器412は、CPU416と相互に作用することができ、CPU416が動作する周波数を管理し得る。CPU416は、ドローコールに基づいてコマンドストリームを生成し、GPU418に提出するためのドローコマンドを構築する。具体的には、CPU416は、ステートおよびドローコールの量へのアクセスを有し、GPU418に送られるグラフィックスライブラリ(GL)コマンドの2進数(binary)を生成し、それは、ドローコールを実行するために作業負荷を処理する。CPU416は、GPU418に送られるための作業負荷(例えば、ドローコールの量)を処理することを決定し得る。しかしながら、GPU418が重い作業負荷を処理している(例えば、ゲームがコンピューティングデバイス410上でプレーされている)とき、CPU周波数最大値でCPU416を動作することは望ましくない可能性がある。CPU416は、ゲーミングモード検出器412にステートおよびドローコールに関する情報をパスすることができ、それは、ユーザースペースの中で実行し、およびそれは、コンピューティングデバイス410がゲーミングの作業負荷を有するか否かを検出するためにステートおよびドローコール情報を使用し得る。20

【0024】

[0026] ゲーミングモード検出器412がコンピューティングデバイス410上でゲーミングの作業負荷を検出するとき、ゲーミングモード検出器412は、コンピューティングデバイス410がもはやゲーミングモードにあるように(to be in the gaming mode)決定されなくなるまでCPU_FMaxを制限し(cap)得る。例において、ゲーミングモード検出器412は、コンピューティングデバイス410上で実行している他のコンポーネントがCPU416に、低減されたCPU_FMaxを超えさせることができないようにCPU_FMaxを低減する。異なるガバナが、異なる周波数で動作し得る。ある実施形態において、CPU416は、1つ以上のコアを含み、個別のCPU_FMaxは、コアの1つ以上のために設定され得る。コンピューティングデバイス410がもはやゲーミングモードにあるように決定されないと、ゲーミングモード検出器412は、CPU_FMax上の制限(cap)を取り除く、または再調整する、あるいは単に低減されたCPU_FMaxをデフォルトレベルまで増加させ得る。30

【0025】

[0027] オープングラフィックスライブラリ(OpenGL)は、二次元および三次元のコンピュータグラフィックスをレンダリングするためのアプリケーションプログラミングインターフェース(API)の仕様(specification)である。OpenGLのインプリメンテーションは、仕様によって定義されたAPIをインプリメントするライブラリである。APIは、クロスランゲージおよびマルチプラットフォームであり、ハードウェア加40

速されたレンダリング (hardware-accelerated rendering) を実現するために G P U と相互に作用するように典型的に使用される。O p e n G L は、C A D、バーチャルリアリティ、サイエンティフィックビジュアライゼーション (scientific visualization)、インフォメーションビジュアライゼーション (information visualization)、ライトシミュレーション、およびビデオゲームで広く使用される。埋め込まれたシステムのためのO p e n G L (OpenGL for Embedded Systems) (O p e n G L E S) は、A P I をレンダリングするO p e n G L コンピュータグラフィックスのサブセットであり、埋め込まれたシステムのために設計される (例えば、スマートフォン、コンピュータタブレット、ビデオゲームコンソールおよびP D A s)。簡潔さのために、O p e n G L またはO p e n G L E S は、コンピュータグラフィックスをレンダリングするための技術を説明するために使用され得るが、このことは、限定していると意図されず、これらの性能を提供する他の技術が本開示の範囲内であることが理解されるべきである。10

【 0 0 2 6 】

A . ゲーミングの作業負荷を検出する

[0028] いくつかの実施形態において、ゲーミングモード検出器 4 1 2 は、コンピューティングデバイス 4 1 0 上でゲーミングの作業負荷を検出し、様々な方法でそのようにし得る。2 個のO p e n G L E S の作業負荷は、ユーザインターフェース (U I) およびゲーミングの作業負荷である。

【 0 0 2 7 】

1 . ステートアップデート / ドロー (S U P D) メトリック

20

[0029] 例において、ゲーミングモード検出器 4 1 2 は、コンピューティングデバイス 4 1 0 上でゲーミングの作業負荷を検出するために平均のステートアップデート / ドロー (S U P D) メトリックを使用する。S U P D メトリックは、ゲーミングの作業負荷に依存し、所与のドローコールに結びつく (bound to)。例えば、グラフィックスライブラリ (G L) コールは、ゲームアプリケーションによって使用されるA P I s である。S U P D メトリックは、(例えば、2 個のe g l S w a p B u f f e r s 間の) レンダリングのフレームごとの単一のG L ドローコール (例えば、g l D r a w E l e m e n t s およびg l D r a w A r r a y) ごとになされたすべてのG L コール (例えば、g l U n i f o r m M a t r i x 、g l B i n d B u f f e r 、g l * 等) の平均の量をトラッキングすることを含み得る。この例において、ゲーミングモード検出器 4 1 2 は、単一のドローコールの前にいくつのG L A P I コール (例えば、ステートアップデート) がなされるかをトラッキングし得る。グラフィックスレンダリングのために、トライアングル (triangle) は典型的に、ドローする (draw) ための最も小さいユニットである。ゲーミングモード検出器 4 1 2 は、特定のトライアングルがドローされる前にいくつのG L A P I コールがなされるかを決定し得る。レンダリングのフレームごとのG L ドローコールごとになされたG L コールのより高い平均の量は、関連付けられた作業負荷がG P U 4 1 8 行き (bound for) であることのより高い可能性を示し得る。30

【 0 0 2 8 】

[0030] S U P D メトリックを使用して、ゲーミングモード検出器 4 1 2 が、単一のレンダリングされたフレームに基づいてコンピューティングデバイス 4 1 0 をノンゲーミングモードからゲーミングモードに (またはゲーミングモードからノンゲーミングモードに) 切り替え得る。このことは、しかしながら、1 フレームごとのベース (frame-by-frame basis) で連続的にゲーミングモードから、およびゲーミングモードに、いったりきたり切り替えることをもたらし得る。それゆえに、特定の状態を満たす持続された重い作業負荷に基づいて、コンピューティングデバイス 4 1 0 をノンゲーミングモードからゲーミングモードに (またはゲーミングモードからノンゲーミングモードに) 切り替えることはより望ましくあり得る。そのような発見的なアプローチ (heuristic approach) を使用して、ゲーミングモードから切り替えること、およびゲーミングモードに切り替えることは、一貫性のある経歴 (consistent history) に基づくことができ、誤検出 (false positives) を避け得る。例えば、ゲーミングの作業負荷の特徴が一貫性があり、散発性がない (n40

50

ot sporadic) ことを確実にするために、スライディングウィンドウ (a sliding window) が使用され得る。この方法において、ゲーミングモード検出器 412 は、重い OpenGL ES ベースのゲーミングの作業負荷対、より軽いユーザインターフェースの作業負荷の間で区別することができ、ゆえにゲーミングの作業負荷のための CPU 周波数最大値をプロアクティブに低減し得る。

【0029】

[0031] 連続するフレームの既定の量におよぶ (spanning) スライディングウィンドウを組み込むいくつかの実施形態において、ゲーミングモード検出器 412 は、レンダリングのフレームごとの GL ドローコールごとの GL コールの平均の量をトラッキングする。例において、連続するフレームの既定のウィンドウのために、レンダリングのフレームごとの GL ドローコールごとの GL コールの平均の量をトラッキングすることの結果が閾値を満たすとき、この状態が満たされるとき GPU バウンドである重い、維持された作業負荷が大いに起こり得るため、ゲーミングモード検出器 412 は、コンピューティングデバイス 410 上のゲーミングの作業負荷を検出したこととなる。例において、レンダリングのフレームごとの GL ドローコールごとの GL コールの平均の量をトラッキングすることの結果は、結果が 25 以上であるとき、閾値を満たす。10

【0030】

[0032] 同様に、GL コールの平均の量をトラッキングすることの結果が、例えば、25 の閾値に達しないとき、ゲーミングモード検出器 412 は、コンピューティングデバイス 410 がゲーミングアクティビティ (gaming activity) に従事していないことを決定することができ、コンピューティングデバイス 410 をノンゲーミングモードに切り替える。ある実施形態において、コンピューティングデバイス 410 をノンゲーミングモードに切り替えることは、CPU の CPU_FMax をリセットすること、または再調整すること（例えば、CPU_FMax の制限を増加させること）を含む。20

【0031】

[0033] SUPD の閾値は、例えば、約 25、正確に 25、または 20 - 25 の範囲であり得る。しかしながら、25 の閾値が例によるものであること、および他の値が様々なコンピューティングデバイス 410 の特徴にしたがって適用され得ることは理解されることとなる。

【0032】

[0034] 例において、スライディングウィンドウのサイズを定義する連続するフレームの既定の量は、5 個のフレームである。OpenGL ES の作業負荷の N 個の（例えば、5 個の）連続するフレームのスライディングウィンドウのために、ゲーミングモード検出器 412 は、レンダリングされるための N 個の（例えば、5 個の）連続するフレームの各々のための GL ドローコールごとになされた GL コールの平均の量をトラッキングし、ここで N は、ゼロより大きい任意の整数である。例えば、SUPD が 25 以上である場合、ゲーミングモード検出器 412 は、（コンピューティングデバイス 410 が、まだゲーミングモードにない場合）コンピューティングデバイス 410 をゲーミングモードに入れ得る。30

【0033】

2. SUPD およびバインドテクスチャドロー (BTPD) メトリック

[0035] 別の例において、ゲーミングモード検出器 412 は、コンピューティングデバイス 410 上でゲーミングの作業負荷を検出するために SUPD および平均バインドテクスチャ / ドロー (BTPD) メトリックを使用する。画像がコンピューティングデバイス 410 のスクリーン上にドローされる前に、テクスチャは、オンスクリーン (onscreen) でレンダリングされるための特定のトライアングルにマップされ得る。ゲームアプリケーションがコンピューティングデバイス 410 上で実行している場合、CPU 416 は、CPU 416 がドローコールを発行し、それを GPU 418 に送る前に複数のテクスチャをルックアップ (look up) し得る。例えば、テクスチャは、トライアングルの 1 つのカラーまたは複数のカラーを示すことができ、複数のテクスチャは、フレームのために決定さ4050

れ得る。B T P D メトリックは、ゲーミングの作業負荷に依存し、所与のドローコールに結びつく。B T P D は、(例えは、2 個の e g l S w a p B u f f e r s 間の) レンダリングのフレームごとの G L ドローコール (例えは、g l D r a w E l e m e n t s および g l D r a w A r r a y) ごとになされたテクスチャコールの平均の量をトラッキングすることを含み得る。この例において、ゲーミングモード検出器 4 1 2 は、いくつの G L A P I コール (例えは、テクスチャコール) が単一のドローコールの前になされるかをトラッキングし得る。レンダリングのフレームごとの G L ドローコールごとのテクスチャコールのより高い平均の量は、より多くのテクスチャがフレームをレンダリングするために使用されることを示す。ゆえに、ゲーミングモード検出状態 (gaming mode detection condition) として使用される S U P D および B T P D メトリックは、O p e n G L E S のゲーミングの作業負荷の広い範囲にわたってよく機能することができ、モバイルデバイス上でゲーミング中にモバイルプラットフォームによって使用される全体の電力を低減するためにプロアクティブに使用されることができる。
10

【 0 0 3 4 】

[0036] S U P D および B T P D メトリックは、単一のレンダリングされたフレームに基づいてコンピューティングデバイス 4 1 0 をノンゲーミングモードからゲーミングモードに (またはゲーミングモードからノンゲーミングモードに) 切り替えるためにゲーミングモード検出器 4 1 2 によって使用され得る。このことは、しかしながら、1 フレームごとのベースで連続的にゲーミングモードから、およびゲーミングモードに、いったりきたり切り替えることをもたらし得る。特定の状態を満たす持続された重い作業負荷に基づいて、コンピューティングデバイス 4 1 0 をノンゲーミングモードからゲーミングモードに (またはゲーミングモードからノンゲーミングモードに) 切り替えることがより望ましくあり得る。そのような発見的なアプローチで、ゲーミングモードから切り替えること、およびゲーミングモードに切り替えることは、作業負荷の経歷に基づくことができ、誤検出を避け得る。
20

【 0 0 3 5 】

[0037] 連続するフレームの既定の量のスライディングウィンドウを組み込むいくつかの実施形態において、ゲーミングモード検出器 4 1 2 は、レンダリングのフレームごとの G L ドローコールごとのテクスチャコールの平均の量をトラッキングする。O p e n G L E S の作業負荷の M 個の (例えは、5 個の) 連続するフレームのスライディングウィンドウのために、ゲーミングモード検出器 4 1 2 は、レンダリングされるための M 個の (例えは、5 個の) 連続するフレームの各々のための G L ドローコールごとのテクスチャコールの平均の量をトラッキングし、ここで M は、ゼロより大きい任意の整数である。例において、連続するフレームの既定の量のために、レンダリングのフレームごとの G L ドローコールごとの G L コールの平均の量をトラッキングすることの結果が第 2 の閾値を満たすとき、およびレンダリングのフレームごとの G L ドローコールごとのテクスチャコールの平均の量をトラッキングすることの結果が第 3 の閾値を満たすとき、ゲーミングモード検出器 4 1 2 は、コンピューティングデバイス 4 1 0 上のゲーミングの作業負荷を検出したこととなる。G P U バウンドである重い、維持された作業負荷は、この状態が満たされるとき大いに起こり得る。多くの U I のケース (UI cases) が、1 5 の S U P D 未満を有する。例において、レンダリングのフレームごとの G L ドローコールごとの G L コールの平均の量をトラッキングすることの結果は、結果が 1 5 以上であるとき第 2 の閾値を満たし、レンダリングのフレームごとの G L ドローコールごとのテクスチャコールの平均の量をトラッキングすることの結果は、結果が 1 より大きいとき第 3 の閾値を満たす。1 5 および 1 の値が例によるものであること、および他の値が様々なコンピューティングデバイス 4 1 0 の特徴にしたがって適し得ることは理解されることとなる。
30
40

【 0 0 3 6 】

[0038] 同様に、グラフィックスライブラリコールの平均の量をトラッキングすることの結果が第 2 の閾値を満たさないとき、またはテクスチャコールの平均の量をトラッキングすることの結果が第 3 の閾値を満たさないとき、ゲーミングモード検出器 4 1 2 が、コ
50

ンピューティングデバイス 410 がゲーミングアクティビティに従事していないことを決定することができ、コンピューティングデバイス 410 をノンゲーミングモードに切り替え得る。コンピューティングデバイス 410 をノンゲーミングモードに切り替えることは、CPU の CPU_FMax をリセットすること（例えば、CPU_FMax の制限を増加させること）を含み得る。例えば、レンダリングのフレームごとの GL ドローコールごとの GL コールの平均の量をトラッキングすることの結果は、結果が 15 未満であるとき第 2 の閾値を満たさず、レンダリングのフレームごとの GL ドローコールごとのテクスチャコールの平均の量をトラッキングすることの結果は、結果が 1 以下である（例えば、1 を越えない）とき第 3 の閾値を満たさない。

【0037】

10

[0039] SUPD の閾値は、例えば、約 15、正確に 15、または 13 - 17 の範囲であり得る。BTPD の閾値は、例えば、約 1、正確に 1、または 0.8 - 1.2 の範囲であり得る。例において、スライディングウィンドウのスパン (span) を定義する連続するフレームの既定の量は、5 個のフレームである。別の例において、フレームの既定の量は、5 個のフレームより大きい、または 5 個のフレームより少ない可能性がある。OpenGL_ES の作業負荷の 5 個の連続するフレームのスライディングウィンドウのために、ゲーミングモード検出器 412 は、レンダリングされるための 5 個の連続するフレームの各々ごとの GL ドローコールごとになされた GL コールの平均の量、およびレンダリングされるための 5 個の連続するフレームの各々ごとの GL ドローコールごとになされたテクスチャコールの平均の量をトラッキングし得る。SUPD が 15 以上であり、かつ BTPD が 1 より大きいとき、ゲーミングモード検出器 412 は、（コンピューティングデバイス 410 が、まだゲーミングモードにない場合）コンピューティングデバイス 410 をゲーミングモードに入れ得る。

20

【0038】

[0040] ゲームは典型的に、1.0 の平均 BTPD より大きく、それらの作業負荷のトラッキングの全体にわたって 2.0 より大きい最大値を有し得る。その一方、UI のケースは典型的に、1.0 の平均 BTPD 未満、または 1.0 の平均 BTPD と同じである。多くの極端な GPU バウンド重量のゲーム (Many extreme GPU bound heavy-weight games)（例えば、Asphalt-7 (商標) および NFS-MW (商標)）は、25 以上の SUPD カウントを有し、一方他のゲーム（例えば、BeamNG-Buggy (商標) および Asphalt-8 (商標)）は、15 - 25 の範囲内で SUPD カウントを有し、それは、UI が使用するケースのうちのいくつかと同様である。ゲームは、UI が使用するケースとは違って、1 より大きいそれらの BTPD カウントを有する可能性があり、それは単一のテクスチャルックアップのみを典型的に有する。

30

【0039】

[0041] 説明されたように、SUPD カウントは、変わり得る。重い作業負荷を要求するゲームに対し、SUPD カウントは、25 まで上がる見込みがある。SUPD が 15 より大きい場合、作業負荷は、ゲームの広いカテゴリを満たす。SUPD カウントが 15 と 25 の間である場合、ゲーミングモード検出器 412 は、それらが 1 より大きいか否かを決定するためにテクスチャアップデート (BTPD) をチェックし得る。これら 2 個の状態は、作業負荷がゲーミングの作業負荷であることの高い可能性に達するために、およびコンピューティングデバイス 410 をゲーミングモードに入れるために組み合わされ得る。言い換えれば、以下の状態が GLES の作業負荷の 5 個の連続するフレームのスライディングウィンドウのために満たされる場合、ゲーミングモード検出器 412 は、コンピューティングデバイス 410 をゲーミングモードに入れ得る：(SUPD >= 25) の場合、または (SUPD >= 15 かつ BTPD > 1.0) の場合）。

40

【0040】

B. ゲーミングモードに切り替える

[0042] 説明されたように、ゲーミングモード検出器 412 は、コンピューティングデバイス 410 をゲーミングモードおよび / またはノンゲーミングモードに入れ得る。いく

50

つかの実施形態では、コンピューティングデバイス 410 をノンゲーミングモードからゲーミングモードに切り替えることは、CPU_FMax を低減することによって CPU 416 の CPU 周波数を低減することを含む。このようにして、ゲーミングの作業負荷が処理のために GPU 418 行きである間、コンピューティングデバイス 410 のパフォーマンスがおよそ同じままである間、電力消費は、低減され得る。したがって、ユーザは、CPU 416 の CPU 周波数が、それが少しの重い GPU の作業負荷も処理する必要がないため低減された後で、パフォーマンスのあらゆる欠如に気づかないことがある。CPU 416 は、2.15GHz の CPU_FMax で典型的に動作し得る。例において、ゲーミングモード検出器 412 が、コンピューティングデバイス 410 をゲーミングモードに切り替えるとき、ゲーミングモード検出器 412 は、CPU_FMax を 2.15GHz から 1.49GHz と 1.72GHz 間の範囲に低減する。
10

【0041】

[0043] 別の例において、CPU 416 の CPU_FMax は、増加している。ゲーミングモード検出器 412 は、コンピューティングデバイス 410 が処理のために GPU 418 行きであるゲーミングの作業負荷をもはや有さないことを検出し得る。CPU 416 の CPU 周波数は、例えば、CPU_FMax の制限を取り除くことによって増加し得る。別の例において、ゲーミングモード検出器 412 が、コンピューティングデバイス 410 をノンゲーミングモードに切り替えるとき、それは、もとのデフォルトの 2.15GHz に CPU_FMax を（例えば、1.49GHz と 1.72GHz 間の範囲から）増加させる。
20

【0042】

[0044] CPU 周波数最大値は、様々な技法を使用して低減され得る。例において、オペレーティングシステム 414 は、ファイルシステムノード（例えば、「/sys/devices/system/cpu/cpu0/cpufreq/scaling_max_freq」ノード）をエクスポーズする（exposes）カーネルを含む。オペレーティングシステム 414 は、例えば、Linux（商標）オペレーティングシステムであり得る。いくつかの実施形態において、ゲーミングモード検出器 412 は、新しい CPU_FMax をファイルシステムノードに動的に書き込む。ゲーミングモード検出器 412 がコンピューティングデバイス 410 をノンゲーミングモードからゲーミングモードに切り替えるとき、ゲーミングモード検出器 412 は、現在の CPU_FMax 未満の新しい CPU_FMax をノードに書き込む。ゲーミングモード検出器 412 がコンピューティングデバイス 410 をゲーミングモードからノンゲーミングモードに切り替えるとき、ゲーミングモード検出器 412 は、現在の CPU_FMax より大きい新しい CPU_FMax をノードに書き込む。いくつかの実施形態において、ゲーミングモード検出器 412 は、CPU_FMax それ自体を修正するためにプログラムされ、CPU 416 に新しい CPU_FMax を越えないように命令する。
30

【0043】

[0045] 上記に説明され、ここでさらに強調されるように、図 1-2 は、単に例であり、それらは、特許請求の範囲を過度に限定するべきでない。

【0044】

I I I . 例示的な方法

[0046] 図 3 は、いくつかの実施形態にしたがって、少なくとも 1 つの CPU の CPU 周波数をスロットリングする例示的な方法 900 を例示する流れ図である。方法 900 は、限定していると意味されず、他のアプリケーションで使用され得る。
40

【0045】

[0047] ブロック 905において、レンダリングのフレームごとのグラフィックスライブラリドローコールごとにされたグラフィックスライブラリコールの平均の量は、トラッキングされる。例において、このことは、ゲーミングモード検出器 412 によって行われる。ブロック 910において、CPU を含むコンピューティングデバイス上のゲーミングの作業負荷は、レンダリングのフレームごとのグラフィックスライブラリドローコール
50

ごとになされたグラフィックスライブラリコールのトラッキングされた平均の量に基づいて検出される。例において、ゲーミングモード検出器 412 は、C P U 416 を含むコンピューティングデバイス 410 上のゲーミングの作業負荷のためにこの検出を行う。ブロック 920 において、コンピューティングデバイスは、ゲーミングモードに切り替えられ、ここで、コンピューティングデバイスをゲーミングモードに切り替えることは、コンピューティングデバイス上で実行している C P U の C P U 周波数最大値を低減することを含む。例において、ゲーミングモード検出器 412 は、切り替えることをを行い、それは、コンピューティングデバイス 410 上で実行している C P U 416 の C P U 周波数最大値を低減することを含む。

【0046】

10

[0048] 追加のプロセスが上記で説明されたブロック 905 - 920 の前、ブロック 905 - 920 の間、またはブロック 905 - 920 の後で行われ得ることも理解される。本明細書で説明された方法 900 のブロックの 1 つ以上が、所望に応じて異なるシーケンスで省略され、組み合わされ、または行われ得ることも理解される。ある実施形態において、ブロック 905 - 920 は、C P U 416 の任意の数の C P U コアのために行われ得る。

【0047】

IV. 例示的なコンピューティングシステム

[0049] 図 4 は、本明細書で説明された実施形態のうちの任意のものをインプリメントすることに適している例示的なコンピュータシステム 1000 のブロック図である。様々なインプリメンテーションにおいて、コンピュータシステム 1000 は、コンピューティングデバイス 410 であり得る。コンピュータシステム 1000 は、1 つ以上のプロセッサを含み得る。コンピュータシステム 1000 は、フロッピー（登録商標）ディスク、フレキシブルディスク、ハードディスク、磁気テープ、任意の他の磁気媒体、C D - R O M、任意の他の光媒体、R A M、P R O M、E P R O M、F L A S H - E P R O M、任意の他のメモリチップまたはカートリッジ、および / またはプロセッサまたはコンピュータが読み取るために適合される任意の他の媒体を含むグループから各々選択される 1 つ以上の記憶デバイスをさらに含み得る。1 つ以上の記憶デバイスは、コンピュータネットワーク（示されていない）を使用してクライアントまたはサーバと結合される 1 つ以上のコンピューティングデバイスおよび / またはコンピュータプログラム（例えば、クライアント）が利用可能であり得る記憶された情報を含み得る。コンピュータネットワークは、L A N、W A N、インターネット、インターネット、クラウド、ならびに / あるいはシステムの中でコンピューティングデバイスおよび / またはコンピュータプログラムを相互に接続する能力があるそれらのネットワークの任意の組み合わせを含むネットワークの任意のタイプであり得る。

20

【0048】

30

[0050] コンピュータシステム 1000 は、バス 1002、またはコンピュータシステム 1000 の様々なコンポーネント間で情報データ、信号、および情報を通信するための他の通信メカニズムを含む。コンポーネントは、例えば、キーパッド / キーボードからキーを選択すること、または 1 つ以上のボタンまたはリンクを選択すること等のユーザアクション（user actions）を処理するためのインプット / アウトプット（I / O）コンポーネント 1004 を含み、対応する信号をバス 1002 に送る。I / O コンポーネント 1004 は、例えば、ディスプレイ 1011 等のアウトプットコンポーネント、および例えば、カーソルコントロール 1013（例えば、キーボード、キーパッド、マウス等）のようなインプットコントロールも含み得る。

40

【0049】

[0051] オーディオ I / O コンポーネント 1005 は、オーディオ信号を情報信号にコンバートすることによって、ユーザが情報をインプットするために音声を使用することを可能にすることも含まれ得る。オーディオ I / O コンポーネント 1005 は、ユーザが音声を聞くことを可能にし得る。トランシーバまたはネットワークインターフェース 1006

50

は、通信リンク 1018 を介してコンピュータシステム 1000 と他のデバイス間で信号をネットワークに送信および受信する。ある実施形態において、送信は、ワイヤレスであるが、他の送信媒体および方法も適切であり得る。マイクロコントローラ、デジタル信号プロセッサ (DSP)、または他の処理コンポーネントであり得るプロセッサ 416 は、例えば、コンピュータシステム 1000 のディスプレイ 1011 上のディスプレイのための、または通信リンク 1018 を介する他のデバイスへの送信のため等のこれらの様々な信号を処理する。ゲーミングモード検出器 412 は、プロセッサ 416 で実行され得る。プロセッサ 416 は、他のデバイスへの、例えば、クッキーまたは IP アドレス等の情報の送信もコントロールし得る。

【0050】

10

[0052] コンピュータシステム 1000 はまた、GPU 418 を含む。プロセッサ 416 は、バス 1002 を介して、コマンドのストリーム (stream of commands) を GPU 418 に送ることができ、レンダリングのフレームごとのグラフィックスライブラリドローコールごとになされたグラフィックスライブラリコールの平均の量およびテクスチャコールの平均の量をトラッキングし得る。

【0051】

20

[0053] コンピュータシステム 1000 のコンポーネントはまた、システムメモリコンポーネント 1014 (例えば、RAM)、静的な記憶装置コンポーネント (static storage component) 1016 (例えば、ROM)、および / またはコンピュータ可読媒体 1017 を含む。コンピュータシステム 1000 は、プロセッサ 1012 によって特定のオペレーションを、およびシステムメモリコンポーネント 1014 に包含される命令の 1 つ以上のシーケンスを実行することによって他のコンポーネントを行う。ロジックは、コンピュータ可読媒体 1017 で符号化でき、それは、実行のためのプロセッサ 1012 への命令を提供することに関与する任意の媒体を指し得る。そのような媒体は、不揮発性媒体、揮発性媒体、および送信媒体を含むがそれらに限定されない、多くの形式を取り得る。様々な実施形態において、不揮発性媒体は、光、または磁気ディスク、または個体のドライブを含み、揮発性媒体は、例えば、システムメモリコンポーネント 1014 等の動的なメモリを含み、送信媒体は、バス 1002 を含むワイヤを含む、同軸ケーブル、銅線、および光ファイバーを含む。ある実施形態において、ロジックは、非一時的なコンピュータ可読媒体で符号化される。コンピュータ可読媒体 517 は、プロセッサ 512 によって、またはプロセッサ 512 に関連して使用される命令を包含する、記憶する、通信する、伝播する、またはトランスポートすることができる任意の装置であり得る。コンピュータ可読媒体 517 は、電子、磁気、光学、電磁気、赤外線、または半導体デバイスまたは伝播媒体、または任意の他のメモリチップまたはカートリッジ、あるいはコンピュータが読み取るために適合される任意の他の媒体であり得る。ある例において、送信媒体は、例えば、無線波、光学、および赤外線データ通信の間に生成されるそれら等の、音響波または光波の形式を取り得る。

30

【0052】

[0054] バス 1002 がプロセッサ 416 と GPU 418 間でデータを配信する経路であるように例示されるが、それは、限定していると意図されず、プロセッサ 416 と GPU 418 間でデータを配信するための異なるメカニズムを使用する他の実施形態が、本開示の範囲内である。

40

【0053】

[0055] 本開示の様々な実施形態において、本開示を実施するための命令シーケンスの実行が、コンピュータシステム 1000 によって行われ得る。本開示の様々な他の実施形態において、通信リンク 1018 によってネットワーク (例えば、LAN、WLAN、PTSN、および / または電気通信、モバイル、およびセルラ電話ネットワークを含む、様々な他のワイヤードまたはワイヤレスネットワーク等) に結合される複数のコンピュータシステム 1000 は、お互いとの連携で本開示を実施するための命令シーケンスを行い得る。

50

【0054】

[0056] 適用可能な場合、本開示によって提供される様々な実施形態は、ハードウェア、ソフトウェア、またはハードウェアおよびソフトウェアの組み合わせを使用してインプリメントされ得る。また、適用可能な場合、本明細書に記載された様々なハードウェアコンポーネントおよび／またはソフトウェアコンポーネントは、本開示の趣旨から逸脱することなく、ソフトウェア、ハードウェア、および／またはそれら両方を含む複合のコンポーネントに組み合わされ得る。適用可能な場合、本明細書に記載された様々なハードウェアコンポーネントおよび／またはソフトウェアコンポーネントは、本開示の趣旨から逸脱することなく、ソフトウェア、ハードウェア、またはそれら両方を含むサブコンポーネントに分けられ得る。加えて、適用可能な場合、ソフトウェアコンポーネントは、ハードウェアコンポーネントとしてインプリメントされることができ、逆の場合も同じであり得ることは、企図される。10

【0055】

[0057] 本開示にしたがってアプリケーションソフトウェアは、1つ以上のコンピュータ可読媒体上に記憶され得る。本明細書で識別されるアプリケーションソフトウェアは、ネットワーク化されているおよび／またはそうでない、1つ以上の汎用または専用コンピュータおよび／またはコンピュータシステムを使用してインプリメントされ得ることもまた企図される。適用可能な場合、本明細書で説明された様々なステップの順番は、本明細書で説明された特徴を提供するために、変更され得、複合のステップに組み合わされ得、および／またはサブステップに分割され得る。20

【0056】

[0058] 先の開示は、開示されたものに役立つ特定の分野またはまさにその形式に本開示を限定することは意図されない。そのようなものとして、本明細書に明示的に説明されたか暗示的に説明されたかに関わらず、本開示への様々な代替の実施形態および／または修正が開示を踏まえて見込まれることが企図される。本開示の範囲から逸脱することなく、形式において、詳細に、変更がなされ得る。ゆえに、本開示は、特許請求の範囲によつてのみ限定される。

以下に、本願出願の当初の特許請求の範囲に記載された発明を付記する。

[C1]

少なくとも1つの中央処理装置(CPU)のCPU周波数をスケーリングする方法であつて、

レンダリングのフレームごとのグラフィックスライブラリドローコールごとになされたグラフィックスライブラリコールの平均の量をトラッキングすることと、

CPUを含むコンピューティングデバイス上のゲーミングの作業負荷を、レンダリングのフレームごとのグラフィックスライブラリドローコールごとになされたグラフィックスライブラリコールの前記平均の量を前記トラッキングすることに基づいて、検出することと、

前記コンピューティングデバイスをゲーミングモードに切り替えることと、ここにおいて、前記コンピューティングデバイスを前記ゲーミングモードに前記切り替えることは、前記コンピューティングデバイス上で実行している前記CPUのCPU周波数最大値(FMax)を低減することを含む、

を備える、方法。

[C2]

N個の連続するフレームのスライディングウィンドウは、定義され、前記トラッキングすることは、レンダリングの前記N個の連続するフレームの各々ごとのグラフィックスライブラリドローコールごとになされたグラフィックスライブラリコールの平均の量をトラッキングすることを含む、C1に記載の方法。

[C3]

前記トラッキングすることの結果が閾値を満たさないとき、前記コンピューティングデバイスが、ゲーミングモードを処理していないことを検出することと、

10

20

30

40

50

前記コンピューティングデバイスをノンゲーミングモードに切り替えることと、ここにおいて、前記コンピューティングデバイスをノンゲーミングモードに前記切り替えることは、前記CPU FMaxを最大の許可された周波数にリセットすることを含む、
をさらに含む、C1に記載の方法。

[C4]

前記結果が25未満であるとき、前記結果は、前記閾値を満たさない、C3に記載の方法。

[C5]

レンダリングのフレームごとのグラフィックスライブラリドローコールごとになされたテクスチャコールの平均の量をトラッキングすることと、

10

前記CPUを含む前記コンピューティングデバイス上の前記ゲーミングの作業負荷を、
レンダリングのフレームごとのグラフィックスライブラリドローコールごとになされたテクスチャコールの平均の量を前記トラッキングすることに基づいて、検出することと、
をさらに含む、C3に記載の方法。

[C6]

M個の連続するフレームの第2のスライディングウィンドウは、定義され、テクスチャコールの平均の量を前記トラッキングすることは、レンダリングの前記M個の連続するフレームの各々のためのグラフィックスライブラリドローコールごとになされたテクスチャコールの前記平均の量をトラッキングすることを含む、C5に記載の方法。

20

[C7]

グラフィックスライブラリコールの平均の量を前記トラッキングすることの第1の結果が第2の閾値を満たし、かつテクスチャコールの平均の量を前記トラッキングすることの第2の結果が第3の閾値を満たすとき、前記ゲーミングの作業負荷は、前記コンピューティングデバイス上で検出される、C6に記載の方法。

[C8]

グラフィックスライブラリコールの平均の量を前記トラッキングすることの前記第1の結果が前記第2の閾値を満たさないとき、またはテクスチャコールの平均の量を前記トラッキングすることの前記第2の結果が前記第3の閾値を満たさないとき、前記コンピューティングデバイスが、ゲーミングの作業負荷を処理していないことを検出する、をさらに含む、C7に記載の方法。

30

[C9]

連続するフレームの第1のスライディングウィンドウは、5であり、かつ連続するフレームの前記第2のスライディングウィンドウは、5である、C8に記載の方法。

[C10]

前記第1の結果が15以上であるとき、前記第1の結果は、前記第2の閾値を満たし、前記第2の結果が1より大きいとき、前記第2の結果は、前記第3の閾値を満たす、C8に記載の方法。

[C11]

グラフィックスライブラリコールの平均の量を前記トラッキングすることは、N個の連続するフレームのあらゆるスライディングウィンドウのために、レンダリングの前記N個の連続するフレームの各々のためのグラフィックスライブラリドローコールごとになされたグラフィックスライブラリコールの前記平均の量をトラッキングすることを含む、C1に記載の方法。

40

[C12]

レンダリングのM個の連続するフレームの各々のためのグラフィックスライブラリドローコールごとになされたテクスチャコールの平均の量をトラッキングすること、
をさらに含む、C11に記載の方法。

[C13]

グラフィックスライブラリコールの平均の量を前記トラッキングすることの結果が第1の閾値を満たすとき、前記検出することは、前記コンピューティングデバイス上の前記ゲ

50

一ミングの作業負荷を検出することを含む、C 1 2 に記載の方法。

[C 1 4]

グラフィックスライブラリコールの平均の量を前記トラッキングすることの第 1 の結果が第 2 の閾値を満たし、かつテクスチャコールの平均の量を前記トラッキングすることの第 2 の結果が第 3 の閾値を満たすとき、前記検出することは、前記コンピューティングデバイス上の前記ゲーミングの作業負荷を検出することを含む、C 1 3 に記載の方法。

[C 1 5]

前記第 1 の結果が 2 5 以上であるとき、前記第 1 の結果は、前記第 1 の閾値を満たし、前記第 1 の結果が 1 5 以上であるとき、前記第 1 の結果は、前記第 2 の閾値を満たし、前記第 2 の結果が 1 より大きいとき、前記第 2 の結果は、前記第 3 の閾値を満たす、C 1 4 に記載の方法。

10

[C 1 6]

C P U F M a x を前記低減することは、前記 C P U の前記 C P U F M a x を 1 . 4 9 G H z と 1 . 7 2 G H z 間の範囲に制限することを含む、C 1 に記載の方法。

[C 1 7]

前記コンピューティングデバイスは、カーネルを含み、C P U F M a x を前記低減することは、前記カーネルの中で A P I コールを呼び出すことを含む、C 1 に記載の方法。

[C 1 8]

前記コンピューティングデバイスは、モバイルデバイスである、C 1 に記載の方法。

[C 1 9]

前記モバイルデバイスは、スマートフォン、タブレット、ラップトップ、または携帯情報端末である、C 1 8 に記載の方法。

20

[C 2 0]

少なくとも 1 つの中央処理装置 (C P U) の C P U 周波数をスケーリングするためのシステムであって、

C P U と、

レンダリングのフレームごとのグラフィックスライブラリドローコールごとになされたグラフィックスライブラリコールの平均の量をトラッキングし、前記 C P U を含むコンピューティングデバイス上のゲーミングの作業負荷を、グラフィックスライブラリコールの前記トラッキングされた平均の量の結果に基づいて、検出するゲーミングモード検出器と、ここにおいて、前記ゲーミングの作業負荷を検出することに応答して、前記ゲーミングモード検出器は、前記コンピューティングデバイスをゲーミングモードに切り替え、前記コンピューティングデバイス上で実行している前記 C P U の C P U F M a x を低減する

30

を備えるシステム。

[C 2 1]

前記ゲーミングモード検出器は、N 個のフレームのスライディングウィンドウを定義し、レンダリングの前記 N 個の連続するフレームの各々のためのグラフィックスライブラリドローコールごとになされたグラフィックスライブラリコールの前記平均の量をトラッキングし、ここにおいて、前記結果が閾値を満たすとき、前記ゲーミングモード検出器は、前記コンピューティングデバイス上の前記ゲーミングモードを検出する、C 2 0 に記載のシステム。

40

[C 2 2]

前記コンピューティングデバイスが前記ゲーミングモードにあることを決定され、かつ前記結果が前記閾値を満たさないとき、前記ゲーミングモード検出器は、前記コンピューティングデバイスをノンゲーミングモードに切り替える、C 2 1 に記載のシステム。

[C 2 3]

前記結果が 2 5 以上であるとき、前記結果は、前記閾値を満たす、C 2 1 に記載のシステム。

[C 2 4]

50

前記ゲーミングモード検出器は、レンダリングのM個の連続するフレームの各々のためのグラフィックスライブラリドローコールごとになされたテクスチャコールの平均の量をトラッキングし、ここにおいて、グラフィックスライブラリコールの前記平均の量をトラッキングすることの第1の結果が第2の閾値を満たし、かつテクスチャコールの前記平均の量をトラッキングすることの第2の結果が第3の閾値を満たすとき、前記ゲーミングモード検出器は、前記コンピューティングデバイス上の前記ゲーミングの作業負荷を検出する、C21に記載のシステム。

[C 2 5]

前記第1の結果が25以上であるとき、前記第1の結果は、前記第1の閾値を満たし、前記第1の結果が15以上であるとき、前記第1の結果は、前記第2の閾値を満たし、前記第2の結果が1より大きいとき、前記第2の結果は、前記第3の閾値を満たす、C24に記載のシステム。

10

[C 2 6]

前記ゲーミングモード検出器は、前記CPUの前記CPU_FMaxを前記CPUの最大の許可された周波数レベルにリセットし、前記CPUの前記CPU_FMaxを増加させる、C20に記載のシステム。

[C 2 7]

オペレーションを行うためのコンピュータ実行可能な命令を記憶したコンピュータ可読媒体であって、

20

レンダリングのフレームごとのグラフィックスライブラリドローコールごとになされたグラフィックスライブラリコールの平均の量をトラッキングすることと、

CPUを含むコンピューティングデバイス上のゲーミングの作業負荷を、レンダリングのフレームごとのグラフィックスライブラリドローコールごとになされたグラフィックスライブラリコールの平均の量を前記トラッキングすることに基づいて、検出することと、

前記コンピューティングデバイスをゲーミングモードに切り替えることと、ここにおいて、前記コンピューティングデバイスをゲーミングモードに前記切り替えることは、前記コンピューティングデバイス上で実行している前記CPUのCPU_FMaxを低減することを含む、

を備える、コンピュータ可読媒体。

[C 2 8]

30

前記オペレーションは、

N個の連続するフレームのスライディングウィンドウを定義することと、前記トラッキングすることは、レンダリングの前記N個の連続するフレームの各々のためのグラフィックスライブラリドローコールごとになされたグラフィックスライブラリコールの平均の量をトラッキングすることを含む、

グラフィックスライブラリドローコールごとになされたグラフィックスライブラリコールの前記平均の量の結果が閾値を満たすとき、前記コンピューティングデバイス上の前記ゲーミングの作業負荷を検出することと、

をさらに含む、C27に記載のコンピュータ可読媒体。

[C 2 9]

40

前記オペレーションは、

レンダリングのM個の連続するフレームの各々のためのグラフィックスライブラリドローコールごとになされたテクスチャコールの平均の量をトラッキングすることと、

グラフィックスライブラリドローコールごとになされたグラフィックスライブラリコールの前記平均の量の結果が第2の閾値を満たし、かつテクスチャコールの前記平均の量の結果が第3の閾値を満たすとき、前記コンピューティングデバイス上で前記ゲーミングの作業負荷を検出することと、前記第1の閾値が、前記第2の閾値とは異なる、

をさらに含む、C28に記載のコンピュータ可読媒体。

[C 3 0]

50

少なくとも1つの中央処理装置(CPU)のCPU周波数をスケーリングするための装

置であって、

レンダリングのフレームごとのグラフィックスライブラリドローコールごとになされた
グラフィックスライブラリコールの平均の量をトラッキングするための手段と、

CPUを含むコンピューティングデバイス上のゲーミングの作業負荷を検出するための
手段と、

前記コンピューティングデバイス上で実行している前記CPUのCPU FMaxを低
減するための手段と、

を備える、装置。

【図1】

図1

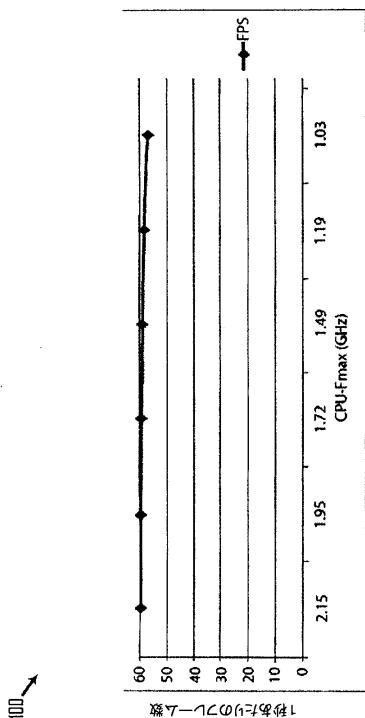


FIG.1

【図2】

図2

400

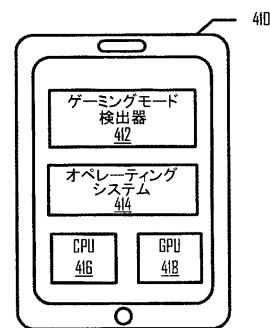


FIG.2

【図3】

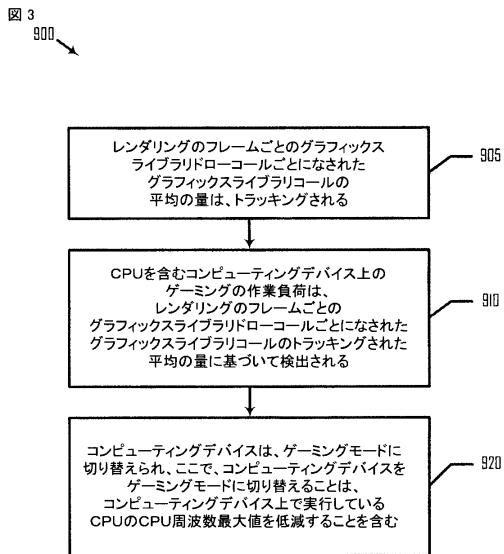


FIG. 3

【図4】

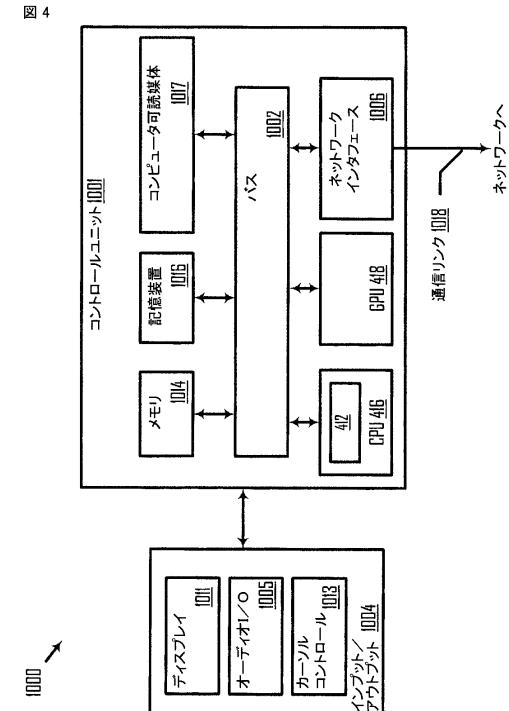


FIG. 4

フロントページの続き

(72)発明者 シャー、プレマル

アメリカ合衆国、カリフォルニア州 92121、サン・ディエゴ、モアハウス・ドライブ 57
75

(72)発明者 エリス、ブライアン

アメリカ合衆国、カリフォルニア州 92121、サン・ディエゴ、モアハウス・ドライブ 57
75

審査官 田川 泰宏

(56)参考文献 Dietrich, Benedikt , Time Series Characterization of Gaming Workload for Runtime Power Management , IEEE Transactions on Computers , 米国 , 2013年10月 8日 , vol. 64, No. 1 , pp. 260-273

Sheng Guo , Practical Game Performance Analysis Using Intel Graphics Performance Analyzer , 米国 , Intel Corporation , 2011年12月31日 , pp.1-22 , [検索日] : 2017/11/13 , URL , https://software.intel.com/sites/default/files/m1/o/c/e/6/26750-PracticalPerformanceAnalysis_GPA.pdf

(58)調査した分野(Int.Cl. , DB名)

G 06 F 1 / 04

G 06 T 15 / 00