

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第5529748号
(P5529748)

(45) 発行日 平成26年6月25日(2014.6.25)

(24) 登録日 平成26年4月25日(2014.4.25)

(51) Int.Cl.

F I

G06F 3/14 (2006.01)
 G09G 5/00 (2006.01)
 G06F 13/38 (2006.01)
 G06F 3/153 (2006.01)

G06F 3/14 310A
 G09G 5/00 555D
 G09G 5/00 510X
 G09G 5/00 550M
 G09G 5/00 510V

請求項の数 7 (全 31 頁) 最終頁に続く

(21) 出願番号 特願2010-536221 (P2010-536221)
 (86) (22) 出願日 平成20年12月1日(2008.12.1)
 (65) 公表番号 特表2011-509445 (P2011-509445A)
 (43) 公表日 平成23年3月24日(2011.3.24)
 (86) 国際出願番号 PCT/US2008/085160
 (87) 国際公開番号 W02009/073617
 (87) 国際公開日 平成21年6月11日(2009.6.11)
 審査請求日 平成23年11月30日(2011.11.30)
 (31) 優先権主張番号 11/987,559
 (32) 優先日 平成19年11月30日(2007.11.30)
 (33) 優先権主張国 米国(US)

(73) 特許権者 508301087
 エーティーアイ・テクノロジーズ・ユーエ
 ルシー
 ATI TECHNOLOGIES UL
 C
 カナダ、オンタリオ エル3ティー 7
 エックス6、マーカム、コマース バリー
 ドライブ イースト 1
 One Commerce Valley
 Drive East, Markha
 m, Ontario, L3T 7X6
 Canada
 (74) 代理人 100108833
 弁理士 早川 裕司

最終頁に続く

(54) 【発明の名称】 高速周辺相互接続バスにおけるビデオレンダリング

(57) 【特許請求の範囲】

【請求項 1】

プロセッサと、メモリと、ディスプレイにビデオフレームを提示するためのディスプレ
 イインタフェースおよび第1のグラフィックプロセッサを備える第1のグラフィックサブ
 システムと、前記プロセッサ、前記メモリおよび前記第1のグラフィックサブシステムと
 相互接続するインタフェース回路と、第2のグラフィックプロセッサを有するが専用のフ
 レームバッファメモリを有しない第2のグラフィックサブシステムと、周辺相互接続バス
 とを有するコンピューティングデバイスであって、前記周辺相互接続バスは、前記第2の
 グラフィックサブシステムへ提供され若しくは前記第2のグラフィックサブシステムから
 提供される全てのデータが前記インタフェース回路および前記周辺相互接続バス経由で提
 供されるように、前記第2のグラフィックサブシステムと前記インタフェース回路とを相
 互接続している、コンピューティングデバイスを動作させる方法であって、

前記ディスプレイインタフェースと相互接続している前記メモリ内にフレームバッファ
 を確立するステップと、

ビデオフレームを、前記周辺相互接続バス経由で、プリミティブごとに、前記フレーム
 バッファ内にレンダリングするように、前記第2のグラフィックプロセッサに対して、前
 記周辺相互接続バス経由で指示を提供するステップであって、前記ディスプレイに提示さ
 れる全てのフレームは、前記第2のグラフィックプロセッサによってレンダリングされて
 いるステップと、

前記フレームバッファからの前記フレームを、前記ディスプレイインタフェース経由で

ディスプレイに提示するステップとを含む、方法。

【請求項 2】

前記フレームバッファ内の前記フレームを前記ディスプレイに提示するように、前記ディスプレイインタフェースをプログラムするステップを更に含む、請求項 1 に記載の方法。

【請求項 3】

コンピューティングデバイスであって、

中央処理装置と、

前記プロセッサと通信するメモリと、

グラフィックメモリ内のフレームバッファと、

前記グラフィックメモリと相互接続されたディスプレイインタフェースであって、ディスプレイにビデオフレームを提示するためのディスプレイインタフェースを備える第 1 のグラフィックサブシステムと、

前記プロセッサ、前記メモリおよび前記第 1 のグラフィックサブシステムと相互接続するインタフェース回路と、

周辺拡張カードに形成された第 2 のグラフィックサブシステムであって、ローカルなフレームバッファを定義するメモリを有しない第 2 のグラフィックサブシステムと、

前記第 2 のグラフィックサブシステムと前記インタフェース回路とを相互接続する周辺相互接続バスであって、前記第 2 のグラフィックサブシステムへ提供され若しくは前記第 2 のグラフィックサブシステムから提供される全てのデータが、前記周辺相互接続バス経由で提供される、周辺相互接続バスとを備え、

前記第 2 のグラフィックサブシステムは、レンダリングの指示を受信し、前記周辺相互接続バス経由で画像をレンダリングするように動作し、

前記メモリは、前記ディスプレイに提示される全てのビデオフレームを、前記周辺相互接続バス経由で、プリミティブごとに、前記フレームバッファ内にレンダリングするように、前記第 2 のグラフィックサブシステムを動作させるコンピュータ実行可能命令を記憶している、コンピューティングデバイス。

【請求項 4】

前記ディスプレイインタフェースは周辺拡張インタフェースの一部を構成している、請求項 3 に記載のコンピューティングデバイス。

【請求項 5】

前記ディスプレイインタフェースは、前記中央処理装置を前記周辺相互接続バスと相互接続している周辺拡張インタフェース上の前記第 1 のグラフィックサブシステムの一部を構成している、請求項 3 に記載のコンピューティングデバイス。

【請求項 6】

グラフィックサブシステムであって、

グラフィックプロセッサと、周辺相互接続バスインタフェースとを備え、前記周辺相互接続バスインタフェースは、前記グラフィックサブシステムへ提供され若しくは前記グラフィックサブシステムから提供される全てのデータが、前記周辺相互接続バスインタフェース経由で送られるように、前記グラフィックサブシステムをホストコンピューティングデバイスの周辺相互接続バスと相互接続するためのものであり、前記ホストコンピューティングデバイスは、前記グラフィックメモリと相互接続されたディスプレイインタフェースであって、ディスプレイにビデオフレームを提示するためのディスプレイインタフェースを備え、前記グラフィックサブシステムは前記グラフィックサブシステムにローカルなフレームバッファを定義するメモリを持たず、

前記グラフィックサブシステムは、レンダリングの指示を受信し、前記ビデオフレームを、前記周辺相互接続バスインタフェース経由で、プリミティブごとに、前記ホストコンピューティングデバイスの前記グラフィックメモリのフレームバッファ内にレンダリングするように動作可能であり、

前記グラフィックプロセッサは、前記ディスプレイに提示される全てのフレームをレン

10

20

30

40

50

ダリングし、全てのグラフィックコマンドは、前記周辺相互接続バスインタフェースを介して前記ホストコンピューティングデバイスから受信される、グラフィックサブシステム。

【請求項 7】

コンピューティングデバイスを請求項 1 に記載の方法を実行するように動作させるコンピューティング実行可能命令を記憶しているコンピュータ可読媒体。

【発明の詳細な説明】

【技術分野】

【0001】

【関連出願への相互参照】

本願は、2005 年 5 月 24 日に出願の米国特許出願第 11 / 136 , 483 号、および 2006 年 5 月 30 日に出願の米国特許出願第 11 / 421 , 005 号の米国一部継続出願であり、そのいずれも参照によりここに援用する。

【0002】

本発明は、一般に、コンピューティングデバイスに関し、より詳細には、高速周辺バスによって相互接続されているグラフィックプロセッサを有するコンピューティングデバイスに関する。

【背景技術】

【0003】

近年、コンピュータハードウェアが次第に一体化されている。コンピュータの周辺機器は、中央処理装置と協働する一体化チップセットの一部として形成されることが多い。多くの場合、コンピュータの音声、グラフィックアダプタおよびインタフェースは、一体化チップセットの一部を構成している。このような一体化により、コンピューティングデバイスのコスト低減と小型化が実現される。

【発明の概要】

【発明が解決しようとする課題】

【0004】

同時に、グラフィックプロセッサは、改善された性能と特徴を提供するために常に進歩している。最先端のグラフィックプロセッサは、中央処理装置と同程度に複雑化している。このため、統合の傾向にもかかわらず、高性能のグラフィックプロセッサは、未だに周辺機器カードに形成されることが一般的である。比較すると、集積されたグラフィックプロセッサは比較的単純であり、高性能の周辺機器カードに形成されるグラフィックプロセッサの特徴または性能を提供しない。一体型グラフィック部品は、アドオンの周辺機器カードが存在している場合には、通常はあっても冗長であり、このため無効にされる。

【0005】

同様に、集積化、外部を問わず、グラフィックアダプタをアップグレードすると、多くの場合、非力なアダプタは役に立たなくなり、廃棄されるか無効にされる。

【0006】

したがって、部品の冗長性を低減する必要性が依然として存在する。

【課題を解決するための手段】

【0007】

本発明の一態様によれば、1つのグラフィックプロセッサによって生成されるグラフィックが、高速拡張 / 相互接続バスを介してフレームバッファに転送される。フレームバッファからのレンダリングされたフレームが、フレームバッファと通信しているディスプレイインタフェース経由でディスプレイで提示される。

【0008】

別の既存の（例えば一体化）グラフィックアダプタ / サブシステムのディスプレイインタフェースは、相互接続されているディスプレイにレンダリングされたフレームを提示するために使用することができ、利便性が高い。

【0009】

10

20

30

40

50

このようにして、グラフィックプロセッサは、ディスプレイインタフェースを有さず、場合によってはフレームバッファさえも備えている必要がないグラフィックアダプタの一部を構成しうる。同様に、グラフィックプロセッサを、最新のグラフィックプロセッサで置き換えることができる。

【0010】

グラフィックプロセッサは、周辺拡張カードに包含されるグラフィックアダプタの一部を構成しうる。

【0011】

一実施形態では、コンピューティングデバイスは、第1のグラフィックアダプタおよび第2のグラフィックアダプタを有しうる。ディスプレイに表示させる画像が、前記第2のグラフィックアダプタによってアクセス可能なメモリから転送される。前記第2のグラフィックアダプタの前記ディスプレイインタフェースが、前記第2のグラフィックアダプタの前記メモリ内の画像を、前記複数のディスプレイの少なくとも1つで提示するために使用される。このようにして、前記第2のアダプタのポートと共に、前記ディスプレイインタフェースを形成しているデバイスの電子回路が使用されうる。レンダリングされたグラフィックは、フレーム単位で転送されるか、グラフィックプリミティブ (graphics primitive) 単位で転送されうる。後者の場合、グラフィックが、前記第2のグラフィックアダプタによってアクセス可能なメモリ内に、効率的に直接レンダリングされる。このメモリは、システムメモリでもアダプタメモリでもよい。

【0012】

別の実施形態によれば、前記第1のグラフィックプロセッサが、前記高速相互接続バスを介して転送されるフレームをレンダリングしている間、前記第2のアダプタのグラフィックプロセッサが無効にされるか、低電力モードに設定される。

【0013】

本発明の別の態様によれば、第1のグラフィックアダプタの一部を構成しているグラフィックプロセッサが、第1の画像を第1のディスプレイに、第2の画像を第2のディスプレイに、それぞれ描画するように動作される。前記第1のディスプレイは、前記第1のグラフィックアダプタの第1のディスプレイインタフェースと相互接続されうる。前記第2のディスプレイは、第2のグラフィックアダプタの第2のディスプレイインタフェースと相互接続されうる。前記方法は、前記第1のグラフィックアダプタのメモリ内に前記第1の画像と前記第2の画像とをレンダリングするステップと、前記第2の画像を前記第1のアダプタの前記メモリから前記第2のグラフィックアダプタのフレームバッファに転送するステップと、前記第2のグラフィックアダプタの前記第2のディスプレイインタフェースを使用して、前記第2のグラフィックアダプタの前記フレームバッファから前記第2のディスプレイに前記第2の画像を表示するステップと、を有する。

【0014】

本発明の別の態様によれば、コンピューティングデバイスは、プロセッサと、前記プロセッサと通信しているコンピュータ可読メモリと、第1のグラフィックアダプタと、前記第1のグラフィックアダプタと通信している第1のアダプタメモリと、第1のディスプレイで画像を提示するための第1のディスプレイインタフェースと、第2のグラフィックアダプタと、第2のディスプレイで画像を提示するための第2のディスプレイインタフェースと、前記第1のグラフィックアダプタと前記第2のグラフィックアダプタとを相互接続している高速バスと、を有する。プログラムコードが、前記コンピュータ可読メモリに記憶され、前記プロセッサによって実行可能である。前記プログラムコードは、実行されると、前記第1のグラフィックアダプタに、前記第1のアダプタメモリ内に第1の画像および第2の画像をレンダリングさせ、前記第2の画像を前記第1のアダプタメモリから前記第2のグラフィックアダプタのフレームバッファに転送させ、前記第2のグラフィックアダプタの前記フレームバッファから前記第2のディスプレイに、前記第2の画像を表示するように、前記第2のグラフィックアダプタの前記第2のディスプレイインタフェースをプログラムさせる。

【 0 0 1 5 】

更に別の態様によれば、前記第 1 のアダプタは、前記第 2 のアダプタによってアクセス可能な前記システムメモリに第 1 の画像を直接レンダリングする。前記第 2 のアダプタは、前記第 2 の画像を同じメモリにレンダリングする。前記第 2 のアダプタは、前記システムメモリから両方の画像を表示する。

【 0 0 1 6 】

本発明の別の態様によれば、コンピューティングデバイスを動作させる方法が提供される。前記コンピューティングデバイスは、プロセッサ、前記プロセッサと通信しているメモリ、前記プロセッサをグラフィックプロセッサと相互接続している周辺相互接続バス、およびディスプレイインタフェースを有する。前記方法は、前記メモリ内にフレームバッファを確立するステップと、ビデオフレームをレンダリングするように前記グラフィックプロセッサに指示するステップと、前記ビデオフレームを前記周辺相互接続バス経由で前記フレームバッファに転送するステップと、前記フレームバッファからの前記フレームを、前記ディスプレイインタフェース経由でディスプレイで提示するステップと、を有する。

10

【 0 0 1 7 】

本発明の更に別の態様によれば、コンピューティングデバイスが提供される。前記コンピューティングデバイスは、中央処理装置と、前記中央処理装置と通信しているメモリと、前記中央処理装置をグラフィックプロセッサと相互接続している周辺相互接続バスと、ディスプレイインタフェースと、前記メモリに記憶されたコンピュータ実行可能命令と、を有し、前記命令は、コンピューティングデバイスを、前記グラフィックプロセッサに、ビデオフレームをレンダリングさせ、前記ビデオフレームを前記周辺相互接続バス経由で前記フレームバッファに転送し、前記フレームバッファからのレンダリングされた前記フレームを、前記ディスプレイインタフェース経由でディスプレイで提示するように適合させる。

20

【 0 0 1 8 】

本発明の別の態様によれば、コンピューティングデバイスが提供される。前記コンピューティングデバイスは、中央処理装置と、前記プロセッサと通信しているメモリと、第 1 のフレームバッファと、前記プロセッサをグラフィックサブシステムと相互接続している周辺相互接続バスであって、前記グラフィックサブシステムは、前記グラフィックサブシステムにローカルなフレームバッファを定義するメモリを有さない周辺相互接続バスと、前記メモリと通信しているディスプレイインタフェースと、を有する。前記メモリは、前記グラフィックプロセッサを、ビデオフレームをレンダリングし、前記ビデオフレームを前記周辺相互接続バス経由で前記第 1 のフレームバッファに転送するように適合させるコンピュータ実行可能命令を記憶している。

30

【 0 0 1 9 】

本発明の別の態様によれば、グラフィックサブシステムが提供され、前記グラフィックサブシステムは、グラフィックプロセッサと、前記グラフィックサブシステムをホストプロセッサと相互接続するための周辺相互接続バスインタフェースと、を有し、前記グラフィックサブシステムは、前記グラフィックサブシステムにローカルなフレームバッファを定義するメモリを有さず、前記グラフィックサブシステムは、ビデオフレームをレンダリングし、前記ビデオフレームを前記周辺相互接続バス経由でフレームバッファに転送するように動作可能である。

40

【 0 0 2 0 】

本発明の別の態様によれば、第 1 の画像を第 1 のディスプレイに、第 2 の画像を第 2 のディスプレイに、それぞれ描画するように、第 1 のグラフィックアダプタを動作させる方法が提供される。前記第 2 のディスプレイは、第 2 のグラフィックアダプタのディスプレイインタフェースと相互接続されている。前記方法は、前記第 1 のグラフィックプロセッサと通信しているメモリ内に前記第 1 の画像と前記第 2 の画像とをレンダリングするステップと、前記第 2 の画像を前記第 2 のグラフィックアダプタのフレームバッファに転送す

50

るステップと、前記第２のグラフィックアダプタの前記フレームバッファから前記第２のディスプレイに、前記第２の画像を表示するように、前記第２のグラフィックアダプタの前記第２のディスプレイインタフェースをプログラムするステップと、を有する。

【図面の簡単な説明】

【００２１】

【図１】本発明の実施形態の例示的なコンピューティングデバイスの簡略ブロック図。

【図２】図１のコンピューティングデバイスのソフトウェアの簡略論理ブロック図。

【図３】図１のコンピューティングデバイスの一部の簡略ブロック図。

【図４】図１のコンピューティングデバイスの一部の別のモードブロック図。

【図５Ａ】本発明の実施形態の例示的な、図１のコンピューティングデバイスで実行されるステップを示すフローチャート。 10

【図５Ｂ】図５Ａのステップの、図１のデバイスの部品への効果を示す図。

【図６Ａ】本発明の実施形態の例示的な、図１のコンピューティングデバイスで実行されるステップを示すフローチャート。

【図６Ｂ】図６Ａのステップの、図１のデバイスの部品への効果を示す図。

【図６Ｃ】図６Ａのステップの、図１のデバイスの部品への効果を示す図。

【図７】本発明の別の実施形態の例示的な、図１のコンピューティングデバイスで実行されるステップを示すフローチャート。

【図８】本発明の別の実施形態の例示的な、図１のコンピューティングデバイスで実行されるステップを示すフローチャート。 20

【図９】本発明の別の実施形態の例示的なコンピューティングデバイスの一部の別の部分簡略ブロック図。

【図１０】本発明の実施形態の例示的な、図９のデバイスのソフトウェアによって実行されるステップを示すフローチャート。

【図１１Ａ】図９のデバイスの動作を示す簡略ブロック図。

【図１１Ｂ】図９のデバイスの動作を示す簡略ブロック図。

【図１２】本発明の別の実施形態の例示的なコンピューティングデバイスの一部の別の簡略モードブロック図。

【発明を実施するための形態】

【００２２】 30

本発明のほかの態様および特徴は、添付の図面を参照して、以下の本発明の特定の実施形態の説明を検討すれば、当業者にとって明らかとなるであろう。

【００２３】

図面において、本発明の実施形態を、例示のみを目的として図示する。

【００２４】

図１は、本発明の実施形態の例示的なコンピューティングデバイスの簡略モードブロック図である。コンピューティングデバイス１０は、従来のインテル×８６アーキテクチャをベースとしている。しかし、明らかなように、本発明は、Power PC、AMD×８６または他のアーキテクチャを有するコンピューティングデバイスにおいても容易に実施することができる。 40

【００２５】

コンピューティングデバイス１０は、プロセッサ１２を有し、プロセッサ１２は一体型インタフェース回路１４、２０を介してシステムメモリ１６および周辺機器と相互接続されている。

【００２６】

例示的なコンピューティングデバイス１０のプロセッサ１２は、従来の中央処理ユニットであり、例えばインテル（登録商標）×８６ファミリのマイクロプロセッサと互換のマイクロプロセッサなどである。

【００２７】

一体型インタフェース回路１４と２０は、それぞれ、ノースブリッジ、サウスブリッジ 50

と呼ばれることが多く、プロセッサ12に対して、周辺機器およびメモリにインタフェースを行う。図に示すように、ノースブリッジ14は、プロセッサ12、メモリ16、高速相互接続バス22経由で複数の拡張スロット18、およびサウスブリッジ20を相互接続している。サウスブリッジ20は、低速の周辺機器およびインターコネクタを相互接続している。これには、例えば、統合IDE/SATAポート24経由のディスクドライブ、統合ユニバーサルシリアルバス(USB)ポート26経由のユニバーサルシリアルバスデバイスや、例えば、公知のPCIまたはISA規格に準拠した、低速相互接続バス28経由の他の周辺機器がある。サウスブリッジ20は、一体化音声部品30、ネットワークインタフェース(図示せず)などを含んでもよい。

【0028】

10

例示的な相互接続バス22は、ギガバイト/秒の範囲の帯域幅を有するPCIエクスプレス(PCIe)バスであり、この帯域幅でのデータ転送の読み出しおよび書き込みに対応している。例えば、現在のPCIeバスの速度により、最大266MHzまたは533MHzの速度での双方向データ転送が可能となり、最大2.1ギガバイト/秒または4.2ギガバイト/秒のデータ速度が可能となる。

【0029】

図の実施形態では、ノースブリッジ14は、コンピューティングデバイス10を、モニタ、LCDパネル、テレビなどの形態の第1のディスプレイ32と相互接続するために適した一体型グラフィックアダプタ34を更に備える。

【0030】

20

下記で詳細に説明するように、相互接続バス22経由で相互接続される追加のグラフィックアダプタ52が、例えば、相互接続バス22上の拡張スロット18内の周辺拡張カード50の一部を構成しており、更にコンピューティングデバイス10の一部も構成している。

【0031】

図に示した実施形態では、コンピューティングデバイス10は、メモリ16に記憶されているソフトウェアを実行する。図2に示すように、例示的なソフトウェア100には、オペレーションシステム102、グラフィックライブラリ104およびアプリケーションソフトウェア106が含まれ、メモリ16に記憶されている。代表的なオペレーティングシステムとしては、Windows Vista、Windows XP、Windows NT 4.0、Windows ME、Windows 98、Windows 2000、Windows 95またはLinuxオペレーティングシステムが挙げられる。代表的なグラフィックライブラリとしては、Microsoft DirectXライブラリやOpenGLライブラリが挙げられる。コンピュータメモリ16(図1)と、相互接続されたディスクドライブ(図示せず)とは、ランダムアクセスメモリ、リードオンリーメモリおよびディスクストレージメモリの適した組み合わせを有し、これは、本発明の実施形態の例示的な方法によりデバイス10を適合させる、オペレーティングシステムとグラフィックアダプタドライバプログラムを記憶および実行するためにデバイス10によって使用される。例示的なソフトウェア100(図2)は、例えば、リードオンリーメモリに記憶されるか、またはIDE/SATAポート24と通信しているディスクドライブ等の外部周辺機器からロードされうる。コンピュータ可読媒体27は、光学記憶媒体、磁気ディスク、テープ、ROMカートリッジなどであってもよい。

【0032】

グラフィックライブラリ104またはオペレーティングシステム102には、グラフィックアダプタ34、52との下位レベルの通信に使用されるグラフィックドライバ108が更に含まれる。ソフトウェアは階層構造を有しており、上位レベルのレイヤが、特定の機能を提供するために下位レイヤを使用する。アプリケーションは、二次元または三次元のグラフィックをレンダリングするために、オペレーティングシステム102とグラフィックライブラリ104を使用しうる。この文脈において、「レンダリング」とは、描画またはその他の表現のためにグラフィック画像を生成することを意味し、例えば、ポリゴン

50

のレンダリング、レイトレイシング、画像のデコーディング、ライン描画などが含まれる。明らかなように、本発明の実施形態の例示的なソフトウェアは、グラフィックライブラリ104および/またはドライバ108の一部を構成しうる。ここに示す実施形態では、例示的なソフトウェアは、OpenGLライブラリおよび関連するドライバの一部を構成しうる。OpenGLについては、「The Official Guide to Learning OpenGL、第1.1版」に詳しく記載されており、その内容をここに参照により援用する。しかし、当業者は、本発明の実施形態の例示的な方法は、Microsoft Direct3Dライブラリ、カスタムライブラリまたはルーチンを使用しないアプリケーション(ゲームなど)、あるいは他の任意の三次元アプリケーションまたはライブラリの一部を構成しうることを認めるであろう。

10

【0033】

一体型グラフィックアダプタ34を示す例示的なコンピューティングデバイス10(および特にノースブリッジ14)の一部の更に詳しい簡略ブロック図が、図3に示される。図に示すように、ノースブリッジ14は、高速バスインタフェース36およびメモリコントローラ38を有し、ブリッジ14を、相互接続バス22とシステムメモリ16とを相互接続している。ノースブリッジ14は、一体型グラフィックアダプタ34を更に有し、一体型グラフィックアダプタ34は、グラフィックプロセッサ40、コマンドプロセッサ42およびディスプレイインタフェース44を有する。

【0034】

グラフィックプロセッサ40は、システムメモリ16内に割り当てられるフレームバッファ46に、三次元画像をレンダリング可能な三次元(3D)グラフィックプロセッサの形態をとってもよく、グラフィックの高速レンダリングのために複数のパイプラインとパイプラインステージを有しうる。

20

【0035】

通常、シングルモニタ動作では、コマンドが、コマンドプロセッサ42によって従来の方法でグラフィックプロセッサ40に渡される。例えば、コマンドプロセッサ42は、メモリ16内にコマンドキュー47を定義するために使用するレジスタを備えうる。アプリケーションソフトウェア106またはオペレーティングシステム102は、ライブラリ104を使用してグラフィック画像をレンダリングする。ドライバソフトウェア108は、グラフィックプロセッサ固有のコマンドを生成して、コマンドキュー47にこのコマンドをキューイングする一方、キューの終点を変更するためにコマンドプロセッサ42のレジスタも更新する。グラフィックプロセッサ42は、とりわけ、フレームバッファ46にレンダリング画像をレンダリングするために、キューイングされたコマンドを、非同期でパイプライン方式で処理する。グラフィックプロセッサ40がキュー内の次のコマンドを処理可能な場合、コマンドプロセッサ42に信号を供給し、これを受けて、コマンドプロセッサ42は、キュー47にある次のコマンドを取り出し、コマンドプロセッサ42のレジスタのキューポインタを進める。

30

【0036】

ディスプレイインタフェース44は、メモリコントローラ38経由でフレームバッファ46をサンプリングし、テレビ、モニタなどの形態のディスプレイ32(図1)等のビデオデバイスに1つ以上のビデオ画像を表示するために、VGAポート、コンポジットビデオポート、DVIポート、HDMI、Display Portなどの形態で、1つ以上のビデオ出力ポート48に画像を提示する。このようにして、フレームバッファ46内の、グラフィックプロセッサ40によってレンダリングされる画像が提示されうる。

40

【0037】

理解されるように、ディスプレイインタフェース44は、バッファ内のデータをディスプレイ装置に表示するために変換するのに適したインタフェースであれば、どのようなものでもよい。例えば、ディスプレイインタフェース44は、RAMDACの形態をとりうる。ディスプレイインタフェース44は、通常、例えば複数のレジスタによってプログラム可能であり、これにより、ドライバソフトウェア108または同様のソフトウェアまた

50

はグラフィックプロセッサ 40 が、そのディスプレイ出力を提示するために、メモリ 16 内に先頭アドレスを設定可能となる。同様に、インタフェース 44 のピクセル深度（すなわちピクセル当たりのビット数）とスクリーン解像度も、通常はプログラム可能である。

【0038】

理解されるように、メモリ 16 内に 1 つのフレームバッファ 46 を割り当てる代わりに、2 つのフレームバッファが割り当てられてもよい。この 2 つのバッファは、レンダリングされるグラフィックをよりスムーズに表現するために交互に使用されうる。ディスプレイインタフェース 44 は、いつの時点でも、2 つのバッファの一方の内容しか表示しない。表示中のバッファは、通常「フロントバッファ」と呼ばれる。グラフィックプロセッサ 40 は、ディスプレイインタフェース 44 によって現在表示されていないフレームバッファである、いわゆる「バックバッファ」に表示するために画像をレンダリングする。フレームのレンダリングが完了すると、フロントバッファとバックバッファが反転されて、ディスプレイインタフェース 44 が、バックバッファを提示するようにプログラムされる。このようにして、グラフィックプロセッサ 40 によってレンダリングされる逐次的なフレーム間の遷移がスムーズに表示されうる。

【0039】

特に、アダプタ 34 によって使用されるレジスタおよびフレームバッファメモリ 46 は全て、グローバルなメモリ空間内のアドレスにマップされている。

【0040】

しかし、本発明の例示的なコンピュータデバイス 10 は、図 1 に示すように、ノースブリッジ 14 と通信している高速相互接続バス 22 上の拡張スロット経由でデバイス 10 に挿入されうる周辺拡張カード 50 に形成されているグラフィックアダプタ 52 を更に有する。ノースブリッジ 14 に相互接続されているグラフィックアダプタ 52 が、図 4 に模式的に概略的に示されている。図に示すように、グラフィックアダプタ 52 は、グラフィックプロセッサ 54、ビデオメモリ 56、コマンドプロセッサ 58、メモリコントローラ 60、およびディスプレイインタフェース 62 を有する。1 つ以上のポート 64、66 により、ディスプレイ 68（図 1）などの 1 台以上のディスプレイ装置とアダプタ 52 を相互接続することができる。

【0041】

コマンドプロセッサ 58 はコマンドプロセッサ 42 と、ディスプレイインタフェース 62 はディスプレイインタフェース 44 と、それぞれ全く同様に機能する。グラフィックプロセッサ 54 は、通常動作モードにおいて、グラフィックプロセッサ 40 と全く同様に動作する。グラフィックアダプタ 52 は、自身のメモリ 56 と、メモリ 56 およびシステムメモリ 16 へのアクセスを提供するメモリコントローラ 60 とを有する。ローカルメモリ 56 は、なかでも、グラフィックプロセッサ 54 によって使用され、フレームバッファとして機能しうる、グラフィックアダプタ 52 用の 1 つ以上のバッファ 70 を保持するために使用される。グラフィックプロセッサ 54 用のコマンドキュー 57 も、システムメモリ 16 内に作成される。

【0042】

この場合も、ローカルメモリ 56 は、メモリコントローラ 60 を介して、グローバルシステムメモリ空間内のアドレスでアドレッシング可能である。同様に、コマンドプロセッサ 54 およびディスプレイインタフェース 62 によって使用されるレジスタは、グローバルアドレス空間内のアドレスでアドレッシング可能である。上で説明したように、アダプタ 34 によって使用されるレジスタおよびフレームバッファ 46 はグローバルメモリ空間にマップされる。このため、コマンドプロセッサ 58 とグラフィックプロセッサ 54 は、ディスプレイインタフェース 44 とコマンドプロセッサ 42 によって使用されるフレームバッファ 46 およびレジスタに書き込むことができる。

【0043】

また、グラフィックプロセッサ 54 は、メモリコントローラ 60 と連動して、メモリアドレスにおいて、システムメモリ 16 およびアダプタメモリ 56 との間でデータを転送す

10

20

30

40

50

るために、ブロック転送操作を実行するように動作可能である。

【 0 0 4 4 】

経済的な一体化部品を提供するために、一体型グラフィックアダプタ 3 4 は機能が限定されている。例えば、一体型グラフィックコントローラの解像度、三次元グラフィック機能などは、比較的制限されている。このため、一体型グラフィックアダプタ 3 4 は、通常、グラフィックアダプタ 5 2 よりもグラフィック処理能力が格段に低い。例えば、アダプタ 3 4 は、アダプタ 5 2 よりも動作が低速であったり、機能が限られていたりなどである。

【 0 0 4 5 】

ここで、一体型アダプタ 3 4 またはアダプタ 5 2 は、単独で、ポート 4 8 などの複数のポートを介して、複数台の物理的ディスプレイ装置の相互接続に対応しても、対応していなくてもよい。しかし、各アダプタは、少なくとも 1 台の物理的ディスプレイと相互接続されうる。このため、本発明の実施形態の例示では、メモリ 1 6 にロードされたアダプタドライバソフトウェア 1 0 8 が、グラフィックアダプタ 5 2 を、ディスプレイ 3 2 と 6 8 の両方で提示するグラフィックをレンダリングするマスタ（または第 1 の）グラフィックアダプタとして動作させ、グラフィックアダプタ 3 4 を、第 2 またはスレーブアダプタとして動作させる。

【 0 0 4 6 】

明らかなように、このマスタ/スレーブ動作モードにおいては、一体型グラフィックアダプタ 3 4 が実質的に無効にされる。しかし、オンボードグラフィックアダプタ 3 4（特にディスプレイインタフェース 4 4）の部品を、相互接続されたモニタや他のディスプレイ装置を駆動するために使用することができる。このため、カード 5 0 が、複数のディスプレイポートおよびディスプレイドライバを有している必要はない。この代りに、ディスプレイインタフェース 4 4 とそのポート 4 8 を、カード 5 0 によって駆動されるディスプレイ 6 8 またはディスプレイに加えて、少なくとも 1 台のモニタまたはディスプレイ 3 2 を駆動するために使用することができる。アダプタ 5 2 が複数のポート（ポート 6 4, 6 6 など）を有する場合、プロセッサ 5 2 が、これらの複数のポートを介して提示するための画像を付加的にレンダリングしてもよい。当然、ドライバソフトウェア 1 0 8 は、上記のように、アダプタ 3 4 を、その従来の動作モードで動作させてもよい。アダプタ 3 4（このためアダプタ 5 2）の動作モードの選択は、アプリケーションソフトウェア 1 0 6 によって行われるか、またはアプリケーションソフトウェア 1 0 6 を介して行われうる。

【 0 0 4 7 】

詳細には、図 4 に模式的に示すように、本発明の実施形態の例示では、マスタ/スレーブ動作モードにおいて、メモリ 1 6 内のドライバソフトウェア 1 0 8 が、アダプタ 5 2 のメモリ 5 6 内に、アクティブフレームバッファ 7 0 とセカンダリアダプタバッファ 7 2 を割り当てる。アクティブフレームバッファ 7 0 は、アダプタ 5 2 用の従来のフレームバッファとして機能し、このため、アダプタ 5 2 のディスプレイ 6 8 に表示するためのフレームを記憶する。このように、グラフィックプロセッサ 5 4 は、従来の方法で、メモリ 5 6 内のアクティブフレームバッファ 7 0 に、二次元または三次元のグラフィックをレンダリングする。カード 5 0 のディスプレイインタフェース 6 2 は、相互接続されたモニタまたはディスプレイ装置に表示させるために、アクティブフレームバッファ 7 0 の内容を変換する。任意選択で、前述のように、アクティブフレームバッファ 7 0 の代りに、フロントバッファとリアバッファが使用されてもよい。

【 0 0 4 8 】

グラフィックプロセッサ 5 4 は、更に、セカンダリアダプタフレームバッファ 7 2 に、ディスプレイ装置 3 2 に表示させるための画像をレンダリングする。セカンダリアダプタフレームバッファ 7 2 へのフレームのレンダリングが終了すると、下記で詳細に説明するように、セカンダリアダプタフレームバッファ 7 2 の内容が、グラフィックアダプタ 3 4 のディスプレイインタフェース 4 4 によって提示するために、フレームバッファ 4 6 から転送される。

【 0 0 4 9 】

第 1 実施形態では、メモリ 16 内のグラフィックアダプタドライバソフトウェア 108 が、図 5 A に示したステップ S 5 0 0 を実行する。詳細には、ステップ S 5 0 2 において、ドライバソフトウェア 108 は、メモリ 56 のセカンダリアダプタバッファ 72 をレンダリングするために、グラフィックプロセッサ 54 宛のコマンドを生成する。上で説明したように、コマンドがキュー 47 にキューイングされるのと全く同じ方法で、これらのコマンドがアダプタ 52 のコマンドキュー 57 にキューイングされ、グラフィックプロセッサ 54 によって実行される。フレームがレンダリングされると、ドライバソフトウェア 108 は更に、1 つ以上のコマンドを生成し、この結果、ステップ S 5 0 4 において、プロセッサ 54 は、メモリ 56 内のセカンダリアダプタフレームバッファ 72 の内容を、グラフィックアダプタ 34 用のフレームバッファ 46 として割り当てられたメモリ 16 の当該領域に、ビットブロック転送 (B I T B L T) する。

10

【 0 0 5 0 】

これに対し、グラフィックアダプタ 34 のディスプレイインタフェース 44 は、ポート 48 において相互接続されたモニタまたはディスプレイに、自身のフレームバッファ 46 の内容を表示するように事前にプログラムされている。このようにして、ポート 48 とポート 64 によって相互接続されたディスプレイ 32 と 68 に表示される画像が、グラフィックプロセッサ 54 によってレンダリングまたは描画され、有利である。バッファ 72 およびフレームバッファ 46 へのステップ S 5 0 4 の効果が図 5 B に示される。

【 0 0 5 1 】

20

第 2 実施態様では、図 6 A に示したステップ S 6 0 0 が実行される。まず、フロントフレームバッファとバックフレームバッファが、アダプタ 52 のメモリ 16 内に割り当てられる。便宜上、これらを、バッファ 46 a , 46 b と呼び、図 6 B に示す。この場合も、ステップ S 6 0 2 において、プロセッサ 54 は、ステップ S 5 0 2 を参照して上で説明したように、ディスプレイインタフェース 44 によって提示させる任意のフレームを、メモリ 56 のセカンダリアダプタフレームバッファ 72 にレンダリングする。

【 0 0 5 2 】

デバイス 32 に表示するためのフレームのレンダリングが終了すると、ステップ S 6 0 4 において、ドライバソフトウェア 108 は、メモリ 56 内のセカンダリアダプタフレームバッファ 72 の内容を、グラフィックアダプタ 34 用のメモリ 16 内の現在のバックバッファにビットブロック転送 (B I T B L T) するように、グラフィックプロセッサ 54 をプログラムする。B I T B L T が終了すると、ステップ S 6 0 6 において、ドライバは、アダプタ 34 のバックバッファとフロントバッファを反転させる (すなわち、バックバッファをフロントバッファとして使用する) ように、ディスプレイインタフェース 44 のレジスタをプログラムするように、プロセッサ 54 をプログラムする。このことは、ディスプレイインタフェース 44 がデータを提示するために使用する開始アドレスを指定するように、ディスプレイインタフェース 44 のレジスタを直接再プログラムすることによって行うことができる。これに対し、グラフィックアダプタ 34 のディスプレイインタフェース 44 は、ポート 48 において相互接続されたモニタ (すなわちディスプレイ 68) に、ディスプレイインタフェース 44 のバッファの内容を提示する。前のフロントバッファが、今度はアダプタ 34 用のバックバッファとして使用され、ステップ S 6 0 4 において、ディスプレイ 32 に表示させるためにプロセッサ 54 によってレンダリングされる次のフレームが、最初にこのバックバッファから転送される。ドライバソフトウェア 108 は、当然、各バッファ 46 a , 46 b の開始位置と、フロントバッファおよびバックバッファとしての状態を保持している。バッファ 46 a , 46 b へのステップ S 6 0 4 , S 6 0 6 の効果が、図 6 B、図 6 C に模式的に示される。

30

40

【 0 0 5 3 】

同期を行わない場合には、1 つのバッファ 46 の使用、あるいはアダプタ 34 のディスプレイインタフェース 44 の直接のプログラミングにより、目に見えるティアリングが発生しうる。すなわち、インタフェース 44 によるフレーム出力の途中でディスプレイパラ

50

メータが変更されるため、ディスプレイインタフェース 44 のレジスタが再プログラムされる垂直帰線消去期間以外の任意の時点でも、ティアリングが見えてしまう。

【0054】

このため、更に別の第3実施形態では、図7に示したステップ S700 が実行されうる。この場合も、最初に、アダプタ 34 によって使用されるメモリ 16 内に、フロントリアバッファ 46a とリアバッファ 46b が割り当てられる。ステップ S502, S602 を参照して上で説明したように、この場合も、プロセッサ 54 は、ローカルメモリ 56 のバッファ 72 に、装置 32 に表示させるための画像をレンダリングする。

【0055】

ディスプレイ 32 に表示させるためのフレームのレンダリングが完了すると、ステップ S704 において、メモリ 16 内のドライバソフトウェア 108 は、メモリ 56 内のセカンダリアダプタフレームバッファ 72 の内容を、グラフィックアダプタ 34 のメモリ 16 内の現在のバックバッファにビットブロック転送 (BITBLT) するように、グラフィックプロセッサ 54 をプログラムする。ステップ S706 において、ドライバソフトウェア 108 は、更に、アダプタ 34 のコマンドキュー 47 に、フロントバッファとバックバッファを反転させるためのコマンドをキューイングする。しかし、コマンドプロセッサ 42 のレジスタが、コマンドキュー 47 内の未処理のコマンドを反映させるように、ドライバソフトウェア 108 によって更新されない。その代わりに、ステップ S708 において、ドライバソフトウェア 108 は、ステップ S706 で開始されたビットブロック転送の終了後に、コマンドプロセッサ 42 のコマンドキューレジスタを更新するためのコマンドを、グラフィックプロセッサ 54 に提供する。これにより、バックバッファをフロントバッファに戻させるための命令が、コマンドプロセッサ 42 によってプロセッサ 40 に提供される。これを受けて、グラフィックプロセッサ 40 は、(すなわち、ポート 48 において相互接続されたモニタまたはディスプレイに前のバックバッファの内容を提示するように、ディスプレイインタフェース 44 のレジスタを再プログラムすることによって)、キューイングされた、アダプタ 34 のバックバッファとフロントバッファを反転させるための命令を実行する。

【0056】

更に別の第4実施態様では、図8に示したステップ S800 が実行される。この場合も、まず、アダプタ 34 のメモリ 16 内に、フロントバッファとバックバッファが割り当てられる。この場合も、プロセッサ 54 は、前述のように、ローカルメモリ 56 のバッファ 72 に、ディスプレイ 32 用の画像をレンダリングする。

【0057】

ドライバソフトウェア 108 は、更に、グラフィックプロセッサ 40 に、メモリ 56 内のセカンダリフレームバッファ 72 の内容を、グラフィックアダプタ 34 用の現在のバックフレームバッファとして割り当てられたシステムメモリ 16 の当該領域にビットブロック転送 (BITBLT) させるコマンドを、アダプタ 34 のキュー 47 にキューイングする。また、ドライバソフトウェア 108 は、リアバッファをフロントバッファとして使用するようにディスプレイインタフェース 44 をプログラムすることにより、グラフィックプロセッサ 40 に、アダプタ 34 のフロントバッファとバックバッファを反転させるコマンドを、キュー 47 にキューイングする。しかし、フレームがバッファ 72 にレンダリングされるまで、キューポインタを定義しているコマンドプロセッサ 42 のレジスタは更新されない。

【0058】

このため、ディスプレイ 32 用のフレームのレンダリングの終了時に、グラフィックプロセッサ 54 は、コマンドプロセッサ 42 用のキューポインタを定義しているコマンドレジスタを更新させる命令を提供して、アダプタ 34 のコマンドキュー内の2つの未処理のコマンドを反映させる。これを受けて、グラフィックプロセッサ 40 は、バッファ 72 の内容をバックバッファにビットブロック転送し、バックバッファとフロントバッファを反転させるためにキューイングされた命令を(すなわち、ポート 48 において相互接続され

たモニタまたはディスプレイに前のバックバッファの内容を提示するために、ディスプレイインタフェース44のレジスタを再プログラムすることにより)実行する。

【0059】

ここで理解されるように、上に記載した実施形態により、マスタグラフィックプロセッサが、複数の独立したフレームバッファへの二次元または三次元のグラフィック画像の描画の責任を負うことが可能となる。スレーブグラフィックアダプタは、1つのフレームバッファに作成されたディスプレイ画像に使用されうる。追加ディスプレイを駆動するために必要なデバイスの電子回路とポートは、スレーブグラフィックアダプタ34によって提供されうる。このようにして、スレーブグラフィックコントローラに設けられた電子回路が、複数台のディスプレイの駆動を可能にするために効率的に使用されうる。

10

【0060】

バッファ70, 72を使用して1つの面を定義することができ、この面に、アプリケーションソフトウェア106が、グラフィックアダプタ52, 34と相互接続された2台以上のディスプレイ48, 64にまたがって表示させる1つのグラフィック画像をレンダリングすることができ、有利である。したがって、ドライバソフトウェア108は、利用可能なスクリーンサイズをアプリケーションソフトウェア106に報告し、この値はバッファ70とバッファ72のサイズと等しい。このため、バッファ70が $m \cdot sub \cdot 1 \times n$ の解像度を有し、バッファ72が $m \cdot sub \cdot 2 \times n$ の解像度を有する場合、ドライバソフトウェア108は、アプリケーションソフトウェア106のアプリケーションに、利用可能なスクリーンサイズとして $(m \cdot sub \cdot 1 + m \cdot sub \cdot 2) \times n$ を報告しうる。バッファ70, 72が、メモリ56内で隣接する $(m \cdot sub \cdot 1 + m \cdot sub \cdot 2) \times n$ ピクセル位置を占める場合には、ドライバソフトウェア108によるレンダリングが簡単になる。アプリケーションソフトウェア106は、ドライバソフトウェア108を使用して、グラフィックプロセッサ40に、バッファ70, 72によって定義される面を構成している、 $(m \cdot sub \cdot 1 + m \cdot sub \cdot 2) \times n$ ピクセルの解像度を有する大きな画像をレンダリングさせる。このため、バッファ70, 72は、画像のペアを記憶し、このペアの各画像が、アプリケーションによってレンダリングされる大きな画像の一部を構成する。大きな画像のそれぞれのレンダリングが終了すると、ドライバソフトウェア108は、上で説明したステップS500、S600、S700またはS800に従って、ディスプレイ32に表示させるために、バッファ72内の画像をフレームバッファ46に転送しうる。バッファ70内の画像は、通常の方法でディスプレイ68に表示されうる。エンドユーザは、ディスプレイ32と68を並べて観ており、アプリケーション106によってレンダリングされる大きな画像を認識する。理解されるように、ゲームの形態のアプリケーション106は、変更を必要とせずに、マルチディスプレイの利点を享受しうる。

20

30

【0061】

ディスプレイ68用にフロントバッファとバックバッファが使用される場合、ディスプレイ32に表示させる面部分を保持しているメモリ56内のバッファが、ディスプレイ68に表示させる面部分をレンダリング中であるバッファの隣に位置することを保証するために、バッファ72の代わりに、2つの異なるバッファが割り当てられて使用されてもよい。したがって、S602、S604、S702、S704またはステップS802、S704は、完成直後の画像を保持しているバッファを、バックバッファ46aまたは46bにビットブロック転送するように変更されてもよい。既存のアプリケーションソフトウェア106を変更する必要がないため、有利である。上記に代えて、アプリケーションソフトウェアが、報告された大きなスクリーンサイズを単に使用してもよい。ドライバソフトウェア108は、複数台のディスプレイにまたがって表面をレンダリングさせる。

40

【0062】

ここで理解されるように、図に示した実施形態は、スレーブグラフィックアダプタとして一体型グラフィックコントローラを使用するが、スレーブグラフィックアダプタが、単に、グラフィックフレームをスレーブフレームバッファに転送させるのに十分な帯域幅のバス経由で、グラフィックアダプタ52と通信する別のグラフィックアダプタであっても

50

よい。このため、スレーブグラフィックプロセッサは、別のP C I e対応拡張カードとして作製されてもよい。このため、例えば、グラフィックカードがアップグレードされた場合も、古い拡張カードは、相互接続されたディスプレイを駆動可能であるため、引き続き使用することができる。同様に、相互接続バスは、P C I e相互接続バスでなくてもよく、ディスプレイインタフェース44が使用するリフレッシュレートに等しい速度で、フレームバッファ間でデータを転送させる帯域幅を有する他の任意の適切なバスであってもよい。同様に、フレームバッファ46（およびフロントバッファおよびバックバッファ46a, 46b）が、システムメモリ16内に作成されると説明したが、アダプタ34に対してメモリなローカルの一部として、容易に作成されてもよい。

【0063】

10

本発明の更に別の代替の実施形態では、バス22を介したレンダリングにより、高速バス22によって相互接続されたグラフィックアダプタ（またはサブシステム）が、一体型アダプタ/サブシステムの代理となることが可能となってもよい。

【0064】

この点において、図9は、本発明の別の実施形態の代表的なコンピューティングデバイス10'の一部の例の代表的な簡略ブロック図である。コンピューティングデバイス10'は、米国特許出願第11/421,005号明細書に詳細に記載されている。

【0065】

しかし簡単に説明すると、デバイス10'は2つのグラフィックサブシステム30', 40'を備える。グラフィックサブシステム30'は、グラフィックエンジン/プロセッサ32'、メモリコントローラ72'、ディスプレイインタフェース74'およびバスインタフェース78'を備える。第2のグラフィックサブシステム40'は、P C I eバスなどの高速バス22'経由でグラフィックサブシステム30'と通信している。グラフィックサブシステム40'は、自身のグラフィックエンジン/プロセッサ42'、メモリコントローラ52'、ディスプレイインタフェース54'を備える。グラフィックサブシステム40'は更にグラフィックメモリ50'と通信している。明らかなように、サブシステム40'は、バス22'を介してメモリ14'にグラフィックをレンダリングするように適合されている。

20

【0066】

デバイス10'はラップトップまたは小型コンピューティングデバイスの形態のポータブルコンピューティングデバイスとして好適に形成することができる。このように、1つのハウジングがD C電源、ディスプレイ26'および上述のマザーボードおよび構成要素を収容してもよい。第2のグラフィックサブシステム40'は、コンピューティングデバイスの他の部分を収容する1つのハウジングに追加されても、デバイス10'が物理的に相互接続された時にデバイス10'の一部のみを構成するドッキングステーションの一部を構成してもよい。

30

【0067】

デバイス10'は、少なくとも2つの電力消費モード、すなわち、高電力消費モードと低電力消費モードで動作することができる。図に示した実施形態では、デバイス10'の高電力モードとは、デバイス10'がA C（主）電源に接続された電源によって給電されている時に入るモードであり、低電力消費モードとは、デバイス10'が1つ以上のバッテリー、燃料電池などを使用するD C電源によって給電されている時に入るモードでもよい。別例では、電力消費モードが、例えばユーザの好み、実行中のソフトウェアアプリケーションのタイプ、バッテリーレベルなどに基づき、ユーザによって選択されるか、ソフトウェアによって制御されるか、または別の方法で選択されてもよい。

40

【0068】

デバイス10'のソフトウェア制御動作の構成は、米国特許出願第11/421,005号明細書に開示されており、上記のソフトウェア100と類似する点を有する。

【0069】

デバイス10'としてのデバイス10'の動作を制御しているソフトウェアの一部は、

50

高電力消費状態と低電力消費状態の間を遷移する。詳細には、図10は、デバイス10'のシステムメモリ内部のソフトウェアの制御下で中央処理装置12'(図1のプロセッサ12など)によって実行されうる本発明の実施形態の例示的なソフトウェアブロックS800'を示す。ブロックS800'はデバイス10'が状態変化を受けるたびに実行され、これに対して、サブシステム30', 40'を相応に構成する必要がある。図に示すように、ブロックS802'において、ソフトウェアは、デバイス10'が高電力消費モードまたは低電力消費モードのいずれを取るべきかを判定する。

【0070】

デバイス10'が高電力消費モードを再開(またはこれに移行)すべき場合、ブロックS804'~S810'が実行される。ブロックS804'において、グラフィックサブシステム40'は、まだフル動作(高電力消費)モードに設定されていなければ、高電力消費モードに設定される。このことは、グラフィックサブシステム40'を制御するドライバを介して、電力コントローラ60'に適切な信号を供給することによって実行することができる。次に、ブロックS806', S808'において、グラフィックサブシステム40'が有効にされる。この場合も、このことは、ブロックS804'において、グラフィックサブシステム30'に関連する、相互接続されたディスプレイを論理的に無効にし、ブロックS808'において、グラフィックサブシステム40'と接続されたディスプレイを論理的に有効にすることによって実行することができる。この場合も、ブロックS806', S808'は、米国特許第11/421,005号明細書に記載のEnumDisplayDevices()コールおよびChangeDisplaySettingsEX()コールなどの適切なオペレーティングシステムAPIコールによってか、またはハードウェアとの直接通信によって実行することができる。

【0071】

特に言えば、グラフィックサブシステム40'に接続されている物理的ディスプレイは存在しない。グラフィックサブシステム40'の動作を制御しているドライバソフトウェアは、ステップS810'において、関連するメモリ50'内部の代わりに、グラフィックサブシステム30'のバッファ14'に画像をレンダリングするように構成される。(例えば、PCIバスとして実施された)高速バス22が存在している場合、このようなレンダリングが、一部には、このバスによって実現される転送速度のおかげで、バス22'を介して可能となる。

【0072】

このレンダリングは、バスを介してフレームバッファにプリミティブごと(primitive by primitive)に送られ、このため、バス22'を介してレンダリングされた画像が転送されてもよい。このことは、サブシステム40'によってアクセス可能なメモリ内にバッファ14'を割り当て、サブシステム40'用のドライバソフトウェアに、バッファ14'のアドレスを提供することにより容易に行うことができる。別法として、明らかなように、フレームが1つのバッファでレンダリングされ、バス22'を介してダイレクトメモリアクセスなどによって転送されてもよい。

【0073】

同様に、グラフィックサブシステム30'用のドライバは更に、グラフィックサブシステム40'によってメモリ14'内のフレームバッファにレンダリングされた画像を、相互接続されたディスプレイ26'で提示するために、グラフィックサブシステム30'のディスプレイインタフェース74'に、メモリ14'のフレームバッファをサンプリングさせるように構成されている。同時に、グラフィックサブシステム30'のドライバは、グラフィックサブシステム30'のグラフィックエンジン32'に対し、実質的に活動停止状態またはアイドル状態に留まるように指示することができる。この動作モードは、図11Aに模式的に示されており、グラフィックサブシステム40'およびグラフィックサブシステム30'のアクティブなブロックのみが網掛けされている。

【0074】

明らかなように、図11Aの実施形態では、メモリ50'とディスプレイインタフェー

10

20

30

40

50

ス 5 4 ' は使用されていない。このため、これらの機能ブロックはサブシステム 4 0 ' から削除することができ、コスト低減が可能となる。これにより得られるサブシステム 4 0 ' は、ディスプレイインタフェースを有する必要はなく、サブシステム 4 0 ' にローカルなフレームバッファを定義するメモリがないこともある。このようなグラフィックサブシステムを作成することは、サブシステム 3 0 ' によって提供される機能をサブシステム 4 0 ' が補完するように作成されうるので、有益であろう。例えば、サブシステムは、3 D グラフィックまたはビデオデコーディング機能を提供するグラフィックエンジン 4 2 ' を提供してもよい。グラフィックエンジン 3 2 ' はこれらの機能を搭載していなくてもよい。同時に、グラフィックエンジン 3 2 ' によって提供される 2 D グラフィック機能を、サブシステム 4 0 ' に搭載する必要はない。消費者は、追加の機能が必要な場合にのみグラフィックサブシステム 3 0 ' を追加してもよい。

10

【 0 0 7 5 】

デバイス 1 0 ' が低電力消費モードに移行するか、このモードを再開した場合、ブロック S 8 1 2 ' ~ S 8 1 8 ' が実行される。概説すると、グラフィックサブシステム 4 0 ' は、部分的または完全に無効にされて低電力消費モードに設定され、この場合も、レンダリングがグラフィックサブシステム 3 0 ' によって実行される。これを行うため、ブロック S 8 1 2 ' において、グラフィックサブシステム 3 0 ' に関連する、相互接続されたディスプレイが有効にされ、ブロック S 8 1 4 ' において、グラフィックサブシステム 4 0 ' と物理的に接続されたあらゆるディスプレイが論理的に無効にされる。次に、グラフィックサブシステム 3 0 ' の動作を制御するドライバソフトウェアが、グラフィックサブシステム 3 0 ' にメモリ 1 4 ' に画像をレンダリングさせるように再度構成される。ディスプレイインタフェース 7 4 ' は、ポート 7 8 ' と相互接続されたディスプレイ 2 6 ' で画像を提示するために、メモリ 1 4 ' をサンプリングし続ける。同様に、ブロック S 8 1 8 ' において、プロセッサ 1 2 ' は最初に電力コントローラ 6 0 ' に適切な信号を供給し、グラフィックサブシステム 4 0 ' を低電力状態に設定する。最も単純な形態では、電力コントローラ（図示なし）がグラフィックサブシステム 4 0 ' への電力を遮断するか、またはグラフィックサブシステム 4 0 ' を低電力スリープモードに設定する。この場合も、この低電力消費モードでは、電圧が抑えられるか、グラフィックサブシステム 4 0 ' の全部または一部が電源切断されるか、グラフィックサブシステム 4 0 ' が使用する選択されたクロックが減速されるか、この組み合わせが行われる。詳細には、グラフィックサブシステム 4 0 ' のグラフィックエンジン 4 2 ' が、アイドル状態に留まるか、または実質的にアイドル状態に留まる（例えば低速化されるか、無効にされるか、または電源切断される）。この動作モードは、図 1 1 B に模式的に示されており、アダプタ 4 0 ' とグラフィックサブシステム 3 0 ' のアクティブな機能ブロックのみが網掛けされている。非作動状態 / アイドル状態の機能ブロックは、完全に無効化されるか、または低電圧または低クロック速度で動作される。

20

30

【 0 0 7 6 】

任意選択で、グラフィックサブシステム 3 0 ' の一部が、グラフィックエンジン 3 2 ' の不使用時に無効にされてもよい。このことは、グラフィックサブシステム 4 0 ' が画像のレンダリングを担っている任意の時に、G P I O または類似の回路によって無効にできる 1 つ以上のボルテージアイランド（voltage island）に、グラフィックエンジン 3 2 ' および他の構成要素を配置することによって容易に行うことができる。

40

【 0 0 7 7 】

他の変更例もまた明白であろう。例えば、図 1 1 A に示す高電力モードにおいて、グラフィックサブシステム 3 0 ' とグラフィックサブシステム 4 0 ' の両方が、メモリ 1 4 ' またはメモリ 5 0 ' にレンダリングしてもよい。このようにして、2 つのグラフィックサブシステム 3 0 ' , 4 0 ' が、それぞれメモリ 1 4 ' の交番フレームにレンダリングするか、またはメモリ 1 4 ' の各フレームの交番部分（例えばスキャンライン）にレンダリングして、協調して動作することができる。

【 0 0 7 8 】

50

更に別の実施形態では、上で説明したように、追加ディスプレイをグラフィックサブシステム30'、40'に接続して、高電力消費モードでの複数台のディスプレイの同時使用を可能にすることができる。このようにして、ディスプレイインタフェース54'を、第2のディスプレイを駆動するために使用することができる。デバイス10'を、低電力消費モードへの移行時に、図11Bに示すように動作するように構成してもよい。

【0079】

同様に、デバイス10'（または10）は、バス22'（または22）に接続された複数の追加のグラフィックサブシステムを有してもよく、高電力消費モードにおいては、その全てがアクティブとなり、グラフィックサブシステム30'のディスプレイインタフェース74'を介してグラフィックをレンダリングしてもよい。低電力消費モードへの移行時に、このグラフィックサブシステムが無効化され、グラフィックサブシステム30'のグラフィックエンジン32'にレンダリングが任せられうる。

【0080】

図12に示す更に別の実施形態では、コンピューティングデバイス10'はダイレクトメモリアクセス（DMA）コントローラ90を備えてもよい。DMAコントローラ90はデータをメモリ50'からメモリ14'に転送することができる。このように、デバイス10'の高電力消費モードにおいて、グラフィックサブシステム40'は画像をメモリ50'にレンダリングしうる。レンダリングされたこれらの画像はその後、DMAコントローラ90によってメモリ14'内のフレームバッファに転送されうる。DMAコントローラ90'は、（例えばグラフィックエンジン32'または42'のDMAエンジンとして）グラフィックサブシステム30'または40'の一部を構成するか、またはコンピューティングデバイス10'内に別の方法で配置されうる。データは、メモリ50'からメモリ14'に、バス20'を介して転送されるか、または別の方法で直接転送されうる。ディスプレイインタフェース74'は、上述のように動作を続け、レンダリングされた画像をディスプレイ26'で提示するためにメモリ14'のフレームバッファをサンプリングする。この場合も、高電力消費モードにある図10のデバイス10'のアクティブなブロックが、図12で網掛けして図示されている。

【0081】

当然、上記の実施形態は、例示のみを目的としており、限定を意図するものではない。本発明を実施する記載した実施形態は、形状、構成要素の配置、操作の詳細および順序がさまざまに変更される。むしろ、本発明は、請求の範囲によって規定される範囲に、このような変更の全てを含むことを意図する。

【図 1】

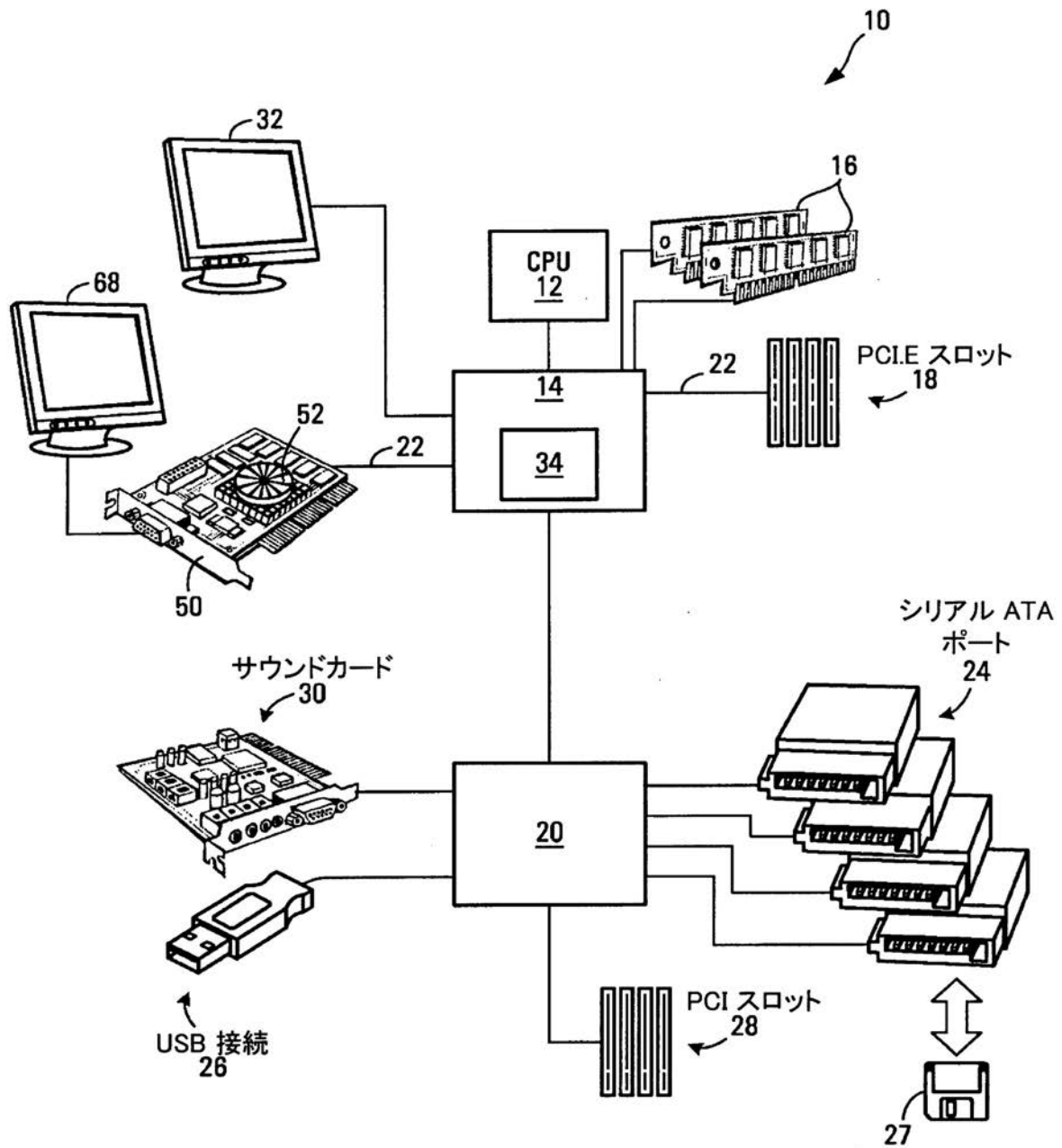
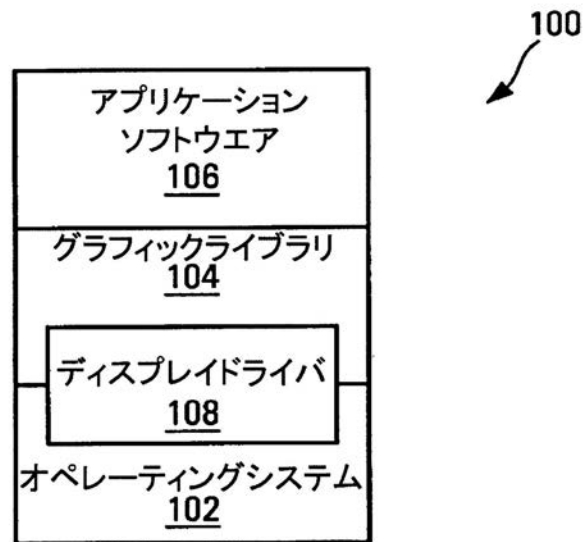


FIG. 1

【図 2】

**FIG. 2**

【 図 3 】

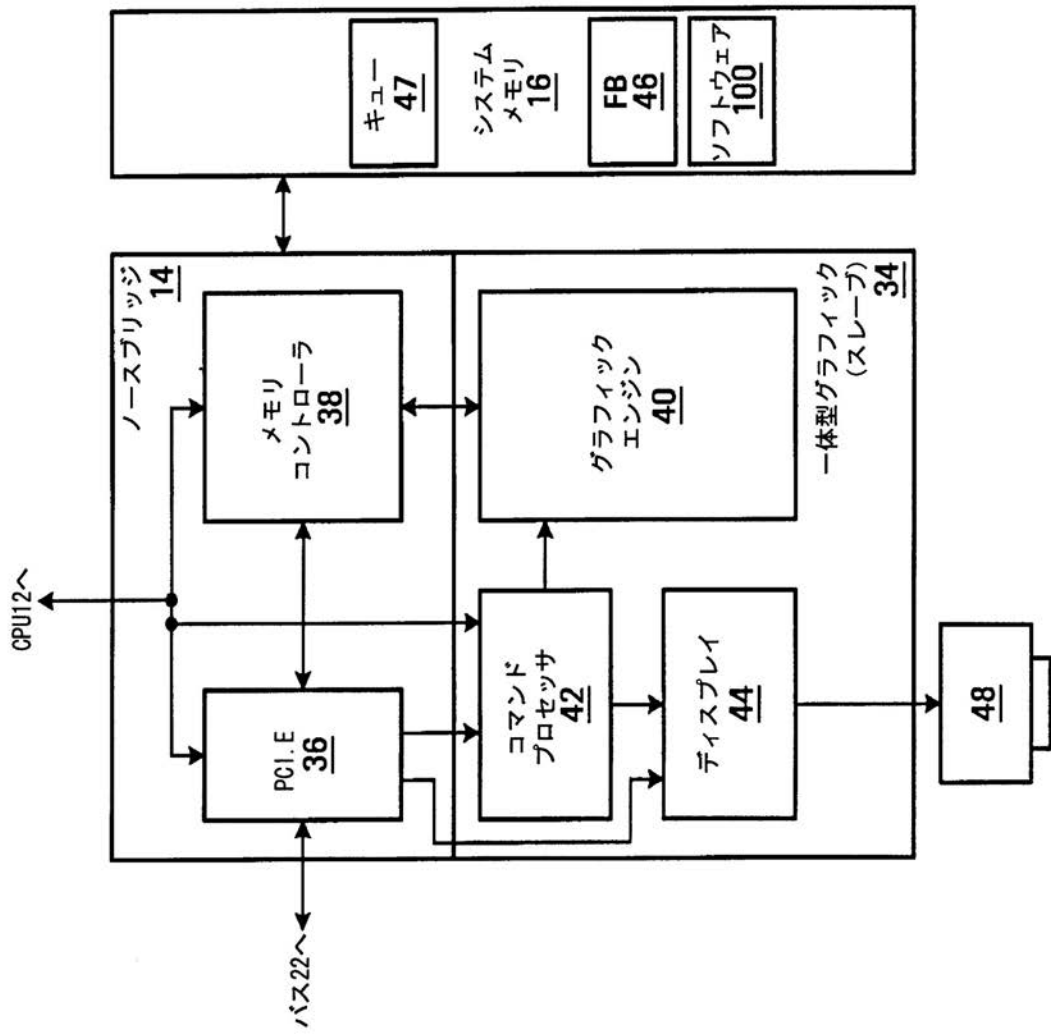


FIG. 3

【図4】

10

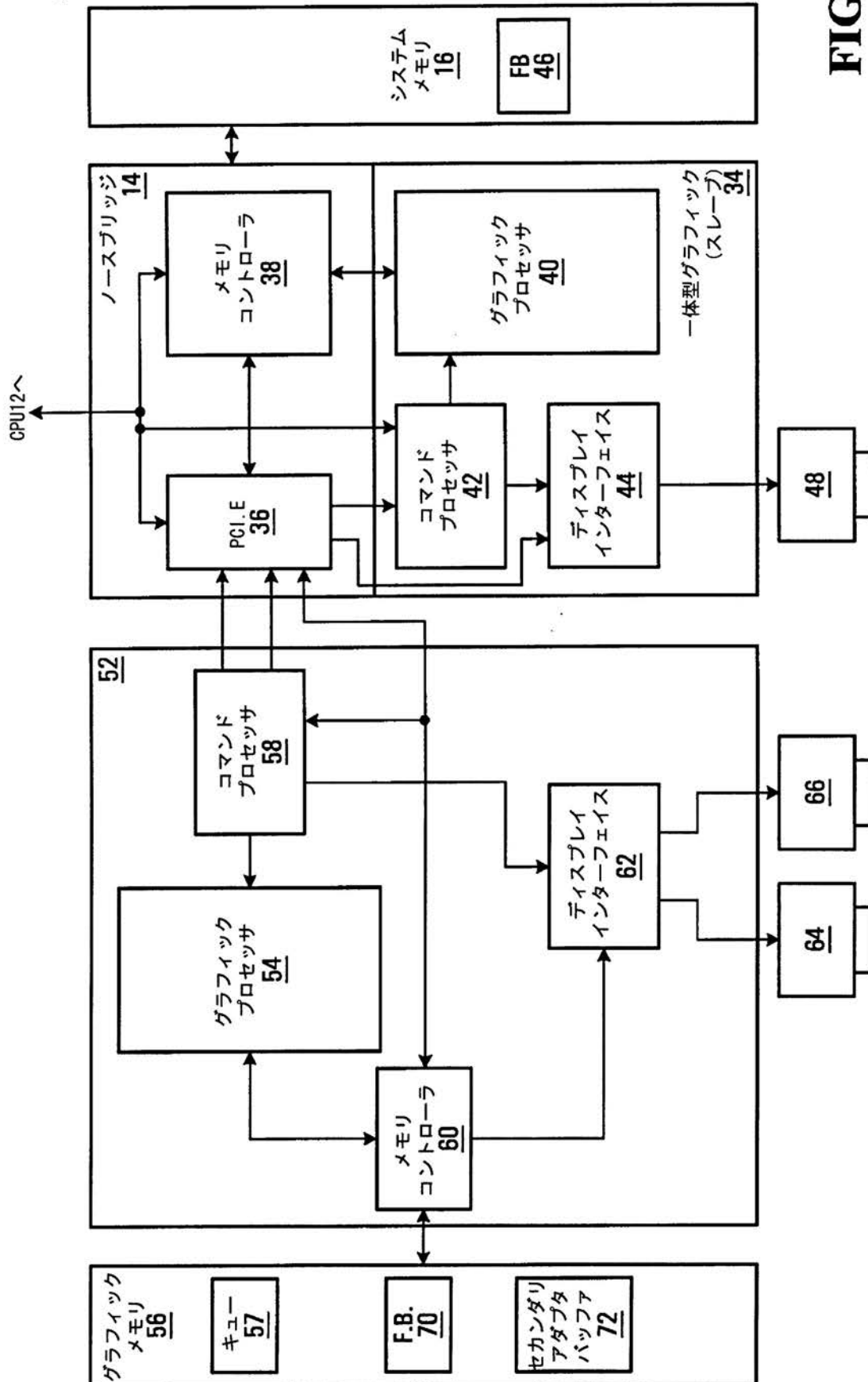
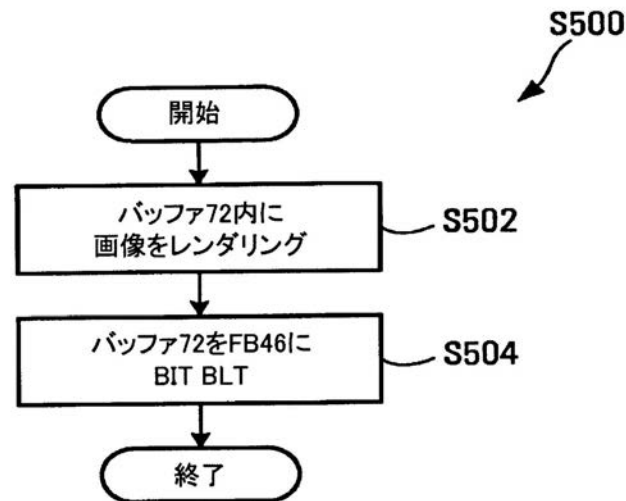
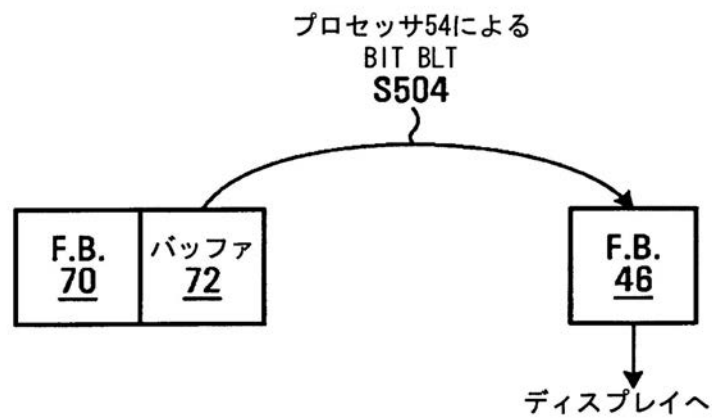


FIG. 4

【図 5 A】

**FIG. 5A**

【図 5 B】

**FIG. 5B**

【図 6 A】

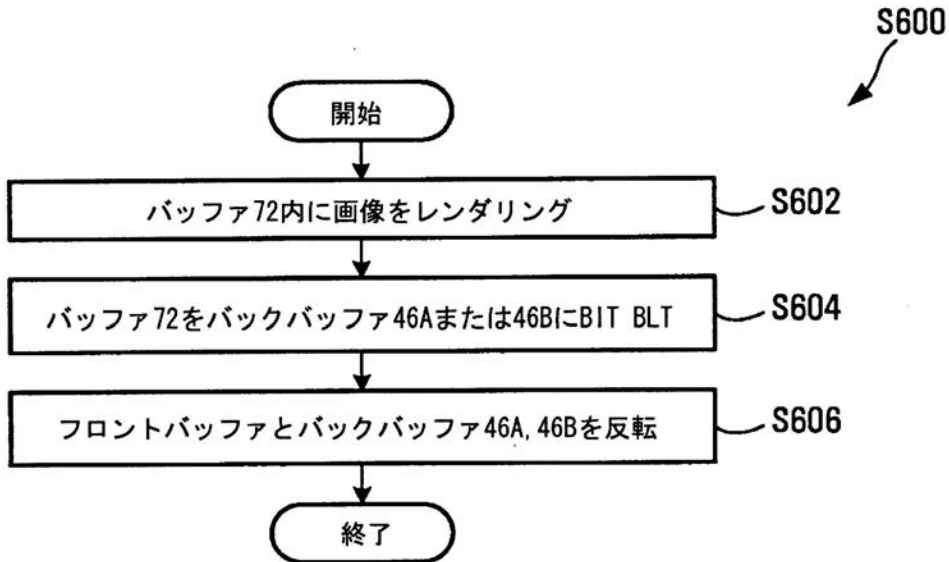


FIG. 6A

【図 6 B】

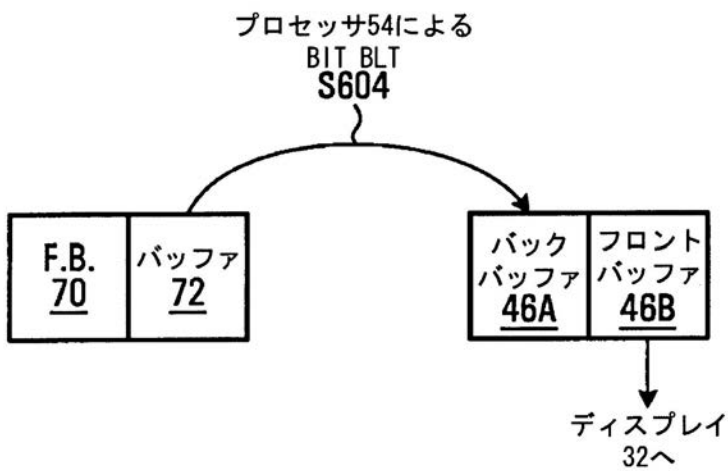


FIG. 6B

【図 6 C】

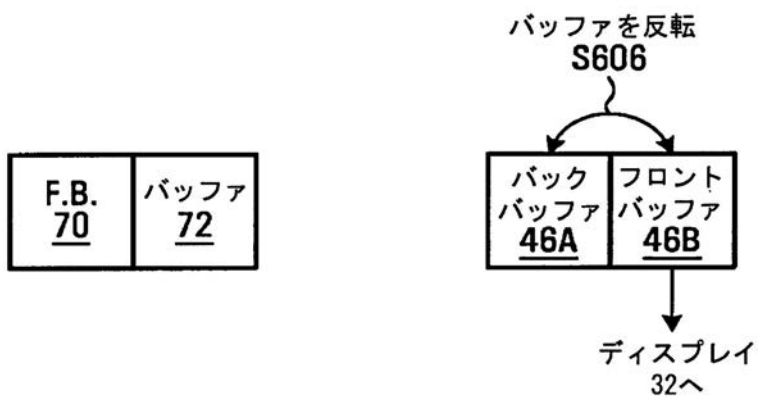


FIG. 6C

【図7】

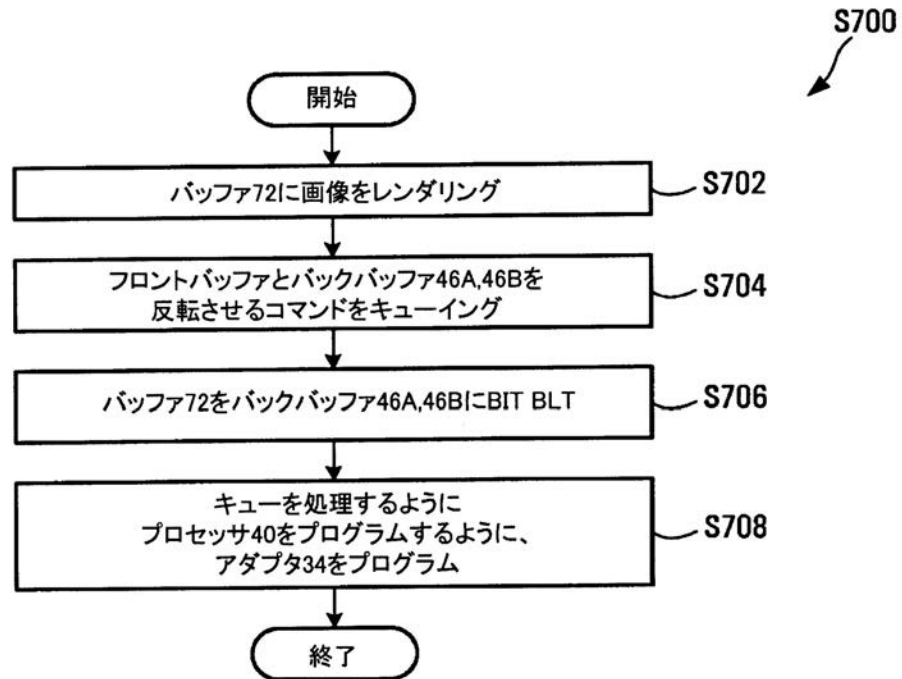


FIG. 7

【図8】

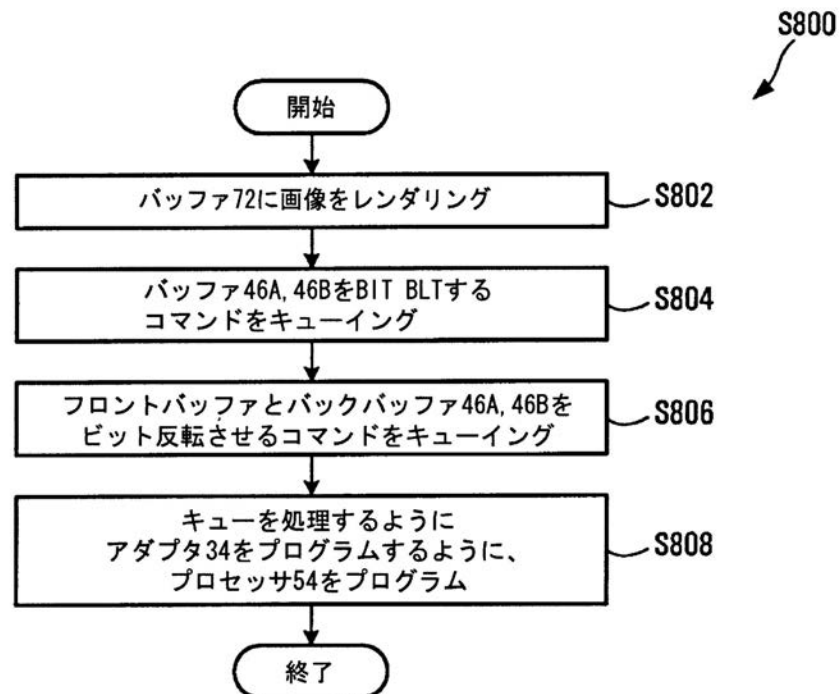


FIG. 8

【図 9】

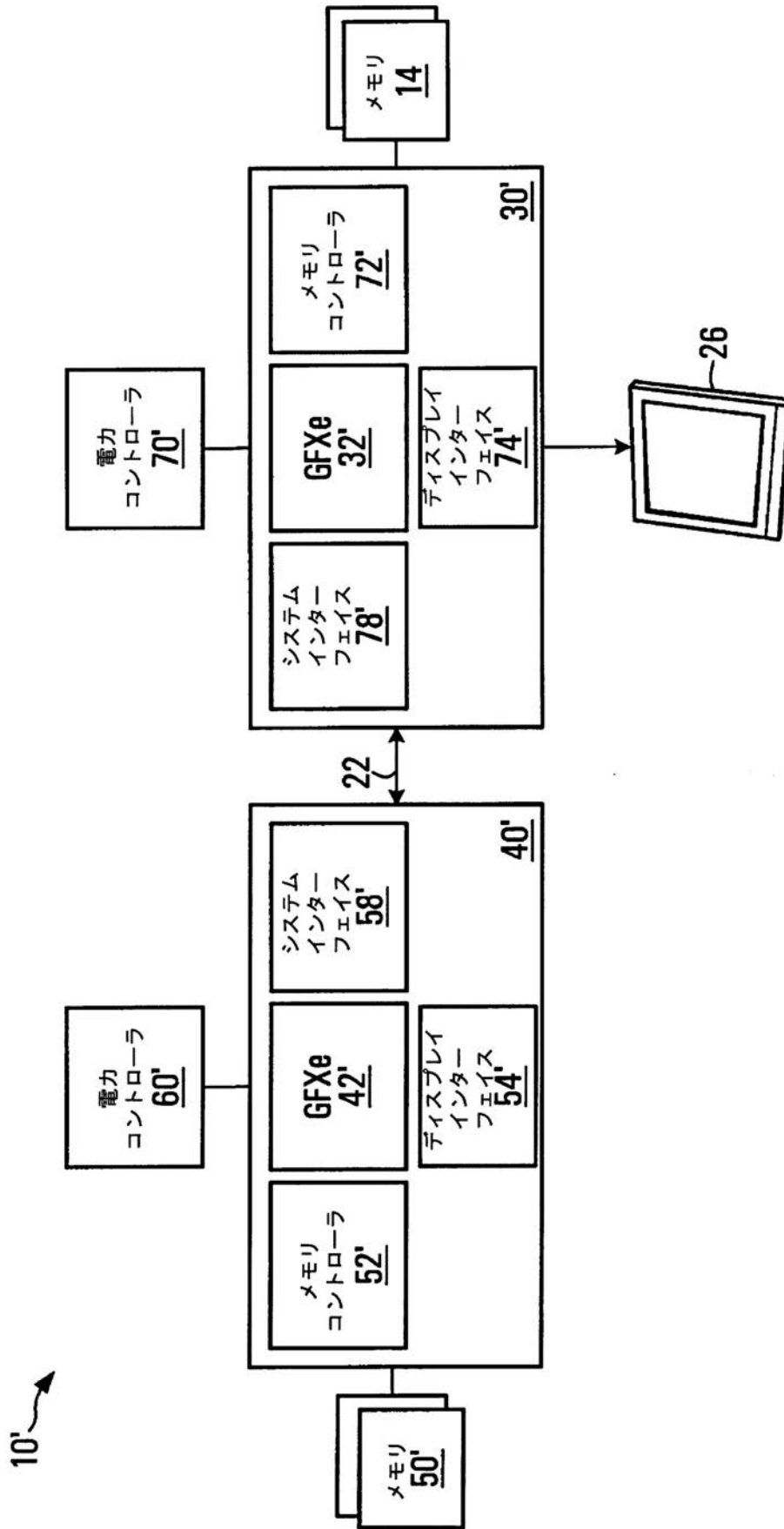


FIG. 9

【図10】

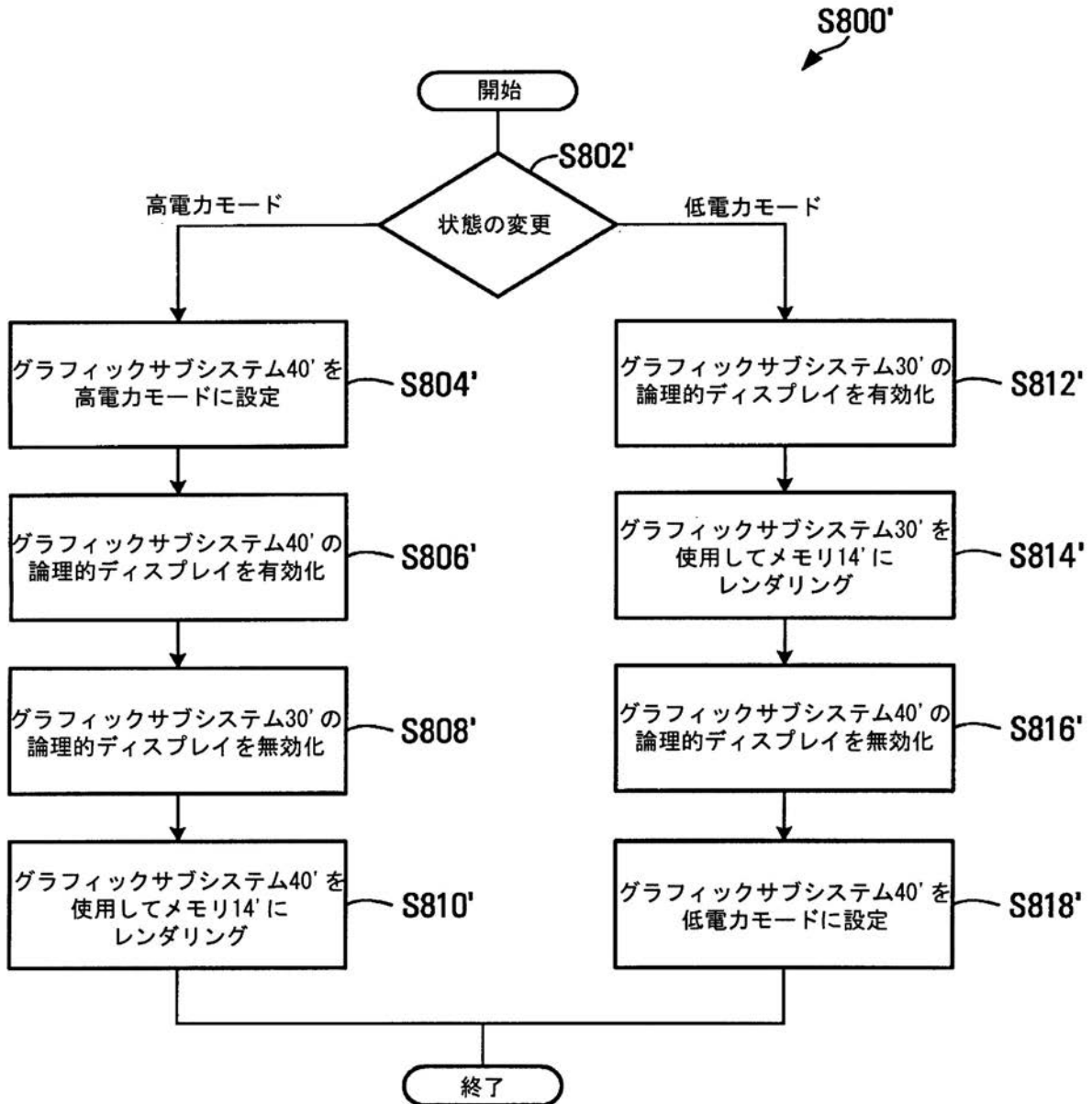


FIG. 10

【図 11A】

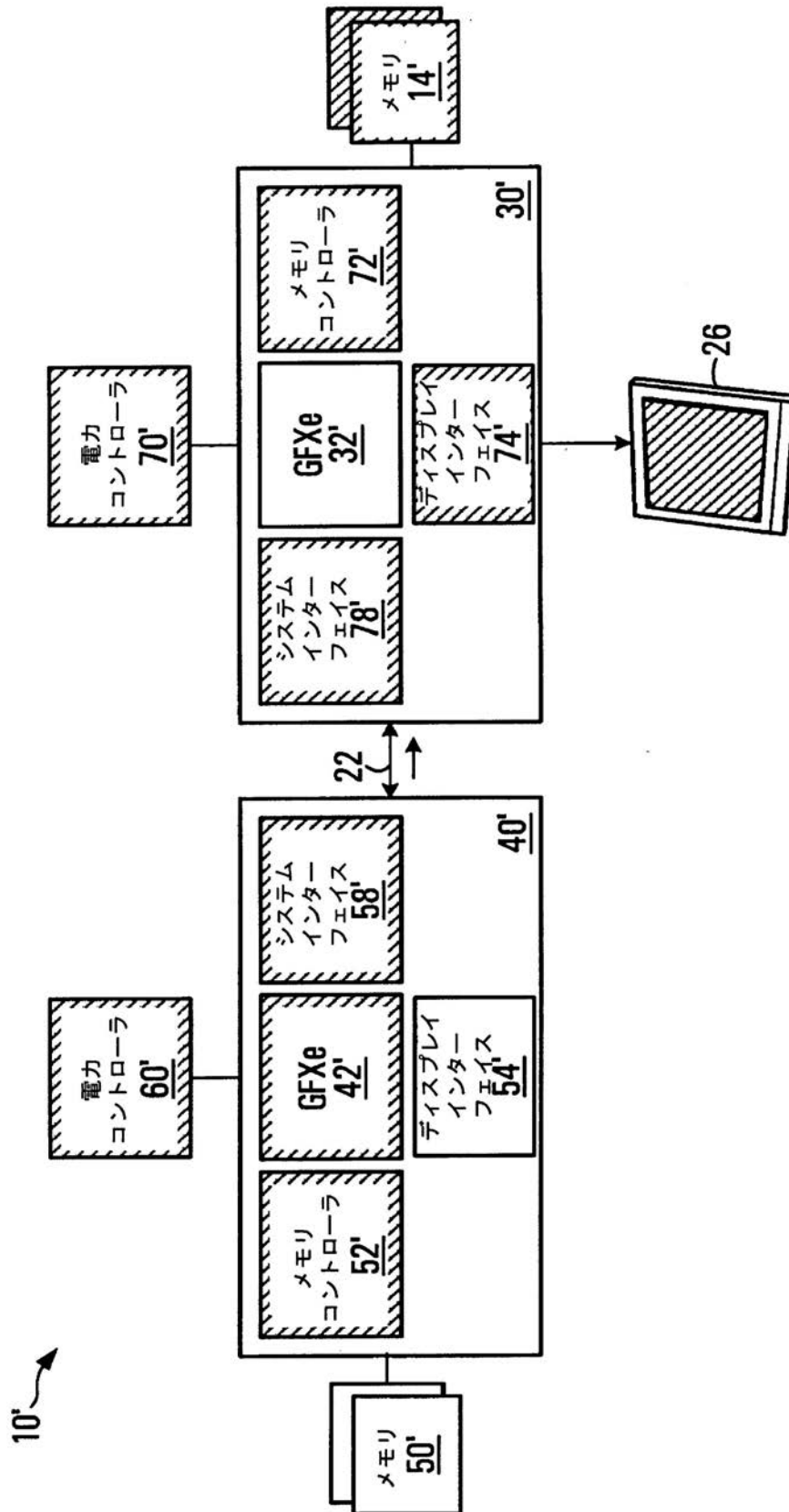


FIG. 11A

【図 11B】

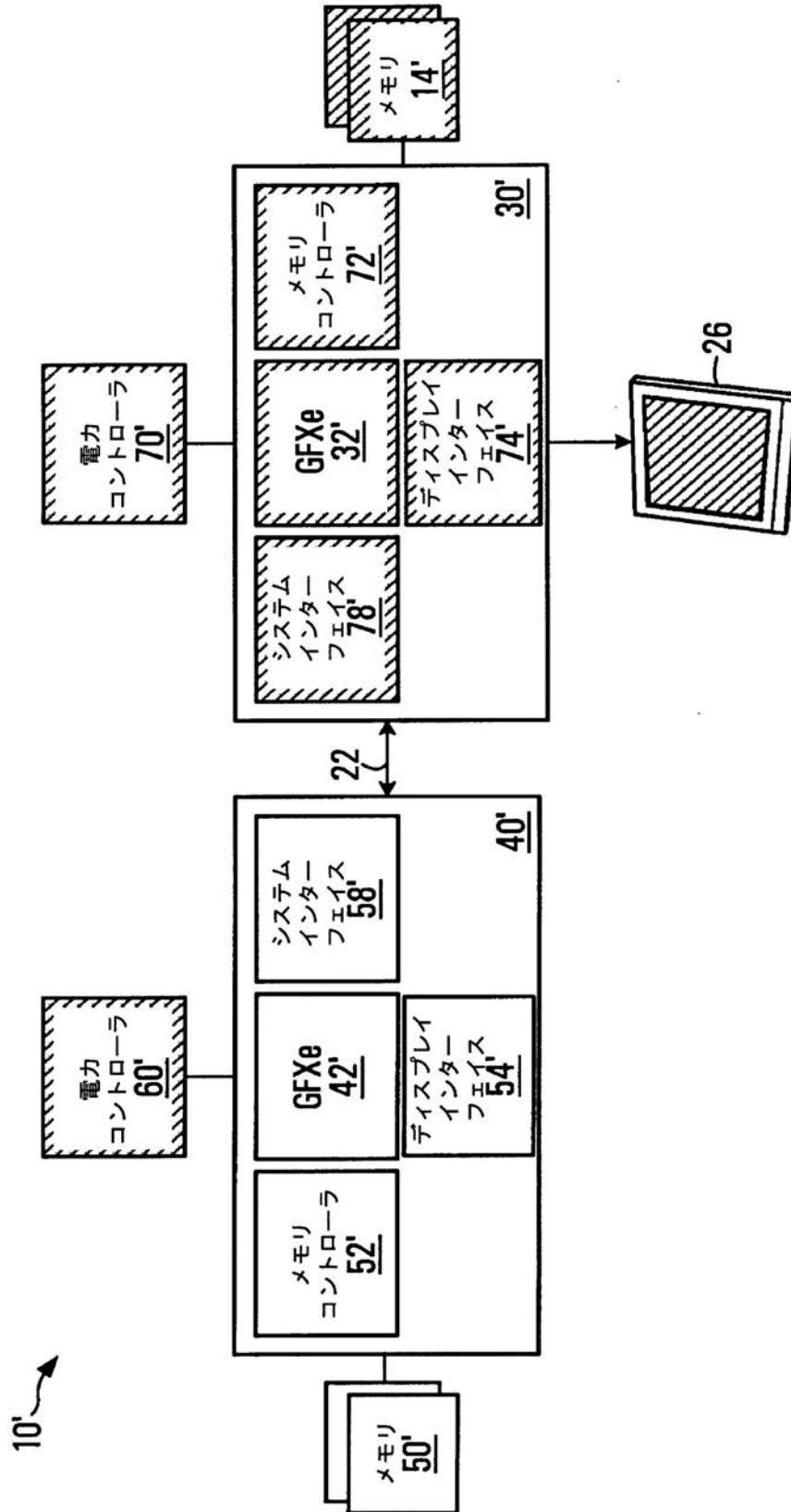


FIG. 11B

【図12】

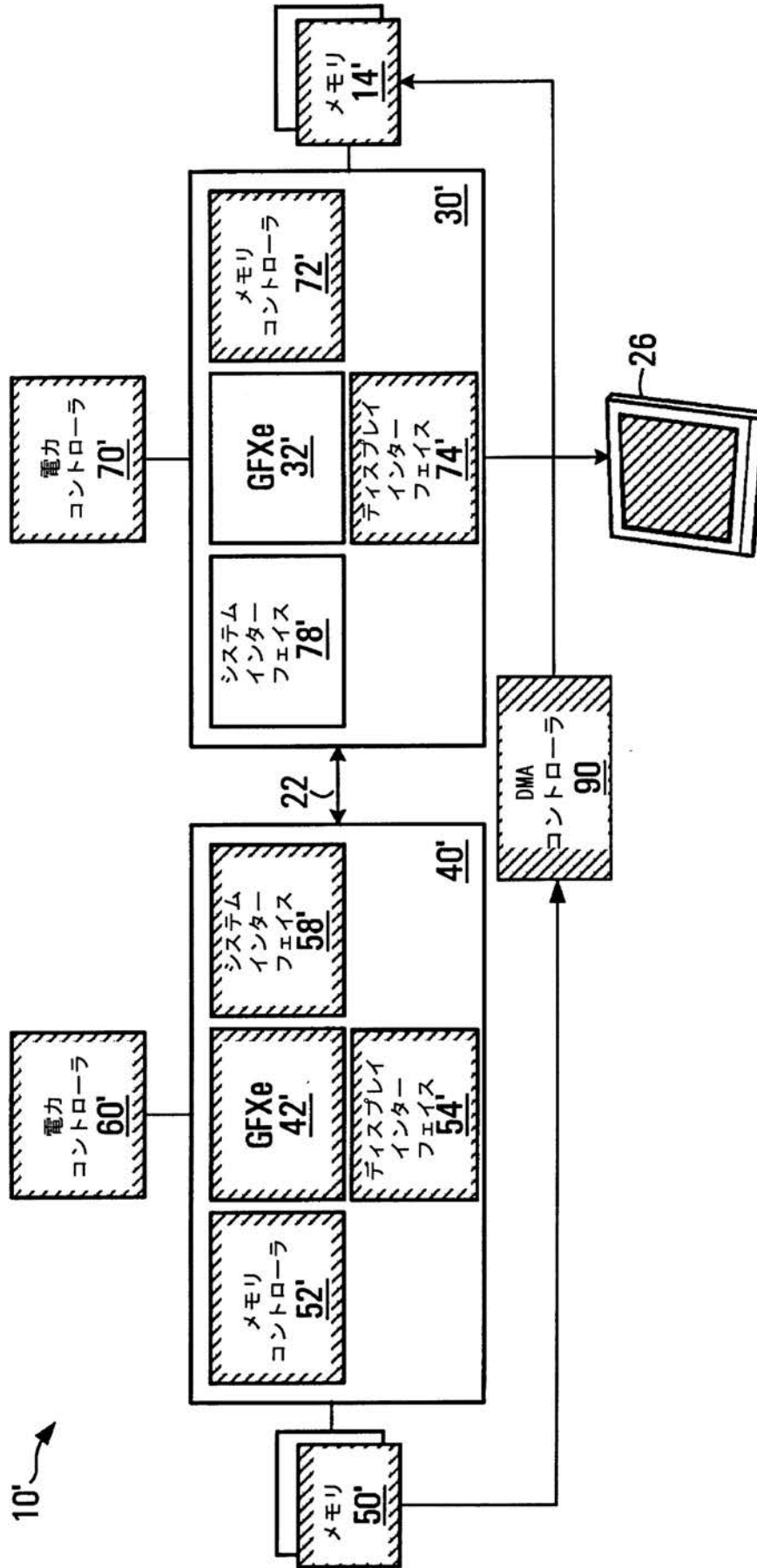


FIG. 12

フロントページの続き

(51)Int.Cl.

F I

G 0 6 F 13/38 3 1 0 B

G 0 6 F 3/153 3 3 0 A

(74)代理人 100111615

弁理士 佐野 良太

(74)代理人 100132207

弁理士 太田 昌孝

(74)代理人 100162156

弁理士 村雨 圭介

(72)発明者 ジェフリー チェン

カナダ国 M 2 K 2 A 4 オンタリオ、トロント オスカーコート 8

(72)発明者 テリー ラヴィオレット

カナダ国 L 1 E 2 C 3 オンタリオ、コーティス オーバーンレーン 4 1

(72)発明者 ジェームス ファン

カナダ国 M 2 N 4 M 7 オンタリオ、ノースヨーク ホームズアベニュー 1 7 1 a

(72)発明者 ロバート ザブルジッキ

カナダ国 L 5 H 4 A 1 オンタリオ、ミシソーガ レインツリーレーン 1 0 1 9

(72)発明者 ジェyson ロン

カナダ国 L 4 S 2 J 5 オンタリオ、リッチモンドヒル ファームステッドロード 1 8 5

(72)発明者 シアンチュエン ウェン

カナダ国 M 4 A 2 J 9 オンタリオ、トロント ウィンドクリフクレセント 9 1

(72)発明者 ササ マリンコヴィック

カナダ国 M 6 B 4 B 5 オンタリオ、トロント マーリーアベニュー 1 5 5

(72)発明者 フィル ムンマー

アメリカ合衆国 9 4 0 6 3 カリフォルニア、レッドウッドシティ ピー . オー . ボックス
5 8 2 6

(72)発明者 ミンウェイ チェン

カナダ国 M 2 J 5 G 5 オンタリオ、トロント ゴッドストーンロード 7 6

(72)発明者 マイケル トレシッダー

カナダ国 L 3 X 1 L 6 オンタリオ、ニューマーケット マリナーレーン 6 6 0

(72)発明者 ルーメン サルチェフ

カナダ国 L 3 R 4 C 4 オンタリオ、マークハム ブキャナンドライブ 7 3

(72)発明者 ジョージ シエ

カナダ国 M 2 J 1 G 3 オンタリオ、ノースヨーク ジョージヘンリーブールバード 4 1 4
- 1 3 0

(72)発明者 ユーリ リチマノフ

カナダ国 L 4 C 5 N 2 オンタリオ、リッチモンドヒル ユニット 2 0 キャッスルロック
ドライブ 1 4 1

審査官 萩島 豪

(56)参考文献 米国特許出願公開第2005/0285863(US, A1)

米国特許第05896141(US, A)

米国特許出願公開第2006/0267987(US, A1)

特表2007-535042(JP, A)

特開平11-109937(JP, A)

特開平03-233780(JP, A)

(58)調査した分野(Int.Cl. , D B 名)

G 0 6 F	3 / 1 4
G 0 6 F	3 / 0 4 8
G 0 6 F	3 / 1 5 3
G 0 6 F	1 3 / 3 8
G 0 9 G	5 / 0 0