(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第3707950号
(P3707950)
(45)発行日　平成17年10月19日(2005.10.19)　(24)登録日　平成17年8月12日(2005.8.12)

(51)Int.Cl.[7]　F I
G06F　3/153　G06F　3/153　３３６Ｂ

請求項の数 5　(全 11 頁)

(21)出願番号　特願平11-89030
(22)出願日　平成11年3月30日(1999.3.30)
(65)公開番号　特開2000-76039(P2000-76039A)
(43)公開日　平成12年3月14日(2000.3.14)
審査請求日　平成13年3月6日(2001.3.6)
(31)優先権主張番号　052,866
(32)優先日　平成10年3月31日(1998.3.31)
(33)優先権主張国　米国(US)

(73)特許権者　398038580
ヒューレット・パッカード・カンパニー
ＨＥＷＬＥＴＴ－ＰＡＣＫＡＲＤ　ＣＯＭＰＡＮＹ
アメリカ合衆国カリフォルニア州パロアルト　ハノーバー・ストリート　３０００
(74)代理人　100099623
弁理士　奥山　尚一
(74)代理人　100096769
弁理士　有原　幸一
(74)代理人　100107319
弁理士　松島　鉄男
(72)発明者　マイケル・アール・ディール
アメリカ合衆国コロラド州ラブランド　チャンスリー・ドライブ２００８
最終頁に続く

(54)【発明の名称】コンピュータ・グラフィックス用フレーム・バッファ転送方法

(57)【特許請求の範囲】
【請求項１】
ラスタ型コンピュータ・グラフィックス・ディスプレイ・システムにおいて、画素データのブロックを送り側マルチライン・フレーム・バッファ領域から宛先マルチライン・フレーム・バッファ領域に転送する方法であって、
（ａ）前記宛先マルチライン・フレーム・バッファ領域の最初の画素データ書き込みラインに対応したラスタ・ラインの位置が、現在のラスタ・ラインの位置から少なくとも最小数のラスタ・ラインだけ先行するか否かを判断し、先行するならステップ(b)へ進み、そうでないならステップ(c)へ進むステップと、
（ｂ）前記画素データのブロックにおける画素データの最後のラインが転送されるまで、該画素データのラインを前記送り側マルチライン・フレーム・バッファ領域から前記宛先マルチライン・フレーム・バッファ領域へ一度に１ラインずつ転送し、その後転送動作を停止するステップと、
（ｃ）前記宛先マルチライン・フレーム・バッファ領域の最初の画素データ書き込みラインに対応したラスタ・ラインの位置が、現在のラスタ・ラインの位置より後であるか否かを判断し、後でない場合にはステップ（ａ）を再び開始し、後である場合にはステップ（ｄ）へ進むステップと、
（ｄ）次の送り側ライン指示および次の宛先ライン指示に従って、前記送り側マルチライン・フレーム・バッファ領域から前記宛先マルチライン・フレーム・バッファ領域へ１ラインの画素データを転送するステップと、

（ｅ）ステップ（ｄ）で転送された前記１ラインの画素データが前記画素データのブロックの最後のラインであった場合に、転送動作を停止し、最後でない場合には、前記次の送り側ライン指示および前記次の宛先ライン指示を更新して、ステップ（ｆ）へ進むステップと、

（ｆ）前記更新された次の宛先ライン指示に基づく前記宛先マルチライン・フレーム・バッファ領域の画素データ書き込みラインに対応したラスタ・ラインの位置が現在の読み出しラスタ・ラインの位置より前であるか否かを判断し、前であればステップ（ｂ）へ進み、前でなければステップ（ｇ）へ進むステップと、

（ｇ）前記更新された次の宛先ライン指示に基づく前記宛先マルチライン・フレーム・バッファ領域の画素データ書き込みラインに対応したラスタ・ラインの位置が現在の読み出しラスタ・ラインの位置より後であるか否かを判断し、後でない場合には再びステップ（ｇ）を開始し、後である場合にはステップ（ｄ）へ進むステップと、

を含むことを特徴とする方法。

【請求項２】

前記ステップ(b)及び前記ステップ(d)は１ラインＢＬＴコマンドをＢＬＴエンジンに対して発行することを含むことを特徴とする請求項１記載の方法。

【請求項３】

前記ステップ（ｂ）および（ｄ）を、送り側ＸＹアドレス、宛先ＸＹアドレス、および１フル・ラスタ・ラインのＸ長よりも短いＸ長に対応して実行することを特徴とする請求項１記載の方法。

【請求項４】

前記送り側のマルチライン・フレーム・バッファ領域は第１のフレーム・バッファ・メモリ中に置かれ、

前記宛先のマルチライン・フレーム・バッファ領域は第２のフレーム・バッファ・メモリ中に置かれる

ことを特徴とする請求項１記載の方法。

【請求項５】

前記送り側のマルチライン・フレーム・バッファ領域及び前記宛先のマルチライン・フレーム・バッファ領域は同一のフレーム・バッファ・メモリ中の第１の領域及び第２の領域に夫々対応することを特徴とする請求項１記載の方法。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】

本発明は、コンピュータ・グラフィックス・ディスプレイ・システムに係り、特に、コンピュータ・グラフィックス・ディスプレイ・システムにおいてフレーム・バッファ内またはフレーム・バッファ間でデータのブロック転送を実行する方法および装置に関する。

【０００２】

【従来の技術】

ラスタ型コンピュータ・グラフィックス・ディスプレイ・システムにおいて、画素・データを格納するためにフレーム・バッファ・メモリが使用されている。一般に、フレーム・バッファ・メモリは、アレイとして構成されており、そのアレイの特定の行および列におけるエントリは、ディスプレイ装置の対応する画素の輝度値を表している。（ＲＧＢディスプレイ装置の場合、フレーム・バッファの各エントリは、画素のＲ、Ｇ、Ｂ成分の各々に対応する輝度情報を含む。）動作中、フレーム・バッファ・メモリに格納された画素輝度値は、繰り返しラスタ・パターンで読み出され、ディスプレイ装置を駆動するデジタル・アナログ・コンバータ・モジュールに送信される。

【０００３】

また、ディスプレイ装置は、繰り返しラスタ・パターンを採用して、ディスプレイを構成する画素をリフレッシュする。例えば、ＣＲＴディスプレイ装置では、水平および垂直デフレクタを使用して、左から右、上端から下端のラスタ・パターンで電子ビームを走査し

、各画素について短期間励起させる。ビームは、所定の画素の行における右端の画素に到達した後、次の行の左端の画素に素早く戻る。この帰線動作中に、ビームの強度は弱まりすなわちブランクとなり、ビームが次の行の新たな先頭位置に到達するまでにいかなる画素も励起されない。（ここで、説明の目的で、「水平走査期間」という語を、ビームがある行ですべての画素を走査し、および次の行の先頭に戻るために必要な期間を述べるために使用する。）同様に、ビームは、最後の行の最後の画素に到達した後、上端の行の左端の画素に素早く戻り、それによって、ラスタ・パターンが再び開始する。この垂直帰線動作中に、ビームの強度は、表示された画像に影響を与えないように再度ブランクとなる。

【０００４】

更なる背景として、従来から、垂直帰線には、ビームがディスプレイの上左端部に戻った後で可視画素を再び走査し始める前の待ち時間が含まれるということが、ディスプレイ技術の発展によるアーティファクトである。より古い技術によるディスプレイ・システムでは、この待ち時間は、アナログ回路がビームを移動させて帰線後に安定させるために必要であった。通常、この待ち時間の長さは、複数の水平走査期間の長さと等しい。ここでの説明のために、「ラスタ走査期間」という語は、ビームがディスプレイ全体のすべての画素を走査した後、次の行の先頭に戻るために必要な期間を示すのに用いる。

【０００５】

ラスタ走査期間中のいつでもフレーム・バッファ・メモリの画素データを変更することができるが、走査期間の途中でフレーム・バッファ・データを変更することにより、表示中の画像にノイズまたは裂け目のような不要な視覚的効果が現れることとなる。

【０００６】

この問題を解決するために用いられてきた１つの方法として、フレーム・バッファ・メモリへの書き込みを垂直帰線区間においてのみ行う方法がある。そのような技術により、表示された画像にアーティファクトを導入するのが防止されるが、フレーム・バッファへのすべての書込みが、垂直ブランキングが実行されるまで待たなければならない。このため、この技術を採用した場合、このような書込みすべてについての平均待ち時間が、１ラスタ走査期間の長さの半分となる。この待ち時間は、画質を向上させる代りに速度を妥協することを示している。

【０００７】

この問題を解決するために過去に使用されてきた第２の方法として、第１および第２のフレーム・バッファを使用する方法がある。第１および第２のフレーム・バッファを使用するこの第２の方法は、ダブル・バッファリングとして知られている。ハイヤ・エンド（higher－end）ダブルバッファ・システムにおいて、第１のフレーム・バッファ（可視フレーム・バッファ、すなわちフロント・フレーム・バッファ）の画素データは、画像が第２のフレーム・バッファ（不可視フレーム・バッファ、すなわちバック・フレーム・バッファ）内にレンダリングされる間に表示される。一旦画像が第２のフレーム・バッファに完全にレンダリングされると、２つのバッファは「スワップされ」、第２のフレーム・バッファが可視バッファとなり第１のフレーム・バッファが不可視バッファとなる。そして、新たなフレームが第１のフレーム・バッファにレンダリングされ、２つのバッファは再びスワップされる。このような解決法では、不可視フレーム・バッファに対する書込みをいつでも行うことができるという利点がある。しかしながら、このシステムでも、バッファがスワップされる前に垂直ブランキングが発生するまで待たなければならない。さらに、このようなシステムを実現するためには、冗長なフレーム・バッファ・メモリのコストと２つのフレーム・バッファをスワップさせることに関連するオーバーヘッドとにより、よりコストがかかる。

【０００８】

ローワ・エンド（lower－end）ダブル・バッファ・システムでは、バッファはまったくスワップされず、一方のバッファは常に可視であり他方のバッファは常に不可視である。フレームが完全にレンダリングされると、不可視バッファの画素データは可視バッファに単純にコピーされる。この転送を効率よく行うために、一般に、ブロック転送エンジン（Bl

ock Transfer Engine、以下ＢＬＴエンジン）が使用される。このようなローワ・エンド・ダブル・バッファシステムは、真のバッファ・スワッピング・システムより実現にコストがかからないが、上述した欠点、すなわち、視覚的な欠陥を発生させないようにするためにはＢＬＴ動作を垂直リフレッシュ中にのみ行うという欠点がある。

【０００９】

【発明が解決しようとする課題】

従って、本発明の目的は、コンピュータ・グラフィックス・ディスプレイ・システムにおいてフレーム・バッファＢＬＴ動作を実行するために必要な平均待ち時間を短縮することである。

【００１０】

本発明の他の目的は、表示画像に不要な視覚的アーティファクトを発生することなく待ち時間を短縮したフレーム・バッファＢＬＴ動作を実行することである。

【００１１】

【課題を解決するための手段】

これらの目的および他の目的は、ラスタ型コンピュータ・グラフィックス・ディスプレイ・システムにおいて送り側マルチライン・フレーム・バッファ領域から宛先マルチライン・フレーム・バッファ領域へ画素データのブロックを転送する方法および装置により達成される。好ましい実施の形態では、コンピュータ・グラフィックス・サブシステム内のマクロ・ファンクション・ユニットに組込まれた状態機械によって、上記方法のステップが実行される。本方法によれば、最初に、宛先マルチライン・フレーム・バッファ領域に対応する最初のラスタ・ラインが、少なくとも現在のラスタ・ラインに最小のラスタ・ライン数を加算した値と等しいかを判断（４０８）し、等しければ、本願方法および状態機械は高速パスを選択し、その間画素データは、そのブロックの画素データの最終ラインが転送されるまでに、送り側マルチライン・フレーム・バッファ領域から宛先マルチライン・フレーム・バッファ領域に一度に１ラインずつ転送される。そして、転送動作は停止する。一方、宛先マルチライン・フレーム・バッファ領域に対応する最初のラスタ・ラインが、少なくとも現在のラスタ・ラインに最小のラスタ・ライン数を加算した値と等しくなければ、本願方法および状態機械は低速パスを選択する。低速パスでは、状態機械は、画素データのライン毎の転送のタイミングを、転送動作を通して常に現在のラスタより後であるように設定する。状態機械は、各ラインが転送される前に、宛先ラインが現在のラスタ・ラインより後であることを確かめるようチェックする。しかしながら、各ラインが書込まれた後、状態機械はまた、次の宛先ラインが現在のラスタラインより前であるか否かを判断する。これは、ラスタラインが垂直帰線を開始した直後に発生する。上記宛先ラインが現在のラスタラインより前である場合、低速パスプロシージャは中断され、状態機械が転送動作の合図に応じて高速パスに切換える。

【００１２】

本発明の実施の形態について、図面を参照して説明する。

【００１３】

【発明の実施の形態】

図１は、グラフィックス・サブシステムにおいて実現される本発明の好ましい実施の形態を有するコンピュータ・システムのブロック図である。コンピュータ・システム１００は、従来からのものでよく、コンピュータ・システム１００内に示す特定の要素は、単に例として与えている。当業者は、本発明が、図示しているようなコンピュータ・システムだけでなく他のバス・タイプおよびバス・アーキテクチャを有するコンピュータ・システムにおいて実現する場合にも適しているということを認めるであろう。ＣＰＵ１０２およびシステム・メモリ１０４が、システム・バス１０６に接続されている。複数のＩ／Ｏ装置１１２が、ＰＣＩバス１０８に接続されており、ＰＣＩバス１０８は、ＰＣＩバス・ブリッジ１１０を介してシステム・バス１０６に接続されている。グラフィックス・サブシステム１１８は、バス・インタフェース１２０を介してＡＧＰバス１１４に接続されている。更に、ＡＧＰバス１１４は、ＡＧＰバス・ブリッジ１１６を介してシステム・バス１０

６に接続されている。

【００１４】

グラフィックス・サブシステム１１８内のグラフィックス・パイプラインは、マクロ・ファンクション・ユニット１２２、ラステライザ（rasterizer）／テクスチャ・マッピング・サブシステム１２４、フレーム・バッファ・コントローラ１２６およびフレーム・バッファ・メモリ１２８を含む。フレーム・バッファ・コントローラ１２６は、フレーム・バッファ・メモリ１２８に対する読出しおよび書込みを制御し、フレーム・バッファ・メモリ１２８からの画素データをビデオ・コントローラ１３０に供給する。ビデオ・コントローラ１３０は、従来からのものでよく、ＲＡＭＤＡＣと、モニタ１３２を駆動するための必要な水平同期信号および垂直ブランキング信号を生成する回路とを含むものとする。また、上記水平同期信号（hsync）および垂直ブランキング信号（vblank）は、ワイヤ１３４によりマクロ・ファンクション・ユニット１２２に供給される。フレーム・バッファ・コントローラ１２６は、ステータス情報を、ワイヤ１３６によりマクロ・ファンクション・ユニット１２２に供給する。例えば、フレーム・バッファ・コントローラ１２６は、フレーム・バッファ・コントローラ１２６自身が、先に発行したグラフィックス・コマンドを実行中であるか、または新たなグラフィックス・コマンドを実行することが可能であるかを示すステータス信号を、マクロ・ファンクション・ユニット１２２に対して供給する。

【００１５】

ここで、フレーム・バッファ・コントローラ１２６が、あるフレーム・バッファ・メモリから他のフレーム・バッファ・メモリへ、または同じフレーム・バッファ・メモリ内のある位置から他の位置へ、１ラインＢＬＴ動作を実行することができるものと仮定する。このような１ラインＢＬＴ機能は、本技術分野では周知であり、ここでは詳述しない。但し、一般に、このような機能を使用して１ラインＢＬＴを実行するには、送り側のＸ，Ｙ先頭アドレス、宛先のＸ，Ｙ先頭アドレス、およびライン長を特定することだけが必要である。好ましくは、本発明を実現するのに必要なレジスタおよび状態機械は、マクロ・ファンクション・ユニット１２２に設けられ、フレーム・バッファ・コントローラ１２６によって与えられる機能を利用する。

【００１６】

図２は、バック（不可視）フレーム・バッファ・メモリ２００およびフロント（可視）フレーム・バッファ・メモリ２０２を示す図である。ここで、図２は、後述する説明において使用するいくつかの用語を定義するのを視覚的に支援するものである。送り側フレーム・バッファ領域２０４は、図示しているように、ｘ方向およびｙ方向の長さxlength, ylengthを有すると共に、最上左端の画素２０６を有している。ここで規則として、画素２０６は送り側フレーム・バッファ領域２０４の先頭アドレスを示す。また、先頭アドレスは、図示しているようにsourcex、sourceyとして書込まれる。同様に、宛先フレーム・バッファ領域２０８は、ｘ方向の長さおよびｙ方向の長さxlength, ylengthを有すると共に、最上左端の画素２１０を有している。また、規則として、画素２１０は宛先フレーム・バッファ領域２０８の先頭アドレスを示す。この先頭アドレスは、図示しているようにdestx、destyとして書込まれる。ラスタ・パターンにおいて最初の可視画素に対応するフレーム・バッファ・メモリ２０２内のロケーションは、２１２で示す。ラスタ・パターンにおいて最後の可視画素に対応するロケーションは、２１４で示す。ラスタは、一旦ロケーション２１４に対応する画素に到達すると、ディスプレイの最上左端の位置に素早く戻り、上述した待ち時間「vblank」の後、走査を再び開始する。「lines_in_vblank」という語は、vblank中の水平走査期間の数を示す。明確にするために、全ラスタ走査期間に必要な合計時間は、１水平走査期間に必要な時間に「lines_in_vblank＋画素２１２および画素２１４間（それらも含む）のライン数」を掛けた時間に等しいものとする。図中のライン２２０は、宛先フレーム・バッファ領域２０８の先頭位置からのオフセットとして計算されたｙ座標を示している。このオフセットを計算する目的は、図４の説明において後述する。

【００１７】
図３は、マクロ・ファンクション・ユニット１２２内に含まれる好ましいレジスタ３００のセットを示すブロック図である。フレーム・バッファＢＬＴ動作を要求する前に、ソフトウエアは、sourcex,ｙレジスタ３０２が画素２０６を示すように予め設定しておく。ソフトウエアは、destx,ｙレジスタ３０４が画素２１０を示すように予め設定しておく。ソフトウエアは、xlengthレジスタ３０６およびylengthレジスタ３０８を予め設定することにより、移動するブロックの所望の寸法を定義しておく。（好ましい実施の形態では、送り側フレーム・バッファ領域２０４および宛先フレーム・バッファ領域２０８は、常に同じ寸法および座標であるものとする。）
【００１８】
current_raster_lineレジスタ３１０は、計数値である。current_raster_lineレジスタ３１０の値を常に現在の値に保持する回路が必要である。計数値は、vblankの開始時はゼロにリセットし、各水平走査期間後に一度インクリメントしなければならない。これにより、画素２１２が表示される（フレーム・バッファの０のｙ座標に対応する）時間まで、current_raster_lineレジスタ３１０の値はlines_in_vblankに等しくなる。このような不一致のために、本方法を実現する状態機械は、vblankにいくつのラインがあるかを知っていなければならない。lines_in_vblankレジスタ３１２は、この目的で設けられている。これは、フレーム・バッファＢＬＴ動作を要求する前に、ソフトウエアによって予め設定されていなければならない。next_dest_lineレジスタ３１４もまた、カウンタとして使用される。その目的は、フレーム・バッファＢＬＴ動作で画素書込みを行う次のラインが、現在のラスタ位置より前または後のディスプレイ・ラインのいずれに対応するか、という判断を容易にすることである。図４の説明において後述するように、next_dest_lineレジスタ３１４の値は、フレーム・バッファのｙ座標およびディスプレイ・ラインのｙ座標の不一致（その差は、lines_in_vblankである）を考慮している。offset_lineレジスタ３１６の値は、ディスプレイ・ラインの座標を用いてライン２２０の位置を表している。それは、next_dest_lineの初期値－offsetレジスタ３１８の値として計算される。好ましい実施の形態では、使用するオフセットは１０ラインである。他の実施の形態として、他のオフセットを使用してもよい。このオフセットの目的は、図４の説明において後述する。
【００１９】
図４は、本発明に従い、送り側フレーム・バッファ領域２０４から宛先フレーム・バッファ領域２０８へ画素データのブロックを転送する方法を説明する流れ図である。好ましい実施の形態では、マクロ・ファンクション・ユニット１２２内において、状態機械が、レジスタ３０２～３１８に格納された値を使用して図４に示す方法ステップを実現するように構成されている。この明細書、特に図４およびそれを説明する本文を参照している当業者は、このような状態機械を容易に設計することができるであろう。この図面では、始めに、ステップ４００において、ソフトウエアは、フレーム・バッファＢＬＴ動作を要求する前に、offsetレジスタ３１８、lines_in_vblankレジスタ３１２、sourcex,ｙレジスタ３０２、destx,ｙレジスタ３０４、xlengthレジスタ３０６およびylengthレジスタ３０８に適当な値を予めロードする。なお、xlengthレジスタ３０６の値は、ディスプレイ・ラインにおける画素数以下である。これらの値を設定するとすぐに、ソフトウエアは、フレーム・バッファＢＬＴ動作を要求する。図４のステップ４０２は、状態機械がフレーム・バッファＢＬＴ動作が要求されるのを待っているアイドル状態に対応している。上記動作が要求されるとすぐに、ステップ４０４に進み、残りの必要な値を初期化する。すなわち、next_dest_lineレジスタ３１４の初期値は、desty＋lines_in_vblankに等しく設定される。offset_lineレジスタ３１６の初期値は、next_dest_line－offsetに等しく設定される。ステップ４０６では、状態機械は、フレーム・バッファ・コントローラ１２６が（ステータス・ライン１３６を介して）コマンドを受信する準備ができたことを示すまで待機する。この確認が受信されるとすぐに、ＢＬＴ動作の内容が開始される。
【００２０】
ステップ４０８において、offset_line２２０が現在のラスタ位置より前であるかが判断

される。これは、offset_lineレジスタ３１６とcurrent_raster_lineレジスタ３１０との差が、正、負またはゼロのいずれであるかを判断することによって行れる。オフセットより後であるということで、next_dest_lineが少なくとも、現在のラスタより前の最少のライン数であるならば、ＢＬＴ動作は、たとえ最悪状態であってもラスタに追いつくことなく実行される、ということが推定される。従って、ステップ４０８においてoffset_line２２０が現在のラスタより前であると判断された場合、ＢＬＴ動作は可能な限り高速に実行することができる。ブロックは、各ライン転送間で現在のラスタ位置を再度判断するというチェックを行うことなく、ライン毎に転送することができる。このような動作モードは、ステップ４２２～ステップ４２６（「高速パス」）に対応している。一方、ステップ４０８において、offset_line２２０が現在のラスタより前でないと判断された場合、高速パスを使用すると、ＢＬＴ動作が現在のラスタを追い越すか、もしくは現在のラスタがＢＬＴ動作を追い越すという可能性が生じる。どちらのイベントが発生しても、画像に裂け目が生じることがある。従って、ステップ４０８においてoffset_line２２０が現在のラスタより前でないと判断された場合、ステップ４１０～ステップ４２０（「低速パス」）を選択する。

【００２１】

高速パスの場合、ステップ４２２において、マクロ・ファンクション・ユニット１２２が、sourcex,ｙレジスタ３０２、destx,ｙレジスタ３０４およびxlengthレジスタ３０６の現在値を使用して、１ラインＢＬＴコマンドをフレーム・バッファ・コントローラ１２６に対して発行する。その後、ステップ４２６に示すとおり、sourcex,ｙレジスタ３０２（ｙの値のみ）、destx,ｙレジスタ３０４（ｙの値のみ）、ylengthレジスタ３０８およびnext_dest_lineレジスタ３１４の値を更新する。ステップ４２６において、ylengthレジスタ３０８の値がデクリメントされてゼロになったか否かを判断することにより、ＢＬＴ動作が終了したか否かをチェックする。終了した場合、動作は停止して、状態機械はアイドル状態４０２に戻り、終了していない場合、動作はステップ４２２を続行する。

【００２２】

低速パスの場合、状態機械は、１ラインＢＬＴがすべて現在のラスタ位置の後に発生するように、１ラインＢＬＴのタイミングを合わせる。従って、ステップ４１０において、状態機械は、next_dest_lineが現在のラスタ・ラインより後であるか否かをチェックする。これは、next_dest_lineレジスタ３１４の値をcurrent_raster_lineレジスタ３１０の値と比較することによって行う。ステップ４１０において、next_dest_lineが現在のラスタより後であると判断された場合、ステップ４１２において１ラインＢＬＴコマンドが発行される。そして、ステップ４２４と同様に、ステップ４１４において適切な値が更新される。ステップ４１６において、状態機械は、ylengthがデクリメントされてゼロになったか否かを判断することにより、ＢＬＴ動作が終了したか否かをチェックする。終了した場合、動作は停止して、状態機械はアイドル状態４０２に戻る。終了していない場合、動作はステップ４１８を続行する。

【００２３】

重要なことは、ステップ４１８において、状態機械が、next_dest_lineが現在のラスタ位置の前であるか否かをチェックすることである。これは、ラスタラインが垂直帰線を開始した直後に発生する。現在のラスタ位置の前である場合、next_dest_lineは現在のラスタ位置より前のlines_in_vblankの最小値となっているが、僅か前には現在のラスタより後であったものである。このような場合、高速パスを使用してＢＬＴ動作を終了する方が安全である。従って、ステップ４１８において、状態機械が、next_dest_lineが現在のラスタより前であると判断した場合、アルゴリズムはＢＬＴ動作の残りの動作について高速パス（ステップ４２２～ステップ４２６）への切替を行う。一方、next_dest_lineが現在のラスタより前でない場合、アルゴリズムはステップ４２０を続ける。ステップ４２０において、next_dest_lineが現在のラスタの後であるか否かが再び判断される。後である場合、動作はステップ４１２を続行する（次の１ラインＢＬＴ）。後でない場合、アルゴリズムはステップ４１８を繰返す。

【００２４】
以上、本発明の実施例について詳述したが、以下、本発明の各実施態様の例を示す。
【００２５】
（実施態様１）
ラスタ型コンピュータ・グラフィックス・ディスプレイ・システムにおいて、画素データのブロックを送り側マルチライン・フレーム・バッファ領域（２０４）から宛先マルチライン・フレーム・バッファ領域（２０８）に転送する方法であって、該方法は以下の（ａ）から（ｇ）のステップを含むことを特徴とする、
ａ）前記宛先マルチライン・フレーム・バッファ領域に対応する最初のラスタ・ラインが、少なくとも現在のラスタ・ラインに最小のラスタ・ライン数を加算した値と等しいかを判断（４０８）し、前記判断がが等しい場合には、ステップｂ）へ進み、前記判断が等しくない場合にはステップｃ）へ進むステップと、
ｂ）前記画素データのブロックにおける画素データの最後のラインが転送されるまで、画素データのラインを、前記送り側マルチライン・フレーム・バッファ領域（２０４）から前記宛先マルチライン・フレーム・バッファ領域（２０８）へ、一度に１ラインずつ転送し（４２２～４２６）、その後転送動作を停止するステップと、
ｃ）前記宛先マルチライン・フレーム・バッファ領域（２０８）に対応する最初のラスタ・ラインが現在のラスタ・ラインより後であるかを判断し（４１０）、後でない場合にはステップｃ）を再び開始し、後である場合にはステップｄ）へ進むステップと、
ｄ）次の送り側ライン指示手段（３０２）および次の宛先ライン指示手段（３０４）に従って、前記送り側マルチライン・フレーム・バッファ領域から前記宛先マルチライン・フレーム・バッファ領域へ画素データを１ライン転送するステップ（４１２）と、
ｅ）ステップｄ）で転送された画素データのラインが前記画素データのブロックの最後のラインであった場合、転送動作を停止（４１６）し、最後でない場合には、前記次の送り側ライン指示手段（３０２）および前記次の宛先ライン指示手段（３０４）を更新し、ステップｆ）へ進むステップと、
ｆ）前記次の宛先ライン指示手段に対応するラスタ・ラインが現在のラスタ・ラインより前であるかを判断し（４１８）、前であればステップｂ）へ進み、前でなければステップｇ）に進むステップと、
ｇ）前記次の宛先ライン指示手段に対応するラスタ・ラインが現在のラスタ位置より後であるかを判断し（４２０）、後でない場合には再びステップｇ）を開始し、後である場合にはステップｄ）へ進むステップ。
【００２６】
（実施態様２）
前記ステップｂ）およびｄ）は、ＢＬＴエンジンに対して１ラインＢＬＴコマンドを発行する（４２２，４１２）ステップを含むことを特徴とする実施態様１記載の方法。
【００２７】
（実施態様３）
前記ステップｂ）およびｄ）を、送り側ＸＹアドレス（３０２）、宛先ＸＹアドレス（３０４）、および１フル・ラスタ・ラインのＸ長より短いＸ長（３０６）に対応して実行することを特徴とする実施態様１記載の方法。
【００２８】
（実施態様４）
前記送り側マルチライン・フレーム・バッファ領域（２０４）は、第１のフレーム・バッファ・メモリ（２００）に配置し、前記宛先マルチライン・フレーム・バッファ領域（２０８）は、第２のフレーム・バッファ・メモリ（２０２）に配置することを特徴とする実施態様１記載の方法。
【００２９】
（実施態様５）
前記送り側マルチライン・フレーム・バッファ領域（２０４）および前記宛先マルチライ

ン・フレーム・バッファ領域（２０８）は、同じフレーム・バッファ・メモリ（２００または２０２）内の第１の領域および第２の領域にそれぞれ対応していることを特徴とする実施態様１記載の方法。

【００３０】

本発明について、その好ましい実施の形態に関連して詳細を説明してきたが、説明した実施の形態は単なる例として示しているだけであり、限定するためのものではないということは理解されるべきである。当業者には、特許請求の範囲およびそれらの等価物によって定義されるような本発明の精神および範囲を離脱せずに、説明した実施の形態の形式および詳細についてあらゆる変更が可能であることは理解されるであろう。たとえば、説明した実施の形態は１つのフレーム・バッファ・メモリから他のフレーム・バッファ・メモリへブロックを移動させることを企図しているが、本発明は、表示画像に不要なアーティファクトをもたらすことなく、同じフレーム・バッファ・メモリにおける１つのロケーションから他のロケーションへ迅速にブロックを移動させることで実現することもできる。

【図面の簡単な説明】

【図１】　本発明の好ましい実施の形態を実現するコンピュータシステムを示すブロック図である。

【図２】　送り側および送り先フレーム・バッファ領域をそれぞれ含むバックおよびフロント・フレーム・バッファ・メモリを示す図である。

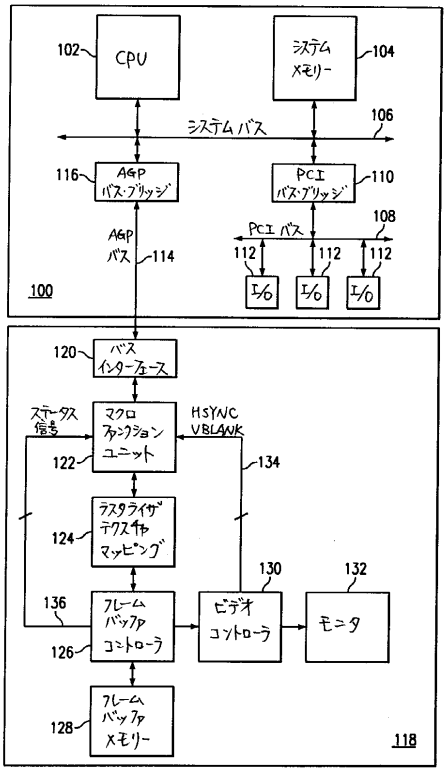
【図３】　図１に示すマクロ・ファンクション・ユニット内に含まれるレジスタのセットを示すブロック図である。

【図４】　本発明の好ましい実施の形態による送り側フレーム・バッファ領域から送り先フレーム・バッファ領域への画素・データのブロックを転送する方法を示す流れ図である。

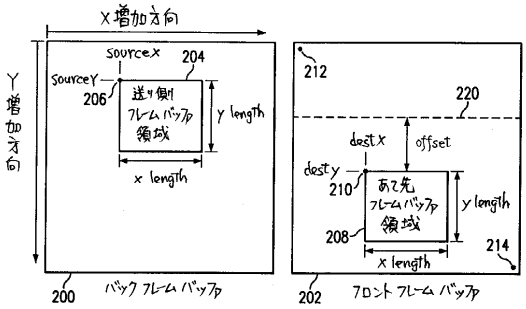
【符号の説明】

１００：コンピュータ・システム
１０２：ＣＰＵ
１０４：システム・メモリ
１０６：システム・バス
１１２：Ｉ／Ｏ装置
１０８：ＰＣＩバス
１１０：ＰＣＩバス・ブリッジ
１１４：ＡＧＰバス
１１６：ＡＧＰバス・ブリッジ
１１８：グラフィックス・サブシステム
１２０：バス・インタフェース
１２２：マクロ・ファンクション・ユニット
１２４：ラステライザ／テクスチャ・マッピング・サブシステム
１２６：フレーム・バッファ・コントローラ
１２８：フレーム・バッファ・メモリ
１３０：ビデオ・コントローラ
１３２：モニタ
１３４：ワイヤ

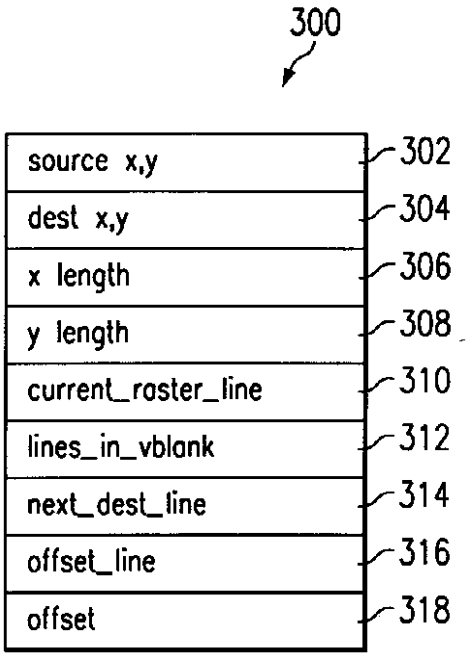
【図１】


CPU
102
システム
メモリー
104
システムバス
106
AGP
バス・ブリッジ
116
PCI
バス・ブリッジ
110
AGP
バス
114
PCI バス
108
112
112
112
I/O
I/O
I/O
100
バス
インターフェース
120
ステータス
信号
マクロ
ファンクション
ユニット
122
HSYNC
VBLANK
134
ラスタライザ
テクスチャ
マッピング
124
130
132
136
フレーム
バッファ
コントローラ
ビデオ
コントローラ
モニタ
126
フレーム
バッファ
メモリー
128
118

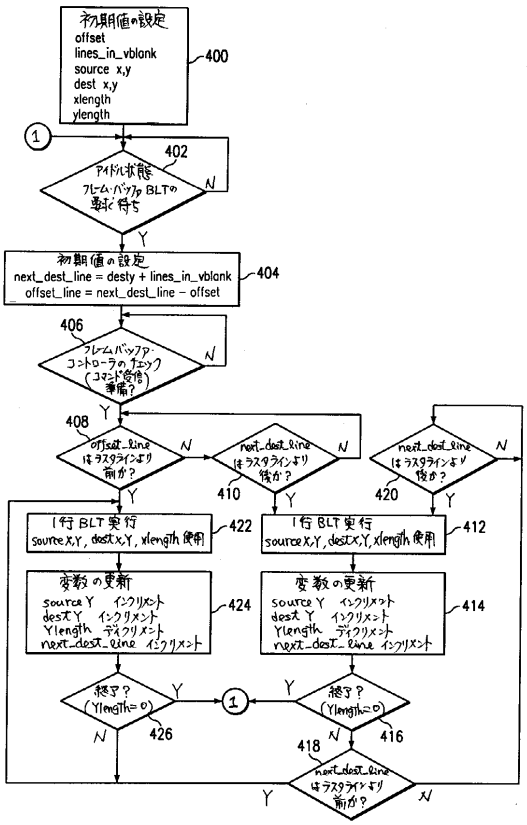
【図２】


X増加方向
Y増加方向
source x
204
source y
206
送り側
フレームバッファ
領域
y length
x length
200
バックフレームバッファ
212
220
dest x
offset
dest y
210
あて先
フレームバッファ
領域
y length
208
214
x length
202
フロントフレームバッファ

【図３】


300
source x,y
302
dest x,y
304
x length
306
y length
308
current_raster_line
310
lines_in_vblank
312
next_dest_line
314
offset_line
316
offset
318

【図４】


初期値の設定
offset
lines_in_vblank
source x,y
dest x,y
xlength
ylength
400
1
アイドル状態
フレームバッファ BLTの
受け待ち
402
N
Y
初期値の設定
next_dest_line = desty + lines_in_vblank
offset_line = next_dest_line − offset
404
406
フレームバッファ
コントローラのチェック
(コマンド受信)
準備?
408
offset_line
はラスタラインより
前か?
next_dest_line
はラスタラインより
後か?
410
next_dest_line
はラスタラインより
後か?
420
1行 BLT 実行
source x,y, dest x,y, xlength 使用
422
1行 BLT 実行
source x,y, dest x,y, xlength 使用
412
変数の更新
source Y インクリメント
dest Y インクリメント
Ylength ディクリメント
next_dest_line インクリメント
424
変数の更新
source Y インクリメント
dest Y インクリメント
Ylength ディクリメント
next_dest_line インクリメント
414
終了?
(Ylength=0)
426
終了?
(Ylength=0)
416
418
next_dest_line
はラスタラインより
前か?

フロントページの続き

(72)発明者　ジョエル・ディ・バックジェングラー
アメリカ合衆国コロラド州ロングモント　ブキャナン・レーン８２４

審査官　圓道　浩史

(56)参考文献　特開昭５８－１２３１３８（ＪＰ，Ａ）
特開昭６０－１６９９３５（ＪＰ，Ａ）
特開昭６３－２００１８９（ＪＰ，Ａ）
特開昭６３－２３４３１９（ＪＰ，Ａ）
特開平０１－２９００３０（ＪＰ，Ａ）
特開平０５－１２７６６２（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.[7]，ＤＢ名)
G06F 3/14 - 3/153
G09G 5/00 - 5/40