

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第3969017号

(P3969017)

(45) 発行日 平成19年8月29日(2007.8.29)

(24) 登録日 平成19年6月15日(2007.6.15)

(51) Int. Cl.

F I

G09G 5/00 (2006.01)

G09G 5/00 550R

G06T 1/60 (2006.01)

G09G 5/00 555J

G09G 5/02 (2006.01)

G09G 5/00 550T

G06T 1/60 450E

G06T 1/60 450F

請求項の数 8 (全 35 頁) 最終頁に続く

(21) 出願番号 特願2001-170165 (P2001-170165)
 (22) 出願日 平成13年6月5日(2001.6.5)
 (65) 公開番号 特開2002-366131 (P2002-366131A)
 (43) 公開日 平成14年12月20日(2002.12.20)
 審査請求日 平成16年9月24日(2004.9.24)

(73) 特許権者 000004075
 ヤマハ株式会社
 静岡県浜松市中区中沢町10番1号
 (74) 代理人 100092820
 弁理士 伊丹 勝
 (72) 発明者 榎本 保宏
 静岡県浜松市中区中沢町10番1号 ヤマハ株
 式会社内

審査官 後藤 亮治

最終頁に続く

(54) 【発明の名称】 画像データ転送装置及び画像表示処理システム

(57) 【特許請求の範囲】

【請求項1】

指定された転送元領域及び転送先領域に関するパラメータに基づいて画像データ記憶装置に記憶された前記転送元領域の画像データと前記転送先領域の画像データとを演算処理した後に前記転送先領域に転送する画像データ転送装置において、

前記転送元領域の画像データは、1ピクセルが1ビットで構成されたモノクロームの画像データ、前記転送先領域の画像データは、1ピクセルが1又は複数のバイトで構成された画像データであり、

前記画像データ記憶装置の転送元領域及び転送先領域から1ピクセルの画像データよりも大きいブロック単位で画像データを読み出すアドレスカウンタと、

前記画像データ記憶装置から読み出されたブロック単位の画像データを記憶すると共に、転送元領域の1ピクセルをそのビット値に応じて予め設定された2色のデータのうちの1色を表す1又は複数のバイトで構成された画像データに拡張するデータ拡張機能付きのデータ格納手段と、

前記拡張された転送元領域の画像データと前記転送先領域の画像データとでブロック内における位置合わせを行って前記演算処理を実行する演算手段とを備え、

前記データ格納手段は、転送元領域の各ピクセルの1ビットのデータを転送先領域の各ピクセルのバイト数に応じてコピーして、コピー後の各ビットにカラータグを設定して、予め設定された2色のデータのうちの1色を表す1又は複数のバイトのカラーピクセルデータのうち前記カラータグで指定されたバイトを割り当てることにより画像データを拡張

10

20

するモノクローム拡張手段を備えてなる

ことを特徴とする画像データ転送装置。

【請求項 2】

前記指定された転送元領域及び転送先領域に関するパラメータ及び1ピクセル当たりのバイト数に基づいて、前記転送元領域及び転送先領域を含む各走査ライン毎に転送開始ブロックアドレス及び転送終了ブロックアドレスを算出し前記アドレスカウンタにセットするアドレス計算手段を更に備え、

前記アドレスカウンタは、転送開始ブロックアドレスを含む転送開始ブロックから前記転送終了ブロックアドレスを含む転送終了ブロックまで連続的に前記画像データを転送する

10

ことを特徴とする請求項1記載の画像データ転送装置。

【請求項 3】

前記アドレスカウンタは、前記各ブロック内の転送画像開始バイトアドレス及び転送画像終了バイトアドレスに基づいて前記転送元領域及び転送先領域以外の領域をマスクするマスクパターンを生成し、

このマスクパターンを使用して前記転送元領域と転送先領域とでブロック内における位置合わせを行うマスク演算手段を更に備えた

ことを特徴とする請求項1又は2記載の画像データ転送装置。

【請求項 4】

前記1ピクセル分の画像データは、バイト毎にカラータグが設定されたものであり、

20

前記アドレスカウンタは、前記ブロック内の転送画像開始バイトアドレスに基づいて各バイトのカラータグを設定するものである

ことを特徴とする請求項1～3のいずれか1項記載の画像データ転送装置。

【請求項 5】

画像データを記憶する画像データ記憶装置と、

この画像データ記憶装置内に記憶された画像データの転送元領域及び転送先領域に関するパラメータを出力する中央処理装置と、

この中央処理装置から出力された転送元領域及び転送先領域に関するパラメータに基づいて画像データ記憶装置に記憶された前記転送元領域の画像データと前記転送先領域の画像データとを演算処理した後に前記転送先領域に転送する画像データ転送装置と、

30

前記画像データ記憶装置に記憶された画像データを表示する画像表示装置と

を備えた画像表示処理システムにおいて、

前記転送元領域の画像データは、1ピクセルが1ビットで構成されたモノクロームの画像データ、前記転送先領域の画像データは、1ピクセルが1又は複数のバイトで構成された画像データであり、

前記中央処理装置は、前記1ピクセルを構成するバイト数を前記画像データ転送装置に出力し、

前記画像データ転送装置は、前記1ピクセルの画像データよりも大きいブロック単位で画像データを転送し、且つ転送元領域の1ピクセルをそのビット値に応じて予め設定された2色のデータのうちの1色を表す1又は複数のバイトで構成された画像データに拡張し、この拡張された転送元領域の画像データと転送先領域の画像データとでブロック内における位置合わせを行って前記演算処理を実行したのち、前記画像データ記憶装置に格納すると共に、

40

前記画像データ転送装置は、転送元領域の各ピクセルの1ビットのデータを転送先領域の各ピクセルのバイト数に応じてコピーして、コピー後の各ビットにカラータグを設定して、予め設定された2色のデータのうちの1色を表す1又は複数のバイトのカラーピクセルデータのうち前記カラータグで指定されたバイトを割り当てることにより画像データを拡張するものである

ことを特徴とする画像表示処理システム。

【請求項 6】

50

前記画像データ転送装置は、

前記指定された転送元領域及び転送先領域に関するパラメータ及び1ピクセル当たりのバイト数に基づいて、前記転送元領域及び転送先領域を含む各走査ライン毎に転送開始ブロックアドレス及び転送終了ブロックアドレスを算出し、

前記転送開始ブロックアドレスを含む転送開始ブロックから前記転送終了ブロックアドレスを含む転送終了ブロックまで連続的に前記画像データを転送するものである

ことを特徴とする請求項5記載の画像表示処理システム。

【請求項7】

前記画像データ転送装置は、

前記各ブロック内の転送画像開始バイトアドレス及び転送画像終了バイトアドレスに基づいて前記転送元領域及び転送先領域以外の領域をマスクするマスクパターンを生成すると共に、このマスクパターンを使用して前記転送元領域と転送先領域とでブロック内における位置合わせを行うものである

ことを特徴とする請求項5又は6記載の画像表示処理システム。

【請求項8】

前記1ピクセル分の画像データは、バイト毎にカラータグが設定されたものであり、

前記画像データ転送装置は、前記ブロック内の転送画像開始バイトアドレスに基づいて各バイトのカラータグを設定するものである

ことを特徴とする請求項5～7のいずれか1項記載の画像表示処理システム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

この発明はCRT (Cathode Ray Tube) 等のラスタースキャン型表示装置において、平面上に表現された画像を構成するデータの画像表示処理システムに関し、特に特定領域の画像データを転送する画像データ転送装置及び画像表示処理システムに関する。

【0002】

【従来の技術】

従来より、CRT等のラスタースキャン型表示装置を使用した2次元画像データ処理装置としては、例えば特開昭60-214392号公報に開示されているものが知られている。この画像表示処理システムは、図41に示すように、中央演算処理装置(CPU)202の負担を軽減するため、画像表示処理のためのディスプレイコントローラ201が備えられる。ディスプレイコントローラ201内部の画像データ処理回路210は、CRT表示装置205の画面の走査速度に対応してビデオRAM(以下、VRAMと呼ぶ。)204内に記憶されている静止画像データ及び動画像データ等をインターフェイス211を介して読み出すと共に、CRT表示装置205へ画像の走査に必要な同期信号SYNCを出力する。

【0003】

この場合の静止画及び動画像データは、表示画面上のドットの色を指定する2, 4又は8ビットのカラーコードからなり、画像データ処理回路210は、読み出したカラーコードをカラーパレット212に出力する。カラーパレット212は、読み出したカラーコードをRGB(赤, 緑, 青)信号に変換してCRT表示装置に供給する。

【0004】

また、画像データ処理回路210は、CPU202からインターフェイス213を介して供給される画像データを、画面の非表示期間(垂直帰線期間等)にVRAM204に書き込む。更に、VRAM204をアクセスしているとき(書き込み及び読み出し時)は、信号S1をコマンド処理回路215に供給してアクセス中であることを知らせる。コマンド処理回路215は、CPU2からインターフェイス213を介して供給される各種のコマンドに対応する処理を行う。

【0005】

上記の構成の画像表示処理システムにおいては、静止画像を含む矩形領域の移動を短時間

10

20

30

40

50

でCPUを介さずに行うことや、上述したカラーコードの転送に際して、転送すべきドットのカラーコードの各ビットと転送先のドットのカラーコードの各ビットとの間の論理演算を行い、この結果を転送先のドットに対応するVRAM204の記憶エリアへ書き込む論理演算処理及び転送元領域の各ドットのカラーコードのうち透明のカラーコードについてはカラーコードの転送を行わず、それ以外のカラーコードについてのみ転送を行う透明処理(トランスペアレント処理)等を行うことができる。また、このシステムでは、カラーコードをドット単位で転送するドット単位転送モードと、カラーコードをバイト単位で転送するバイト単位転送モードとを備えている。

【0006】

【発明が解決しようとする課題】

しかしながら、上述した従来の画像表示処理システムでは、高速転送を実現しつつ所定の論理演算処理や透明処理を実行することができなかった。特に現在では画像処理技術の著しい向上やそれに伴う画像データ転送量の急激な増大等によって、上述したような装置ではもはや高速転送と細かな画像処理の要請とに因應することが困難になってきた。更に、定型的なパターンであっても、そのカラーコードが種々変化する場合には、カラーコード毎にソースデータを記憶しなければならず、記憶すべきデータ量が多くなるという問題もあった。

【0007】

この発明は、このような問題点に鑑みてなされたもので、更に効率的に画像処理及び転送データ制御等を行うことができると共にメモリ容量の削減を図ることができる画像データ転送装置及び画像表示処理システムを提供することを目的とする。

【0008】

【課題を解決するための手段】

この発明に係る画像データ転送装置は、指定された転送元領域及び転送先領域に関するパラメータに基づいて画像データ記憶装置に記憶された前記転送元領域の画像データと前記転送先領域の画像データとを演算処理した後に前記転送先領域に転送する画像データ転送装置において、前記転送元領域の画像データは、1ピクセルが1ビットで構成されたモノクロームの画像データ、前記転送先領域の画像データは、1ピクセルが1又は複数のバイトで構成された画像データであり、前記画像データ記憶装置の転送元領域及び転送先領域から1ピクセルの画像データよりも大きいブロック単位で画像データを読み出すアドレスカウンタと、前記画像データ記憶装置から読み出されたブロック単位の画像データを記憶すると共に、転送元領域の1ピクセルをそのビット値に応じて予め設定された2色のデータのうちの1色を表す1又は複数のバイトで構成された画像データに拡張するデータ拡張機能付きのデータ格納手段と、前記拡張された転送元領域の画像データと前記転送先領域の画像データとでブロック内における位置合わせを行って前記演算処理を実行する演算手段とを備え、前記データ格納手段は、転送元領域の各ピクセルの1ビットのデータを転送先領域の各ピクセルのバイト数に応じてコピーして、コピー後の各ビットにカラータグを設定して、予め設定された2色のデータのうちの1色を表す1又は複数のバイトのカラーピクセルデータのうち前記カラータグで指定されたバイトを割り当てることにより画像データを拡張するモノクローム拡張手段を備えてなることを特徴とする。

【0009】

また、この発明に係る画像表示処理システムは、画像データを記憶する画像データ記憶装置と、この画像データ記憶装置内に記憶された画像データの転送元領域及び転送先領域に関するパラメータを出力する中央処理装置と、この中央処理装置から出力された転送元領域及び転送先領域に関するパラメータに基づいて画像データ記憶装置に記憶された前記転送元領域の画像データと前記転送先領域の画像データとを演算処理した後に前記転送先領域に転送する画像データ転送装置と、前記画像データ記憶装置に記憶された画像データを表示する画像表示装置とを備えた画像表示処理システムにおいて、前記転送元領域の画像データは、1ピクセルが1ビットで構成されたモノクロームの画像データ、前記転送先領域の画像データは、1ピクセルが1又は複数のバイトで構成された画像データであり、

10

20

30

40

50

前記中央処理装置は、前記1ピクセルを構成するバイト数を前記画像データ転送装置に出力し、前記画像データ転送装置は、前記1ピクセルの画像データよりも大きいブロック単位で画像データを転送し、且つ転送元領域の1ピクセルをそのビット値に応じて予め設定された2色のデータのうちの1色を表す1又は複数のバイトで構成された画像データに拡張し、この拡張された転送元領域の画像データと転送先領域の画像データとでブロック内における位置合わせを行って前記演算処理を実行したのち、前記画像データ記憶装置に格納すると共に、前記画像データ転送装置は、転送元領域の各ピクセルの1ビットのデータを転送先領域の各ピクセルのバイト数に応じてコピーして、コピー後の各ビットにカラータグを設定して、予め設定された2色のデータのうちの1色を表す1又は複数のバイトのカラーピクセルデータのうち前記カラータグで指定されたバイトを割り当てることにより画像データを拡張するものであることを特徴とする。

10

【0010】

この発明によれば、指定された転送元及び転送先領域（以下、この段落のみ「転送領域」とする。）に関するパラメータに基づき、画像データ記憶装置に記憶された転送領域の画像データを演算処理し転送する際に、画像データを1ピクセルが1又は複数のバイトで構成された画像データであるとして、この1ピクセルの画像データよりも大きい複数バイトからなるブロック単位で画像データを転送し、転送領域でブロック内における位置合わせを行い演算処理を実行することで、画像データの高速転送をすることができる。

【0011】

また、この発明では、転送元領域の画像データは、1ピクセルが1ビットで構成されたモノクロームの画像データであり、転送処理の際に、転送元領域の1ピクセルをそのビット値に応じて予め設定された2色のデータのうちの1色を表す1又は複数のバイトで構成された画像データに拡張するようにしているので、画像データ記憶手段に記憶させる転送元領域の画像データのデータ量を削減することができる。特に、同一パターンで異なる複数種類のカラーのパターンを表示するような場合、パターン自体は1種類だけを記憶し、その着色データを種々変えるようにすれば良く、記憶すべきデータ量を格段に削減することができる。

20

【0012】

なお、この発明において、各走査ライン毎に転送開始及び転送終了ブロックアドレスを算出し、転送開始ブロックから転送終了ブロックまで連続的に画像データを転送するようにすると、画像データを高速に転送することができる。更に、この発明において、各ブロック内の転送画像開始及び終了バイトアドレスに基づき転送領域以外をマスクするマスクパターンを生成し、転送領域でブロック内における位置合わせを行うようにすると、ピクセルが何バイトで構成されていても任意の位置に記憶された転送元及び転送先のデータの位置を合わせて演算処理を行うことができる。また、画像データ1ピクセル分のバイト毎にカラータグが設定されブロック内の転送画像開始バイトアドレスに基づき各バイトのカラータグが設定されることにより、画像表示処理効率が向上する。従って、画像表示処理システムにこの画像データ転送装置を組み込むことで、画像表示処理の高速化及び高効率化を図ることが可能となる。

30

【0013】

なお、この発明のより具体的な態様においては、前記画像データの拡張は、前記転送元領域の各ピクセルの1ビットのデータを各ピクセルのバイト数に応じてコピーして、コピー後の各ビットにカラータグを設定して、予め設定された2色のデータのうちの1色を表す1又は複数のバイトのカラーピクセルデータのうち前記カラータグで指定されたバイトを割り当てることによりおこなうようにすれば良い。

40

【0014】**【発明の実施の形態】**

以下、図面を参照して、この発明に係る画像表示処理システムの実施例を説明する。図1は、この発明の一実施例に係る画像表示処理システムの基本構成を説明するためのブロック図である。

50

この画像表示処理システムは、表示すべき画像データを記憶するD R A M (Dynamic Random Access Memory) 等からなるローカルメモリ4と、このローカルメモリ4に記憶された画像データの任意の矩形領域を転送するための種々のパラメータを出力するC P U 1と、このC P U 1から与えられるパラメータに基づいて前記ローカルメモリ4上の矩形領域の画像データを転送処理する画像データ転送装置2と、この画像データ転送装置2とローカルメモリ4との間のインターフェイスであって画像データのアクセスを制御するメモリコントローラ3と、ローカルメモリ4のスクリーンエリアの画像データを表示するC R Tディスプレイ、液晶ディスプレイ等の表示装置5とを備えて構成されている。

【0015】

このうち画像データ転送装置2の機能の概略を図2に示す。画像データ転送装置2は、ローカルメモリ4上の非表示領域に記憶された転送元データ(以下、ソースデータと呼ぶ。)Sの矩形領域及び表示領域に記憶された転送先データ(以下、デスティネーションデータと呼ぶ。)Dの矩形領域をそれぞれ定義するパラメータ、並びにソースデータSに付加される非表示領域に記憶された任意のパターンデータPの矩形領域を定義するパラメータをC P U 1から受け取り、ローカルメモリ4からソース、デスティネーション及びパターンの各データを取り込み、これらのデータ間で所定のラスタ演算処理を施してデスティネーションエリアに書き込む処理を実行する。

【0016】

ここで、ソースデータSは、各ピクセルが1又は複数のバイトからなるカラーデータである場合の他に、各ピクセルが2色で定義されるモノクロームデータである場合も許容する。ソースデータがモノクロームデータの場合、1ピクセルは1ビットで構成される。図3は、8ピクセル×8ラインで“x”印をモノクロームのソースパターンとして記憶した例を示している。図3の場合、8ピクセル×8ラインのパターンには、8バイトの容量しか必要としないので、メモリへの記憶容量を大幅に削減できる。ここで、例えば各ピクセルの値が0の場合はバックグラウンドカラー、1の場合はフォアグラウンドカラーであると定義しておく。バックグラウンドカラー及びフォアグラウンドカラーは、この例では、1ピクセルにつき1~4バイトで構成され、予め後述するレジスタに転送されストアされている。この1ピクセルを構成するバイト数を、ここではB P P (バイト・パー・ピクセル) と呼ぶ。ソースデータがそれ自体カラーデータである場合には、1ピクセルがB P P で定義される。画像データ転送装置2は、ソースデータがモノクロームデータで定義される場合、ソースデータの1ピクセルに相当する1ビットのデータを、0がバックグラウンドカラー、1がフォアグラウンドカラーとしてB P P で定義されているバイト数に拡張し、デスティネーションエリアDに転送する。

【0017】

以下、この画像データ転送装置2について詳細に説明する。

図4は、この画像データ転送装置2の詳細な構成を示すブロック図である。

C P U 1から送られてくるデスティネーションエリア、ソースエリア及びパターンエリアを特定するパラメータは、インターフェイス11を介してデスティネーションアドレス計算回路12、ソースアドレス計算回路13及びパターンアドレス計算回路14にそれぞれ供給される。これらのアドレス計算回路12, 13, 14では、1スキャンライン毎にデスティネーションエリア、ソースエリア及びパターンエリアのローカルメモリ4上での転送開始を示すスタートアドレス(F B S P n)と転送終了を示すエンドアドレス(F B E P n)とを計算する。その値がそれぞれデスティネーションアドレスカウンタ15、ソースアドレスカウンタ16及びパターンアドレスカウンタ17に渡される。

【0018】

なお、ソースアドレス計算回路13は、図5に示すように、カラーアドレス計算回路131と、モノクロームアドレス計算回路132と、これらの出力を選択するマルチプレクサ133とを備える。マルチプレクサ133は、インターフェイス11に設けられたM O N Oレジスタ(図示せず)を参照することにより、ソースデータがカラーデータかモノクロームデータかを判断し、出力を切り換える。これにより、ソースデータがカラーデータで

10

20

30

40

50

ある場合には、カラーソースアドレス計算回路131の出力をソースアドレスカウンタ16に出力し、ソースデータがモノクロームデータである場合には、モノクロームソースアドレス計算回路132の出力をソースアドレスカウンタ16に出力する。

【0019】

一方、この画像データ転送装置2には、ローカルメモリ4から転送された画像データを一時格納するための3つのSRAM(Static Random Access Memory)、即ち、デスティネーションSRAM18、ソースSRAM19及びパターンSRAM20が備えられている。各アドレスカウンタ15, 16, 17は、ローカルメモリ4をセクタ、ブロック及びバイトの単位にそれぞれ階層化し、各SRAM18, 19, 20に転送するセクタのアドレスを出力する。これにより、一セクタ分のデータがブロック単位でSRAM18, 19, 20に転送される。

10

【0020】

また、転送開始ピクセルと転送終了ピクセルとが丁度ブロックの先頭及び末尾であるとは限らないので、転送開始及び転送終了の際のブロックには、転送すべきピクセルデータ以外のデータが含まれることがある。各アドレスカウンタ15, 16, 17は、これらのデータをマスクするためのマスクデータを生成し、マスク演算回路23に供給する。マスク演算回路23は、入力されたマスクデータに基づいて、各SRAM18, 19, 20からデータを読み出して、ラスタ演算回路21にデータを送るための計算を実行する。ラスタ演算回路21は、各SRAM18, 19, 20からセクタデータを1ブロック単位で読み出し、マスク演算回路23からの演算結果も読み出してラスタ演算を行い、その演算結果を出力FIFO22に格納する。FIFO22に格納されたデータは、所定のタイミングでローカルメモリ4のデスティネーションエリアに転送される。コントローラ24は、CPU1からの制御命令に従い、各回路をコントロールする。なお、ソースSRAM19の内部には、後述するモノクローム拡張装置96が設けられており、ソースデータがモノクロームデータで定義される場合、ソースデータの1ピクセルに相当する1ビットのデータを、0がバックグラウンドカラー、1がフォアグラウンドカラーとしてBPPで定義されているバイト数に拡張し、デスティネーションエリアに転送する。

20

【0021】

図6は、この画像データ転送装置2の処理の流れを示すフローチャートである。

まず、各アドレス計算回路12, 13, 14の内部に設定される転送スキャンラインYの初期値として $Y = Y_S$ がロードされる(S1)。次に、そのスキャンラインのスタートアドレス(FBSPn)とエンドアドレス(FBEPn)とが、それぞれアドレス計算回路12, 13, 14で計算され(S2)、その値がそれぞれアドレスカウンタ15, 16, 17に渡される。ソースデータがモノクロームデータの場合、アドレス計算回路13でスタートアドレス(MSPj)とエンドアドレス(MEPj)とが計算され(S2)、その値がアドレスカウンタ16に渡される。各アドレスカウンタ15, 16, 17で生成されるアドレスに従ってローカルメモリ4から各SRAM18, 19, 20に1セクタ分のデスティネーションデータD、ソースデータS及びパターンデータPがそれぞれ転送される(S3, S4)。SRAM18, 19, 20への転送後は、1ブロック単位で、各SRAM18, 19, 20からデータを読み出す。また、ソースデータがモノクロームデータの場合には、BPPに応じてデータが拡張されてソースSRAM19から読み出される。読み出されたデータは、マスク演算回路23による演算結果に従い、ラスタ演算回路21にてラスタ演算され、出力FIFO22にストアされる(S5)。

30

40

【0022】

もし、ソースSRAM19にストアされているセクタデータの処理が終了した場合には(S7)、次のセクタデータを転送する(S4)。また、デスティネーションSRAM18にストアされているセクタデータの処理が終了した場合には、ラスタ演算終了後、出力FIFO22にストアされたデータをローカルメモリ4に書き込み(S8)、新たなセクタデータをデスティネーションSRAM18に転送する(S2)。また、パターンSRAM20にストアされているセクタデータの処理が終了した場合には(S7)、次のセクタデ

50

ータを転送する (S 4)。

【 0 0 2 3 】

以上の処理を繰り返し、1ライン分のデータの処理が終了したら (S 9)、Yを更新し (S 1 0)、次のラインの処理に移る。そして、最後のラインの処理を終了したら、矩形領域の処理は終了する (S 1 1)。

【 0 0 2 4 】

次に、この画像データ転送装置 2 のより具体的な動作について説明する。

図 7 (a) は、転送矩形領域を含む画像データの矩形 (表示) 領域を更に詳細に示す図である。この矩形領域は、デスティネーションデータ D についてはスクリーン領域、ソースデータ S 及びパターンデータ P についてはオフスクリーン領域に対応する。また、ソースデータについては、カラーデータの転送イメージを示している。ここに表示された各パラメータは、前述したように CPU 1 から画像データ転送装置 2 に与えられるものであって、次の通りである。

【 0 0 2 5 】

B A S E : 転送矩形領域を含む矩形領域の基準位置 (通常は領域の最も左上の画素の位置) に対応するローカルメモリ 4 上の座標値をバイトで表したデータ。スクリーンエリアの基点を示す場合もあれば、オフスクリーンエリアの基点を示す場合もある。

P T C H : 転送矩形領域を含む矩形領域の 1 ラインの幅をバイト数で表したデータ。

X S : 転送矩形領域の転送開始 X 座標値をピクセルで表したデータ。

Y S : 転送矩形領域の転送開始 Y 座標値をスキャンラインで表したデータ。

X E X T : 転送矩形領域の X 方向の幅をピクセル数で表したデータ。

Y E X T : 転送矩形領域の Y 方向の幅をスキャンライン数で表したデータ。

【 0 0 2 6 】

X D I R : 転送が X の正・負のいずれの方向から行われるかを示したデータで、0 のとき正 (右向き) で、1 のとき負 (左向き)。即ち、図 8 に示すように、X D I R が 0 の場合、X S は転送矩形領域の左端となり、X は X S から正の方向に更新される。また、X D I R が 1 の場合、X S は転送矩形領域の右端となり、X は X S から負の方向に更新される。

【 0 0 2 7 】

Y D I R : 転送が Y の正・負のいずれの方向から行われるかを示したデータで、0 のとき正 (下向き) で、1 のとき負 (上向き)。即ち、図 8 に示すように、Y D I R が 0 の場合、Y S は転送矩形領域の上端となり、Y は Y S から正の方向に更新される。また、Y D I R が 1 の場合、Y S は転送矩形領域の下端となり、Y は Y S から負の方向に更新される。

【 0 0 2 8 】

ここで、スキャンライン $Y = n$ で与えられたときの第 n ライン内のスタートアドレス (F B S P n) 及びエンドアドレス (F B E P n) は、X D I R = 0 のとき、次式で与えられる。

【 0 0 2 9 】

【 数 1 】

$$F B S P n = B A S E + n \times P T C H + X S \times B P P$$

$$F B E P n = B A S E + n \times P T C H + (X S + X E X T) \times B P P - 1$$

【 0 0 3 0 】

また、X D I R = 1 のときは、次式で与えられる。

【 0 0 3 1 】

【 数 2 】

$$F B S P n = B A S E + n \times P T C H + (X S + 1) \times B P P - 1$$

$$F B E P n = B A S E + n \times P T C H + (X S - X E X T + 1) \times B P P$$

【 0 0 3 2 】

図 7 (b) は、B A S E = 0 としたときの $Y = n$ におけるスタートアドレス (F B S P n) 及びエンドアドレス (F B E P n) をローカルメモリ 4 上の連続したデータとして示した図である。

10

20

30

40

50

【 0 0 3 3 】

一方、ソースデータがモノクロームデータの場合、CPU 1 から画像データ転送装置 2 に与えられるパラメータは、図 9 に示すように、次の通りとなる。

【 0 0 3 4 】

S B A S E : ソースデータ S を格納する基準となる座標値のローカルメモリ 4 上の座標値をバイトで表したデータ。

M O F S T : モノクロームのオフセットであり、S B A S E からのモノクロームデータの格納開始位置をバイト数で表したもの。複数のモノクロームデータを格納した場合、M O F S T の値を変えることにより、その格納開始位置を指定できる。

M P T C H : モノクロームピクセルピッチであり、モノクロームパターンの 1 ライン中のピクセル数を定義する。1 ライン中のピクセル数は、8 の倍数を基準として定義される。即ち、

【 0 0 3 5 】

M P T C H = 1 のとき 8 ピクセル / ライン、
M P T C H = 2 のとき 1 6 ピクセル / ライン、
M P T C H = 3 のとき 2 4 ピクセル / ライン、
M P T C H = 4 のとき 3 2 ピクセル / ライン、

【 0 0 3 6 】

のように定義される。図 3 のパターンは、M P T C H = 1 の例である。

転送先のデスティネーションの領域のサイズと、ソースの領域のサイズとは一致していなくてはならない。転送はバイト単位のアドレスで指定されるため、デスティネーションのサイズが 8 の倍数でない場合は、それ以下である最大の整数が M P T C H として採用される。即ち、 $8 * (M P T C H - 1) < X E X T$ (デスティネーションの X 転送領域幅) $8 * M P T C H$ となる M P T C H が選ばれる。8 の倍数でない場合、各ラインのモノクロームデータは、図 9 に示すように、有効なモノクロームデータがバイトの先頭に合うように格納される。図示の例は、1 ライン中のピクセル数が 1 4 の例であり、この場合、M P T C H として 2 が選択され、2 バイトのデータの 1 ビット目から 1 4 ビット目までにモノクロームデータが割り当てられている例である。

【 0 0 3 7 】

ソースデータがモノクロームデータである場合、デスティネーションへの転送は、常に X , Y とともに正の方向に行われるものとする。デスティネーションのスキャンライン Y に、スタートのスキャンラインである Y S がセットされると同期して、モノクロームパターン内のスキャンラインもインクリメントされる。モノクロームソースアドレス計算回路 1 3 2 では、 $Y = j$ で与えられたときの、第 j ライン内のソースのスタートアドレス (M S P j) 及びエンドアドレス (M E P j) を、以下の式で計算する。

【 0 0 3 8 】

【 数 3 】

$$M S P j = S B A S E + M O F S T + j * M P T C H$$

$$M E P j = S B A S E + M O F S T + (j + 1) * M P T C H - 1$$

【 0 0 3 9 】

ソースで指定されたバイトは、スタートアドレス M S P j から順にビット単位に展開され、デスティネーションのライン単位に処理される。ソーススタートアドレス M S P j 及びソースエンドアドレス M E P j の計算は、初期値に M P T C H を順次加算することで可能となる。

【 0 0 4 0 】

このように、デスティネーションアドレス計算回路 1 2 、ソースアドレス計算回路 1 3 及びパターンアドレス計算回路 1 4 でそれぞれ計算されたスタートアドレスとエンドアドレスとは、デスティネーションアドレスカウンタ 1 5 、ソースアドレスカウンタ 1 6 及びパターンアドレスカウンタ 1 7 に 1 スキャンライン毎にセットされる。

【 0 0 4 1 】

10

20

30

40

50

各アドレスカウンタ15, 16, 17は、ローカルメモリ4を階層化し、画像データ転送装置2とローカルメモリ4とのインターフェイスを、メモリコントローラ3を経由して一連の連続したデータ単位毎に行うことで、効率的にデータを転送する。このため、アドレスを次のように分解する。

【0042】

即ち、ソースデータがカラーデータの場合、各アドレスカウンタ15, 16, 17は、受け取ったスタートアドレス(FBSP_n)とエンドアドレス(FBEP_n)とを、図10(a)に示すように、上位側からセクタアドレスUビット、ブロックアドレスVビット及びバイトアドレスWビットに分解することで、ローカルメモリ4を階層化する。全体のビット数は、同図(b)に示すように、U+V+Wビットとなり、ローカルメモリ4の容量は、最大 2^{U+V+W} バイトとなる。換言すると、ローカルメモリ4は、 2^U 個のセクタで構成され、1セクタは 2^V 個のブロックで構成される。1ブロックは、 2^W バイトから構成される。図10(a)の例は、V=3, W=3の例である。

10

【0043】

1ブロックのバイト数 2^W バイトは、ローカルメモリ4のデータバス幅と等しい。即ち、ローカルメモリ4への1アドレスのアクセスで 2^W バイト(1ブロック)分のデータを転送することができる。ローカルメモリ4へのアクセスは、 2^W バイト(1ブロック)を単位として連続的に行われる。 2^W バイト(1ブロック)分のデータ転送が最小で1回、最大で 2^V 回連続し、最大の場合には、1セクタ分のデータが連続して転送されることになる。各SRAMのデータバス幅は、ローカルメモリ4のバス幅と等しい 2^W バイトであり、アドレスはVビットとなる。これは1セクタ分のデータのサイズと等しい。

20

【0044】

図10(b)に示すように、ローカルメモリ4のアドレスの上位をロードアドレス、下位をカラムアドレスとし、ロードアドレスを固定、カラムアドレスのみを連続的に変化させるページモード伝送によって更に高速のデータ転送が実現される。

【0045】

ソースデータがモノクロームデータで定義される場合、ソースデータの1ピクセルに相当する1ビットのデータを、0がバックグラウンドカラー、1がフォアグラウンドカラーとしてBPPで定義されているバイト数に拡張し、デスティネーションエリアに転送する。図11は、このデータ拡張の様子を示している。ここに示された例は、1セクタが8ブロックから構成され、1ブロックが8バイトで構成される場合の拡張の例である。図11に示すように、1バイト(8ビット)のモノクロームのソースデータSは、BPPが1のときは8バイトに、BPPが2のときは16バイトに、BPPが3のときは24バイトに、BPPが4のときは32バイトのデータに拡張され、デスティネーションエリアに転送される。

30

【0046】

図12は、ソースアドレスカウンタ16の基本構成を示す図である。なお、デスティネーションアドレスカウンタ15及びパターンアドレスカウンタ17も基本的にはソースアドレスカウンタ16と同様の構成であるが、信号NXMQD、MONOによる選択はない。また、パターンアドレスカウンタ17には、カラータグ計算用カウンタ40はない。

40

【0047】

スタートアドレス(FBSP)は、スタートアドレスレジスタ30にロードされる。同時にFBSPは、アドレス更新回路31を経由して、上位Uビット(FBSP[U+V+W-1:V+W])がセクタアドレスレジスタ32に、中央のVビット(FBSP[V+W-1:W])がブロックアドレスレジスタ33に、下位Wビット(FBSP[W-1:0])がバイトアドレスレジスタ34にそれぞれロードされる。エンドアドレス(FBEP)は、エンドアドレスレジスタ35にロードされる。

【0048】

最初に、ソースSRAM19(デスティネーションSRAM18, パターンSRAM20)にローカルメモリ4内のデータを転送するために、レジスタ32に格納されたローカル

50

メモリ4のセクタのアドレスを示すセクタアドレス (SEC) がメモリコントローラ3に出力されると共に、そのセクタにおいて転送すべきブロック数 (BLKCNT) 及びセクタ内の転送すべき最初のブロックのアドレス (セクタ内のブロックがスタートするアドレス) を示すブロックスタートアドレス (BLKSTR) がセクタ内ブロック演算回路45等で計算され、メモリコントローラ3に送られる。また、セクタ内ブロック演算回路45は、上述したブロック数 (BLKCNT) 及びブロックスタートアドレス (BLKSTR) の他、セクタ内のブロックが終了するアドレスを示すブロックエンドアドレス (BLKEND) も計算する。

【0049】

このセクタ内ブロック演算回路45での計算には、セクタスタートコンパレータ37の出力であるセクタスタートフラグ (SECSTRF)、セクタエンドコンパレータ36の出力であるセクタエンドフラグ (SECENDF)、スタートアドレスレジスタ30に格納されたブロックスタートアドレス (FBSP[V+W-1:W])、エンドアドレスレジスタ35に格納されたブロックエンドアドレス (FBEP[V+W-1:W]) 及び転送のXの方向を表すXDIR等が入力され用いられる。

10

【0050】

セクタスタートコンパレータ37はスタートアドレスレジスタ30からのスタートアドレスのセクタアドレス (FBSP[U+V+W-1:V+W]) とセクタアドレスレジスタ32からのセクタアドレス (SEC) とを比較し、等しいときはその出力データ (SECSTRF) を1とし、等しくないときは0とする。また、セクタエンドコンパレータ36はエンドアドレスレジスタ35からのエンドアドレスのセクタアドレス (FBEP[U+V+W-1:V+W]) とセクタアドレスレジスタ32からのセクタアドレス (SEC) とを比較し、等しいときはその出力データ (SECENDF) を1とし、等しくないときは0とする。

20

【0051】

図13は、セクタアドレスレジスタ32に格納されているセクタデータを説明するための図である。

【0052】

例えば、XDIR = 0と定義され、SECSTRF, SECENDFが0のとき、セクタアドレスレジスタ32に格納されているセクタアドレスは、図13(a)に示すように、矩形領域を構成するラインの最初のセクタではなく、最後のセクタでもないことがわかる。この場合、セクタ内にあるブロックの数 (BLKCNT) は 2^V 個で、ブロックのスタートアドレス (BLKSTR) は0、エンドアドレス (BLKEND) は 2^{V-1} となる。

30

【0053】

SECSTRFが1でSECENDFが0のとき、セクタアドレスレジスタ32に格納されているセクタは、矩形領域を構成するラインの最初のセクタであることが同図(b)から分かる。この場合、セクタ内にあるブロックの数は、 $2^V - \text{FBSP}[V+W-1:W]$ 個となり、ブロックのスタートアドレスは $\text{FBSP}[V+W-1:W]$ 、エンドアドレスは 2^{V-1} となる。

【0054】

また、SECSTRFが0でSECENDFが1のとき、セクタアドレスレジスタ32に格納されているセクタは、矩形領域を構成するラインの最後のセクタであることが同図(c)から分かる。この場合、セクタ内にあるブロックの数は $\text{FBEP}[V+W-1:W] + 1$ 個となり、ブロックのスタートアドレスは0でエンドアドレスは $\text{FBEP}[V+W-1:W]$ となる。

40

【0055】

更に、SECSTRF, SECENDFともに1のとき、セクタアドレスレジスタ32に格納されているセクタは、矩形領域を構成するラインの最初のセクタであり、最後のセクタでもあることが同図(d)から分かる。この場合、ブロックの数は、 $\text{FBEP}[V+W-1:W] - \text{FBSP}[V+W-1:W] + 1$ 個となり、ブロックのスタートアドレスは

50

FBSP[V+W-1:W]、エンドアドレスはFBEP[V+W-1:W]となる。上記の結果と、XDIR=1のときのそれぞれの場合とを表にまとめると、図14のようになる。

【0056】

ブロックスタートコンパレータ39は、スタートアドレスレジスタ30のブロックアドレス(FBSP[V+W-1:W])部分とブロックアドレスレジスタ33に格納されたブロックアドレス(BLK)とを比較し、等しいときはその出力データ(BLKSTRF)を1、等しくないときは0として出力する。また、ブロックエンドコンパレータ38は、

10

【0057】

上述の各コンパレータ36~39の出力SECENDF, SECSTRF, BLKENDF, BLKSTRFに基づき、スタートマスク演算回路46でスタートマスク(STRMSK)、エンドマスク演算回路47でエンドマスク(ENDMSK)がそれぞれ計算される。図15に示すように、スタート及びエンドマスクは共に0又は1のデータからなる連続した 2^W ビットのパターンであり、各マスクの1ビットが各SRAM18, 19, 20から読み出された1バイト分のデータに対応し、 2^W ビットのパターンは1ブロック分のデータに対応するものである。

20

【0058】

マスク論理積演算回路48は、スタート及びエンドマスク演算回路46, 47からの出力データであるスタート及びエンドマスクを入力し、これらの論理積データ(AMSK)を計算することで、スタートアドレス(FBSP)とエンドアドレス(FBEP)との間に存在するバイトには1を、それ以外には0をフラグとして与える役割を担っている。

【0059】

例えば、図16(a)に示すように、スタートマスク演算回路46でのスタートマスクの計算は、XDIR=0のときは、Wビット内のすべてが1である2進数パターン($2^W - 1$)をバイトアドレスレジスタ34内のバイトアドレス(BYT)で示される数だけ右にシフトして左から0を詰めることで行われる。同様に、同図(b)に示すように、XDIR=1のときのスタートマスクの計算は、Wビット内のすべて1である2進数パターン($2^W - 1$)を $2^W - 1 - \text{BYT}$ (バイトアドレス)で示される数だけ左にシフトし、右から0を詰めることで行われる。この場合、スタートアドレス(FBSP)がブロック内にあるときはSECSTRF及びBLKSTRFが共に1となる。また、ブロック内スタートアドレス演算回路49は、このときのマスクがスタートするブロック内のアドレス(BYTSTR)を計算する。

30

【0060】

一方、同図(c)に示すように、エンドマスク演算回路47でのエンドマスクの計算は、XDIR=0のとき、エンドアドレス(FBEP)のバイトアドレスがFBEP[W-1:0]とすると、Wビット内のすべて1である2進数パターン($2^W - 1$)を $2^W - 1 - \text{FBEP}[W-1:0]$ だけ左にシフトし、右から0を詰めることで行われる。同様に、同図(d)に示すように、XDIR=1のときのエンドマスクの計算は、Wビット内のすべて1である2進数パターン($2^W - 1$)をFBEP[W-1:0]だけ右にシフトし、左から0を詰めることで行われる。この場合もスタートアドレスがブロック内にあるときはSECSTRF及びBLKSTRFが共に1となる。このようにして計算された各マスクは、ブロック毎にマスク演算回路23に送られる。

40

【0061】

スタートフラグ回路50は、セクタスタートコンパレータ37の出力SECSTRF及びブロックスタートコンパレータ39の出力BLKSTRFを入力してスタートフラグXSTRFを出力する。エンドフラグ回路51は、セクタエンドコンパレータ36の出力SE

50

C E N D F 及びブロックエンドコンパレータ 3 8 の出力 B L K E N D F を入力してエンドフラグ X E N D F を出力する。

【 0 0 6 2 】

カラータグ計算用カウンタ 4 0 は、トランスペアレント処理（透明処理）の計算のために後段で用いられるものであり、スタートアドレス（F B S P）及びエンドアドレス（F B E P）が入力される時に初期化するように設定されている。このカラータグ計算用カウンタ 4 0 は、ブロックアドレスレジスタ 3 3 からの出力 B L K が更新されるのに同期して、その出力タグブロックアドレス（T G B L K）を更新する。

【 0 0 6 3 】

その更新パターンは、B P P と X D I R とによって決まるもので、初期値と更新パターンとは、図 1 7 に示すようになる。例えば、X D I R が 0 の場合、1 ピクセル当たり 1 バイト（B P P が 1）のときは初期値 = 0 で常に 0 が出力され、1 ピクセル当たり 2 バイト（B P P が 2）のときは、初期値 = 0 で、T G B L K は 0, 1, 0, 1... を繰り返すように更新され、1 ピクセル当たり 3 バイト（B P P が 3）のときは、初期値 = 0 で、T G B L K は 0, 1, 2, 0, 1, 2... を繰り返すように更新され、1 ピクセル当たり 4 バイト（B P P が 4）のときは、初期値 = 0 で、T G B L K は 0, 1, 2, 3, 0, 1, 2, 3... を繰り返すように更新される。また、X D I R が 1 の場合、1 ピクセル当たり 1 バイト（B P P が 1）のときは初期値 = 0 で常に 0 が出力され、1 ピクセル当たり 2 バイト（B P P が 2）のときは、初期値 = 1 で、T G B L K は 1, 0, 1, 0... を繰り返すように更新され、1 ピクセル当たり 3 バイト（B P P が 3）のときは、初期値 = 2 で、T G B L K は 2, 1, 0, 2, 1, 0... を繰り返すように更新され、1 ピクセル当たり 4 バイト（B P P が 4）のときは、初期値 = 3 で、T G B L K は 3, 2, 1, 0, 3, 2, 1, 0... を繰り返すように更新される。なお、デスティネーションアドレスカウンタ 1 5 及びソースアドレスカウンタ 1 6 内のカラータグ計算用カウンタ 4 0 の出力 T G B L K は、後述するように、それぞれデスティネーション S R A M 1 8, ソース S R A M 1 9 へ送られ、各 S R A M 内にあるタグ選択回路 9 3 の出力を選択するのに利用される。

【 0 0 6 4 】

アドレス更新回路 3 1 は、セクタアドレスレジスタ 3 2 からの出力セクタアドレス（S E C）とブロックアドレスレジスタ 3 3 からの出力ブロックアドレス（B L K）とを一連の値として捉え、コントローラ 2 4 からの指示によりエンドアドレスレジスタ 3 5 内の値と一致するまで B L K 単位でアドレスをインクリメントし、上記セクタアドレス（S E C）とブロックアドレス（B L K）を更新する。この場合、エンドアドレスレジスタ 3 5 内の値と一致するかどうかは、エンドフラグ回路 5 1 にて S E C E N D F 及び B L K E N D F が共に 1 であることにより検出される。

【 0 0 6 5 】

以上は、ソースデータがカラーデータの場合であるが、ソースデータがモノクロームデータである場合のソースアドレスカウンタ 1 6 の特有の機能について説明する。

アドレス更新回路 3 1 は、ソースデータにカラーデータが選択されているときには、上述したように 1 ブロック（2^W バイト）単位でアドレスをインクリメントするが、ソースデータにモノクロームデータが選択された場合、タグブロックカウンタ 4 0 からの信号 N X M Q D により、アドレスを 1 バイト単位でインクリメントする。信号 N X M Q D は、タグブロックカウンタ 4 0 が示すタグブロックが、ブロックの最後になったときに出力され、B P P によってその出力タイミングが異なる。その詳細を図 1 8 に示す。B P P = 1 のときには、信号 N X M Q D は常に O N、B P P = 2 のときは、タグブロックが 0 で O F F、1 で O N、B P P = 3 のときは、タグブロックが 0, 1 で O F F、2 で O N、B P P = 4 のときは、タグブロックが 0, 1, 2 で O F F、3 で O N となる。

【 0 0 6 6 】

スタートマスク演算回路 4 6 及びエンドマスク演算回路 4 7 は、モノクロームデータに対応するために、M O N O 信号が O N になるとその機能を切り換え、スタートマスク（S T R M S K）及びエンドマスク（E N D M S K）として常にオール 1 のデータを出力する。

10

20

30

40

50

これにより、ソースデータのマスク (AMSK) の初期値は、常にオール 1 となる。同様に、バイトのスタートアドレス (BYSTR) は、モノクロームデータの場合、常に BYSTR = 0 として出力される。タグブロックカウンタ 40 は、信号 NME X を入力する。信号 NXME X はコントローラ 24 から出力され、ソースデータがモノクロームデータであり、且つソースのマスク値が 0 になると、次のマスク (オール 1) をマスク演算回路 23 にロードし、タグブロックカウンタ 40 をインクリメントするために出力される。ソースデータがモノクロームデータの場合、タグブロックカウンタ 40 は、信号 NXME X によって更新される。タグブロックカウンタ 40 は、最後のブロックに更新されたとき、アドレス更新回路 31 をインクリメントするため、信号 NXMQD を出力する。

【0067】

次に、この画像データ転送装置 2 内のマスク演算回路 23 について説明する。マスク演算回路 23 は、主に各アドレスカウンタ 15, 16, 17 から送られてきたデスティネーション、ソース、パターンの各マスクデータ AMSK (以下、これをそれぞれ DSTMSK, SRCMSK, PATMSK と呼ぶ。) 及び各ブロック内スタートアドレス BYSTR (以下、これをそれぞれ DBYSTR, SBYSTR, PBYSTR と呼ぶ。) 等に基づき、画像表示装置 5 のデータ転送を制御するための計算を行うものである。なお、各スタートアドレスは、そのときのマスクがスタートするアドレスを表しているものである。

【0068】

図 19 に示すように、デスティネーション、ソース、パターンの各スタートアドレス (FBSP) は、ローカルメモリ 4 内の任意のアドレスを指している。マスク演算回路 23 は、これらのアドレスに基づいて、データ転送時にデスティネーション、ソース、パターンの各バイトが転送のスタートバイトからそれぞれ 1 対 1 で転送されるように対応させ、転送を制御している。

【0069】

図 20 は、このマスク演算回路 23 の内部構成を示すブロック図である。マスク演算回路 23 は、ローカルメモリ 4 から SRAM 18 ~ 20 にそれぞれ転送されたデスティネーション、ソース、パターンのいずれかのセクタデータがなくなるまで、各 SRAM 18, 19, 20 からそのデータを読み出してラスタ演算回路 21 に出力するための計算を行う。

【0070】

はじめに、図 21 (a) に示すように、デスティネーションアドレスカウンタ 15 からの DSTMSK と、ソースアドレスカウンタ 16 からの SRCMSK と、パターンアドレスカウンタ 17 からの PATMSK とがマスク演算回路 23 に入力され、各マスクセクタ 53, 54, 55 を通過し、それぞれのレジスタ 56, 57, 58 に格納される。一方、上記各アドレスカウンタ 15, 16, 17 からの各ブロック内スタートアドレス DBYSTR, SBYSTR, PBYSTR は、各アドレスセクタ 60, 61, 62 を通って後段の減算器 63, 64, 65, 66 に入力される。

【0071】

減算器 63 は、DSTMSK と SRCMSK のマスクがスタートする 1 ブロック内のアドレス差 (SRCST) を計算しレジスタ 67 に出力する。同様に減算器 65 は、DSATMSK と PATMSK のマスクがスタートする 1 ブロック内のアドレス差 (PATST) を計算しレジスタ 69 に出力する。減算器 63, 65 から出力されレジスタ 67, 69 に格納されたアドレス差 SRCST, PATST と XDIR とをシフタ 71, 72 は入力し、シフタ 71 がソースマスクデータ SRCMSK を、シフタ 72 がパターンマスクデータ PATMSK をそれぞれそのマスクのスタートが DSTMSK に一致するまでシフトする。このときのシフトは、DSTMSK に対して正・負の両方向に可能なものである。そして、その結果は、調整済みのマスクデータ (SRCADJ, PATADJ) としてそれぞれシフタ 71, 72 から論理演算回路 73 に出力される。論理演算回路 73 は、同図 (a) に示すように、これらのマスクデータの他、DSTMSK を入力し論理積を計

10

20

30

40

50

算することでDSTMSK内で処理すべきバイトを決定する。

【0072】

同時に、減算器64, 66は、シフタ71, 72がアドレス差SRCST, PATSTに基づきシフトした方向と反対の方向へのシフトにあたるシフト数(SRCREV, PATREV)を計算してレジスタ68, 70に出力する。シフタ74, 75は、レジスタ68, 70から格納されたシフト数SRCREV, PATREVを入力し、論理演算回路73からの出力マスクデータ(PRCMSK)をそれぞれ上記シフト数だけシフトしてマスク(SRCRMV, PATRMV)を算出しソースマスク論理演算回路77, パターンマスク論理演算回路79にそれぞれ出力している。このマスクSRCRMV, PATRMVは、ソースマスクSRCMSKとパターンマスクPATMSKのうち、デスティネーションマスクDSTMSKとの計算に寄与した部分を示している。

10

【0073】

図21(c)及び(d)に示すように、ソースマスク論理演算回路77及びパターンマスク論理演算回路79は、ソースマスクSRCMSK, パターンマスクPATMSKから、それぞれマスクSRCRMV及びPATRMVに対応する部分のマスクを取り除く演算をする。その結果、取り除かれ出力されたマスク(SRCUDT, PATUDT)が次に計算すべきSRCMSK, PATMSKとして更新されると共に、これらはそれぞれソースマスクセクタ54及びパターンマスクセクタ55を通り、レジスタ57, 58に格納される。また、両論理演算回路77, 79は、マスクが存在しないことが分かった時点でマスクがゼロとなったことを表す信号(SRCZR, PATZR)をそれぞれコントローラ24に対して出力する。この信号SRCZR及びPATZRを受けたコントローラ24は、新たな1ブロック分のマスクデータをソース及びパターンマスクセクタ54, 55を通してレジスタ57, 58に格納するように制御を行う。

20

【0074】

また、このときコントローラ24は、ソース及びパターンアドレスカウンタ16, 17にブロックを更新するように指示を出し、各アドレスカウンタ16, 17から出力された新たな1ブロック分のマスクデータがレジスタ57, 58に格納される。このマスクデータがレジスタ57, 58に格納されるのと同時に、新たなスタートアドレスSBYSTR, PBYSTRが各アドレスカウンタ16, 17からソース及びパターンアドレスセクタ61, 62に入力され、その後マスクスタートアドレス(SSSTAD, PSTAD)として前述のように減算器63~66に出力される。

30

【0075】

このように更新されたソース及びパターンマスク(SRCMSK, PATMSK)は、それぞれソースプライオリティエンコーダ(SPRIENC)81, パターンプライオリティエンコーダ(PPRIENC)80に入力され、マスクがスタートするアドレスSXUDT, PXUDTが計算される。また、更新されたマスクがゼロでない場合、ソース及びパターンアドレスセクタ61, 62は、先に計算されたアドレスSXUDT, PXUDTをマスクスタートアドレスSSSTAD, PSTADとして、SSSTADは減算器63, 64へ、PSTADは減算器65, 66へとそれぞれ出力する。

【0076】

一方、図21(b)に示すように、デスティネーションマスク論理演算回路76は、入力したデスティネーションマスクデータDSTMSKの論理演算回路73からのPRCMSKと一致する部分を取り除く処理をする。その出力である取り除かれたマスクDSTUDTは、次に計算すべきDSTMSKとして更新され、デスティネーションマスクセクタ53を通過してレジスタ56に格納されるものである。この論理演算回路76もまた、マスクが存在しないとわかった場合、マスクがゼロとなったことを表す信号DSTZRをコントローラ24に出力する。この信号DSTZRを受けたコントローラ24は、デスティネーションアドレスカウンタ15にブロックを更新させ、新たな1ブロック分のマスクデータを出力させる。このマスクデータは、デスティネーションマスクセクタ53に入力され、レジスタ56に格納される。

40

50

【 0 0 7 7 】

この更新されたデスティネーションマスク D S T M S K は、デスティネーションブライオリティエンコーダ (D P R I E N C) 8 2 に入力され、そこでマスクがスタートするアドレス D X U D T が計算される。更新されたマスクがゼロでない場合、デスティネーションアドレスセクタ 6 0 は、先に計算されたアドレス D X U D T をマスクスタートアドレス D S T A D として、減算器 6 3 ~ 6 6 へとそれぞれ出力する。

【 0 0 7 8 】

上述のようなマスク演算処理過程において、例えばソースのマスクがゼロとなり、且つソース S R A M 1 9 にローカルメモリ 4 から転送された 1 セクタ内の全ブロックの更新が終了した場合、コントローラ 2 4 は、次のセクタデータをソース S R A M 1 9 に転送するように制御を行う。また、デスティネーションのマスクがゼロとなり、且つデスティネーション S R A M 1 8 にローカルメモリ 4 から転送された 1 セクタ内の全ブロックの更新が終了した場合、コントローラ 2 4 は、ラスタ演算回路 2 1 によるラスタ演算の終了を待って出力 F I F O 2 2 に格納された更新後のデスティネーションデータをローカルメモリ 4 に書き込み、次のセクタデータをデスティネーション S R A M 1 8 に転送する制御を行う。更に、パターンのマスクがゼロとなり、且つパターン S R A M 2 0 にローカルメモリ 4 から転送された 1 セクタ内の全ブロックの更新が終了した場合、コントローラ 2 4 は、次のセクタデータをパターン S R A M 2 0 に転送するように制御を行う。

【 0 0 7 9 】

また、このとき論理演算回路 7 8 は、ソースデータのバイトパターンの中で、P R C M S K に寄与のある部分を示すマスク (S R C P R C) を計算する。このマスク S R C P R C は、以下のような計算によって得ることができる。即ち、S R C P R C がゼロのバイトは P R C M S K に対して寄与しないが、S R C P R C が 1 のバイトは P R C M S K に対して寄与することが前提としてある場合は、 $S R C P R C = S R C M S K \& S R C R M V$ として求めることができる。これは、後段のラスタ演算回路 2 1 にて、イネーブルフラグの計算に用いられるものである。

【 0 0 8 0 】

次に、この画像データ転送装置 2 内の各 S R A M 1 8 ~ 2 0 で行われるカラータグ演算処理について説明する。図 2 2 は、ソース S R A M 1 9 の構成を示すブロック図である。なお、デスティネーション S R A M 1 8 及びパターン S R A M 2 0 も基本的にはソース S R A M 1 9 と同様の構成であるが、デスティネーション S R A M 1 8 及びパターン S R A M 2 0 には、モノクローム拡張装置 9 6 及びこれに付随するカラー/モノクロデータ選択用の選択回路 (M U X) 9 7 , 9 8 が備えられていない。また、パターン S R A M 2 0 内には、スタートバイトレジスタ 9 0 , タグブロックレジスタ 9 1 , タグ計算回路 9 2 , タグ選択回路 9 3 及びトランスペアレントフラグ計算回路 (T R P) 9 5 はない。

【 0 0 8 1 】

ソース S R A M 1 9 (デスティネーション S R A M 1 8 及びパターン S R A M 2 0) 内のラインアドレスカウンタ (C N T R) 8 5 は、それぞれ対応するデスティネーション、ソース及びパターンアドレスカウンタ 1 5 , 1 6 , 1 7 からブロックスタート信号 (B L K S T R) 及びブロックカウンタ信号 (B L K C N T) を受け取る。これらの信号は、同時にメモリコントローラ 3 へも出力される。

【 0 0 8 2 】

メモリコントローラ 3 は、ローカルメモリ 4 からセクタアドレス S E C と、B L K S T R により指定されるアドレスから B L K C N T で指定されるだけのデータを読み出し、画像表示装置 5 内にある S R A M 等のメモリ (図示せず) にこのデータを転送する。この B L K S T R は、S R A M 8 8 のライトアドレスのスタートとなり、そのアドレスから B L K C N T で与えられるアドレス分だけローカルメモリ 4 からデータが転送され、その転送されたデータが S R A M 8 8 に書き込まれる。データが S R A M 8 8 に転送された後に、今度はデータの読み出しが行われる。このデータの読み出しは、各アドレスカウンタ 1 5 , 1 6 , 1 7 から送られるそれぞれの対応するブロックアドレス (B L K) に基づき行われ

10

20

30

40

50

る。

【 0 0 8 3 】

このとき、S R A M 8 8 のアドレスは、コントローラ 2 4 からの指示により、セクタ 8 6 で各アドレスカウンタ 1 5 ~ 1 7 のそれぞれが出力する B L K 側に切り換えられ、この B L K により S R A M 8 8 から読み出された 1 ブロック分のデータは、一旦、レジスタ 8 9 に格納されたのち、それがカラーデータの場合には、選択回路 9 7 及びレジスタ 9 4 を介して後段のラスタ演算回路 2 1 に転送される。また、モノクロームのデータの場合には、S R A M 8 8 から読み出された 1 ブロック分のデータは、モノクローム拡張装置 9 6 でカラーデータに拡張されたのち、選択回路 9 7 及びレジスタ 9 4 を介して後段のラスタ演算回路 2 1 に転送される。

10

【 0 0 8 4 】

またこのとき、カラータグを用いて、各 S R A M 1 8 ~ 2 0 でトランスペアレント処理（透明処理）のための計算も同時に行われる。以下にこれを説明する。

【 0 0 8 5 】

スタートバイトレジスタ 9 0 は、矩形領域を構成する 1 ラインの最初の F B S P のバイトアドレスを 1 ラインの処理が継続する間中保持し続ける。タグブロックレジスタ 9 1 は、各ブロックごとのタグブロックを保持するレジスタであり、ブロックの処理ごとに更新されるものである。このスタートバイトレジスタ 9 0 の出力データ B Y T S T R は、続くタグ計算回路（T A G T B L）9 2 をアクセスし、タグブロックレジスタ 9 1 の出力データ T G B L K は、タグ選択回路（M U X）9 3 をアクセスする。このタグ選択回路 9 3 は、タグ計算回路 9 2 からの複数の出力データ T A G T B L のうちの 1 つを選択する役割を担っている。

20

【 0 0 8 6 】

ここで、図 2 3 に示すように、1 ピクセルは最大 4 バイトのデータで構成され、この最大 4 バイトのピクセルデータに対して、各バイトに 0 から 3 までのカラータグ（C T）を定義すると、1 ピクセルが 1 バイトのときカラータグは 0 となることが分かる。同図より、1 ピクセルが 2 バイトのときは上位側のバイトが 1 で下位側が 0、3 バイトのときは上位側から 2, 1, 0、4 バイトのときは上位側から 3, 2, 1, 0 となることが分かる。

【 0 0 8 7 】

このカラータグは、図 2 4 に示すように、スタートアドレス F B S P で指定される最初のバイトから順番に割り付けられ、B P P の単位でエンドアドレス F B E P まで繰り返される。ここで、B P P の最大が 4 バイトであることから、タグ計算回路 9 2 は、最大で $2^W \times 4$ バイト分のカラータグを出力することが分かる。 2^W バイトのデータを同時に処理することから、 2^W バイト分ごとのカラータグをタグブロック T G B L K とすると、その構成は図 2 5 に示すように定義される。

30

【 0 0 8 8 】

このようにして定義されたカラータグ、タグブロック、B P P の関係をタグ計算回路 9 2 の出力 T A G T B L として考えると、図 2 6 ~ 2 8 に示すようになる。なお、この場合の下位ビット W は、W = 3 とする。

【 0 0 8 9 】

図 2 6 (a) に示すように、B P P が 1 バイトのときのカラータグの値は常にゼロとなる。B P P が 2 バイトのときのカラータグの値は、X D I R 及びスタートアドレスのバイト部分 B Y T S T R の L S B (Least Significant Bit/Byte) により決まる。

X D I R = 0 で B Y T S T R [0] = 0 のとき、又は

X D I R = 1 で B Y T S T R [0] = 1 のときは同図 (b) に示すようになり、

X D I R = 0 で B Y T S T R [0] = 1 のとき、又は

X D I R = 1 で B Y T S T R [0] = 0 のときは同図 (c) に示すようになる。

40

【 0 0 9 0 】

また、B P P が 3 バイトのときのカラータグの値は、図 2 7 に示すように X D I R 及びスタートアドレスのバイト部分 B Y T S T R の L S B 側 3 ビットにより決まる。

50

XDIR = 0 で BYTSTR [2 : 0] = 0 , 3 , 6 のとき、又は
 XDIR = 1 で BYTSTR [2 : 0] = 1 , 4 , 7 のときは同図 (a) に示すように、
 XDIR = 0 で BYTSTR [2 : 0] = 2 , 5 のとき、又は
 XDIR = 1 で BYTSTR [2 : 0] = 0 , 3 , 6 のときは同図 (b) に示すように、
 XDIR = 0 で BYTSTR [2 : 0] = 1 , 4 , 7 のとき、又は
 XDIR = 1 で BYTSTR [2 : 0] = 2 , 5 のときは同図 (c) に示すようにそれぞれ定義される。

【 0 0 9 1 】

更に、BPP が 4 バイトのときのカラータグの値は、図 2 8 に示すように XDIR 及び BYTSTR の LSB 側 2 ビットにより決まる。

10

XDIR = 0 で BYTSTR [1 : 0] = 0 のとき、又は
 XDIR = 1 で BYTSTR [1 : 0] = 3 のときは同図 (a) に示すように、
 XDIR = 0 で BYTSTR [1 : 0] = 3 のとき、又は
 XDIR = 1 で BYTSTR [1 : 0] = 2 のときは同図 (b) に示すようになる。同様に、

XDIR = 0 で BYTSTR [1 : 0] = 2 のとき、又は
 XDIR = 1 で BYTSTR [1 : 0] = 1 のときは同図 (c) に示すように、
 XDIR = 0 で BYTSTR [1 : 0] = 1 のとき、又は
 XDIR = 1 で BYTSTR [1 : 0] = 0 のときは同図 (d) に示すようにそれぞれ定義される。

20

【 0 0 9 2 】

タグ計算回路 9 2 の出力である TAGTBL は、タグ選択回路 9 3 に入力され、そこでステーション及びソースアドレスカウンタ 1 5 , 1 6 内のカラータグ計算用カウンタ 4 0 で計算された TGBLK で指定された部分が選択され、カラータグ (CT) となる。カラータグは、2 ビットから構成される 2^W 個のデータであり、1 ブロック内の各バイトデータのカラータグを示す。このカラータグは、トランスペアレント計算のためにトランスペアレント計算回路 (TRP) 9 5 に出力される。

【 0 0 9 3 】

次に、モノクローム拡張装置 9 6 について説明する。モノクローム拡張装置 9 6 は、ソースデータがモノクロームデータの場合、各ピクセルのビット数を拡張するもので、バイトアドレスレジスタ (BYTREG) 1 0 1 と、バイトデータ選択回路 (PXLMUX) 1 0 2 と、データ拡張回路 (BYTEXP) 1 0 3 と、ブロック選択回路 (BLKMUX) 1 0 4 と、カラー出力回路 (COLEXP) 1 0 5 とを備えて構成されている。

30

【 0 0 9 4 】

バイトアドレスレジスタ 1 0 1 は、ソースアドレスカウンタ 1 6 からのバイトアドレス BYT をストアするレジスタである。前段のソースアドレスカウンタ 1 6 は、ソースがモノクロームデータの場合、バイト単位でインクリメントされるため、このレジスタ 1 0 1 には、ローカルメモリ 4 から出力される 2^W バイト (図示の例では 8 バイト) のデータのどのバイトを指し示すかのアドレスが格納されることになる。バイトデータ選択回路 1 0 2 は、 2^W バイト分のカラー出力データをバイトレジスタ 1 0 1 に格納されたバイトアドレス BYT によって選択するための回路である。図 2 9 には、モノクローム拡張装置 9 6 の更に詳細を示す。バイトデータ選択回路 1 0 2 は、ローカルメモリ 4 から出力される 2^W バイトの読出データ RDT [6 3 : 0] のうち、バイトアドレス BYT で指定された 1 バイトのみを選択して後段に出力する。ソースアドレスカウンタ 1 6 を 1 バイトずつインクリメントすることで、バイトデータ選択回路 1 0 2 は、読出データ RDT を 1 バイトずつ順番に選択し、後段に出力する。

40

【 0 0 9 5 】

データ拡張回路 1 0 3 は、モノクロームの 1 ビットのデータを BPP の指定に基づいて 1 ビットから 4 ビットまでに拡張する。この拡張の様子を図 3 0 に示す。データ拡張回路 1 0 3 は、読出データ RDT のうちバイトアドレス BYT により選択された 1 バイトのデー

50

タを入力する。この1バイト(8ビット)のデータをBPPにより、最大32ビットまで拡張して拡張バイトBYTEXとして出力する。8ビットのデータは、モノクロームデータであるため、その0がバックグラウンドカラーを示し、1がフォアグラウンドカラーを示している。BPP=1の場合は、8ビットのデータがそのままBYTEXとして出力される。BPP=2の場合には、図30に示すように、8ビットデータの各ビットのデータがそれぞれ2ビットずつコピーされて全体として16ビットデータのBYTEXに拡張される。図で、[0][0]と示されているのは、入力のビット[0]のデータ(0または1)が2ビットにコピーされたことを示している。コピーされた2ビットは、同一のピクセルであることを示している。同様に、BPP=3のときは、入力の1ビットが3ビットにコピーされて24ビットのBYTEXが出力され、BPP=4のときは、入力の1ビットが4ビットにコピーされて36ビットのBYTEXが出力される。

10

【0096】

ブロック選択回路104は、データ拡張回路103で拡張されたモノクロームデータBYTEXのうち、タグブロックTGBLKで指定された部分(8ビット)を選択し、モノクロームの出力ブロックMONFLGとして次段のカラー出力回路105に出力する。拡張されたビットデータ、タグブロックTGBLK、カラータグCTの関係を図31に示す。BPP=1の場合、TGBLK=0として8ビットのBYTEXの全体が選択されて出力される。BPP=2の場合、TGBLK=0で16ビットのBYTEXのうちの下位8ビットが選択され、TGBLK=1で上位8ビットが選択される。以下同様である。

【0097】

カラー出力回路105は、図32にその詳細構成を示すように、バス幅2^Wが8バイトの場合、8つのカラー拡張ユニット(COL__EXP__UNIT)110~117から構成される。1つのカラー拡張ユニット11nに拡張された1ビットのモノクロームデータMONFLGの1ビットを入力し、これを1バイトのカラー成分データMONEXに変換し出力する。各カラー拡張ユニット11nは、拡張されたモノクロームデータMONFLGが0であるか1であるか、並びにそれに対応するカラータグCTnCを参照し、あらかじめレジスタにストアされているフォアグラウンドカラーまたはバックグラウンドカラーの適当な成分を選ぶ。カラー拡張ユニット11nは、バックグラウンドカラーの成分を選ぶバックグラウンド選択回路(BGMUX)118a、フォアグラウンドカラーの成分を選ぶフォアグラウンド選択回路(FGMUX)118b及び出力データを選択するBG/FG選択回路(BFMUX)119から構成される。バックグラウンド選択回路118a及びフォアグラウンド選択回路118bは、ともにそれぞれに対応したカラータグCT0C~CT7Cの値に応じて、バックグラウンドカラー、フォアグラウンドカラーを構成する32ビットのカラーデータのうちの8ビットの成分を選択する。BG/FG選択回路119は、モノクロームデータMONFLGが0のときはバックグラウンドカラー成分を選択し、モノクロームデータMONFLGが1のときはフォアグラウンドカラー成分を選択する。BG/FG選択回路119の出力MONEXは、カラーデータとモノクロームデータとを切り換えて出力するカラー/モノクロデータ選択用の選択回路(MUX)97に入力され、レジスタ94を介して後段のラスタ演算回路21に転送される。また、BG/FG選択回路119の出力MONEXは、同様の選択回路98を介してトランスペアレント計算回路(TRP)95にも入力される。

20

30

40

【0098】

次に、トランスペアレント計算回路95について説明する。トランスペアレント処理は、ピクセル値が予め定義されたトランスペアレントカラーと一致する場合に、そのピクセルの更新を行わない処理である。具体的には、ソースのピクセル値がソースペアレントカラーと一致する場合、そのピクセルの更新を行わないか、又はデスティネーションのピクセル値を上書きする。また、デスティネーションのピクセル値がデスティネーションカラーと一致する場合、そのピクセルの更新を行わないか、又はデスティネーションのピクセル値を上書きする。

【0099】

50

図33は、このトランスペアレント計算回路95の内部構成を示す図である。トランスペアレント計算回路95は、主にFG/BGセクタ120、2W個のセクタ121~126、2W個の比較器127~132、 2^W 個のレジスタ133~138、 2^W 個の演算器139~144、レジスタ145及びR/Lセクタ146から構成される。なお、 2^W 個からなるセクタ、比較器、レジスタ、演算器は、それぞれ1ブロック内にある各バイトデータに対応して設けられているものである。

【0100】

FG/BGセクタ120は、図示しないインターフェイスから送られてくる転送先領域の背景色（FG：フォアグラウンドカラー）と転送元領域の背景色（BG：バックグラウンドカラー）とを、同じく送られてくる前トランスペアレントカラーデータ（FGTR）を利用して選択し、どちらをトランスペアレントカラーとするかを決定する。その結果、このFG/BGセクタ120の出力TRCOLが、新たなトランスペアレントカラーとなる。このトランスペアレント計算回路95は、デスティネーション領域又はソース領域に格納されているピクセルデータとトランスペアレントカラーとをピクセル単位で比較し、その結果等しい場合は、そのピクセルデータはトランスペアレントであるとし、図22に示すように新しいデータに書き換えないようにするためのフラグ（TRPF）を後段に出力する。

【0101】

FG/BGセクタ120から出力されたTRCOLは、BPPの定義より、最大で4バイトとなる。各セクタ121~126は、カラータグ（CT0~CT 2^W-1 ）により、TRCOLを構成するバイト成分をカラータグの値で選択し、各比較器127~132に出力する。これを受けて各比較器127~132は、図示しないメモリからの出力データ（RDT）と選択されたトランスペアレントカラー成分とをバイト単位で比較し、等しいときには1、等しくないときには0と定義して、比較結果NEQを出力する。各レジスタ133~138は、各比較器127~132で比較された結果NEQとそれに対応するカラータグ（CT）とを一時的に保持し、保持されたNEQとCTとをカレントデータのCEQ、CCTとして出力する。レジスタ145は、このカレントの比較結果CEQを保持し、それらをPEQとしてR/Lセクタ146に出力する。

【0102】

ここで、図34に示すように、1ブロック内でバイトアドレスのゼロバイトに近づく方向を左、 2^W-1 バイトに近づく方向を右とすると、BPPの最大が4バイトであることから、現在のカラータグを含めて最大で左右に3バイト分の各比較器127~132での比較結果を参照すれば、そのピクセルがトランスペアレントとして与えられたカラーと等しいかどうかを判断することができる。

【0103】

また、このとき、1ピクセル内のバイトデータがブロック間をまたぐ場合があり、この場合、図35に示すように、現在処理中のブロックの一つ前又は一つ後のブロックデータの各比較器127~132での比較結果を参照する。

【0104】

更に、PEQは、現在の処理に対して一つ前に処理されたブロックの各比較器127~132での比較結果となり、NEQは、現在の処理に対して一つ後に処理されるブロックの各比較器127~132での比較結果となる。現在の処理の各比較器127~132での比較結果CEQのゼロバイトから 2^W-1 バイトまでの各バイトに対する左右に隣接するそれぞれ3バイトは、図36で示すように表すことができる。

【0105】

この図36の表（a）、（b）から明らかなように、CEQ0に対するleft3、left2、left1と、CEQ1に対するleft3、left2と、CEQ2に対するleft3と、CEQ 2^W-3 に対するright3と、

10

20

30

40

50

$CEQ2^W - 2$ に対する $right3$, $right2$ と、
 $CEQ2^W - 1$ に対する $right3$, $right2$, $right1$ とが、 $XDIR$ により異なっている。

【0106】

R/L セクタ146は、上述の NEQ , PEQ 及び $XDIR$ を入力して選択し、適切なものを $CEQ2^W - 1$, $CEQ2^W - 2$, $CEQ2^W - 3$, $CEQ2$, $CEQ1$, $CEQ0$ の各演算器139~144に出力する。各演算器139~144は、入力した CCT と BPP により、左右のどの各比較器127~132での比較結果を参照すればよいかを知ることができる。例えば、 BPP が1のとき、ゼロバイトから $2^W - 1$ バイトまでの各比較器127~132での比較結果が、そのままトランスペアレントかどうかを示しているとい

10

【0107】

例えば、図37(a)に示すように、 BPP が2のときは、現在の処理に対して左右に1バイトずつの各比較器127~132での比較結果を参照すれば、左右どちらかの比較結果を参照すべきかは、現在処理中のカラータグの値により決まる。現在の処理のカラータグがゼロの場合、右の1バイトが同一ピクセルであり、現在の処理のカラータグが1の場合、左の1バイトが同一ピクセルであることが分かる。もし、各比較器127~132での比較結果が共に1であるならば、このピクセルはトランスペアレントであるとい

20

【0108】

また、 BPP が3のときは、現在の処理に対して左右に最大2バイトずつの各比較器127~132での比較結果を参照する。もし、同一ピクセル部分の各比較器127~132での比較結果がすべて1ならば、このピクセルはトランスペアレントであるとい

【0109】

更に、 BPP が4のときは、現在の処理に対して左右に最大3バイトずつの各比較器127~132での比較結果を参照する。この場合も同様に同一ピクセル部分の各比較器127~132での比較結果がすべて1であるならば、このピクセルはトランスペアレントであるとい

30

【0110】

最後に、この画像データ転送装置2内のラスタ演算回路21について簡単に説明する。図38は、ラスタ演算回路21の内部構成を示す図であり、このラスタ演算回路21は、デスティネーション、ソース、パターンの各 $SRAM18$, 19 , 20 から読み出したデータに対してラスタ演算するものである。

【0111】

ラスタ演算回路21は、主にソース $SRAM19$ からの1ブロック分のデータを、バイトデータを単位としてシフトする SB シフタ150、パターン $SRAM20$ からの1ブロック分のデータを、同じくバイトデータを単位としてシフトする PB シフタ151、8ビットのラスタ演算を実行する 2^W 個の8ビットラスタ演算回路1521~152n、書き込みイネーブルのデータを計算するイネーブルデータ計算回路153、このイネーブルデータ計算回路153内で、ソーストランスペアレントフラグをシフトするシフタ(図示せず)、ラスタ演算の結果を格納する 2^W 個のレジスタ1541~154n等から構成されている。

40

【0112】

まず、デスティネーション $SRAM18$ から読み出された 2^W バイトのデスティネーションデータは、それぞれ対応する8ビットラスタ演算回路1521~152nに入力される。ソース $SRAM19$ から読み出された 2^W バイトのソースデータは、マスク演算回路23で計算されたシフト分データである1ブロック内のアドレス差 $SRCST$ と等しい分

50

だけS Bシフタ1 5 0にてバイト単位でシフトされ、デスティネーションデータと対応させられる。同様に、パターンS R A M 2 0から読み出された2^Wバイトのパターンデータも、マスク演算回路2 3で計算されたP A T S F Tと等しい分だけP Bシフタ1 5 1にてバイト単位でシフトされ、デスティネーションデータと対応させられる。8ビットラスタ演算回路1 5 2 1 ~ 1 5 2 nは、指定されたコードにより、これらのデスティネーション、ソース、パターンデータでラスタ演算を行い、この結果をレジスタ1 5 4 1 ~ 1 5 4 nに格納する。このとき、マスク演算回路2 3で計算されたP R C M S Kを各レジスタ1 5 4 1 ~ 1 5 4 nは入力し、P R C M S Kが1になるデータのみを格納している。

【0 1 1 3】

イネーブルデータ計算回路1 5 3は、デスティネーションS R A M 1 8からのトランスペアレントフラグ(D T R P F)、ソースS R A M 1 9からのトランスペアレントフラグ(S T R P F)、マスク演算回路2 3からのS R C P R C、S R C S F T、P R C M S K等を入力する。これらの入力情報に基づき、イネーブルデータ計算回路1 5 3は、各バイトデータ毎のローカルメモリ4への書き込みが行われるかどうかを決定するイネーブルフラグE Nを計算する。トランスペアレントの場合は、ローカルメモリ4への書き込みが行われず、以前の値がそのままローカルメモリ4内で維持されるように、イネーブルフラグE Nを変更する。

【0 1 1 4】

このラスタ演算回路2 1の後段にある出力F I F O 2 2は、1ブロック内のすべてのバイトデータのラスタ演算が終了したところで、レジスタ1 5 4 1 ~ 1 5 4 nに格納されている結果を自身のメモリ(図示せず)に書き込む。デスティネーションの1セクタ分のデータがすべて出力F I F O 2 2内にあるメモリに書き込まれたところで、メモリコントローラ3に1セクタ分のデータを連続的に出力し、そのデータをローカルメモリ4に書き込むことで、データの転送を行っている。

【0 1 1 5】

最後に、ソースデータがモノクロームデータで与えられた場合の具体的な処理例について説明する。

1セクタが8ブロック、1ブロックが8バイトからなり、B P P = 3として、図3 9 (a)で示す、1 6ピクセル×1 6ラインの×印のモノクロームデータの転送を、例を挙げて考える。

【0 1 1 6】

インターフェイス1 1は、転送先のデスティネーションデータのパラメータ、モノクロームソースデータのパラメータを受け取り、デスティネーションアドレス計算回路1 2、ソースアドレス計算回路1 3内のモノクロームソースアドレス計算回路1 3 2の対応したレジスタにデータをセットする。計算されたデスティネーションアドレス(スタートアドレス及びエンドアドレス)、ソースモノクロームアドレス(スタートアドレス及びエンドアドレス)は、デスティネーションアドレスカウンタ1 5、ソースアドレスカウンタ1 6に転送される。その後、コントローラ2 4からの制御により、デスティネーションアドレスカウンタ1 5からのアドレスを用い、ローカルメモリ4にアクセスし、最初の1ライン分のデータを含むセクタをデスティネーションS R A M 1 8に転送する。デスティネーションS R A M 1 8に全データが転送出来ない場合は、セクタ単位で転送が繰り返される。同様に、ソースS R A M 1 9にも最初の1ライン分のモノクロームデータを転送する。

【0 1 1 7】

デスティネーションデータ、ソースデータともにローカルメモリ4から転送された後、S R A M 1 8、1 9から読み出される。×印のデータは、図3 9 (b)のようになる。たとえば、×印のモノクロームソースデータが、ローカルメモリ4の表示領域外のアドレスA 0 0 0 0 4番地から、図示のように格納されているものとする。ソースアドレスカウンタ1 6はアドレスA 0 0 0 0 4を分解し、バイトアドレスレジスタ3 4にはB Y T = 4が、ブロックアドレスレジスタ3 3には0が、セクタアドレスレジスタ3 2には2 8 0 0 0 0がそれぞれセットされる。マスクがスタートするブロック内のスタートアドレス(B

10

20

30

40

50

)となる[図40(c)]。

【0123】

この状態で、ソースマスクが0となると、ソースアドレス計算回路13内のカラータグ計算用カウンタ40から信号NXMQDがアドレス更新回路31に送られる。これにより、アドレス更新回路31は、1バイトだけインクリメントされる。アドレス更新回路31がインクリメントされることにより、モノクローム拡張装置96のバイトアドレスレジスタ101にはBYT=5がセットされる。バイトアドレスレジスタ101に新たにセットされたBYT=5は、バイトデータ選択回路102に入力され、SRAM88から出力されたデータの対応した部分の1バイトのデータが選択される。この場合は、ブロック内のアドレスが5の01のデータが出力される。

10

【0124】

以下、同様の処理が行われる。1ラインの処理が終了する場合は、次のラインのアドレスが、デスティネーションアドレス計算回路12、ソースアドレス計算回路13で計算される。最初のラインと同様に、ローカルメモリ4からデスティネーションSRAM18、ソースSRAM19にデータが転送される。これにより、ソースアドレスカウンタは、A000006番地に更新される。バイトアドレスレジスタ101にはBYT=6がセットされる。ソースSRAM19内のバイトデータ選択回路102は、SRAM88から読み出されたデータのBYTにより、対応した部分の1バイトのデータを選ぶ。ブロック内のアドレスが6のデータ“40”が出力される。

以上を繰り返すことにより、モノクロームソースデータが指定されたデスティネーション

20

【0125】

【発明の効果】

以上述べたように、この発明によれば、指定された転送元及び転送先領域転に関するパラメータに基づき、画像データ記憶装置に記憶された転送領域の画像データを演算処理し転送する際に、画像データを1ピクセルが1又は複数のバイトで構成された画像データであるとして、この1ピクセルの画像データよりも大きい複数バイトからなるブロック単位で画像データを転送し、転送領域でブロック内における位置合わせを行い演算処理を実行することで、画像データの高速転送が可能であると共に、転送元領域の画像データは、1ピクセルが1ビットで構成されたモノクロームの画像データであり、転送処理の際に、転送元領域の1ピクセルをそのビット値に応じて予め設定された2色のデータのうちの1色を表す1又は複数のバイトで構成された画像データに拡張するようにしているので、画像データ記憶手段に記憶させる転送元領域の画像データのデータ量を削減することができるという効果を奏する。

30

【図面の簡単な説明】

【図1】 この発明の一実施例に係る画像表示処理システムの基本構成を説明するためのブロック図である。

【図2】 同システムにおける画像データ転送装置の機能を概略的に示す図である。

【図3】 同システムにおける画像データ転送装置のモノクローム転送の概要を示す図である。

40

【図4】 同装置の詳細な構成を示すブロック図である。

【図5】 同装置におけるソースアドレス計算回路の要部を示すブロック図である。

【図6】 同装置の動作を示すフローチャートである。

【図7】 同装置における転送矩形領域を含む画像データの矩形(表示)領域を更に詳細に示す図である。

【図8】 同装置における転送矩形領域のXDIR, YDIRを概略的に示す図である。

【図9】 同装置におけるモノクロームの画像データの構成を示す図である。

【図10】 同装置におけるローカルメモリ内に構築されている画像データの構成を示す図である。

【図11】 モノクロームの画像データの拡張を説明するための図である。

50

【図12】 同装置内のアドレスカウンタ内部の基本構成を示す図である。

【図13】 同カウンタ内部のセクタアドレスレジスタに格納されているセクタデータを説明するための図である。

【図14】 同カウンタ内部のセクタアドレスレジスタに格納されているセクタデータを示す図である。

【図15】 同カウンタ内部におけるマスク演算処理時の各データの構成を説明するための図である。

【図16】 同カウンタ内部におけるマスク演算処理の結果を説明するための図である。

【図17】 同カウンタ内部におけるカラータグ計算用カウンタの初期値と更新パターンとを示す図である。

10

【図18】 モノクロームの画像データ処理時のタグブロックとアドレス更新信号との関係を示す図である。

【図19】 同装置におけるデスティネーション、ソース、パターンの各スタートアドレスのローカルメモリ内の構成を示す図である。

【図20】 同装置におけるマスク演算回路の内部構成を示すブロック図である。

【図21】 同装置におけるマスク演算回路での各データの処理を説明するための図である。

【図22】 同装置における各SRAMの内部構成を示す図である。

【図23】 同装置における各SRAMでのカラータグ演算処理のデータ構成を説明するための図である。

20

【図24】 同処理のデータ構成を説明するための図である。

【図25】 同処理のデータ構成を説明するための図である。

【図26】 同処理におけるカラータグ、タグブロック、BPPの関係を示す図である。

【図27】 同処理におけるカラータグ、タグブロック、BPPの関係を示す図である。

【図28】 同カラータグ演算処理におけるカラータグ、タグブロック、BPPの関係を示す図である。

【図29】 同装置におけるモノクローム拡張装置の詳細機能ブロック図である。

【図30】 同モノクローム拡張装置のビット拡張動作を説明するための図である。

【図31】 同モノクローム拡張装置のデータ拡張動作を説明するための図である。

【図32】 同モノクローム拡張装置のカラー拡張ユニットの詳細ブロック図である。

30

【図33】 各SRAMにあるトランスペアレント計算回路の内部構成を示す図である。

【図34】 同回路におけるトランスペアレント計算処理のデータの構成を説明するための図である。

【図35】 同処理におけるデータの比較方法を説明するための図である。

【図36】 同処理におけるデータの比較結果の構成を表した図である。

【図37】 同処理におけるデータの比較結果を表した図である。

【図38】 同装置におけるラスタ演算回路の内部構成を示す図である。

【図39】 同装置によるモノクロームデータの処理例を示すモノクロームの画像データとそのメモリへの格納状態を示す図である。

【図40】 同モノクロームデータの拡張されたデータを示す表である。

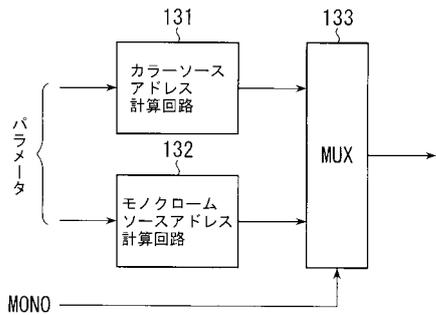
40

【図41】 従来の画像表示処理システムの構成を示すブロック図である。

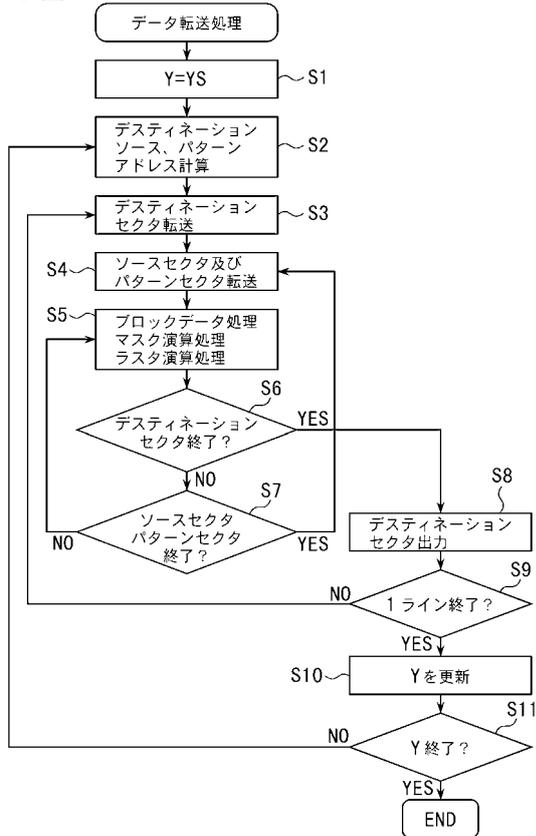
【符号の説明】

1 ... CPU、2 ... 画像データ転送装置、3 ... メモリコントローラ、4 ... ローカルメモリ、5 ... 表示装置、11 ... インターフェイス、12 ... デスティネーションアドレス計算回路、13 ... ソースアドレス計算回路、14 ... パターンアドレス計算回路、15 ... デスティネーションアドレスカウンタ、16 ... ソースアドレスカウンタ、17 ... パターンアドレスカウンタ、18 ... デスティネーションSRAM、19 ... ソースSRAM、20 ... パターンSRAM、21 ... ラスタ演算回路、22 ... 出力FIFO、23 ... マスク演算回路、24 ... コントローラ。

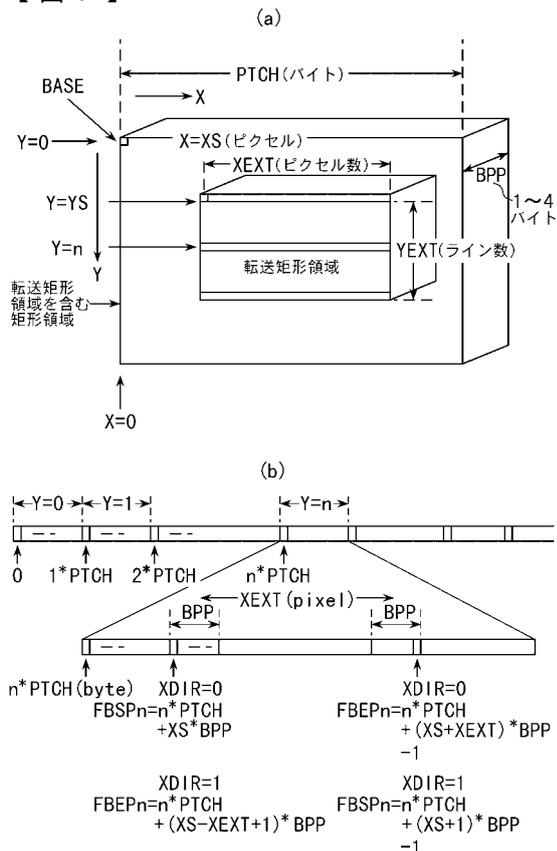
【 図 5 】



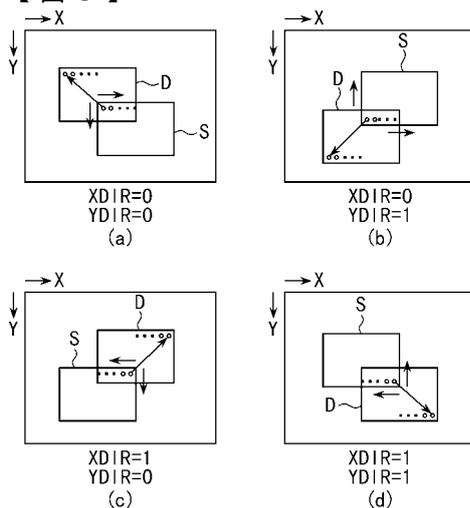
【 図 6 】



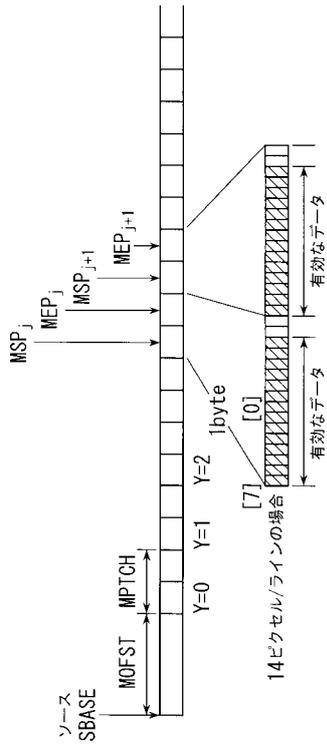
【 図 7 】



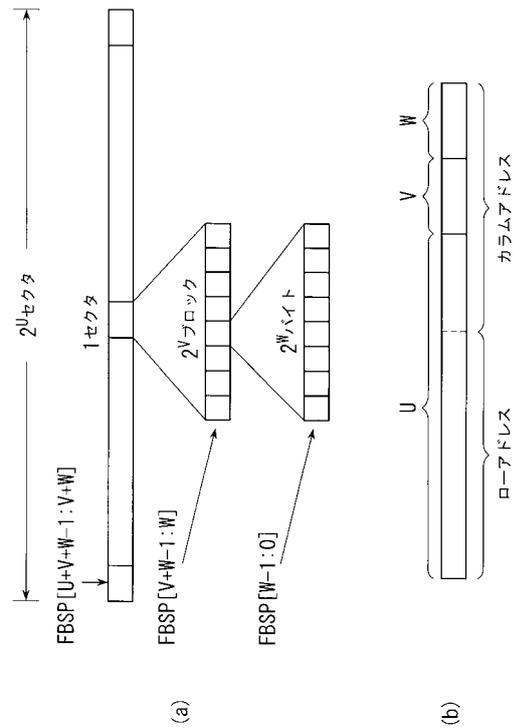
【 図 8 】



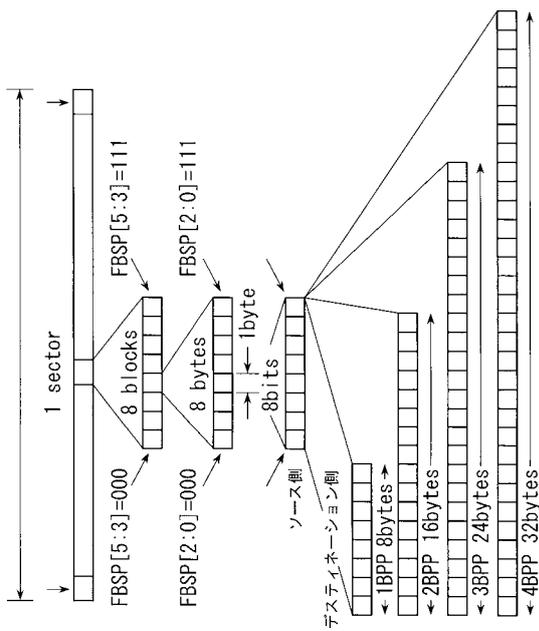
【 図 9 】



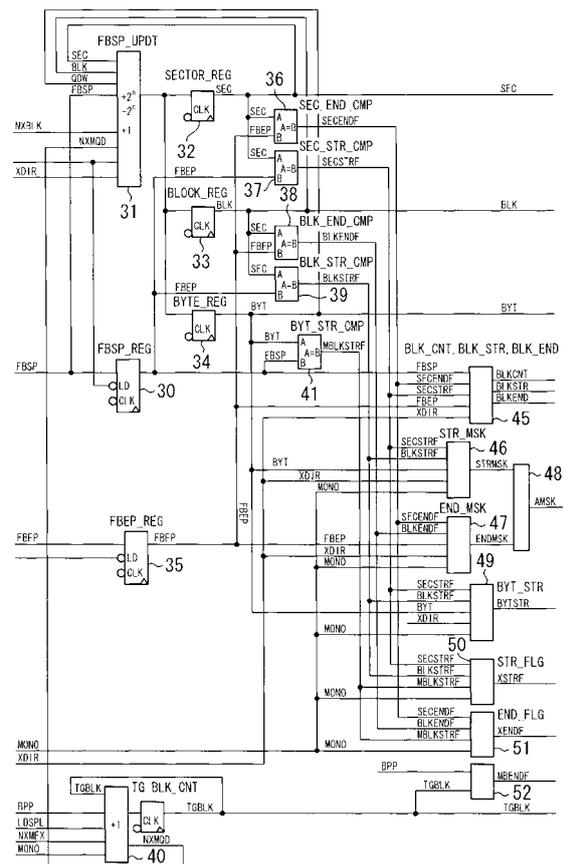
【 図 10 】



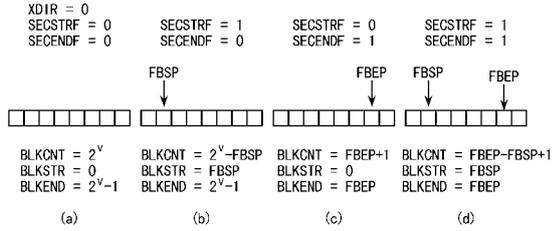
【 図 11 】



【 図 12 】



【 13 】



【 14 】

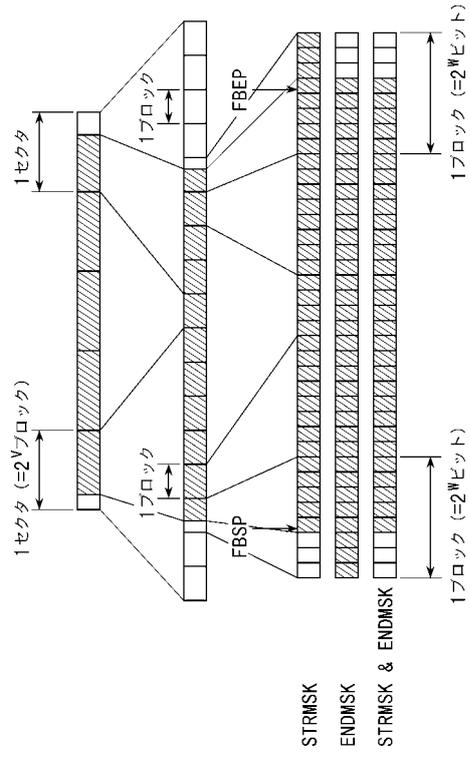
XDIR = 0

SECSTRF	SECENDF	BLKCNT	BLKSTR	BLKEND
0	0	2^v	0	$2^v - 1$
1	0	$2^v - \text{FBSP}[V+W-1:W]$	$\text{FBSP}[V+W-1:W]$	$2^v - 1$
0	1	$\text{FBEP}[V+W-1:W]+1$	0	$\text{FBEP}[V+W-1:W]$
1	1	$\text{FBEP}[V+W-1:W] - \text{FBSP}[V+W-1:W]+1$	$\text{FBSP}[V+W-1:W]$	$\text{FBEP}[V+W-1:W]$

XDIR = 1

SECSTRF	SECENDF	BLKCNT	BLKSTR	BLKEND
0	0	2^v	$2^v - 1$	0
1	0	$\text{FBSP}[V+W-1:W]+1$	$\text{FBSP}[V+W-1:W]$	0
0	1	$2^v - \text{FBEP}[V+W-1:W]$	$2^v - 1$	$\text{FBEP}[V+W-1:W]$
1	1	$\text{FBSP}[V+W-1:W] - \text{FBEP}[V+W-1:W]+1$	$\text{FBSP}[V+W-1:W]$	$\text{FBEP}[V+W-1:W]$

【 15 】



【 16 】

スタートマスク計算

XDIR = 0

SECSTRF	BLKSTRF	STRMSK	BYTSTR
0	0	$2^m - 1$	0
1	0	$2^m - 1$	0
0	1	$2^m - 1$	0
1	1	$2^m - 1 \gg \text{BYT}$	BYT

(a)

XDIR = 1

SECSTRF	BLKSTRF	STRMSK	BYTSTR
0	0	$2^m - 1$	$2^m - 1$
1	0	$2^m - 1$	$2^m - 1$
0	1	$2^m - 1$	$2^m - 1$
1	1	$2^m - 1 \ll 2^m - 1 - \text{BYT}$	BYT

(b)

エンドマスク計算

XDIR = 0

SECENDF	BLKENDF	ENDMSK
0	0	$2^m - 1$
1	0	$2^m - 1$
0	1	$2^m - 1$
1	1	$2^m - 1 \ll 2^m - 1 - \text{FBEP}[W-1:0]$

(c)

XDIR = 1

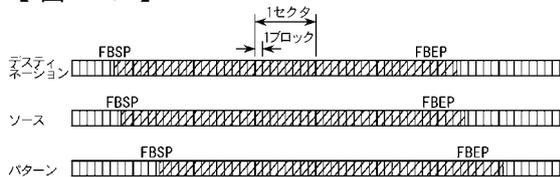
SECENDF	BLKENDF	ENDMSK
0	0	$2^m - 1$
1	0	$2^m - 1$
0	1	$2^m - 1$
1	1	$2^m - 1 \gg \text{FBEP}[W-1:0]$

(d)

【 17 】

	XDIR=0		XDIR=1	
	初期値	TGBLKの更新	初期値	TGBLKの更新
BPP=1	0	常に0。	0	常に0。
BPP=2	0	0, 1, 0, 1, ... と繰り返す。	1	1, 0, 1, 0, 1, ... と繰り返す。
BPP=3	0	0, 1, 2, 0, 1, 2, ... と繰り返す。	2	2, 1, 0, 2, 1, 0, ... と繰り返す。
BPP=4	0	0, 1, 2, 3, 0, 1, 2, 3, ... と繰り返す。	3	3, 2, 1, 0, 3, 2, 1, 0, ... と繰り返す。

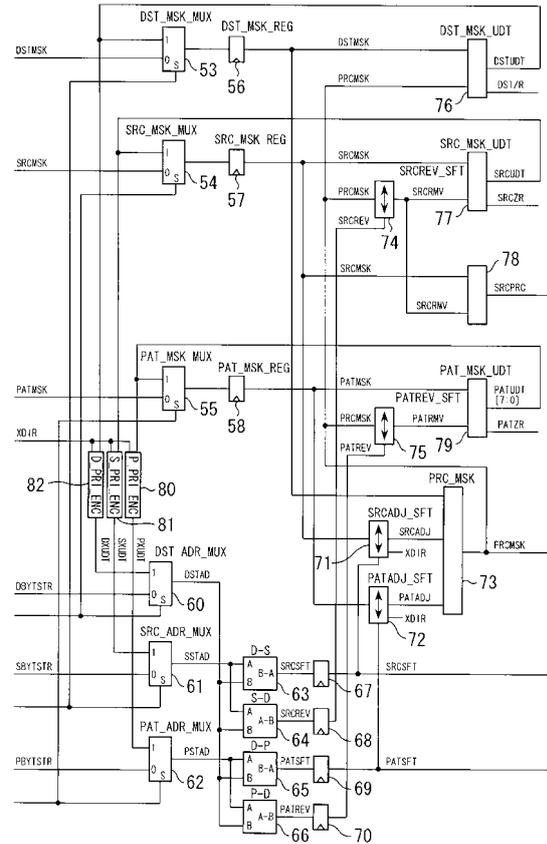
【 19 】



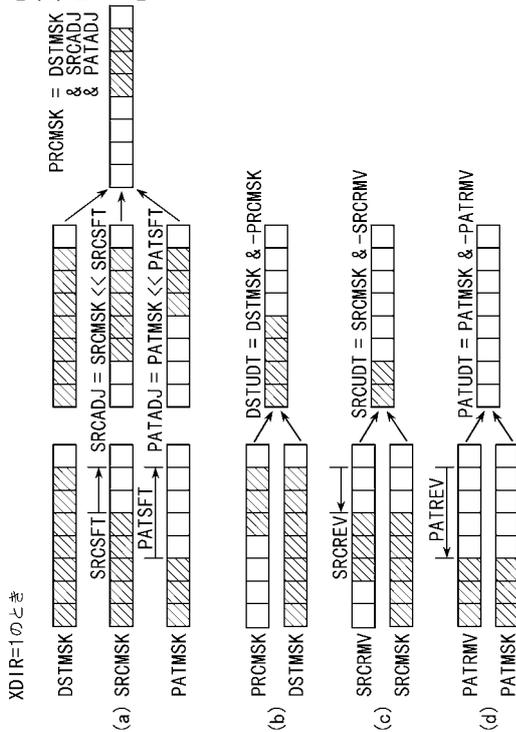
【 図 1 8 】

BPP=1		BPP=2		BPP=3		BPP=4	
TG_BLK_CNT	IXMMD	TG_BLK_CNT	IXMMD	TG_BLK_CNT	IXMMD	TG_BLK_CNT	IXMMD
0	ON	0	OFF	0	OFF	0	OFF
1	OFF	1	ON	1	OFF	1	OFF
2	OFF	2	ON	2	ON	2	OFF
3	ON	3	ON	3	ON	3	ON

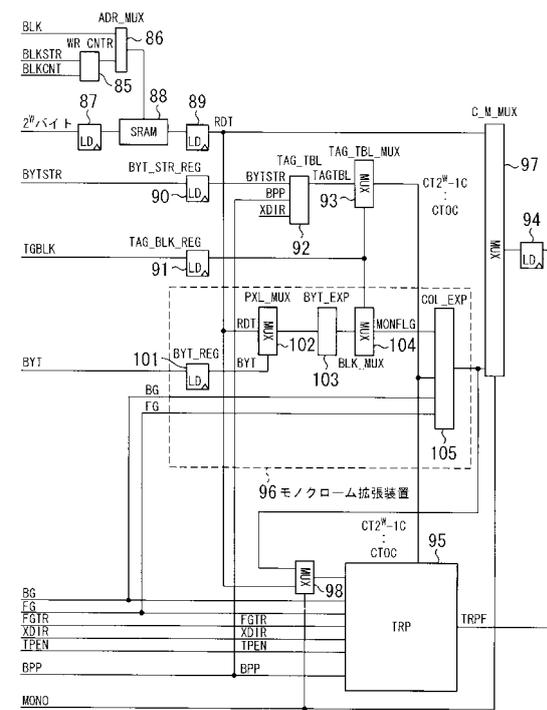
【 図 2 0 】



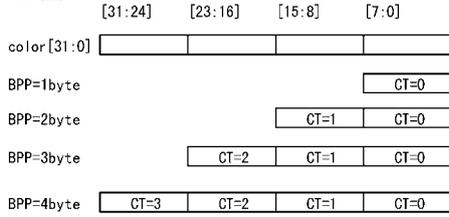
【 図 2 1 】



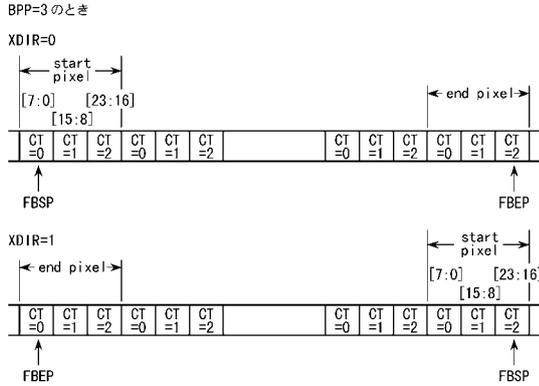
【 図 2 2 】



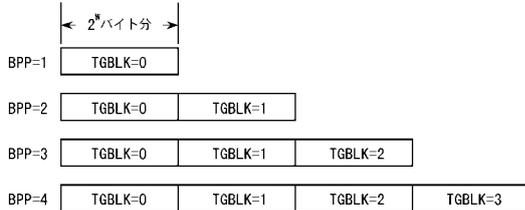
【 2 3 】



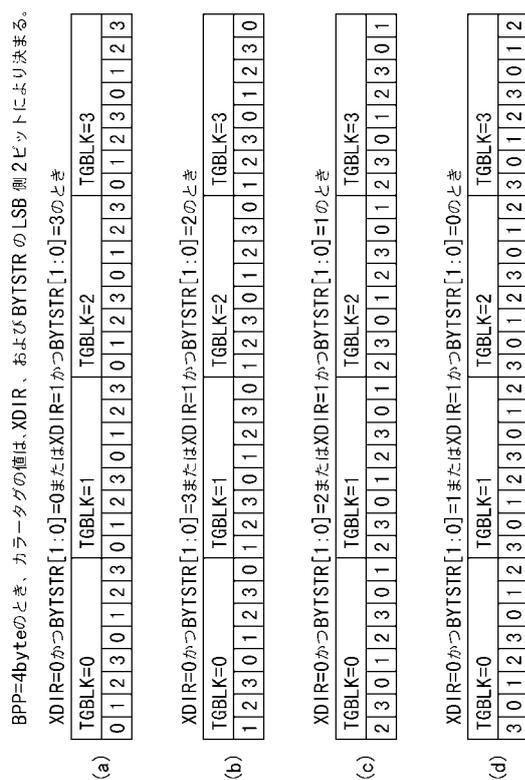
【 2 4 】



【 2 5 】



【 2 8 】



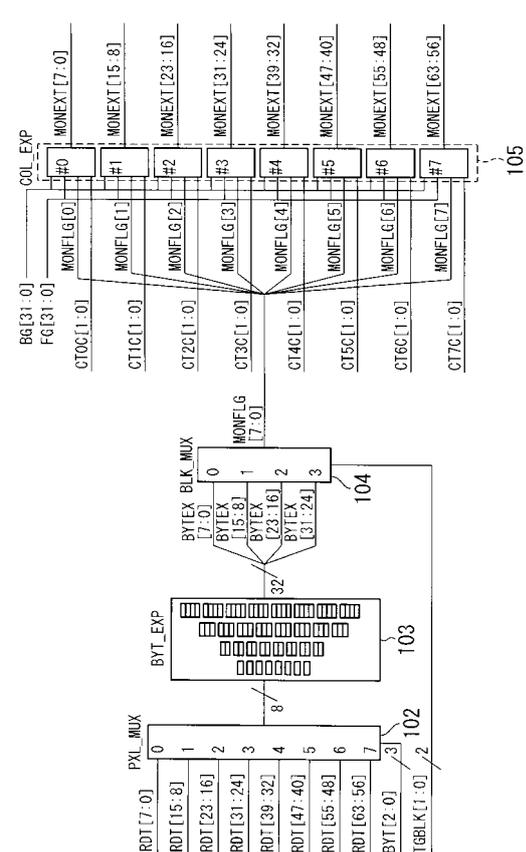
【 2 6 】



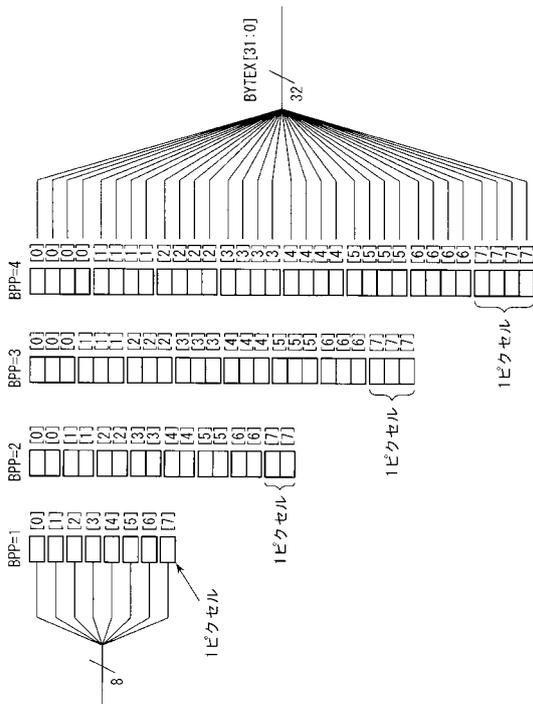
【 2 7 】



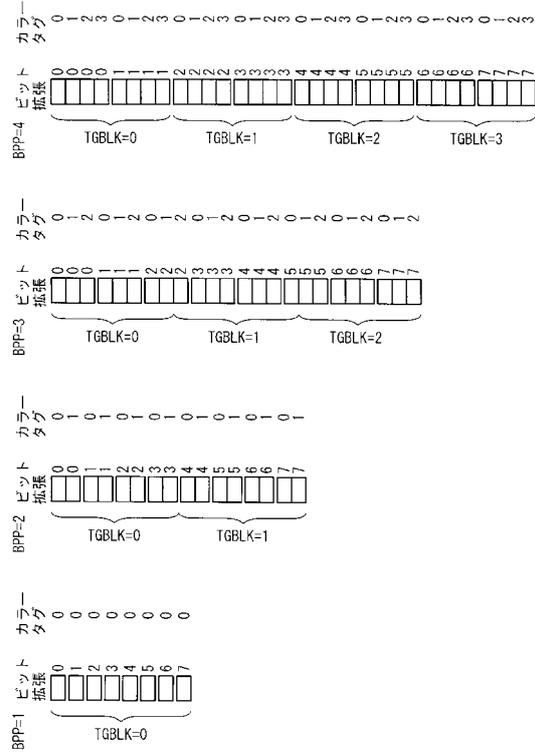
【 2 9 】



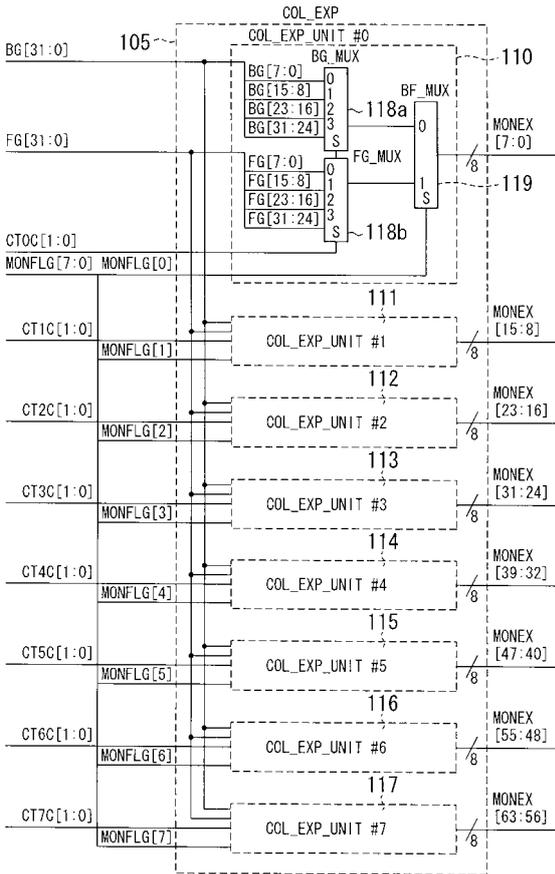
【図 30】



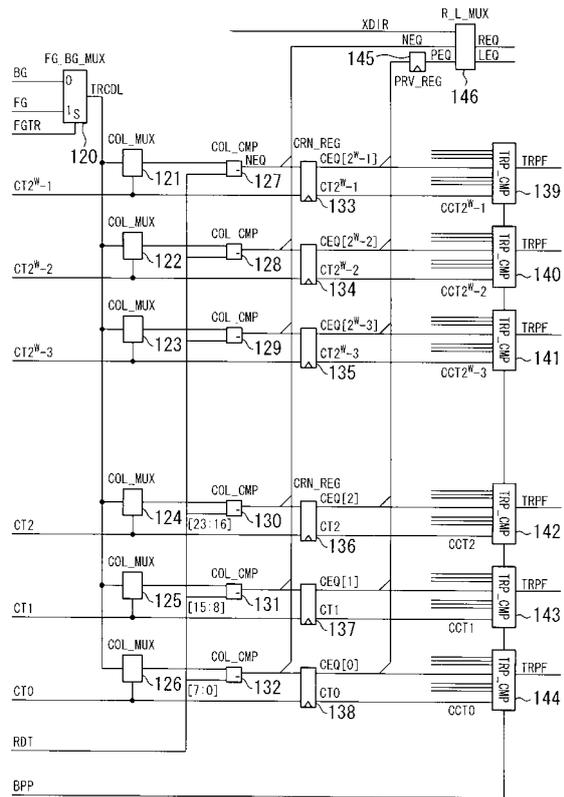
【図 31】

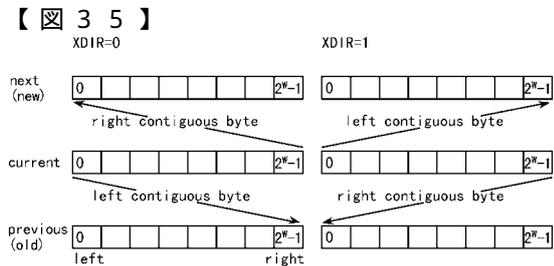
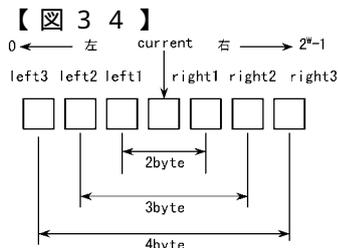


【図 32】



【図 33】





【 3 7 】

(a)

left3	left2	left1	CCT	right1	right2	right3
1	0	1	0	1	0	1
0	1	0	1	0	1	0

(b)

left3	left2	left1	CCT	right1	right2	right3
0	1	2	0	1	2	0
1	2	0	1	2	0	1
2	0	1	2	0	1	2

(c)

left3	left2	left1	CCT	right1	right2	right3
1	2	3	0	1	2	3
2	3	0	1	2	3	0
3	0	1	2	3	0	1
0	1	2	3	0	1	2

【 3 6 】

(a)

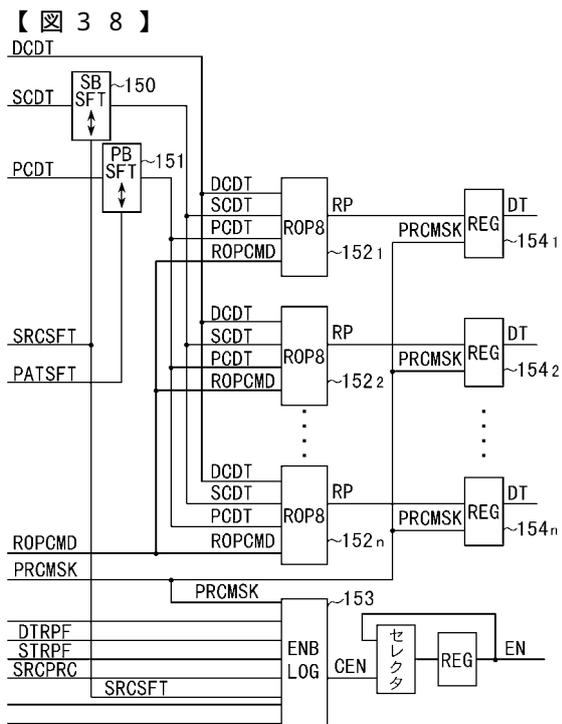
XDIR=0 のとき

	left3	left2	left1	current	right1	right2	right3
0	PEQ2 ^{m-3}	PEQ2 ^{m-2}	PEQ2 ^{m-1}	CEQ0	CEQ1	CEQ2	CEQ3
1	PEQ2 ^{m-2}	PEQ2 ^{m-1}	CEQ0	CEQ1	CEQ2	CEQ3	CEQ4
2	PEQ2 ^{m-1}	CEQ0	CEQ1	CEQ2	CEQ3	CEQ4	CEQ5
3	CEQ0	CEQ1	CEQ2	CEQ3	CEQ4	CEQ5	CEQ6
:	:	:	:	:	:	:	:
2 ^{m-3}	CEQ2 ^{m-6}	CEQ2 ^{m-5}	CEQ2 ^{m-4}	CEQ2 ^{m-3}	CEQ2 ^{m-2}	CEQ2 ^{m-1}	NEQ0
2 ^{m-2}	CEQ2 ^{m-5}	CEQ2 ^{m-4}	CEQ2 ^{m-3}	CEQ2 ^{m-2}	CEQ2 ^{m-1}	NEQ0	NEQ1
2 ^{m-1}	CEQ2 ^{m-4}	CEQ2 ^{m-3}	CEQ2 ^{m-2}	CEQ2 ^{m-1}	NEQ0	NEQ1	NEQ2

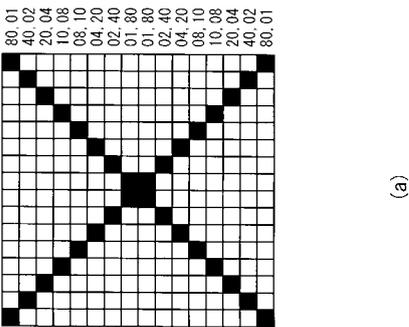
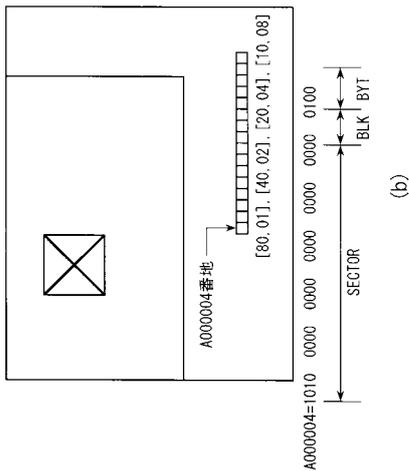
(b)

XDIR=1 のとき

	left3	left2	left1	current	right1	right2	right3
0	NEQ2 ^{m-3}	NEQ2 ^{m-2}	NEQ2 ^{m-1}	CEQ0	CEQ1	CEQ2	CEQ3
1	NEQ2 ^{m-2}	NEQ2 ^{m-1}	CEQ0	CEQ1	CEQ2	CEQ3	CEQ4
2	NEQ2 ^{m-1}	CEQ0	CEQ1	CEQ2	CEQ3	CEQ4	CEQ5
3	CEQ0	CEQ1	CEQ2	CEQ3	CEQ4	CEQ5	CEQ6
:	:	:	:	:	:	:	:
2 ^{m-3}	CEQ2 ^{m-6}	CEQ2 ^{m-5}	CEQ2 ^{m-4}	CEQ2 ^{m-3}	CEQ2 ^{m-2}	CEQ2 ^{m-1}	PEQ0
2 ^{m-2}	CEQ2 ^{m-5}	CEQ2 ^{m-4}	CEQ2 ^{m-3}	CEQ2 ^{m-2}	CEQ2 ^{m-1}	PEQ0	PEQ1
2 ^{m-1}	CEQ2 ^{m-4}	CEQ2 ^{m-3}	CEQ2 ^{m-2}	CEQ2 ^{m-1}	PEQ0	PEQ1	PEQ2



【 図 3 9 】



【 図 4 0 】

(a)

COL_EXP_UNIT	MONFLG	CT	MONEX
#0	MONFLG[0]=1	CT0C[1:0]=0	MONEX[7:0]=56
#1	MONFLG[1]=1	CT1C[1:0]=1	MONEX[15:8]=34
#2	MONFLG[2]=1	CT2C[1:0]=2	MONEX[23:16]=12
#3	MONFLG[3]=0	CT3C[1:0]=0	MONEX[31:24]=BC
#4	MONFLG[4]=0	CT4C[1:0]=1	MONEX[39:32]=9A
#5	MONFLG[5]=0	CT5C[1:0]=2	MONEX[47:40]=78
#6	MONFLG[6]=0	CT6C[1:0]=0	MONEX[55:48]=BC
#7	MONFLG[7]=0	CT7C[1:0]=1	MONEX[63:56]=9A

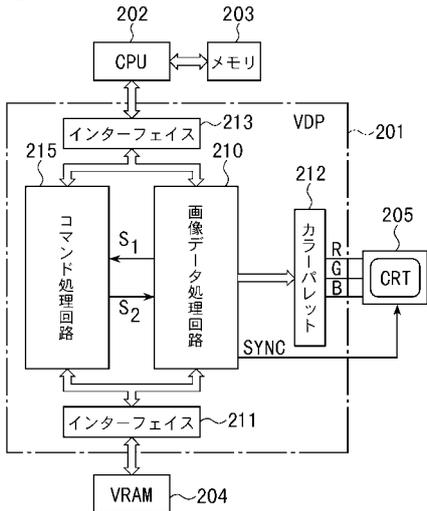
(b)

COL_EXP_UNIT	MONFLG	CT	MONEX
#0	MONFLG[0]=0	CT0C[1:0]=2	MONEX[7:0]=78
#1	MONFLG[1]=0	CT1C[1:0]=0	MONEX[15:8]=BC
#2	MONFLG[2]=0	CT2C[1:0]=1	MONEX[23:16]=9A
#3	MONFLG[3]=0	CT3C[1:0]=0	MONEX[31:24]=78
#4	MONFLG[4]=0	CT4C[1:0]=0	MONEX[39:32]=BC
#5	MONFLG[5]=0	CT5C[1:0]=1	MONEX[47:40]=9A
#6	MONFLG[6]=0	CT6C[1:0]=2	MONEX[55:48]=78
#7	MONFLG[7]=0	CT7C[1:0]=0	MONEX[63:56]=BC

(c)

COL_EXP_UNIT	MONFLG	CT	MONEX
#0	MONFLG[0]=0	CT7C[1:0]=1	MONEX[7:0]=9A
#1	MONFLG[1]=0	CT6C[1:0]=2	MONEX[15:8]=78
#2	MONFLG[2]=0	CT5C[1:0]=0	MONEX[23:16]=BC
#3	MONFLG[3]=0	CT4C[1:0]=1	MONEX[31:24]=9A
#4	MONFLG[4]=0	CT3C[1:0]=2	MONEX[39:32]=78
#5	MONFLG[5]=0	CT2C[1:0]=0	MONEX[47:40]=BC
#6	MONFLG[6]=0	CT1C[1:0]=1	MONEX[55:48]=9A
#7	MONFLG[7]=0	CT0C[1:0]=2	MONEX[63:56]=78

【 図 4 1 】



フロントページの続き

(51) Int.Cl.

F I

G 0 9 G 5/02

(56) 参考文献 特開平 1 0 - 1 4 3 4 2 6 (J P , A)

特許第 2 5 7 5 6 9 5 (J P , B 2)

特開平 0 1 - 1 2 8 0 9 5 (J P , A)

特開平 0 2 - 2 7 8 4 7 5 (J P , A)

特開昭 6 3 - 0 2 1 6 9 4 (J P , A)

特開平 0 5 - 1 6 5 7 1 4 (J P , A)

(58) 調査した分野(Int.Cl. , DB名)

G09G 5/00 - 5/42

G06F 12/00

G06T 1/60