(19)日本国特許庁(JP)　(12)公開特許公報(A)　(11)特許出願公開番号
特開2009-200968
(P2009-200968A)
(43)公開日　平成21年9月3日(2009.9.3)

(51)Int.Cl.			F I			テーマコード（参考）
H04N	1/387	(2006.01)	H04N	1/387		2C187
B41J	5/30	(2006.01)	B41J	5/30	Z	2C362
B41J	2/44	(2006.01)	B41J	3/00	M	2H027
G03G	21/14	(2006.01)	G03G	21/00	372	5C076
G03G	21/00	(2006.01)	G03G	21/00	510	

審査請求 未請求　請求項の数 6　ＯＬ　（全 35 頁）

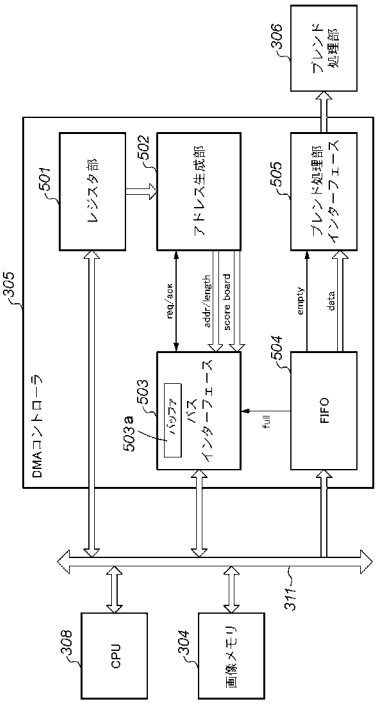
(21)出願番号　特願2008-42071 (P2008-42071)
(22)出願日　平成20年2月22日 (2008.2.22)

(71)出願人　000001007
キヤノン株式会社
東京都大田区下丸子３丁目３０番２号
(74)代理人　100076428
弁理士　大塚　康徳
(74)代理人　100112508
弁理士　高柳　司郎
(74)代理人　100115071
弁理士　大塚　康弘
(74)代理人　100116894
弁理士　木村　秀二
(74)代理人　100130409
弁理士　下山　治
(74)代理人　100134175
弁理士　永川　行光

最終頁に続く

(54)【発明の名称】画像処理装置及びその制御方法

(57)【要約】
【課題】　多くのラインバッファが必要になり、その分メモリ容量が増大し、かつビットマップ画像データを補正するための回路規模が増大する。
【解決手段】　画像データの読み出し開始アドレスとバースト転送長とに基づいて、メモリから走査ラインのライン方向にバーストで画像データを読み出す際の、画像データのラインを切り換えるためのライン方向における切り換えアドレスを設定し（５０１）、その設定された切り換えアドレスにおいて、少なくともライン切り換え前の第１ラインと、ライン切り換え後の第２ラインからバースト転送にて画像データが読み出されるように、第２ラインから画像データを読み出すための第２の読み出し開始アドレスを、その切り換えアドレスに基づいて生成するアドレス生成部５０２を有する。
【選択図】　図３



【特許請求の範囲】
【請求項１】
画像形成時の走査ラインの曲がりを画像データにより補正して画像を形成するための画像処理装置であって、
少なくとも複数ライン分の画像データを記憶する記憶手段と、
画像データの読み出し開始アドレスと、バースト転送長とに基づいて、前記記憶手段から前記走査ラインのライン方向に画像データを読み出してバースト転送するデータ読出し手段と、
前記データ読出し手段が読み出す画像データのラインを切り換えるための前記ライン方向における切り換えアドレスを設定する設定手段と、
前記設定手段により設定された前記切り換えアドレスにおいて、少なくともライン切り換え前の第１ラインと前記ライン切り換え後の第２ラインからバースト転送にて画像データが読み出されるように、前記第２ラインから画像データを読み出すための第２の読み出し開始アドレスを、前記切り換えアドレスに基づいて生成する生成手段と、
前記データ読出し手段が前記記憶手段から読み出してバースト転送した画像データの内の有効領域に対応する画像データに基づいて１ライン分の画像データを出力する出力手段と、
を有することを特徴とする画像処理装置。
【請求項２】
前記切り換えアドレスにおいて、ライン切り換え前の基準のラインに対して上或は下にずれたライン数を示す情報を格納する格納手段を更に有し、
前記生成手段は、前記テーブルに格納された情報に基づいて第２の読み出し開始アドレスを生成することを特徴とする請求項１に記載の画像処理装置。
【請求項３】
前記生成手段は、前記データ読出し手段が読み出す前記第１ラインの画像データの前記ライン方向における読み出しアドレスと、前記データ読出し手段が読み出す前記第２ラインの画像データの前記ライン方向における読出しアドレスが重複するように、前記第２の読み出し開始アドレスを生成し、
前記出力手段は、前記ライン方向における読み出しアドレスが重複する第１ラインの画像データ及び第２ラインの画像データに基づいて階調処理した上で１ライン分の画像データを出力することを特徴とする請求項１又は２に記載の画像処理装置。
【請求項４】
前記データ読出し手段は、リングバッファを介して前記記憶手段から画像データを読み出すことを特徴とする請求項１に記載の画像処理装置。
【請求項５】
前記出力手段により出力された前記画像データに基づいてシート上に画像を形成する画像形成手段を有することを特徴とする請求項１乃至４のいずれか１項に記載の画像処理装置。
【請求項６】
画像形成時の走査ラインの曲がりを画像データにより補正して画像を形成するための画像処理装置の制御方法であって、
画像データの読み出し開始アドレスと、バースト転送長とに基づいて、少なくとも複数ライン分の画像データを記憶するメモリからライン方向に画像データを読み出してバースト転送するデータ読出し工程と、
前記データ読出し工程が読み出す画像データのラインを切り換えるための前記ライン方向における切り換えアドレスを設定する設定工程と、
前記設定工程により設定された前記切り換えアドレスにおいて、少なくともライン切り換え前の第１ラインと前記ライン切り換え後の第２ラインからバースト転送にて画像データが読み出されるように、前記第２ラインから画像データを読み出すための第２の読み出し開始アドレスを、前記切り換えアドレスに基づいて生成する生成工程と、

前記データ読出し工程が前記メモリから読み出してバースト転送した画像データの内の有効領域に対応する画像データに基づいて１ライン分の画像データを出力する出力工程と、
を有することを特徴とする画像処理装置の制御方法。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、画像処理装置及びその制御方法に関するものである。
【背景技術】
【０００２】
従来から、電子写真方式の画像形成装置が知られている。電子写真方式の画像形成装置は、画像データに応じたレーザ光で感光体を走査（照射）するレーザ光源と、レーザ光源から照射されたレーザ光が照射される感光体ドラムと、感光体ドラムを帯電するための帯電ローラを有する。また、帯電された感光体ドラムにレーザ光が照射されたことにより感光体ドラム上の静電潜像をトナーにより現像するための現像器を有する。また、現像器により感光体ドラム上に現像されたトナー像をシート（記録紙）に転写するための転写器を有する。
【０００３】
そして、以上のような画像形成装置においては、１ライン分の画像データに基づいたレーザ光が感光体ドラムに走査（照射）された場合、走査ラインは理想的には感光ドラムの軸方向に平行な直線となる。しかし、レーザ光源や感光ドラムの取付位置等の機械的な誤差により、走査方向に平行な直線とならない（走査ラインが曲がる）ことがある。このような走査ラインの曲がりを補正するための方法として、例えば以下のような技術が提案されている。
【０００４】
特許文献１では、露光ユニットを画像形成装置本体へ組み付ける際に光学センサを用いて走査ラインの傾きの大きさを測定し、その傾きに応じて露光ユニットを機械的に傾かせて走査ラインの傾きを調整している。
【０００５】
また特許文献２には、光学センサを用いて、画像形成装置における走査ラインの傾きと曲がりの大きさを測定し、それらを相殺するようにビットマップ画像データを補正し、その補正した画像データに基づいて画像を形成することが記載されている。この特許文献２の方法は、画像データを処理することで電気的に走査ラインの傾きと曲がりの補正を行っているため、機械的な調整や組立時の調整工程が不要となる。従って、前述の特許文献１及び２に記載の方法に比べて、より安価に走査ラインの傾きと曲がりに対処できるという利点がある。
【特許文献１】特開２００３－２４１１３１号公報
【特許文献２】特開２００４－１７０７５５号公報
【発明の開示】
【発明が解決しようとする課題】
【０００６】
しかし上記特許文献２に記載された方法では、走査ラインの曲がり部分で、隣接する上或は下ラインのデータを階調記録するブレンド処理を行っているため、複数の走査分のビットマップデータを記憶するラインバッファが必要となる。そして、このラインバッファの数は走査ラインの曲がり幅に依存することになる。例えば、走査ラインの曲がり幅が画像データのＮライン分であると、Ｎライン分の画像データを記憶できる分のラインバッファが必要となる。ここで、画像形成装置毎にばらつきがあるため、このＮの値を、ばらつきの最大値を超えるライン数に対応させなければならない。このように多くのラインバッファを確保すると、その分メモリ容量が増大し、かつビットマップ画像データを補正するための回路規模が増大して、コストアップを招くこととなる。

【０００７】
本発明の目的は、上記従来技術の問題点を解決することにある。
【０００８】
本願発明の特徴は、走査ラインの曲がり（湾曲）を補正するラインバッファを不要にして、安価で、かつ簡単な構成で、画像データにより走査ラインの曲がりを補正する画像処理装置及びその制御方法を提供することを目的とする。
【課題を解決するための手段】
【０００９】
上記目的を達成するために本発明の一態様に係る画像処理装置は以下のような構成を備える。即ち、
画像形成時の走査ラインの曲がりを画像データにより補正して画像を形成する画像形成装置であって、
少なくとも複数ライン分の画像データを記憶する記憶手段と、
画像データの読み出し開始アドレスと、バースト転送長とに基づいて、前記記憶手段から前記走査ラインのライン方向に画像データを読み出してバースト転送するデータ読出し手段と、
前記データ読出し手段が読み出す画像データのラインを切り換えるための前記ライン方向における切り換えアドレスを設定する設定手段と、
前記設定手段により設定された前記切り換えアドレスにおいて、少なくともライン切り換え前の第１ラインと前記ライン切り換え後の第２ラインからバースト転送にて画像データが読み出されるように、前記第２ラインから画像データを読み出すための第２の前記読み出し開始アドレスを、前記切り換えアドレスに基づいて生成する生成手段と、
前記データ読出し手段が前記記憶手段から読み出してバースト転送した画像データの内の有効なセグメントに対応する画像データに基づいて１ライン分の画像データを出力する出力手段と、
前記出力手段により出力された画像データに基づいてシート上に画像を形成する画像形成手段とを有することを特徴とする。
【００１０】
上記目的を達成するために本発明の一態様に係る画像形成装置の制御方法は以下のような工程を備える。即ち、
画像形成時の走査ラインの曲がりを画像データにより補正して画像を形成する画像形成装置の制御方法であって、
画像データの読み出し開始アドレスと、バースト転送長とに基づいて、少なくとも複数ライン分の画像データを記憶するメモリからライン方向に画像データを読み出してバースト転送するデータ読出し工程と、
前記データ読出し工程が読み出す画像データのラインを切り換えるための前記ライン方向における切り換えアドレスを設定する設定工程と、
前記設定工程により設定された前記切り換えアドレスにおいて、少なくともライン切り換え前の第１ラインと前記ライン切り換え後の第２ラインからバースト転送にて画像データが読み出されるように、前記第２ラインから画像データを読み出すための第２の前記読み出し開始アドレスを、前記切り換えアドレスに基づいて生成する生成工程と、
前記データ読出し工程が前記メモリから読み出してバースト転送した画像データの内の有効なセグメントに対応する画像データに基づいて１ライン分の画像データを出力する出力工程とを有することを特徴とする。
【発明の効果】
【００１１】
本発明によれば、走査ラインの曲がり（湾曲）を補正するラインバッファを不要にして、安価で、かつ簡単な構成で、画像データにより走査ラインの曲がりを補正する画像処理装置及び当該装置におけるデータ読出し方法を提供することができる。
【発明を実施するための最良の形態】

【００１２】
以下、添付図面を参照して本発明の好適な実施の形態を詳しく説明する。尚、以下の実施の形態は特許請求の範囲に係る本発明を限定するものでなく、また本実施の形態で説明されている特徴の組み合わせの全てが本発明の解決手段に必須のものとは限らない。
【００１３】
［画像形成装置］
図１は、本発明の実施の形態に係るカラー画像形成装置（多機能装置）１００の構成を説明するブロック図である。
【００１４】
この画像形成装置１００は、例えば、複数種類の機能を実現する複合機である多機能処理装置（ＭＦＰ：Multi Function Peripheral）で実現される。またこの画像形成装置１００は、ネットワークＩ／Ｆ１０８を介してネットワークに接続され、このネットワークを利用して、ネットワークに接続された外部機器との間で画像データや各種情報をやりとりすることができる。
【００１５】
図１において、原稿台とオートドキュメントフィーダ（ＡＤＦ）を含む画像読取部１０５は、原稿台上の束状或は１枚の原稿画像を光源で照射し、その原稿からの反射像をレンズで固体撮像素子（不図示）に結像する。これにより画像読取部１０５は、その固体撮像素子からラスタ状の画像信号を基に、所定密度（例えば、６００ｄｐｉ）のページ単位のラスタ画像を得ることができる。尚、本実施の形態では画像読取部１０５で読み取られる原稿として、紙文書を例に挙げて説明するが、紙以外の記録媒体（例えば、ＯＨＰシート、フィルム等の透過原稿、布等）等の印刷物を画像読取部１０５の読取対象としても良い。
【００１６】
また画像形成装置１００は、画像読取部１０５で読み取った画像信号に対応する画像を印刷部１０７で記録媒体に印刷する複写機能を有する。特に、原稿を１部複写する場合には、この画像信号をデータ処理部１０１で画像処理して印刷データを生成し、これを印刷部１０７に出力して記録媒体に印刷させる。一方、原稿を複数部複写する場合には、ＨＤ等の記憶部１０６に一旦、印刷データを記憶させた後、これを指定された部数分繰り返し印刷部１０７に出力して記録媒体上に印刷させる。尚、印刷部１０７を用いる各種印刷制御は、プリンタコントローラ１０３（画像処理装置）によって実現される。印刷部１０７は、この実施の形態では、例えばタンデム方式のレーザプリンタのプリンタエンジンを含むものとする。
【００１７】
画像形成装置１００への操作者の指示は、画像形成装置１００に装備された操作部１０４から行われ、これら一連の動作はデータ処理部１０１内の制御部（図２のＣＰＵ３０８）で制御される。また、操作部１０４における入力状態の表示及び処理中の画像データの表示等は表示部１０２に表示される。尚、この画像形成装置１００では、後述する各種処理を実行するための各種操作及び表示をユーザに提供するユーザインターフェースを、表示部１０２及び操作部１０４によって実現している。
【００１８】
次に、プリンタコントローラ１０３の詳細構成について図２を用いて説明する。
【００１９】
図２は、本発明の実施の形態に係るプリンタコントローラ１０３の詳細構成を示すブロック図である。
【００２０】
このプリンタコントローラ１０３はホストＩ／Ｆ部３０２を有する。このホストＩ／Ｆ部３０２には、データ処理部１０１から送出された印刷データや、この画像形成装置１００の動作を指示する設定を入力する入力バッファ（不図示）が設けられている。またホストＩ／Ｆ部３０２には、データ処理部１０１へ送出する信号や機器情報データを含む出力

データを一時的に保持する出力バッファ（不図示）が設けられている。またホストＩ／Ｆ部３０２は、データ処理部１０１との間で送受信される信号や通信パケットの入出力部を構成するとともに、データ処理部１０１との間の通信制御を行う。

【００２１】

ホストＩ／Ｆ部３０２を介して入力された印刷データは、画像データ発生部３０３に与えられる。ここで、入力される印刷データは、例えば、ＰＤＬ（ページ記述言語）データで構成される。画像データ発生部３０３は、予め定められている解析部により、入力された印刷データの解析（例えば、ＰＤＬ解析処理）を行う。そして、その解析結果から中間言語を生成し、更に印刷部（プリンタエンジン）１０７が処理可能なビットマップデータを生成する。

【００２２】

具体的には、印刷データの解析とその解析による中間言語情報の作成を行うとともに、その中間言語情報の作成と並行してラスタライズ処理を行う。このラスタライズ処理では、印刷データに含まれる表示色ＲＧＢ（加法混色）から印刷部１０７が処理可能なＹＭＣＫ（減法混色）への変換が含まれる。またこの処理には、印刷データに含まれる文字コードから予め格納されているビットパターン、アウトラインフォント等のフォントデータへの変換等の処理が含まれる。その後、ラスタライズ処理では、ページ単位或はバンド単位でビットマップデータを作成し、このビットマップデータに対してディザパターンを用いる疑似階調処理を施し、印刷部１０７において印刷処理が可能なビットマップデータを生成する。こうして作成されたビットマップデータは、少なくとも複数ライン分の画像データを記憶できる画像メモリ３０４に格納される。この画像メモリ３０４に格納されているビットマップデータの読み出しは、ＤＭＡコントローラ３０５で制御されている。このＤＭＡコントローラ３０５による画像メモリ３０４からのビットマップデータの読み出しに対する制御は、ＣＰＵ３０８からの指示に基づき行なわれる。

【００２３】

画像メモリ３０４から読み出されたビットマップデータは、後述するブレンド処理を行うブレンド処理部３０６で処理され、エンジンＩ／Ｆ部３０７を介してビデオ信号として印刷部１０７に転送される。エンジンＩ／Ｆ部３０７は、印刷部１０７へ転送する印刷データ（ビデオ信号）を一時的に保持する出力バッファ（不図示）と、印刷部１０７から送出された信号を一時的に保持する入力バッファ（不図示）とが設けられている。またエンジンＩ／Ｆ部３０７は、印刷部１０７との間で送受信される信号の入出力部を構成するとともに、印刷部１０７との間の通信制御をも司っている。

【００２４】

また操作部１０４におけるユーザの操作入力によって出力されるモード設定に関する指示等の各種指示は、操作部Ｉ／Ｆ部３０１を介して入力される。この操作部Ｉ／Ｆ部３０１は、操作部１０４とＣＰＵ３０８との間のインターフェースを構成する。ＣＰＵ３０８は、操作部１０４もしくはデータ処理部１０１から指示されたモードに応じて、上述の各部に対する制御を行う。この制御はＲＯＭ３０９に格納されている制御プログラムに基づいて実行される。このＲＯＭ３０９に格納されている制御プログラムは、システムクロックによってタスクと称されるロードモジュール単位に時分割制御を行うためのＯＳを含んでいる。またこの制御プログラムには、このＯＳによって機能単位に実行制御される複数のロードモジュールが含まれる。ＣＰＵ３０８による演算処理の作業領域としては、ＲＡＭ３１０が使用される。ＣＰＵ３０８を含む各部は、システムバス３１１に接続されている。このシステムバス３１１は、アドレスバスとデータバス及び制御信号バスとを有している。

【００２５】

次に、ＤＭＡコントローラ３０５の詳細構成について図３を用いて説明する。

【００２６】

図３は、本発明の実施の形態に係るプリンタコントローラ１０３のＤＭＡコントローラ３０５の詳細構成を示すブロック図である。

【００２７】
レジスタ部５０１は、複数のレジスタ（不図示）で構成される。ＤＭＡコントローラ３０５に対するＣＰＵ３０８からの指示は、ＣＰＵ３０８がレジスタ部５０１の各レジスタに適切な値を書き込むことで行なわれる。アドレス生成部５０２は、画像メモリ３０４に記憶されるビットマップデータを読み出すためのアドレスをレジスタ部５０１の各レジスタの内容を参照して生成する。このアドレス生成部５０２は、アドレス信号（addr）と、そのアドレスから読み出すデータ長（データ量）を示すレングス信号（length）をバスインターフェース５０３に出力する。これらアドレス信号とレングス（読出しデータ長）信号は、要求信号（req）と共にバスインターフェース５０３に出力される。さらに、走査ライン上のどの有効なデータを読み出したかを示すスコアボード信号（score_board）をバスインターフェース５０３に出力する。

【００２８】
バスインターフェース５０３は、アドレス生成部５０２からのアドレス信号とレングス信号を入力すると、システムバス３１１に対して画像メモリ３０４のリードトランザクションを発行する。例えば、システムバス３１１のデータバス幅が３２ビットであれば、そのリード要求を３２ビット単位を何回連続リードするかを示すバースト長(バースト転送長）を指定してリードトランザクションを発行する。一組のアドレス信号とレングス信号に対するリード処理が完了すると、バスインターフェース５０３は応答信号（ack）により読み出し完了をアドレス生成部５０２に伝える。この応答信号を受けたアドレス生成部５０２は、次のアドレス信号とレングス信号の組をバスインターフェース５０３に出力して、画像メモリ３０４に対する次のリード要求を発行できる。

【００２９】
画像メモリ３０４から読み出されたビットマップデータは、一時的にバスインターフェース５０３内のバッファ５０３ａに格納され、即座に出力可能なデータはバッファ５０３ａからＦＩＦＯ（データ格納部）５０４に送られて記憶される。ＤＭＡコントローラ３０５は、ブレンド処理部３０６が一時的にデータを入力できない期間が発生しても、画像メモリ３０４からビットマップデータを読み出してＦＩＦＯ５０４に記憶する。これにより、ブレンド処理部３０６がデータ入力可能となった時に、直ちに、ＦＩＦＯ５０４からブレンド処理部３０６にビットマップデータ（data）を供給することができる。

【００３０】
バスインターフェース５０３は、ＦＩＦＯ５０４から出力される、ＦＩＦＯ５０４にデータを書き込むべき空きがないことを示すＦＩＦＯフル信号（full）を監視している。こうしてフル信号が出力されてＦＩＦＯ５０４がフル状態を示している場合には、リードトランザクションを発行せずに、フル状態が解除されるのを待つ。

【００３１】
ブレンド処理部インターフェース５０５は、このＦＩＦＯ５０４に蓄えられたビットマップデータをブレンド処理部３０６に送る。ブレンド処理部インターフェース５０５はＦＩＦＯ５０４から出力される、ＦＩＦＯ５０４に蓄えられたデータがないことを示すＦＩＦＯエンプティ信号（empty）を監視する。ブレンド処理部インターフェース５０５は、ＦＩＦＯ５０４がエンプティ状態でなく、かつブレンド処理部３０６がデータを入力できる状態であれば、ＦＩＦＯ５０４からビットマップデータをリードし、そのデータをブレンド処理部３０６へ送り出す。

【００３２】
［レジストレーションずれ補正の原理］
次に、レジストレーションずれ補正の原理について、図４と図５を用いて説明する。

【００３３】
図４（Ａ）（Ｂ）及び図５（Ａ）～（Ｄ）は、本発明の実施の形態に係るレジストレーションずれ補正の原理を説明するための図である。

【００３４】
図４（Ａ）は、レジストレーションずれの原因となる、画像形成時の走査ラインの曲が

りを説明する図である。図において、横軸は走査ラインの主走査方向（ライン方向）を示し、縦軸は副走査方向を示している。図４（Ｂ）は、この走査ラインの曲がりに合わせて、各走査ラインに対応するラインデータに切り換えながら、ビットマップデータを印刷部１０７へ送り出す様子を示している。ここで、ｎラインが、本来印刷すべき印刷対象の走査ラインを示している。図４（Ｂ）において、黒い部分がｎラインの印刷時に印刷部１０７へ送り出されるラインデータを示している。このように図４（Ａ）に示す走査ラインの曲がりに合わせて、印刷部１０７に出力するデータのラインを切り換えることにより、露光走査に曲がりが生じても、感光ドラム等の像担持体上に生成された静電潜像は歪みのないものにできる。

【００３５】

図５（Ａ）は、このようなレジストレーションずれの原因となる、曲がりが生じた走査ラインの一部を拡大して示す図である。図５（Ｂ）は、この走査ラインの曲がりに合わせてデータを切り換えながら、ビットマップデータを印刷部１０７へ送り出す様子を示す図である。但し、図５（Ａ）のように、単に、ビットマップデータのラインデータを切り換えて印刷部１０７に送ると、その切り換え部分に不自然な段差を生じてしまう。従って図５（Ｂ）に示すように、その切り換え部分のドットサイズを変更し、かつ２つのラインが重複するようにドットを形成することにより、この不自然な段差を解消して画像の平滑化を図っている。こうしてラインの切り換え部分の前後で隣接する２ライン分のデータに階調を付けて（ブレンド処理）段差を目立たなくしている。尚、このようなブレンド処理は、ブレンド処理部３０６で実施される。

【００３６】

本実施の形態に係るＤＭＡコントローラ３０５は、ラインバッファを削減するために走査ラインの曲がりに合わせて、ラインデータを切り換えるようにアドレスを生成する。即ち、ライン切り換え前の第１ラインと、ライン切り換え後の第２ラインからバースト転送にて画像データが読み出されるように、第２ラインから画像データを読み出すための第２の読み出し開始アドレスを、そのライン切り換えアドレスに基づいて生成する。このブレンド処理に必要となるビットマップデータは、１ライン前のデータのみを参照することに注目し、ブレンド処理部３０６では、最大１ライン分のバッファのみを有することにより、ブレンド処理が可能となる。

【００３７】

［実施の形態１］

次に本発明の実施の形態１に係るＤＭＡコントローラ３０５の動作について図６と図７を参照して説明する。

【００３８】

図６は、本発明の実施の形態１に係るＤＭＡコントローラ３０５に走査ラインの曲がりとバースト転送を説明するための図である。尚、ここではビットマップデータの一走査ラインを等分に分割して考える。以下、この分割された一つをセグメントと呼ぶことにする。尚以下の説明において、セグメント長、各セグメントが基準ラインに対して何ラインのずれ量かを示す乗り換え情報、各種アドレス、データ長等の英字で示す名称は、レジスタ部５０１に含まれるレジスタ名を示している。前述のようにＣＰＵ３０８は、これらレジスタ部５０１の各レジスタに値を書き込むことによりＤＭＡコントローラ３０５に対して動作を指示する。

【００３９】

各セグメントの長さを設定するレジスタRegSegLenを、ＤＭＡコントローラ３０５のレジスタ部５０１に設ける。また走査ラインの曲がりに合わせて、各セグメントが基準ラインに対するずれ量を指示するレジスタ群RegSegPosi[i]を必要な数だけ設ける。尚、変数ｉ（＝０～Ｎ）は、出現するセグメントの境界が何番目であるか、即ち，セグメントの番号を示す。尚、本実施の形態１では、１走査ラインにおいて、セグメント長RegSegLenの値は固定としている。

【００４０】

図において、６００は、ｎラインを基準として副走査方向に走査ラインがずれた状態を示している。６０１はｎラインのデータを最大バースト長（fuu_burst_length）で読み取った状態を示す。６０２は、（ｎ＋１）ラインのデータを最大バースト長で読み取った状態を示す。同様に６０３は、（ｎ＋１）ラインのデータを最大バースト長で読み取った状態、そして６０４は（ｎ＋２）ラインのデータを最大バースト長で読み取った状態をそれぞれ示している。

【００４１】

ＤＭＡコントローラ３０５のレジスタ部５０１には、更に、画像メモリ３０４に記憶されるビットマップデータの先頭アドレスを指定するレジスタRegStartAddr、ビットマップデータのライン長を示すレジスタRegLineLenを含む。更に、ビットマップデータの隣り合うラインのアドレスのオフセット値RegLineOffset、ブレンド処理部３０６へ送るべきライン数を指定するRegBeamsを含む。更にＤＭＡコントローラ３０５のレジスタ部５０１には、上記各レジスタ加えて、ＣＰＵ３０８が上記のレジスタの設定を完了した後に、ＤＭＡ動作を開始させるための起動レジスタ（不図示）を含んでいる。

【００４２】

図７は、本発明の実施の形態１に係るＤＭＡコントローラ３０５のアドレス生成部５０２（生成手段）の動作を示すフローチャートである。尚、以下の説明では、説明を簡単にするために各データなどの名称を各対応するレジスタ名で代用する。

【００４３】

まずステップＳ１００で、ＣＰＵ３０８の指示によりＤＭＡ動作が開始されると、各走査ラインの開始アドレスline_start_addrを、ビットマップデータの先頭アドレスRegStartAddrの値で初期化する。またブレンド処理部３０６へ送ったライン数をカウントするline_cntを「０」で初期化する。次にステップＳ１０１に進み、各走査ライン毎の初期化を行う。ここでは読み出しアドレスポインタaddr_pointを、走査ラインの開始アドレスline_start_addrの値で初期化する。また、この走査ライン上のどのセグメントを有効にしてブレンド処理部３０６へデータを転送したかを示すscore_board[0], score_board[1], ..., score_board[N]の各々を「０」で初期化する。尚、score_board[i]が「０」となっている場合は、ｉ番目のセグメントのデータが有効な画像データとして読み出されていないことを示している。またscore_board[i]が「１」となっている場合は、ｉ番目のセグメントのデータが有効な画像データとして読み出されていることを示す。また、参照すべきレジスタ群RegSegPosi[i]の変数ｉ（index）を「０」で初期化する。

【００４４】

次にステップＳ１０２で、読み出し要求毎の初期化を行う。ここではバスインターフェース５０３へ要求すべきアドレスとレングス信号（addrとlength）をそれぞれ、読み出しアドレスポインタaddr_pointと、最大バースト長full_burst_lengthで初期化する。この最大バースト長は、複数のセグメント長を含む長さである。これらの信号は以降のステップで変更されることがある。また開始アドレスに相当するセグメントのラインずれ量seg_posiにレジスタRegSegPosi[i]の値をセットする。

【００４５】

次にステップＳ１０３に進み、バースト転送が有効に行えないようなライン切り替えが発生する場合においても、読み出したデータをバースト転送可能なように、addrの下位ビットをマスクする。これにより、full_burst_lengthでのバースト転送が可能なアドレスとなるようにaddrを変換する。これは例えば図６の６０２で、データの読出し開始アドレスが「0x4048」の場合、このアドレスの下位ビットをマスクして、例えばアドレス「0x4040」に変換する。これにより、最大バースト長でのデータ転送が可能なアドレスに変換する。本実施の形態では、セグメントの切り替わるアドレスが、バースト転送で、きりの良いアドレスになるとは限らないので、最大バースト長full_burst_lengthで効率良くバースト転送するために、このようなアドレス変換を行っている。尚、１回のバースト転送で複数のセグメントのデータが読み出される場合、各セグメントごとに、そのデータが有効であるかどうかを判断して、有効なセグメントのscore_board[i]に「１」を設定する。

【００４６】
例えば図６の(n+1)ラインのデータに着目すると、最大バースト長で読み出すと、２番目のセグメント（ReggSegPosi[1]）と４番目のセグメントに対応するscore_board[3]がともに「１」となる。尚、この(n+1)ラインのデータを読み出す際、３番目のセグメントには有効データがないため、score_board[2]は「０」のままとなる。
【００４７】
次にステップＳ１０４に進み、アドレスaddrと読出しデータ長length（ここでは最大バースト長）をバスインターフェース５０３へ出力して画像メモリ３０４からのビットマップデータの読み出し要求を行う。次にステップＳ１０５で、読み出されたビットマップデータから、score_borad[i]の値が「１」を示すセグメントのデータを、セグメントの順にバスインターフェース５０３のバッファ５０３ａに一時的に格納する。ここで、このバッファ５０３ａの容量は、最大バースト長で読み出されるデータを記憶できる容量である。こうしてバッファ５０３ａに格納されたデータは、セグメント順（走査ラインの左側から順）にＦＩＦＯ５０４へ出力される。但し、ステップＳ１０５では、例えば図６の６０２で読み出したデータは、２番目と４番目のセグメントのデータを含んでいる。しかし、その間の３番目のセグメントのデータの読出しが完了していないので、４番目のセグメントのデータは、６０３で（ｎ＋２）ラインのデータを読み出した後にバッファ５０３ａに格納する。従って図６に示すように、ＦＩＦＯ５０４には、この走査ラインで記録される画像データがセグメント順に格納される。
【００４８】
次にステップＳ１０６に進み、バスインターフェース５０３からackを受信すると、読み出しアドレスポインタaddr_pointと変数ｉとを更新するために、走査ラインの読み出されていないデータの位置をscore_boardの値に基づいて判定する。そして、次の読み出しアドレスがいくつ先のセグメントであるかを算出し、その算出分（Ｍ）、変数ｉを＋Ｍしてアドレスaddr_pointを（＋RegSegLen×Ｍ）に設定する
ここでaddr_pointは、最初は走査ラインの開始アドレスline_start_addrで初期化されているので、図６の６０１でｎラインのデータを読み出す際は0x3000となっている。ｎラインのデータを読み出した時点でのscore_board[0]は「１」であるが、score_board[1], score_board[2]はともに「０」のままである。そこで、score_boardの値から読み出されていないデータの位置として、次に２番目のセグメント（RegSegPosi[1]）を特定する。そして、addr_pointを１つ先のセグメントに進めるために、RegSegLen×１をaddr_pointに加算する。
【００４９】
次にステップＳ１０７で、次のRegSegPosi[i]と、前のRegSegPosi[i-M]とを比較し、切り替え情報の差分を算出する。この切り替え情報の差分とは、何ライン分の切り替えを行うかを示すものである。例えば、６０１で示すデータから、６０２で示すデータに切り替える場合は、ｎラインから（ｎ＋１）ラインへの切替なので、そのライン数Ｌ＝１となる。ここで切り替え情報の差分が負の場合はステップＳ１０８へ進む。一方、切り替え情報の差分Ｌが正の場合はステップＳ１０９へ進む。また切り替え情報の差分が「０」のときステップＳ１１０へ進む。ステップＳ１０８では、図６の６００で上側のラインへの切り替えを意味するためアドレスポインタaddr_pointから、上述したラインオフセットRegLineOffset×Ｌ分を減じる。一方、ステップＳ１０９では、図６の６００の下側のラインへの切り替えを意味するためアドレスポインタaddr_pointに、RegLineOffset×Ｌを加算する。こうしてステップＳ１０８或はステップＳ１０９を実行するとステップＳ１１０に進む。
【００５０】
ステップＳ１１０では、走査ライン上のどの有効なデータを読み出したかを示すscore_boardを参照し、その走査ラインで記録するデータを全て読み出したかどうかを判断する。ここでは、score_board[0], score_board[1], ..., score_board[N]は、各セグメントから有効データが読み出されると「１」が設定される。図６の例では、６０１と６０２で

データを読み出した時点では、score_board[0], score_board[1], score_board[3]が「１」となる。そしてscore_board[2]を含む他のscore_boardの値は「０」のままである。そして６０３で示すデータを読み出した時点で、score_board[2]が「１」となる。同様に６０４で示すデータを読み出すと、core_board[4]が「１」となる。こうして、この１走査ライン分のscore_board[0], score_board[1]～score_board[N]が全て「１」になるとステップＳ１１０からステップＳ１１１に進むことになる。ここで、走査ライン上で読み出し完了していない場合はステップＳ１０２に戻り、前述の読み出し処理を繰り返す。

【００５１】

そしてステップＳ１１１、次のラインデータを処理するために各データを更新する。ここでは各ラインの開始アドレスline_start_addrに、ビットマップデータの隣り合うラインのアドレスのオフセット値RegLineOffsetを加えて、次のラインの先頭アドレスを設定する。また、ブレンド処理部３０６へ出力したライン数line_cntを＋１する。次にステップＳ１１２で、出力したライン数line_cntと、ブレンド処理部３０６へ送るべきライン数RegBeamsとを比較する。ここで出力済みのライン数line_cntが、送るべき全ライン数RegBeams未満であれば、まだ処理すべきラインデータが残っているのでステップＳ１０１に戻り、前述のラインの処理を繰り返す。そうでないときは、１ページのデータの処理が終了したのでＤＭＡ動作を終了する。尚、ＤＭＡコントローラ３０５はＤＭＡ動作を終了した時に割り込み信号（不図示）によりＣＰＵ３０８に伝える。ＣＰＵ３０８はこの割り込みを検出することで、ＤＭＡ転送を終了したことを検知する。

【００５２】

以上のように、ＤＭＡコントローラ３０５は、複数のデータセグメントで構成される１走査ラインの画像データを読み出して画像メモリ３０４から印刷部１０７へ転送する。そして、印刷部１０７は、感光体ドラム（像担持体）を１走査ラインの画像データに基づいてレーザ光にて走査して、感光ドラム上に１走査ラインの静電潜像を形成する。ここで、静電潜像とは、所定の電位に帯電された感光体ドラムをレーザ光を照射（走査）することで、感光ドラム上に生成される画像データに応じた電位差のある像をいう。そして、印刷部１０７は、静電潜像に対してトナーを付着させることで、電位差に応じたトナー像が感光体ドラム上に形成し、更にトナー像を用紙上に転写することでシート上に画像を形成する。

【００５３】

尚、上記説明では、本実施の形態に係るデータ転送の制御をＤＭＡコントローラ３０５で実現した例で説明したが、ＣＰＵ３０８の処理時間に余裕があればＣＰＵ３０８により制御しても良い。また或は、ＣＰＵ３０８以外の別のＣＰＵや、ＤＳＰ（デジタルシグナルプロセッサ）等の制御の下に実現されても良い。更に、カラー画像形成装置において、複数色（ＣＹＭＫ）の各色に対応する像形成ユニットを有する場合には、各色の像形成ユニット毎に本実施の形態に係るデータ転送方式を実施することで、各色毎のレジストレーションずれによる色ずれを防ぐことができる。これは以下に説明する各実施の形態においても同様である。

【００５４】

以上説明したように本実施の形態１によれば、各色毎に走査ラインの曲がりが発生している場合でも、各色毎のレジストレーションずれ量に応じた数のラインバッファを設けなくても、それらレジストレーションずれによる色ずれを防ぐことができる。また各ラインの切り換え時に、そのライン切り換えによる段差を目立たなくするためのブレンド処理を、１ライン分のラインバッファを設けることで実現することができる。

【００５５】

［実施の形態２］

次に本発明の実施の形態２に係るＤＭＡコントローラ３０５の動作について図８と図９を用いて説明する。尚、この実施の形態２に係る画像形成装置のハードウェア構成は、前述の実施の形態１と同様であるため、その説明を省略する。この実施の形態２では、１走査ラインにおいて、セグメント長RegSegLenの値は可変としている。

【００５６】
図８は、本発明の実施の形態２に係るＤＭＡコントローラ３０５に走査ラインの曲がりを指定する方法を説明するための図である。
【００５７】
ここでは走査ラインの曲がりに合わせて、ラインの切り換え位置までの長さ（セグメント長）を指定するレジスタ群RegSegLen[i]を設ける。従って、各セグメントの長さは一定ではなく、各セグメントの境界がライン切替えが発生する位置となる。更に、各セグメントが基準ラインに対するずれ量を指示するレジスタ群RegSegPosi[i]を必要な数だけＤＭＡコントローラ３０５のレジスタ部５０１に設ける。このＤＭＡコントローラ３０５のレジスタ部５０１は更に、画像メモリ３０４に記憶されるビットマップデータの先頭アドレスRegStartAddr、ビットマップデータの各走査ラインの長さRegLineLenがセットされる。またレジスタ部５０１には、ビットマップデータの隣り合うラインデータ同士のアドレスのオフセット値RegLineOffset、ブレンド処理部３０６へ送るべきライン数RegBeamsがセットされる。更に、ＤＭＡコントローラ３０５のレジスタ部５０１は、ＣＰＵ３０８が上記レジスタの値の設定を完了した後に、ＤＭＡ動作を開始させるための起動レジスタ（不図示）も含んでいる。
【００５８】
図９は、本発明の実施の形態２に係るＤＭＡコントローラ３０５のアドレス生成部５０２の動作を示すフローチャートである。
【００５９】
まずステップＳ２００で、ＣＰＵ３０８の指示によりＤＭＡ動作が開始されると、走査ラインの開始アドレスline_start_addrを、画像メモリ３０４のビットマップデータの先頭アドレスRegStartAddrの値で初期化する。また、ブレンド処理部３０６へ送ったライン数line_cntを「０」で初期化する。次にステップＳ２０１で、ライン毎の初期化を行う。読み出し開始アドレスaddr_pointを、走査ラインの開始アドレスline_start_addrの値で初期化する。この走査ライン上のどの有効なデータをブレンド処理部３０６へ転送したかを示すscore_boardを初期化する。また、参照すべきレジスタ群RegSegLen[i]とRegSegPosi[i]のインデックスを示す変数ｉを「０」で初期化する。
【００６０】
次にステップＳ２０２に進み、読み出し要求毎の初期化を行う。バスインターフェース５０３へ要求すべきアドレスaddrとデータ長lengthをそれぞれ、読み出しアドレスポインタaddr_pointと、最大バースト長full_burst_lengthで初期化する。これらの信号は以降のステップで変更される。読み出し開始アドレスに相当するセグメントのラインずれ量seg_posiにレジスタRegSegPosi[i]の値を設定する。
【００６１】
次にステップＳ２０３に進み、ステップＳ１０３と同様に、addrの下位ビットをマスクすることによりバースト転送が可能なアドレス境界に設定する。ここでRegSegLen[i]に対応するscore_board[i]を「１」に設定する。
【００６２】
次にステップＳ２０４に進み、アドレスaddrとデータ長length（バースト長）をバスインターフェース５０３へ出力して画像メモリ３０４からのビットマップデータの読み出し要求を行う。次にステップＳ２０５で、前述の図７のＳ１０５と同様に、読み出されたビットマップデータから、score_boradの「１」に対応するデータを順に抽出してバスインターフェースが持つデータバッファ５０３ａに一時的に格納する。こうしてデータバッファ５０３ａに格納されたデータは走査ライン順にＦＩＦＯ５０４に出力される。ここで、上述するデータバッファ５０３ａの容量は最大バースト長で読み出されるデータを格納できる容量である。
【００６３】
ここで、この実施の形態２では、各セグメントの長さは可変長であるため、最大バースト長full_burst_lengthがセグメント長RegSegLen[i]に対して短い場合がある。このとき

、一つのセグメント長に対して、複数回のトランザクション要求を行う必要がある。ステップＳ２０６では、addr、lengthを参照し、前記読み出し要求が１セグメント分RegSegLen[i]処理したかを判断する。そのセグメントで読み出しが完了していない場合、ステップＳ２０７において、バスインターフェース５０３からackを受信すると、アドレスaddrをデータ長length分足した値に更新する。こうして読み出しが完了していないビットマップデータを読み出せるようにする。各セグメントの読出しが完了した場合、ステップＳ２０８に進む。

【００６４】

ステップＳ２０８では、バスインターフェース５０３からackを受信すると、読み出しアドレスポインタaddr_pointと変数ｉを更新するために、score_boardの値から読み出されていないセグメントの位置を判定する。そして、そのセグメントの先頭を次の読み出しアドレスaddr_pointに設定する。

【００６５】

次にステップＳ２０９に進み、現セグメント（RegSegPosi[i]）と、次のセグメントRegSegPosi[i+1]との差分）×RegLineOffsetをaddr_pointに加える。そして変数ｉに１を加えて、次の読み出しラインの切り替えを行う。

【００６６】

図８の例で説明すると、ｎラインのデータを読み出す場合、最大バースト長の２倍がRegSegLen[0]に等しいとする。これによりscore_board[0]は「１」となるが、１度の読み出し要求ではセグメント０の半分のデータがバッファ５０３ａに格納される。次にセグメント０のデータを全て読み出すために、アドレスをバースト長分足した値に更新する(Ｓ２０７)。次にセグメント１のデータを読み出すために、（RegSegPosi[0]と次のセグメントRegSegPosi[1]との差分）×RegLineOffsetによりセグメント１の先頭アドレスを求める。

【００６７】

次にステップＳ２１０に進み、図７のステップＳ１１０と同様に、走査ライン上のどの有効なデータを読み出したかを示すscore_boardを参照し、そのラインのデータを全て読み出し済みかどうかを判断する。その走査ラインで読み出し完了していない場合、ステップＳ２０２に戻り、前述の読み出し処理を繰り返す。一方、その走査ラインの読出しを完了したときはステップＳ２１１に進み、次のラインデータを処理するために各データを更新する。ここでは各ラインの開始アドレスline_start_addrに、ビットマップデータの隣り合うラインのアドレスのオフセット値RegLineOffsetを加えて、次のラインの先頭アドレスを設定する。また、ブレンド処理部３０６へ出力したライン数line_cntを＋１する。次にステップＳ２１２で、出力したライン数line_cntと、ブレンド処理部３０６へ送るべきライン数RegBeamsとを比較する。ここで出力済みのライン数line_cntが、送るべき全ライン数RegBeams未満であれば、まだ処理すべきラインデータが残っているのでステップＳ２０１に戻り、前述のラインの処理を繰り返す。そうでないときは、１ページのデータの処理が終了したのでＤＭＡ動作を終了する。尚、ＤＭＡコントローラ３０５はＤＭＡ動作を終了した時に割り込み信号（不図示）によりＣＰＵ３０８に伝える。ＣＰＵ３０８はこの割り込みを検出することで、ＤＭＡ転送を終了したことを検知する。

【００６８】

以上説明したように本実施の形態２によれば、前述の実施の形態１の効果に加えて、走査線の曲がりに応じて各セグメントの長さを可変にして、各色毎のレジストレーションずれによる色ずれを防ぐことができる。

【００６９】

［実施の形態３］

本発明の実施の形態３に係るＤＭＡコントローラ３０５の動作について、図１０（Ａ）（Ｂ）、図１１及び図１２を用いて説明する。尚、この実施の形態３に係る画像形成装置のハードウェア構成は、ＤＭＡコントローラ３０５以外は前述の実施の形態１と同様であるため、その説明を省略する。

【００７０】

図１０（Ａ）は、本発明の実施の形態３に係るＤＭＡコントローラ３０５に対して走査ラインの曲がりを指定する方法を説明するための図である。ここでは前述の実施の形態１と同様に、ビットマップデータの１ラインを等分に分割する場合を考える。各セグメントの長さRegSegLenを、ＤＭＡコントローラ３０５のレジスタ部５０１に設定する。ここでは実施の形態１と異なり、走査ラインの曲がりに合わせて、各セグメントが基準ラインに対して何ラインのずれ量かを示す乗り換え情報SegPosi[i]を用いる。ここで変数ｉは、セグメントの境界の順番を示す。

【００７１】

図１０（Ｂ）は、この情報SegPosi[i]を、画像メモリ３０４或はＲＡＭ３１０にテーブルとして格納した状態を示している。

【００７２】

このテーブルの先頭アドレスRegTableStartAddrを、ＤＭＡコントローラ３０５のレジスタ部５０１にセットする。このＤＭＡコントローラ３０５のレジスタ部５０１には、更に、画像メモリ３０４に記憶されるビットマップデータの先頭アドレスRegStartAddr、ビットマップデータのライン長RegLineLenがセットされる。更に、ビットマップデータの隣り合うラインのアドレスのオフセット値RegLineOffset、ブレンド処理部３０６へ送るべきライン数RegBeamsを含んでいる。更にＤＭＡコントローラ３０５のレジスタ部５０１には、ＣＰＵ３０８が上記のレジスタ値の設定を完了した後に、ＤＭＡ動作を開始させるための起動レジスタ（不図示）を含んでいる。

【００７３】

図１１は、本発明の実施の形態３に係るＤＭＡコントローラ３０５の構成を示すブロック図で、前述の図３と共通する部分は同じ記号で示している。

【００７４】

レジスタ部５０１は、複数のレジスタ（不図示）で構成される。ＤＭＡコントローラ３０５に対するＣＰＵ３０８からの指示はレジスタ部５０１の各レジスタに適切な値を書き込むことで行なわれる。アドレス生成部５０２'は、画像メモリ３０４に記憶されるビットマップデータを読み出すためのアドレスを、レジスタ部５０１の各レジスタの内容を参照して生成する。また画像メモリ３０４のライン切り替え情報のテーブルをリードするためのアドレスも生成する。

【００７５】

アドレス生成部５０２'は、アドレスaddrと、そのアドレスから読み出すデータ量を示す読み出しデータ長lengthにより、バスインターフェース５０３へ画像メモリ３０４からのデータ読み出しを要求する。画像メモリ３０４からビットマップデータを読み出す場合は、要求信号reqを用いて要求し、テーブルのデータを読み出す場合には要求信号req_tableを用いて要求する。

【００７６】

バスインターフェース５０３'は、アドレス生成部５０２'からのアドレスとデータ長を受け、バス３１１に対して画像メモリ３０４のリードトランザクションを発行する。例えば、バス３１１のデータバス幅が３２ビットであれば、アドレスとデータ長から複数の３２ビット単位を何回連続リードするかを示すバースト長を指定し、リードトランザクションを発行する。バスインターフェース５０３'は、１組のアドレスとデータ長に対する処理が完了すれば、応答信号ackを用いて、処理完了をアドレス生成部５０２'に伝える。この応答信号ackを受けた生成部５０２'は、次のアドレスとデータ長をバスインターフェース５０３'に要求できる。バスインターフェース５０３'は、画像メモリ３０４から読み出したデータを受け取り、それがビットマップデータであればＦＩＦＯ５０４に書き込む。一方、テーブルのデータであれば、その読み出したテーブルデータをアドレス生成部５０２'へ送る。

【００７７】

こうして読み出されたビットマップデータから、score_boradの「１」に対応するデータを、一時的にバスインターフェース５０３'内のデータバッファ５０３ａに格納する。

そしてデータはデータバッファ５０３ａからＦＩＦＯ５０４に記憶される。ＤＭＡコントローラ３０５は、ブレンド処理部３０６が一時的にデータを入力できない期間が発生しても、ビットマップデータをＦＩＦＯ５０４に記憶できる。これにより、ブレンド処理部３０６がデータ入力可能となった時に、即座にＦＩＦＯ５０４からブレンド処理部３０６にビットマップデータを供給することができる。

【００７８】

バスインターフェース５０３'は、ＦＩＦＯ５０４から出力される、ＦＩＦＯ５０４にデータを書き込むべき空きがないことを示すＦＩＦＯフル信号fullを監視している。そして、ＦＩＦＯ５０４がフル状態を示している場合は、リードトランザクションを発行せずに、フル状態が解除されるのを待つ。またブレンド処理部インターフェース５０５は、ＦＩＦＯ５０４に蓄えられたビットマップデータをブレンド処理部３０６に送る。ブレンド処理部インターフェース５０５は、ＦＩＦＯ５０４から出力される、ＦＩＦＯ５０４にデータが無いないことを示すＦＩＦＯエンプティ信号emptyを監視する。そしてブレンド処理部インターフェース５０５は、ＦＩＦＯ５０４がエンプティ状態でなく、かつブレンド処理部３０６がデータを入力できる状態であれば、ＦＩＦＯ５０４からビットマップデータを読み出してブレンド処理部３０６へ送り出す。

【００７９】

図１２は、本発明の実施の形態３に係るＤＭＡコントローラ３０５のアドレス生成部５０２'の動作を示すフローチャートである。尚、この図１２のフローチャートは、ずれ量をテーブルに記憶しているための処理が異なる以外は、前述の図７のフローチャートと同様である。

【００８０】

まずステップＳ３００で、ＣＰＵ３０８の指示により、ＤＭＡ動作が開始されると、各ラインの開始アドレスline_start_addrを、ビットマップデータの先頭アドレスRegStartAddrの値で初期化する。また、ブレンド処理部３０６へ送ったライン数を記憶するline_cntを「０」で初期化する。次にステップＳ３０１で、ライン毎の初期化を行う。読出しアドレスポインタaddr_pointを、ライン開始アドレスline_start_addrで初期化する。この走査ライン上のどの有効なデータをブレンド処理部３０６へ転送したｔかを示すscore_boardを初期化する。また参照すべきライン切り替え情報SegPosi[i]のテーブルのアドレスtable_addrを、テーブルの先頭アドレスRegTableStartAddの値で初期化する。

【００８１】

次にステップＳ３０２で、読出し要求毎の初期化を行う。バスインターフェース５０３'へ要求すべきアドレスaddrと、読み出しデータ長lengthをそれぞれ、読出しアドレスポインタaddr_pointと、最大バースト長の値で初期化する。ここで、望ましくは最大バースト長は複数のセグメント長を含む長さである。これらの信号は以降のステップで変更されることがある。

【００８２】

ステップＳ３０３で、図７のステップＳ１０３と同様に、addrの下位ビットをマスクすることによりバースト転送可能なアドレス境界に設定する。次にステップＳ３０４に進み、テーブルアドレスtable_addrで、図１０（Ｂ）に示すテーブルデータを読み出すためにバスインターフェース５０３'へ要求する。このとき、上述したaddrに設定したアドレス境界から最大バースト長の範囲に含まれるセグメントの分のテーブルデータを要求する。そして、その読み出したテーブルデータからSegPosi[i]の値を、読出し開始アドレスに相当するセグメントのラインずれ量seg_posiに設定する。更に、addrに設定したアドレス境界から最大バースト長の範囲において、seg_posiと同じラインずれ量（SegPosi[i]）をもつセグメントに対応するデータがあるかを判断する。その判断に該当するデータに対応するscore_boardを「１」に設定する。ここで、該当するデータが含まれる長さのバースト長にlengthを更新する。

【００８３】

ステップＳ３０５で、アドレスaddrとデータ長lengthとで、バスインターフェース５０

３'を介して画像メモリ３０４よりビットマップデータを読み出す。次にステップＳ３０６で、読み出されたビットマップデータから、score_boradの値が「１」の対応するデータを、バスインターフェースが持つデータバッファ５０３ａに一時的に格納する。こうしてデータバッファ５０３ａに格納されたデータは、走査ライン順にＦＩＦＯ５０４へ出力される。ここで、上述するデータバッファ５０３ａの容量は、最大バースト長で読み出されるデータの容量を有する。

【００８４】

次にステップＳ３０７に進み、バスインターフェース５０３'からackを受信すると、読み出しアドレスポインタaddr_pointと変数ｉを更新するために、score_boardの値から読み出されていないデータの位置を判定する。そして次の読み出しアドレスがいくつ先のセグメントかを算出する。その算出分（Ｍ）、変数ｉに＋Ｍし、アドレスaddr_pointを（RegSegLen×Ｍ）に設定する。次にステップＳ３０８で、次のSegPosi[i]と前のSegPosi[i-M]とを比較し、切り替え情報の差分Ｌを算出する。ここで切り替え情報の差分Ｌが負の場合はステップＳ３０９へ進む。一方、切り替え情報の差分Ｌが正の場合はステップＳ３１０へ進む。また切り替え情報の差分Ｌが「０」のときステップＳ３１１へ分岐する。

【００８５】

ステップＳ３０９では、下のラインへの切り替えるため、アドレスポインタaddr_pointから、上述したラインオフセット（RegLineOffset×Ｌ）を減じる。またステップＳ３１０では、上のラインへの切り替えるため、アドレスポインタaddr_pointに（RegLineOffset×Ｌ）を加える。そしてステップＳ３１１に進み、走査ライン上のどの有効なデータを読み出したかを示すscore_boardを参照して、その走査ラインのデータを全て読み出したかを判断する。ここで、走査ライン上で読出し完了していないデータがある場合、ステップＳ３０２に戻り、前述の読出し処理を繰り返す。

【００８６】

次にステップＳ３１２に進み、次のラインの処理のために各データを更新する。ここでは各ラインの開始アドレスline_start_addrに、上記オフセット値RegLineOffsetを加えて、次のラインの先頭アドレスにする。またブレンド処理部３０６へ送ったライン数line_cntを＋１する。そしてステップＳ３１３で、ブレンド処理部３０６へ送ったライン数line_cntと、ブレンド処理部３０６へ送るべきライン数RegBeamsの値を比較して、１ページ分のデータ処理が終了したかを判断する。ここでライン数line_cntの値が、RegBeamsの値未満であれば、まだ処理すべきラインが残っているのでステップＳ３０１に戻り、次のラインの処理を繰り返す。そうでないときはＤＭＡ動作を終了する。ＤＭＡコントローラ３０５は、ＤＭＡ動作を終了した時に割り込み信号をＣＰＵ３０８に送る。ＣＰＵ３０８はこの割り込み信号を検出して、ＤＭＡが終了したことを検知する。これらステップＳ３０７～Ｓ３１３の処理は、前述の図７のステップＳ１０６～Ｓ１１２の処理と同じである。

【００８７】

以上説明したように本実施の形態３によれば、前述の実施の形態１の効果に加えて、各セグメントの長さを一定にしながら、走査線の曲がりに応じて各セグメントごとにラインを上或は下にずらすか、ずらさないかを指定することができる。これによりセグメント長を一定にしながらも、走査線の曲がりに応じたレジストレーションずれによる色ずれを防ぐことができる。

【００８８】

［実施の形態４］

本発明の実施形態４に係るＤＭＡコントローラ３０５の動作について、図１３と図１４を用いて説明する。尚、この実施の形態４に係る画像形成装置のハードウェア構成は、前述の実施の形態３と同様であるため、その説明を省略する。

【００８９】

図１３（Ａ）（Ｂ）は、本発明の実施の形態４に係るＤＭＡコントローラ３０５に走査ラインの曲がりを指定する方法を説明するための図である。ここで１ラインにおけるセグメント長を可変とし、各セグメントごとに、基準ラインに対するずれ量を設定できるもの

とする。

【００９０】

ここでは前述の実施の形態２，３と異なり、走査ラインの曲がりに合わせて、ライン切り換え位置までの長さ（セグメント長）SegLen[i]と、基準ラインに対するずれ量SegPosi[i]を変更している。図１３（Ｂ）は、セグメント長SegLen[i]と、各セグメントにおいて、基準ラインに対して何ラインずれているかを示すSegPosi[i]を格納しているテーブルデータの構成を示している。このテーブルは、画像メモリ３０４或はＲＡＭ３１０に格納されている。このテーブル読み出し時には、この読み出し開始アドレスRegTableStartAddrを、ＤＭＡコントローラ３０５のレジスタ部５０１にセットする。この実施の形態４においても、ビットマップデータの先頭アドレスRegStartAddr、ビットマップデータのライン長RegLineLenがセットされる。またビットマップデータの隣り合うラインのアドレスのオフセット値RegLineOffset、ブレンド処理部３０６へ送るべきライン数RegBeamsもレジスタ部５０１にセットされる。更に、ＤＭＡコントローラ３０５のレジスタ部５０１には、ＣＰＵ３０８が上記の各レジスタ値の設定を完了した後に、ＤＭＡ動作を開始させるための起動レジスタ（不図示）を含んでいる。

【００９１】

図１４は、本発明の実施の形態４に係るＤＭＡコントローラ３０５のアドレス生成部５０２'の動作を示すフローチャートである。尚、この図１４のフローチャートは、ずれ量をテーブルに記憶しているための処理が異なる以外は、前述の図９フローチャートと同様である。

【００９２】

まずステップＳ４００で、ＣＰＵ３０８の指示によりＤＭＡ動作が開始されると、各ラインの開始アドレスline_start_addrを、ビットマップデータの先頭アドレスRegStartAddrで初期化する。またブレンド処理部３０６へ送ったライン数line_cntを「０」で初期化する。次にステップＳ４０１で、ライン毎の初期化を行う。ここでは読出しアドレスポインタaddr_pointを、ラインの開始アドレスline_start_addrで初期化する。この走査ライン上のどの有効なデータをブレンド処理部３０６へ転送したかを示すscore_boardを初期化する。また参照すべきセグメントの長さSegLen[i]と、ライン切り替え情報SegPosi[i]のテーブルのアドレスtable_addrを、テーブルの先頭アドレスRegTableStartAddrの値で初期化する。また変数ｉを「０」にする。

【００９３】

次にステップＳ４０２で、読出し要求毎の初期化を行う。ここではバスインターフェース５０３'へ要求すべきアドレスaddrを、読出しアドレスポインタaddr_pointで初期化する。また読み出しデータ長lengthを、最大バースト長で初期化する。これらの信号は以降のステップで変更される。

【００９４】

次にステップＳ４０３に進み、ステップＳ２０３と同様に、addrの下位ビットをマスクすることによりバースト転送が可能なアドレス境界に設定する。次にステップＳ４０４で、アドレスtable_addrからテーブルデータを読み出すために、バスインターフェース５０３'へデータの読み出し要求を発行する。そして、そのテーブルから読み出したデータに基づき、addr_pointに相当するセグメントのSegPosi[i]をseg_posiに設定する。更に、seg_posiに対応するscore_boardを「１」に設定する。

【００９５】

次にステップＳ４０５で、アドレスaddrと読み出しデータ長（バースト長）lengthとを基に、バスインターフェース５０３'へリード要求を行う。これにより画像メモリ３０４のアドレスaddrから、指定されたデータ長length分のビットマップデータが読み出される。次にステップＳ４０６で、読み出されたビットマップデータから、score_boardが「１」の対応するデータを、バスインターフェース５０３'が持つデータバッファ５０３ａに一時格納する。データバッファ５０３ａに格納されたデータは走査ライン順にＦＩＦＯ５０４へ出力される。ここで上述したデータバッファ５０３ａの容量は、最大バースト長で

読み出されるデータを収容できる容量である。

【００９６】

次にステップＳ４０９に進み、バスインターフェース５０３'からackを受信すると、読み出しアドレスポインタaddr_pointと変数ｉを更新するために、score_boardの値を基に読み出されていないデータの位置を判定する。そして、その位置までの値を次の読出しアドレスポインタaddr_pointに設定する。次にステップＳ４１０に進み、（SegPosi[i]とSegPosi[i+1]との差分）×RegLineOffsetを、addr_pointに加える。また変数ｉに＋１することにより、次の読み出しラインの切り替えを行う。

【００９７】

次にステップＳ４１１に進み、走査ライン上のどの有効なデータを読み出したかを示すscore_boardを参照し、その走査ラインのデータをすべて読み出し済みかどうかを判断する。ここで、走査ライン上で読み出し完了していない場合、ステップＳ４０２に戻り、前述の読み出し処理を繰り返す。そしてステップＳ４１１で、次のラインデータを処理するために各データを更新する。ここでは各ラインの開始アドレスline_start_addrに、ビットマップデータの隣り合うラインのアドレスのオフセット値RegLineOffsetを加えて、次のラインの先頭アドレスを設定する。また、ブレンド処理部３０６へ出力したライン数line_cntを＋１する。次にステップＳ４１２で、出力したライン数line_cntと、ブレンド処理部３０６へ送るべきライン数RegBeamsとを比較する。ここで出力済みのライン数line_cntが、送るべき全ライン数RegBeams未満であれば、まだ処理すべきラインデータが残っているのでステップＳ４０１に戻り、前述のラインの処理を繰り返す。そうでないときは、１ページのデータの処理が終了したのでＤＭＡ動作を終了する。尚、ＤＭＡコントローラ３０５はＤＭＡ動作を終了した時に割り込み信号（不図示）によりＣＰＵ３０８に伝える。ＣＰＵ３０８はこの割り込みを検出することで、ＤＭＡ転送を終了したことを検知する。

【００９８】

以上説明したように本実施の形態４によれば、前述の実施の形態１の効果に加えて、各セグメントの長さを可変にしながら、走査線の曲がりに応じて各セグメントごとにラインを上或は下にずらすか、ずらさないかを指定することができる。これにより走査線の曲がりに応じたレジストレーションずれによる色ずれを防ぐことができる。

【００９９】

［実施の形態５］

本発明のＤＭＡコントローラ３０５の動作について、図１５、図１６及び図１７を用いて説明する。尚、この実施の形態５に係る画像形成装置のハードウェア構成は、前述の実施の形態３と同様であるため、その説明を省略する。

【０１００】

図１５は、本発明の実施の形態５に係るＤＭＡコントローラ３０５に有効画像領域を、メモリアドレスで指定する方法を説明するための図である。

【０１０１】

このＤＭＡコントローラ３０５のレジスタ部５０１には、ある走査ラインに相当するデータの読み出し開始位置RegBeamStartAddr、有効画像領域の先頭アドレスRegLowerAddrがセットされる。更に、有効画像領域の後端後のアドレスRegUpperAddr、１ページ内における全読み出し走査ライン数RegBeamLinesがセットされる。更に、ライン間のアドレスのオフセット値RegLineOffset、印刷有効領域外のデータRegFillDataが設定できる。

【０１０２】

図１６は、本発明の実施の形態５に係るＤＭＡコントローラ３０５のアドレス生成部５０２'の動作を示すフローチャートである。

【０１０３】

まずステップＳ５０１で、１ページのＤＭＡ処理に対して初期化を行う。ラインの開始アドレスline_start_addrに、走査ラインに相当するデータの読み出し開始位置RegBeamStartAddrを設定し、ページ内の処理済みライン数lines_cntに「０」をセットする。次にス

テップＳ５０２で、ＤＭＡ開始アドレスが、画像の有効領域内のアドレスか否かを判定する。もし有効領域外の場合はステップＳ５０３に進み、ライン開始位置状態line_start_stateに有効領域外であることを示す値を入力する。一方ステップＳ５０２で有効領域内であればステップＳ５０４に進み、line_start_stateに有効領域内であることを示す値をセットする。

【０１０４】

次にステップＳ５０５に進み、各ラインごとの処理の初期化を行う。ここではステップＳ５０５以降の処理を、このページのライン数分繰り返すことになる。ここでは読み出しアドレスポインタaddr_pointに、ラインの開始アドレスline_start_addr、読み出しデータ長lengthを最大バースト長にセットする。また状態stateには、上述のライン開始位置状態line_start_stateをセットし、テーブルアドレスtable_addrには、テーブルの先頭アドレスRegTableStartAddrをセットする。

【０１０５】

次にステップＳ５０６に進み、バスインターフェース５０３'により、テーブルアドレスに基づいてテーブルから曲がり情報を読み出する。そしてステップＳ５０７に進み、その読み出したテーブルデータに基づいて、セグメント長seg_lenと、境界部分での上下ラインへの移動があるかを示す情報UpDown[i]をセットする。

【０１０６】

次にステップＳ５０８に進み、前述の状態stateに基づいて、読み出しアドレスポインタが有効領域内であるか否かを判定する。ここで領域内であればステップＳ５０９に進み、バスインターフェース５０３'に対して読み出し要求を発行する。一方、領域外であればステップＳ５１０に進み、印刷有効領域外のデータRegFillDataのデータをセグメント長seg_len分、後段の画像処理ブロックに転送する。ここでステップＳ５０９においては、seg_lenが読み出しデータ長lengthに対して大きい場合、実施の形態３で示したＳ２０４，Ｓ２０５，Ｓ２０６，Ｓ２０７同様、複数回読み出し要求を行い、各セグメント長の読み出しを完了する。こうしてステップＳ５０９或はＳ５１０を実行した後ステップＳ５１１に進み、アドレスポインタaddr_pointを更新する。ここではアドレスポインタaddr_pointに、セグメント長seg_lenを加算し、テーブルアドレスtable_addrに＋８する。そしてステップＳ５１２に進み、１ライン分の終了が終了したかどうかを判定する。この時、曲がり判定が終了していなければステップＳ５１３に進み、状態state、アドレスポインタaddr_point、上下ラインへの切り換え情報UpDownを更新してステップＳ５０６へ戻る。またステップＳ５１２で１ライン分の処理を終了している場合はステップＳ５１４に進む。

【０１０７】

図１７は、図１６のステップＳ５１３における状態state及びアドレスの更新処理の詳細を示すフローチャートである。

【０１０８】

まずステップＳ５２１で、上下ラインへの切り換え情報UpDownをチェックする。ここで曲がり情報が上ラインであればステップＳ５２２に進み、アドレスポインタaddr_pointから、ライン間のアドレスのオフセット値RegLineOffsetを差し引く。一方、曲がり情報が下ラインであればステップＳ５２８に進み、アドレスポインタaddr_pointに、このオフセット値RegLineOffsetを加算する。

【０１０９】

上ラインの場合はステップＳ５２２からステップＳ５２３に進み、状態stateが有効領域内か、有効領域の上側、或は下側のいずれを示しているかを判定する。ここで有効領域の上側であれば、それよりも上のラインは有効領域外で、有効領域の上側のままであるため処理を終了する。状態stateがが有効領域内を示していればステップＳ５２４に進み、更新した上ラインのアドレスポインタaddr_pointと、有効画像領域の先頭アドレスRegLowerAddrとの大小関係をチェックする。ここで有効画像領域の先頭アドレスRegLowerAddrが、アドレスポインタaddr_pointより大きい場合は更新後の上ラインアドレスは有効領域で

あるためステップＳ５２５に進み、状態stateに有効範囲外であることを示す値をセットする。一方ステップＳ５２４で、先頭アドレスRegLowerAddrが、アドレスポインタaddr_pointに等しいか、或はアドレスポインタaddr_pointより小さい場合は有効領域のままであるため、状態stateを変更せずに処理を終了する。

【０１１０】

またステップＳ５２３で、状態stateが有効領域の下側である場合はステップＳ５２６に進み、アドレスポインタaddr_pointと、有効画像領域の後端後のアドレスRegUpperAddrとの大小関係を判定する。ここで後端後のアドレスRegUpperAddrが、アドレスポインタaddr_pointより大きい場合は下ラインは有効領域内であるためステップＳ５２７に進み、状態stateを有効範囲内を示す値にセットする。一方、後端後のアドレスRegUpperAddrが、アドレスポインタaddr_pointより小さい場合は、その下ラインは有効領域の下のままであるため状態stateを変更せずに処理を終了する。

【０１１１】

また一方、下方向の場合はステップＳ５２８からステップＳ５２９に進み、状態stateが有効領域内か、有効領域の上側か、下側かのいずれを示しているかをみる。有効領域の下側であれば、それよりも下のラインは無条件に有効領域外であるため、状態stateを変更せずに処理を終了する。一方、有効領域の上側の場合はステップＳ５２９からステップＳ５３２に進み、更新後のアドレスポインタaddr_pointと有効画像領域の先頭アドレスRegLowerAddrとの大小関係をチェックする。ここで先頭アドレスRegLowerAddrがアドレスポインタaddr_pointより大きい場合は、有効領域外であるため状態stateを変更せずに処理を終了する。またステップＳ５３２で、RegLowerAddrがアドレスポインタaddr_pointより小さい場合はステップＳ５３３に進み、状態stateを有効領域内を示す値に変更する。またステップＳ５２９で、状態stateが領域内を示していればステップＳ５３０に進み、アドレスポインタaddr_pointと、後端後のアドレスRegUpperAddrとの大小関係をチェックする。ここで後端後のアドレスRegUpperAddrがアドレスポインタaddr_pointより大きい場合は、状態stateを変更せずに処理を終了する。一方、後端後のアドレスRegUpperAddrがアドレスポインタaddr_pointより小さい場合は、状態stateに有効領域の下側であることを示す値をセットする。こうして状態stateを更新した後図１６のステップＳ５０６に戻る。

【０１１２】

再び図１６に戻り、ステップＳ５１４では、次のライン処理へ移動するための処理として、ライン開始位置状況line_start_stateと、ラインの開始アドレスline_start_addrとを更新する。そしてステップＳ５１５に進み、ページの処理済みライン数lines_cntをカウントアップ（＋１）する。そしてステップＳ５１６で、１ページの処理が終了したか否かを判定する。ここで１ページ分の処理が全て終了していれば処理を終了し、終了していない場合はステップＳ５０５へ戻って前述の処理を実行する。

【０１１３】

以上説明したように本実施の形態５によれば、上述の実施の形態１に係る効果に加えて、ＤＭＡコントローラが走査ラインの曲がりを相殺するように曲がった画像を読み出し、有効領域外では、指定したデータを生成できる。これにより、例えばプリンタエンジンがマスク機能を有していないような、走査ラインに曲がりが発生する小型のプリンタ部でも高品位な出力を得ることができる。

【０１１４】

[実施の形態６]

次に本発明の実施の形態に係るＤＭＡコントローラ３０５の動作について、図１８及び図１９を用いて説明する。尚、この実施の形態６に係る画像形成装置のハードウェア構成は、前述の実施の形態３と同様であるため、その説明を省略する。

【０１１５】

図１８は、本発明の実施の形態６に係るＤＭＡコントローラ３０５に有効画像領域を、メモリ上のアドレスで指定する方法を説明するための図である。

【０１１６】
曲がり情報を格納するテーブルの先頭アドレスをRegTableStartAddrで、画像データが格納されるメモリの先頭アドレスをRegStartAddrで表す。また、ビームの開始位置を示すインデックスをRegStartLineIndex（本実施の形態６では、上下２ライン分の曲がりを例とし「－２」を設定している。「０」が有効領域の先頭ラインに対応している）としている。更に、１ページの走査ライン数をRegBeamLines、ライン間のアドレスオフセット値をRegLineOffset、印刷有効領域外のデータRegFillDataを設定可能である。
【０１１７】
図１９は、本発明の実施の形態６に係るＤＭＡコントローラ３０５のアドレス生成部５０２'の動作を示すフローチャートである。
【０１１８】
まずステップＳ６０１で、この処理のための初期化処理を行う。ここではラインの開始アドレスline_start_addrに、画像データが格納されるメモリの先頭アドレスRegStartAddrをセットする。開始ラインインデクスstart_line_indexに、１ページの開始位置を示すインデックスRegStartLineIndexをセットし、ページにおける処理済みライン数lines_cntに「０」を代入する。次にステップＳ６０２に進み、各ライン処理の初期化を行う。ここではアドレスaddrにライン開始アドレスline_start_addrをセットし、ライン数line_indexに、開始ラインインデクスstart_line_indexをセットする。またテーブルアドレスtable_addrに、曲がり情報を格納するテーブルの先頭アドレスRegTableStartAddrを代入する。
【０１１９】
次にステップＳ６０３に進み、バスインターフェース５０３'に対してテーブルの読み出し要求を行う。次にステップＳ６０４に進み、ステップＳ６０３で読み出したテーブルデータから、セグメント長seg_len及び上下ラインへの切り換え情報up_down_infoを設定する。次にステップＳ６０５に進み、ライン数line_indexをチェックする。ここで現ラインのライン数が０以上でＮ（印刷有効領域のライン数）以内の場合は印刷有効領域にあるためステップＳ６０６に進み、バスインターフェース５０３'に対して画像データの読み出し要求を、図１６のステップＳ５０９同様に行う。一方、ステップＳ６０５で、印刷有効領域内でないと判断するとステップＳ６０７に進み、セグメント長seg_len分のRegFillDataを後段の画像処理ブロックに対して転送する。これにより、印刷有効領域外では、特定のデータRegFillDataが出力されるため、後段の処理におけるマスク処理などが不要になる。
【０１２０】
こうしてステップＳ６０６或はＳ６０７を実行した後ステップＳ６０８に進み、アドレスポインタaddr_pointにセグメント長seg_lenを加算し、テーブルアドレスtable_addrを＋８する。そしてステップＳ６０９で、上下ラインへの切り換え情報up_down_infoをチェックする。ここで下ラインに切り換えるときはステップＳ６１０に進み、ライン数line_indexを－１する。そしてステップＳ６１１で、その結果が「０」以上であればステップＳ６１２に進み、アドレスaddrからライン間のオフセットアドレス値RegLineOffsetを差し引いてステップＳ６０３に進む。またステップＳ６１１で、ライン数ine_indexが「０」より小さければステップ６０３へ戻る。
【０１２１】
一方、ステップＳ６０９で、上ラインへの切り換えの場合はステップＳ６１３に進み、ライン数line_indexに＋１する。そしてステップＳ６１４に進み、その加算結果が印刷有効領域のライン数Ｎよりも小さければ有効領域内であるためステップＳ６１５に進む。ここでアドレスポインタaddr_pointにライン間のアドレスオフセット値RegLineOffsetを加算してステップＳ６０３に進む。またステップＳ６１４で、ライン数ine_indexがＮより大きければ有効領域外であるためステップＳ６０３に戻る。またステップＳ６０９で、上下ラインへの切り換えがない場合はステップＳ６１６に進み、現ライン数start_line_indexが有効領域内であるか、即ち、０以上全ライン数Ｎ以下であるかを判定する。そうであればステップＳ６１７に進み、現ライン数line_start_addrに、ライン間のオフセットア

ドレス値RegLineOffsetを加算してステップＳ６１８へ進む。またステップＳ６１６で、有効領域外であると判定するとステップ６１８に進み、現ライン数start_line_index及び処理済みライン数lines_cntをそれぞれ＋１する。そしてステップＳ６１９に進み、処理済みライン数linec_cntが１ページ分のライン数RegBeamLinesよりも大きいか否かで、１ページの処理が終了したか否かを判定する。こうして１ページの処理が終了していないときはステップＳ６０２に進むが、１ページの処理を終了したと判断すると、この処理を終了する。

【０１２２】

以上説明したように本実施の形態６によれば、上述の実施の形態１に係る効果に加えて、ＤＭＡコントローラが走査ラインの曲がりを相殺するように曲がった画像を読み出し、有効領域外では、指定したデータを生成できる。これにより、例えばプリンタエンジンがマスク機能を有していないような、走査ラインに曲がりが発生する小型のプリンタ部でも高品位な出力を得ることができる。

【０１２３】

［実施の形態７］

本発明の実施の形態７に係るＤＭＡコントローラ３０５ａの動作について、図２０、図２１、図２２、図２３、図２４、図２５を用いて説明する。

【０１２４】

図２０は、本発明の実施の形態７に係るプリンタコントローラ１０３ａの詳細構成を示すブロック図で、前述の図２のブロック図と共通する部分は同じ記号で示し、それらの説明を省略する。

【０１２５】

また図２１は、本発明の実施の形態７に係るプリンタコントローラ１０３ａのＤＭＡコントローラ３０５ａの詳細構成を示すブロック図で、前述の図３のブロック図と共通する部分は同じ記号で示し、それらの説明を省略する。尚、このＤＭＡコントローラ３０５ａは、アドレス生成部５０２ａがリングバッファカウンタ３２０のカウント値のインクリメントやデクリメントを更に指示する点だけが、前述の図２のＤＭＡコントローラ３０５と相違している。

【０１２６】

図２０では、前述した図２のプリンタコントローラ１０３に、更に、ＤＭＡコントローラ３２１，３２２及び画像処理部３２３、そしてリングバッファカウンタ３２０が追加されている。ここでＤＭＡコントローラ３２２は、画像メモリ３０４に格納されているビットマップ画像データを画像処理部３２３に読み出す。またＤＭＡコントローラ３２１は、画像処理部３２３で処理した画像データを画像メモリ３０４に書き込む。画像処理部３２３による処理は、具体的には回転処理や変倍処理等が含まれる。これらＤＭＡコントローラ３２１，３２２に対する制御は、ＣＰＵ３０８からの指示に基づき行われる。更にリングバッファカウンタ３２０が、ＤＭＡコントローラ３２１とＤＭＡコントローラ３０５ａとの間に接続されている。これによりＤＭＡコントローラ３０５ａ，３２０がともにリングバッファカウンタ３２０の制御とその状態を参照することにより、画像メモリ３０４との間でリングバッファを用いたデータ転送が可能となる。このリングバッファカウンタ３２０は、リングバッファに対して書き込むＤＭＡコントローラ３２１からインクリメント指示を受信すると内蔵カウンタ（不図示）をカウントアップする。またリングバッファからデータを読み出すＤＭＡコントローラ３０５ａからデクリメント指示を受信すると内蔵カウンタをカウントダウンする。ＤＭＡコントローラ３０５ａ，ＤＭＡコントローラ３２１はリングバッファカウンタ３２０のカウント値を基に動作することで、リングバッファを用いて画像メモリ３０４に１ページ分に満たない画像データを送出する。また、画像メモリ３０４の画像データのない領域を読み出すことなくＤＭＡコントローラ３２１からＤＭＡコントローラ３０５ａへの画像データ転送が可能となる。

【０１２７】

画像メモリ３０４からリングバッファを介して読み出されたビットマップデータは、ブ

レンド処理部３０６で処理され、エンジンＩ／Ｆ部３０７を介してビデオ信号として印刷部１０７に転送される。尚、ＤＭＡコントローラ３２１と本実施の形態７に係るＤＭＡコントローラ３０５ａとのリングバッファによるデータ転送の場合の構成について説明したが、ＣＰＵ３０８とのリングバッファによるデータ転送も可能である。

【０１２８】

図２２（Ａ）は、本発明の実施の形態７に係るＤＭＡコントローラ３０５ａに走査ラインの曲がりを指定する方法を説明するための図である。

【０１２９】

図において、ｎライン(n line)は、本来の像形成の対象領域である走査ラインを示す。ここでは走査ラインの曲がりに合わせて、１走査ライン（長さRegLineLen）を複数のセグメントに分割している。そして各セグメントの長さはSegLen[i]に設定されている。ここで変数ｉは、１走査ラインにおけるセグメントの順番に対応している。また各ライン切り換え位置で、走査ラインの曲がりが上方向か下方向かを示すUpDown[i]が設定されている。ここでも変数ｉは、１走査ラインにおけるセグメントの順番に対応しており、UpDown[i]＝０は上方向、UpDown[i]＝１は下方向への切り換えを示している。

【０１３０】

図２２（Ｂ）は、走査ラインの曲がりに合わせて、ライン切り換え位置までの長さ（セグメント長）SegLen[i]と、上又は下のラインに切り替えるかを示すUpDown[i]を記憶しているテーブルを説明する図である。このテーブルは、画像メモリ３０４かＲＡＭ３１０上にテーブルとして格納される。

【０１３１】

このテーブルの先頭アドレスRegTableStartAddrを、ＤＭＡコントローラ３０５ａのレジスタ部５０１にセットする。ここでUpDown[i]のとり得る値は、上或は下の２通りであるため、上のラインに切り替える場合は「０」を、下のラインに切り替える場合は「１」を割り当てている。

【０１３２】

またＤＭＡコントローラ３０５ａのレジスタ部５０１は、更にビットマップデータのライン長RegLineLenや、ＣＰＵ３０８が上記のレジスタの設定を完了した後に、ＤＭＡ動作を開始させるための起動レジスタ（不図示）を含む。

【０１３３】

図２３（Ａ）～（Ｃ）は、本実施の形態７に係るＤＭＡコントローラ３０５ａにリングバッファ領域を指定する方法を説明するための図である。

【０１３４】

図２３（Ａ）は、Ｍラインのリングバッファを説明する図である。

【０１３５】

レジスタ部５０１には、リングバッファ上のビットマップデータの隣り合うラインのアドレスのオフセット値RegLineOffsetがセットされる。更に、リングバッファの下限値RegRingBufferLowerAddr、リングバッファの上限値RegRingBufferUpperAddrがセットされる。更にレジスタ部５０１には、画像メモリ３０４に確保したリングバッファ領域の先頭アドレスRegStartAddr、リングバッファ領域のライン数RegLinesがセットされる。

【０１３６】

図２３（Ｂ）は、走査ラインの曲がりを説明する図である。

【０１３７】

ここでは読出ラインの行頭から、上方向に切り替わる最大ライン数RegUpMaxと、読み出しラインの行頭から下方向に切り替わる最大ライン数RegDownMaxが規定されている。これらの値はレジスタ部５０１にセットされる。図２３（Ｂ）の例では、RegUpMax＝２，RegDownMax＝３である。

【０１３８】

図２３（Ｃ）は、図２３（Ｂ）に示すような走査ラインの曲がりが発生している場合、走査ラインが印刷有効領域内にあるか否かの判定を説明する図である。

【０１３９】
ここでレジスタ部５０１は、リングバッファの先頭アドレスのある行に対して読み出し開始位置を示すRegStartLineIndexがセットされる。更に、ブレンド処理部３０６へ送るべきライン数RegBeamLines、印刷有効領域外を出力する際に使用するダミーデータRegFillDataもセットされる。図２３（Ｂ）では、RegUpMaxが「２」で、RegDownMaxが「３」であるため、読み出し開始ラインRegStartLineIndexは、印刷有効領域の先頭ラインから－３ラインとなる。また読み出し終了ラインは、印刷有効領域の最終ラインから＋２ラインとなる。
【０１４０】
図２４及び図２５は、本実施の形態７に係るＤＭＡコントローラ３０５ａのアドレス生成部５０２ａの動作を示すフローチャートである。
【０１４１】
この処理は、ＤＭＡコントローラ３２１により画像メモリ３０４のリングバッファに、画像処理部３２３で処理されたビットマップデータが格納されて、リングバッファカウンタ３２０がカウントアップされるのと並行して実行される。
【０１４２】
まずステップＳ７０１で、ＣＰＵ３０８の指示によりＤＭＡ動作が開始されると、各ラインの開始アドレスline_start_addrを、リングバッファ領域の先頭アドレスRegStartAddrに初期化する。またブレンド処理部３０６へ送った処理済みのライン数line_cntを「０」で初期化する。更に、画像データの読み出し開始ラインstart_line_indexを、読み出し開始ラインRegStartLineIndex（図２３の例では「－３」）で初期化する。次にステップＳ７０２に進み、ライン毎の初期化を行う。ここでは読み出し開始アドレスポインタaddr_pointを、ライン開始アドレスline_start_addrで初期化する。またライン中の処理済みのデータ量line_data_cntを「０」で初期化し、また現在の処理ライン位置line_indexsを、画像データの読み出し開始位置start_line_indexで初期化する。また、参照すべきセグメントの長さSegLen[i]と、ライン切り替え情報UpDown[i]のテーブルのアドレスを示すtable_addrを、テーブルの先頭アドレスRegTableStartAddで初期化する。
【０１４３】
次にステップＳ７０３に進み、画像データの読み出し開始ラインstart_line_indexが、図２３（Ｃ）に示す印刷有効領域に対してどのラインであるかを判定する。ステップＳ７０３で、有効領域の上側、例えば図２３（Ｃ）のライン－３から２まではステップＳ７０４に進み、リングバッファに図２３（Ｃ）のライン－３から２までのデータが格納されるのを待つ。具体的には、リングバッファカウンタＲＢＣの値と、ＢＨ（＝RegUpMax+RegDownMax+1)＋１（この実施の形態７ではＢＨ＝６）と（start_line_index－１）との加算値とを比較する。こうしてリングバッファにライン－３からライン２までのデータが格納されるとステップＳ７０５に進み、１ライン処理を実行してステップＳ７１１に進む。この１ライン処理は図２５を参照して後述する。ここで、ステップＳ７０４～Ｓ７０５の処理は、図２３（Ｃ）のエリアＡで示すラインのデータ処理に該当している。
【０１４４】
またステップＳ７０３で、画像データの読み出し開始ラインstart_line_indexが図２３（Ｃ）に示す印刷有効領域内であればステップＳ７０６に進む。ここでリングバッファに、走査ラインの曲がりを考慮した６（ＢＨ）ライン分のイメージデータが格納されるのを待ってステップＳ７０７に進む。ステップＳ７０７では後述の１ライン処理を実行してステップＳ７０８に進み、リングバッファカウンタＲＢＣを－１してステップＳ７１１に進む。ここで、ステップＳ７０６～Ｓ７０７の処理は、図２３（Ｃ）のエリアＢで示すように、そのラインデータが全て印刷有効領域に含まれるラインのデータ処理に該当している。
【０１４５】
またステップＳ７０３で、画像データの読み出し開始ラインstart_line_indexが図２３（Ｃ）に示す印刷有効領域の下側でラインＮ－３からラインＮ＋１までの間ではステップ

Ｓ７０９に進む。ここで後述の１ライン処理を実行してステップＳ７１０に進む。ステップＳ７１０ではリングバッファカウンタＲＢＣを－１してステップＳ７１１に進む。ここで、ステップＳ７０９の処理は、図２３（Ｃ）のエリアＣで示すラインのデータ処理に該当している。

【０１４６】

次に図２５のフローチャートを参照して、ステップＳ７０５，Ｓ７０７，Ｓ７０９の１ラインの処理を説明する。

【０１４７】

図２５は、本実施の形態７に係る１ライン処理を説明するフローチャートである。

【０１４８】

ステップＳ８０１では、テーブルアドレスをもとに、そのライン（start_line_indexで示されるライン）の曲がり情報を取得する。次にステップＳ８０に進み、そのラインのセグメント長seg_lenと切り換え情報up_down_infoをレジスタにセットする。次にステップＳ８０３に進み、読み出し開始ラインline_indexが印刷有効領域内であるかを判定する。ステップＳ８０３で印刷有効領域内であればステップＳ８０４に進み、そのラインの該当するセグメントのデータを読み出す。ステップＳ８０３で印刷有効領域内でないときはステップＳ８０５に進み、そのラインの該当するセグメント長分のダミーデータRegFillDataを出力する。こうしてステップＳ８０４或はステップＳ８０５を実行した後ステップＳ８０６に進み、イメージデータの読み出しアドレスポインタaddr_point及びテーブルアドレスtable_addrを更新する。ここでは読み出しアドレスに、処理済みのセグメント長を加算したアドレスに更新し、テーブルアドレスは次のセグメントの曲がり情報を得るために＋８される。

【０１４９】

次にステップＳ８０７に進み、上下切り換え情報up_down_infoを調べる。ここで切り換え情報が「０」、即ち上下方向への切り換えでないときは、ラインを切り換えてラインデータを読み出す必要が無いため処理を終了する。また曲がりが発生しているラインであっても、最終のセグメントの後では切り換え情報up_down_infoが「０」であるため、処理を終了する。

【０１５０】

一方ステップＳ８０７で、上方向の曲がりがあると判定するとステップＳ８０８に進み、読み出しライン番号line_indexを－１し、ステップＳ８０９で、その結果が負の値になったかをみる。負の値でなければステップＳ８１０に進み、読み出しアドレスポインタaddrから、リングバッファの１ラインのオフセットアドレスRegLineOffsetを差し引いて、一つ前のラインのアドレスに更新する。そしてステップＳ８１１で、そのアドレスがリングバッファの先頭アドレスRegRingBufferLowerAddr以下でないか、即ち、リングバッファ内のアドレスかどうかを判定する。リングバッファ内のアドレスであればステップＳ８０１に戻って前述の処理を実行する。しかしそうでないときはステップＳ８１２に進み、その読み出しアドレスポインタaddr_pointにリングバッファのメモリエリア(RegRingBufferUpperAddr－RegRingBufferLowerAddr)を加算する。そして読み出しアドレスポインタをリングバッファ内のアドレスに更新する。そしてステップＳ８０１に進む。

【０１５１】

一方ステップＳ８０７で、下方向の曲がりがあると判定するとステップＳ８１３に進み、読み出しライン番号line_indexを＋１し、ステップＳ８１４で、その結果がリングバッファのライン数Ｎを越えたかをみる。越えなければステップＳ８１５に進み、読み出しアドレスポインタaddrに、リングバッファの１ラインのオフセットアドレスRegLineOffsetを加算して次ラインのアドレスに更新する。そしてステップＳ８１６で、そのアドレスがリングバッファの最終アドレスRegRingBufferUpperAddr以上でないか、即ち、リングバッファ内のアドレスかどうかを判定する。リングバッファ内のアドレスであればステップＳ８０１に戻って前述の処理を実行するが、そうでないときはステップＳ８１７に進む。ここで、その読み出しアドレスポインタaddrから、リングバッファのメモリエリア(RegRing

BufferUpperAddr－RegRingBufferLowerAddr)を差し引く。こうして読み出しアドレスをリングバッファ内のアドレスに更新する。そしてステップＳ８０１に進む。尚、ステップＳ８０９，Ｓ８１４で、計算結果が負或はＮ以上の時はステップＳ８０１に戻る。

【０１５２】

このようにこの１ライン処理では、リングバッファに格納されている印刷有効領域に含まれるラインのセグメント単位でイメージデータを読み出すことができる。また印刷領域に入っていないセグメントでは、ダミーデータRegFillDataが読み出される。

【０１５３】

次に図２４に戻って、ステップＳ７０５，Ｓ７０８，Ｓ７１０のいずれかを実行した後ステップＳ７１１に進み、読み出しラインが印刷有効領域にあるか否かを判定する。印刷有効領域にあればステップＳ７１２に進み、読み出しラインアドレスにリングバッファの１ラインのオフセットアドレス(RegLineOffset)を加算して次ラインのアドレスに更新する。そしてステップＳ７１３で、その加算結果のアドレスポインタがリングバッファの最終アドレスRegRingBufferUpperAddr以上でないか、即ち、リングバッファ内のアドレスかどうかを判定する。ここでリングバッファ内のアドレスであればステップＳ７１５に進むが、そうでないときはステップＳ７１４で、その読み出しアドレスポインタをリングバッファの先頭アドレスに初期化してステップＳ７１５に進む。

【０１５４】

ステップＳ７１１で、読み出しラインが印刷有効領域でないときはステップＳ７１５に進む。ステップＳ７１５では、読み出し回路ラインstart_line_indexを＋１し、処理済みライン数line_cntを＋１する。そしてステップＳ７１６で、その処理済みのライン数が、１ページのライン数RegBeamLinesになったか、即ち、１ページ分のイメージデータの処理が終了したかをみる。終了していれば、このＤＭＡ動作を終了するが、そうでないときはステップＳ７０２に進み、次の走査ラインに対する処理を実行する。ここでＤＭＡコントローラ３０５ａは、ＤＭＡ動作を終了した時に、割り込み（不図示）をＣＰＵ３０８に送る。ＣＰＵ３０８はこの割り込みを検出することで、このＤＭＡを終了したことを検知する。

【０１５５】

以上説明したように本実施の形態７によれば、ＤＭＡＣがリングバッファを使用して、走査ラインの曲がりを補正するようにライン間でのりかえを行いながら曲がった画像データを読み出すことができる。これにより、画像データの１ページ分よりも小さな容量のバッファを用いて、レーザビームの曲がりを補正できるという効果がある。

【図面の簡単な説明】

【０１５６】

【図１】本発明の実施の形態に係るカラー画像形成装置（多機能装置）の構成を説明するブロック図である。

【図２】本発明の実施の形態に係るプリンタコントローラの詳細構成を示すブロック図である。

【図３】本発明の実施の形態に係るプリンタコントローラのＤＭＡコントローラの詳細構成を示すブロック図である。

【図４】本発明の実施形態に係るレジストレーションずれとその補正を説明するための図である。

【図５】本発明の実施形態に係るレジストレーションずれとその補正を説明するための図である。

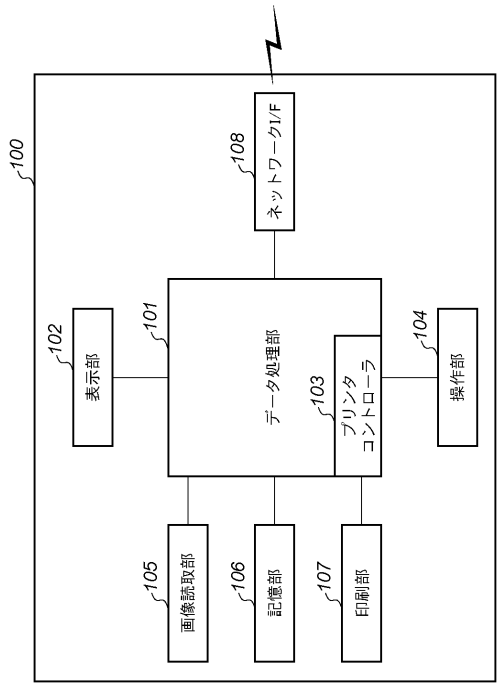
【図６】本発明の実施の形態１に係るＤＭＡコントローラ３０５に走査ラインの曲がりとバースト転送を説明するための図である。

【図７】本発明の実施の形態１に係るＤＭＡコントローラのアドレス生成部の動作を示すフローチャートである。

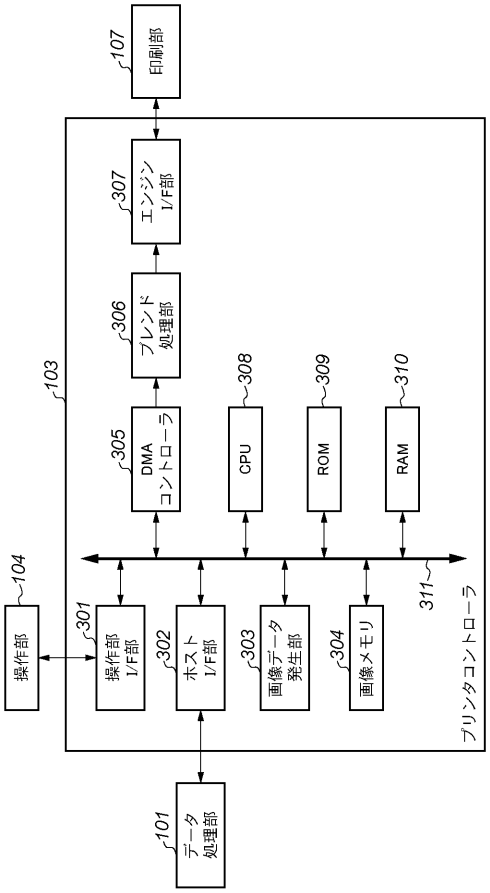
【図８】本発明の実施の形態２に係るＤＭＡコントローラに走査ラインの曲がりを指定する方法を説明するための図である。

【図９】本発明の実施の形態２に係るＤＭＡコントローラのアドレス生成部の動作を示すフローチャートである。
【図１０】本発明の実施の形態３に係るＤＭＡコントローラに対して走査ラインの曲がりを指定する方法を説明するための図（Ａ）、及び情報SegPosi[i]をテーブルとして格納した状態を示す図（Ｂ）である。
【図１１】本発明の実施の形態３に係るＤＭＡコントローラの構成を示すブロック図である。
【図１２】本発明の実施の形態３に係るＤＭＡコントローラのアドレス生成部の動作を示すフローチャートである。
【図１３】本発明の実施の形態４に係るＤＭＡコントローラに走査ラインの曲がりを指定する方法を説明するための図である。
【図１４】本発明の実施の形態４に係るＤＭＡコントローラのアドレス生成部の動作を示すフローチャートである。
【図１５】本発明の実施の形態５に係るＤＭＡコントローラに有効画像領域のメモリアドレスを指定する方法を説明するための図である。
【図１６】本発明の実施の形態５に係るＤＭＡコントローラのアドレス生成部の動作を示すフローチャートである。
【図１７】図１６のステップＳ５１３における状態state及びアドレスの更新処理の詳細を示すフローチャートである。
【図１８】本発明の実施の形態６に係るＤＭＡコントローラに有効画像領域を、メモリ上のアドレスで指定する方法を説明するための図である。
【図１９】本発明の実施の形態６に係るＤＭＡコントローラのアドレス生成部の動作を示すフローチャートである。
【図２０】本発明の実施の形態７に係るプリンタコントローラの詳細構成を示すブロック図である。
【図２１】本発明の実施の形態７に係るプリンタコントローラのＤＭＡコントローラの詳細構成を示すブロック図である。
【図２２】本発明の実施形態７に係るＤＭＡコントローラでの走査ラインの曲がりの指定を説明するための図である。
【図２３】本実施の形態７に係るＤＭＡコントローラにリングバッファ領域を指定する方法を説明するための図である。
【図２４】本実施の形態７に係るＤＭＡコントローラのアドレス生成部の動作を示すフローチャートである。
【図２５】図２４のフローチャートの１ライン処理動作を説明するフローチャートである。

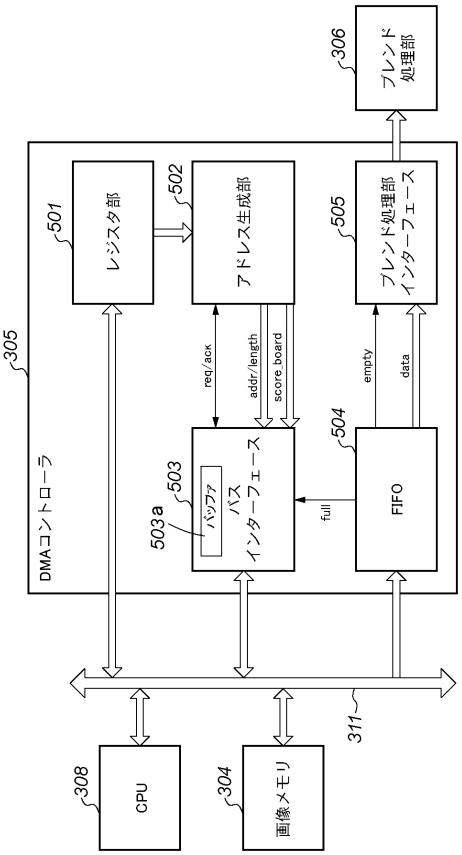
【図１】


100
108
ネットワークI/F
102
表示部
101
データ処理部
103
プリンタコントローラ
104
操作部
105
画像読取部
106
記憶部
107
印刷部

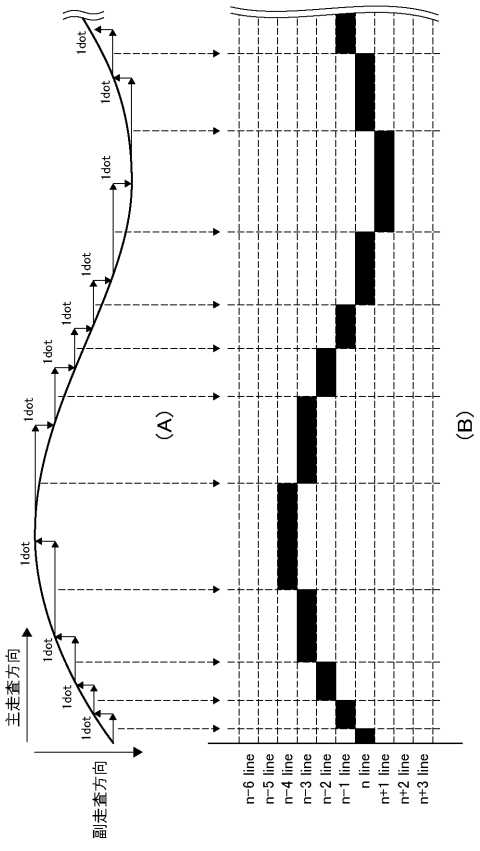
【図２】


107
印刷部
307
エンジンI/F部
306
プレンド処理部
103
305
DMAコントローラ
308
CPU
309
ROM
310
RAM
104
操作部
301
操作部I/F部
302
ホストI/F部
303
画像データ発生部
304
画像メモリ
311
プリンタコントローラ
101
データ処理部

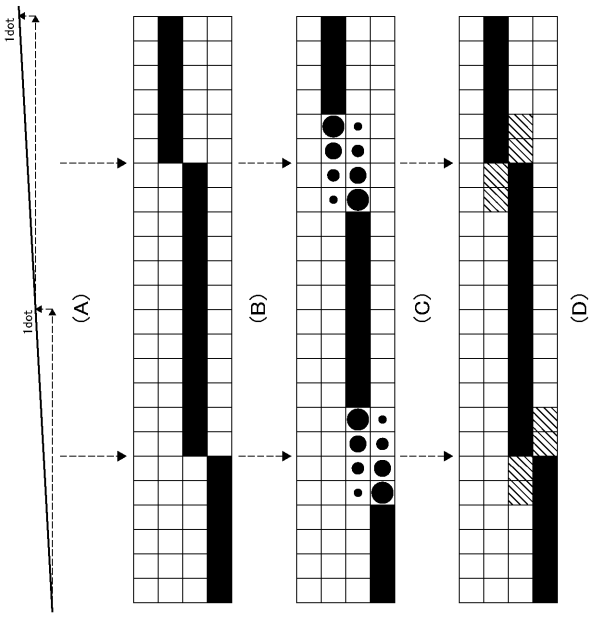
【図３】


306
プレンド処理部
305
DMAコントローラ
501
レジスタ部
502
アドレス生成部
505
プレンド処理部インターフェース
req/ack
addr/length
score board
empty
data
503
503a
バッファ
バスインターフェース
504
FIFO
full
311
308
CPU
304
画像メモリ

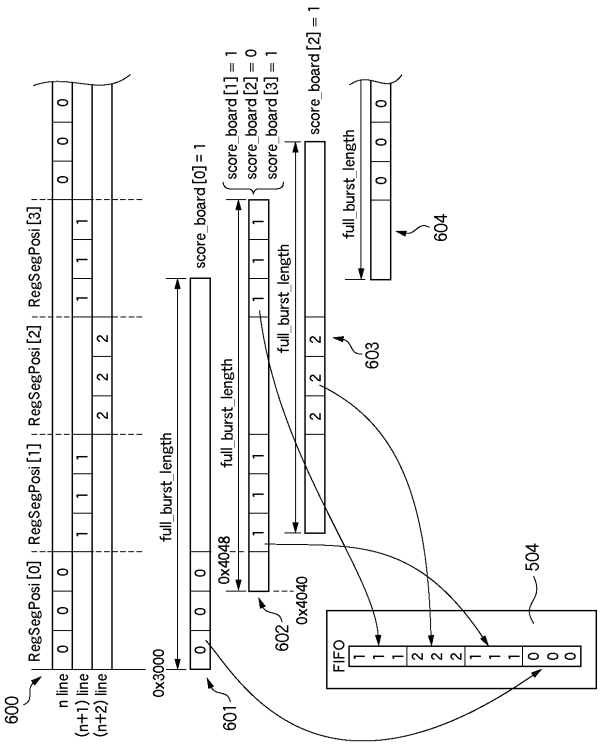
【図４】


1dot
(A)
(B)
主走査方向
副走査方向
n-6 line
n-5 line
n-4 line
n-3 line
n-2 line
n-1 line
n line
n+1 line
n+2 line
n+3 line

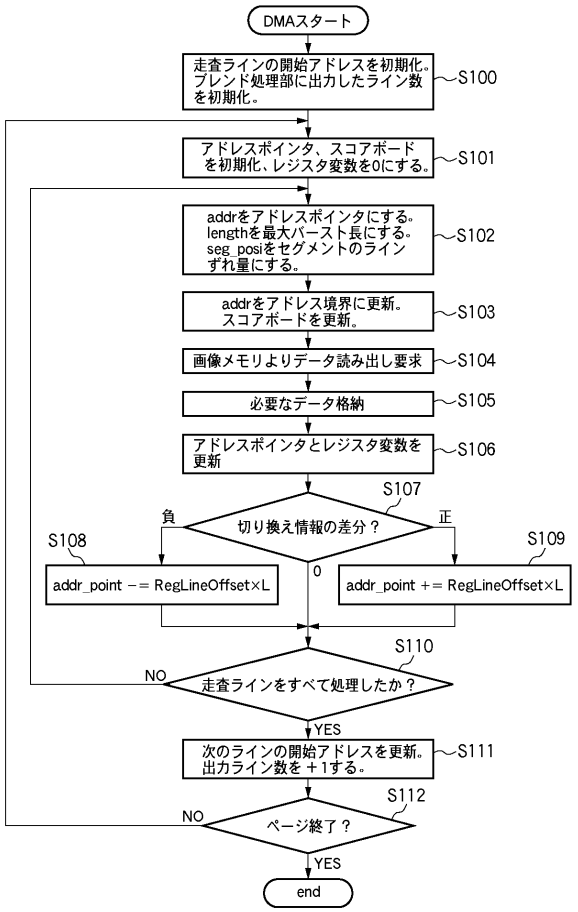
【図５】


1dot
1dot
(A)
(B)
(C)
(D)

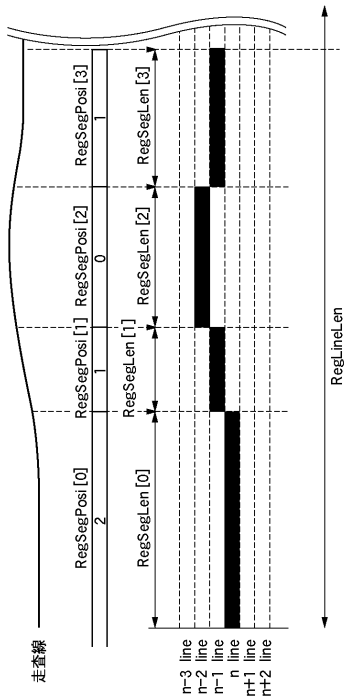
【図６】


RegSegPosi [0]
RegSegPosi [1]
RegSegPosi [2]
RegSegPosi [3]
n line
(n+1) line
(n+2) line
600
0x3000
full_burst_length
score_board [0] = 1
601
0x4048
0x4040
602
score_board [1] = 1
score_board [2] = 0
score_board [3] = 1
full_burst_length
603
score_board [2] = 1
full_burst_length
604
FIFO
504

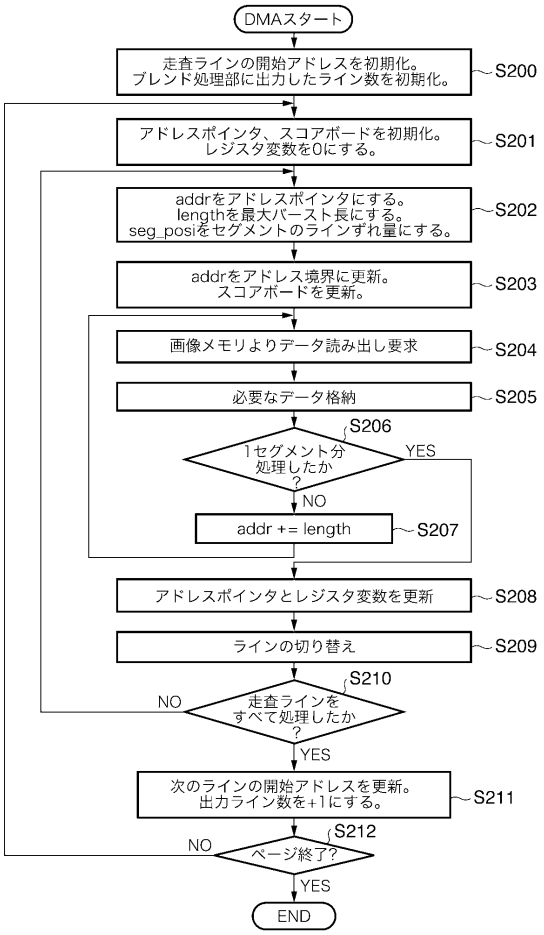
【図７】


DMAスタート
走査ラインの開始アドレスを初期化。ブレンド処理部に出力したライン数を初期化。
S100
アドレスポインタ、スコアボードを初期化、レジスタ変数を0にする。
S101
addrをアドレスポインタにする。lengthを最大バースト長にする。seg_posiをセグメントのラインずれ量にする。
S102
addrをアドレス境界に更新。スコアボードを更新。
S103
画像メモリよりデータ読み出し要求
S104
必要なデータ格納
S105
アドレスポインタとレジスタ変数を更新
S106
S107
切り換え情報の差分？
負
正
0
S108
addr_point −= RegLineOffset×L
S109
addr_point += RegLineOffset×L
S110
NO
走査ラインをすべて処理したか？
YES
次のラインの開始アドレスを更新。出力ライン数を＋1する。
S111
S112
NO
ページ終了？
YES
end

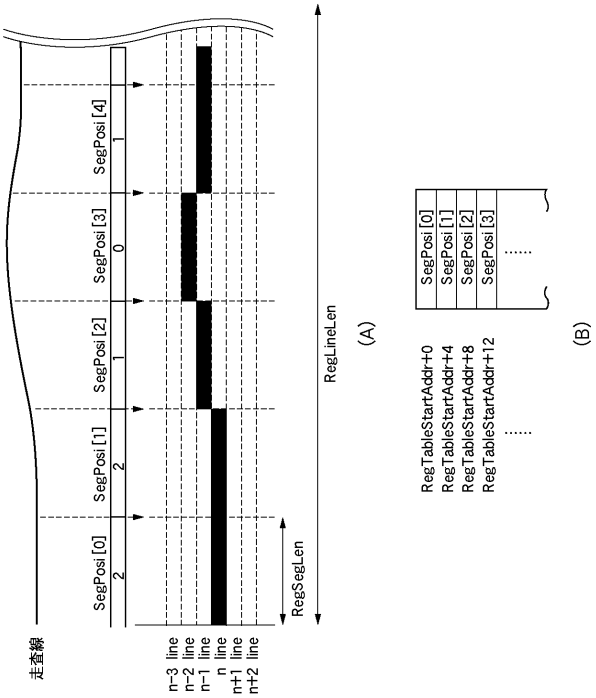
【図８】


RegSegPosi [3]
RegSegPosi [2]
RegSegPosi [1]
RegSegPosi [0]
RegSegLen [3]
RegSegLen [2]
RegSegLen [1]
RegSegLen [0]
RegLineLen
走査線
n−3 line
n−2 line
n−1 line
n line
n+1 line
n+2 line

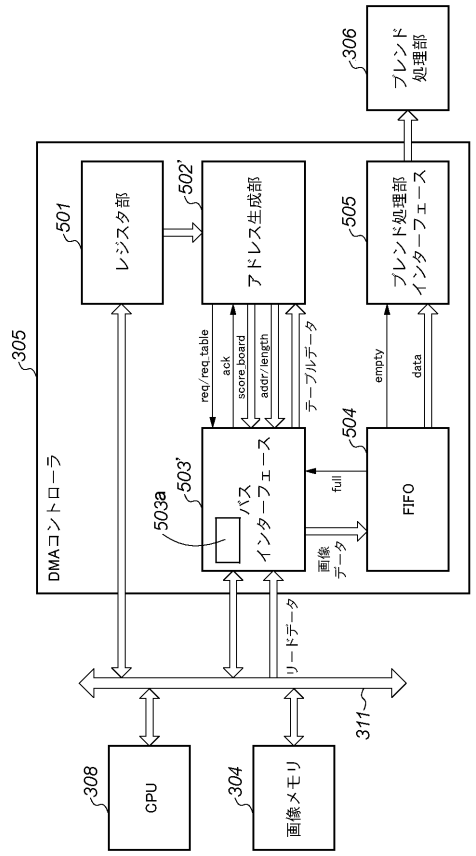
【図９】



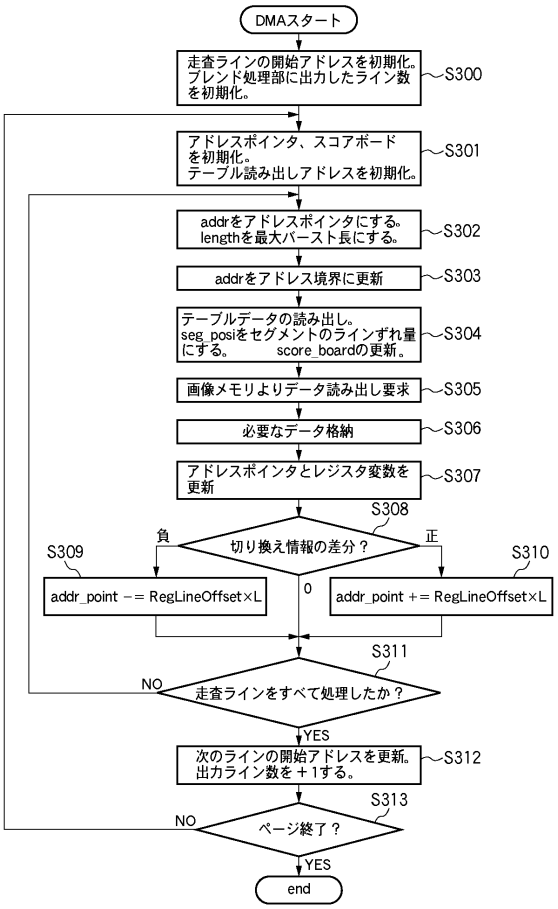
【図１０】



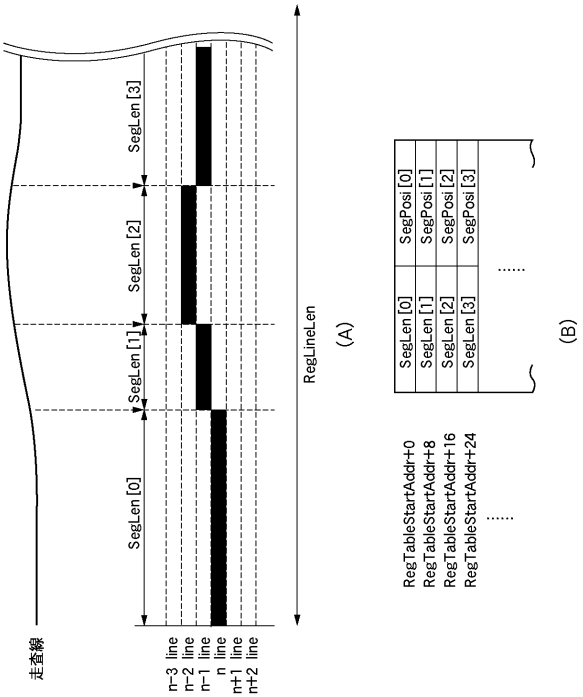
【図１１】



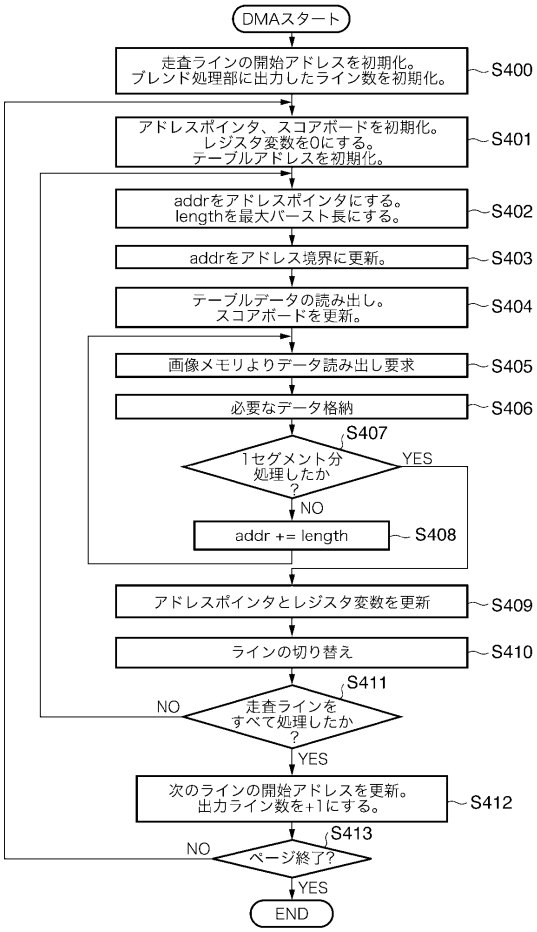
【図１２】



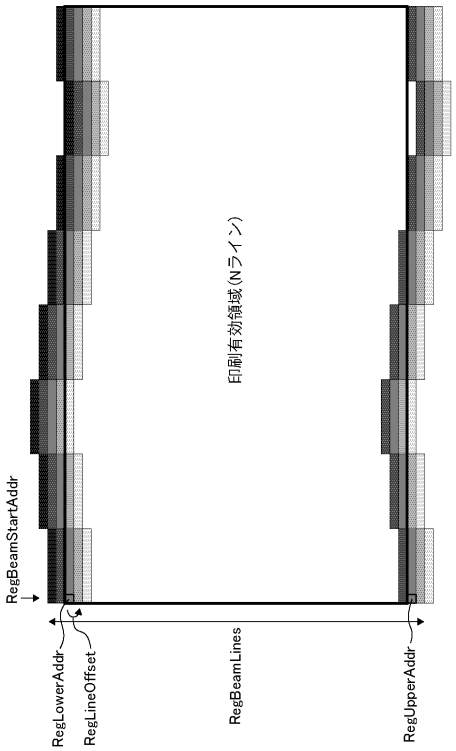
【図１３】



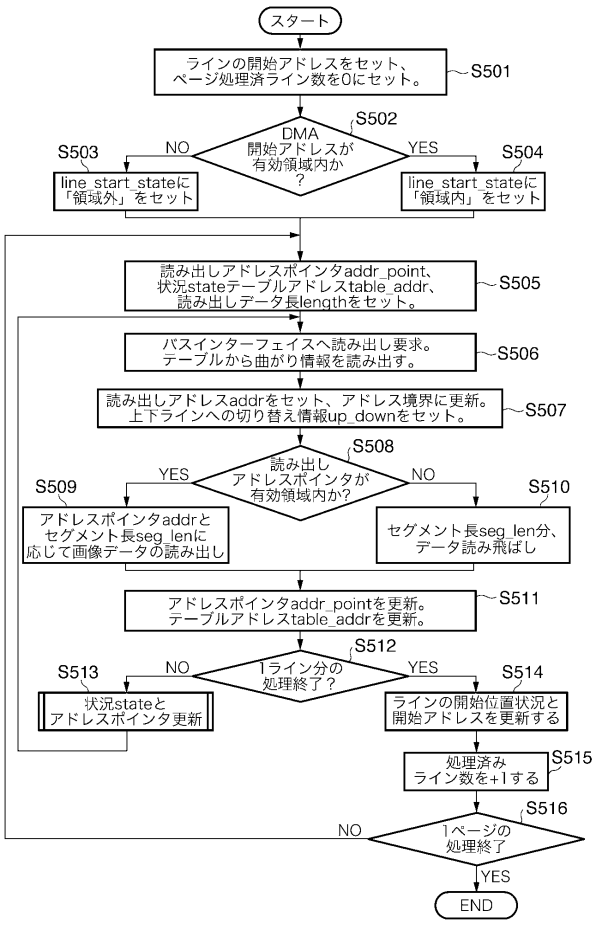
【図１４】



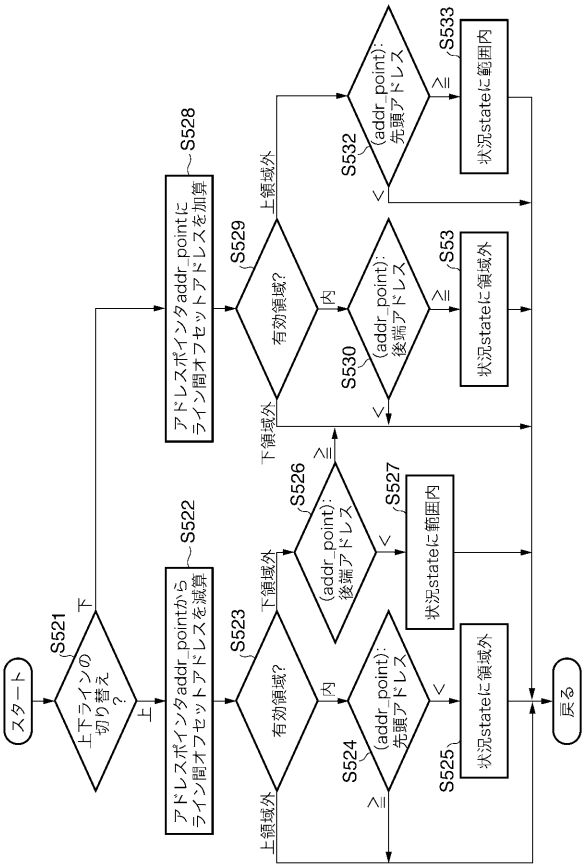
【図１５】



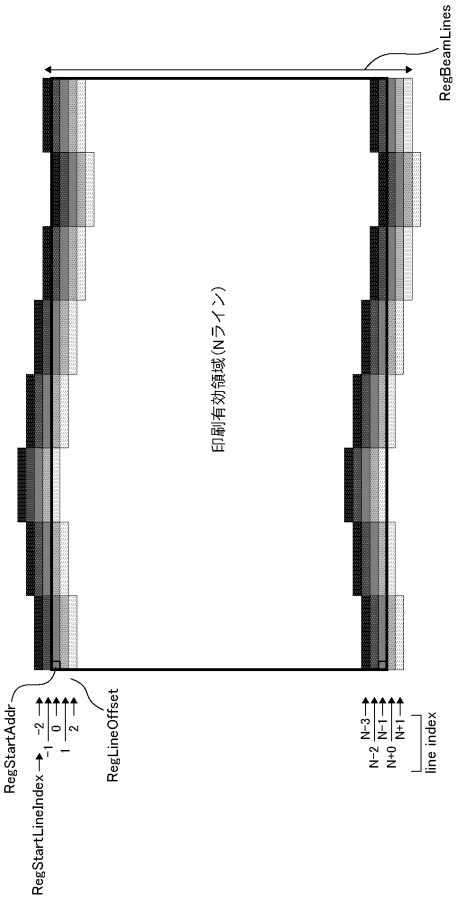
【図１６】



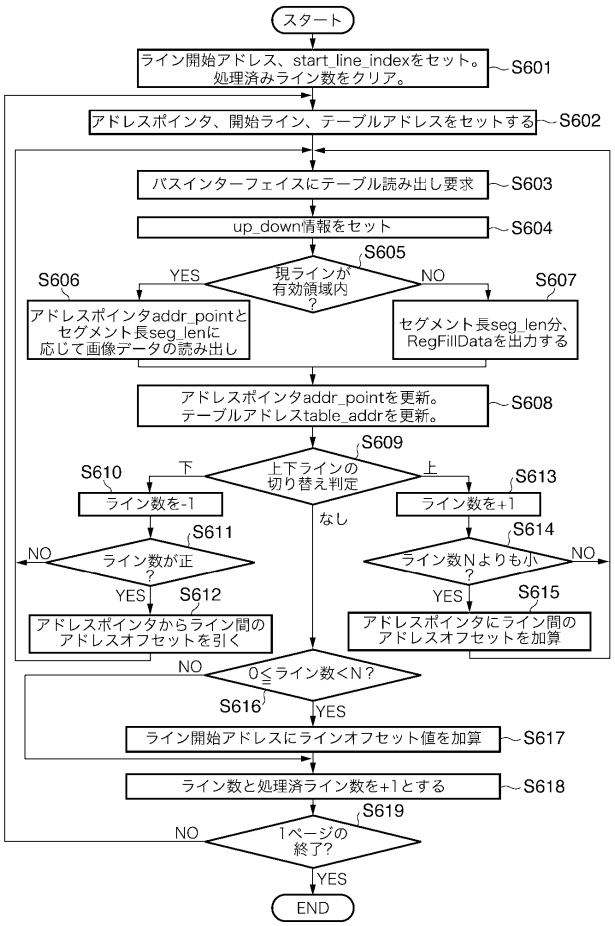
【図１７】



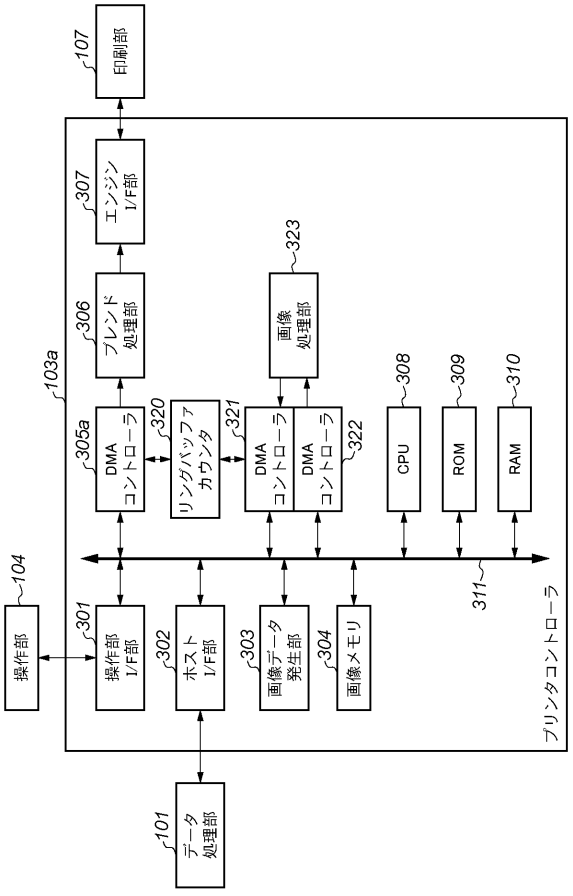
【図１８】



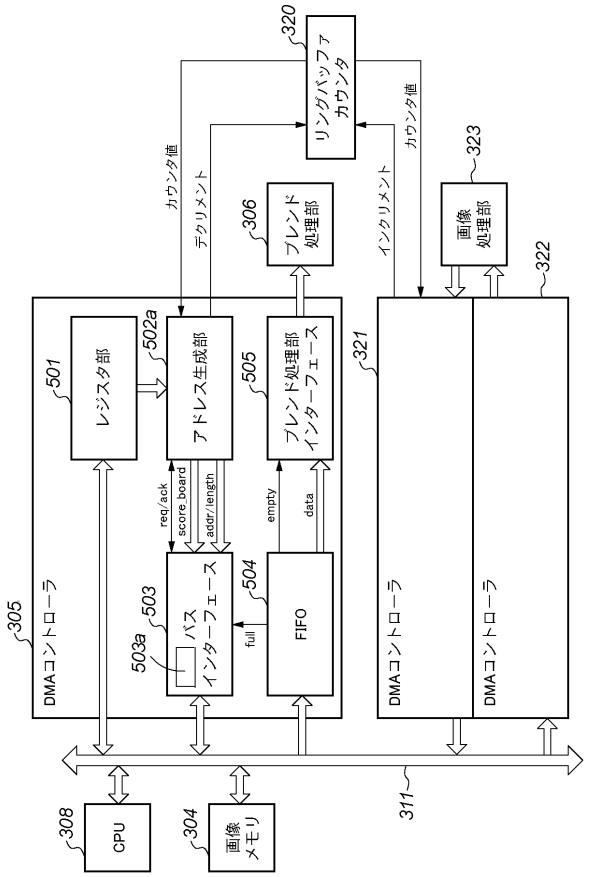
【図１９】



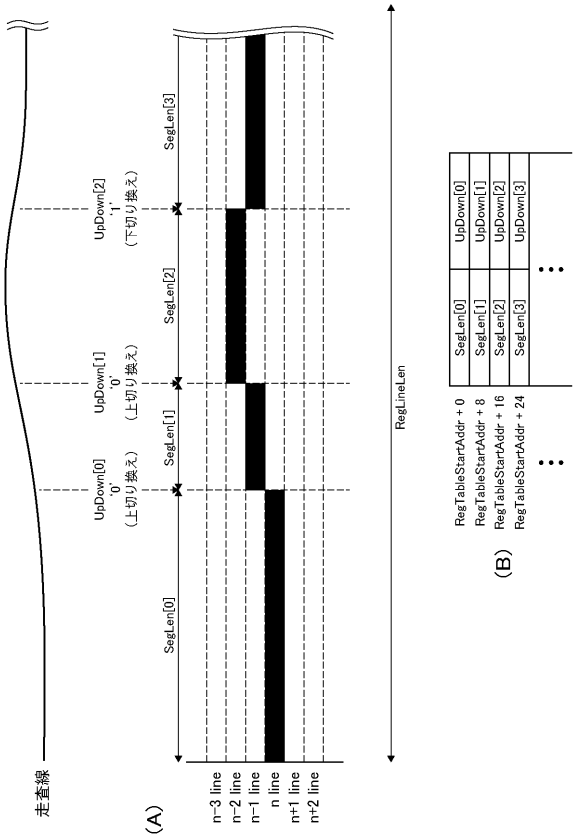
【図２０】



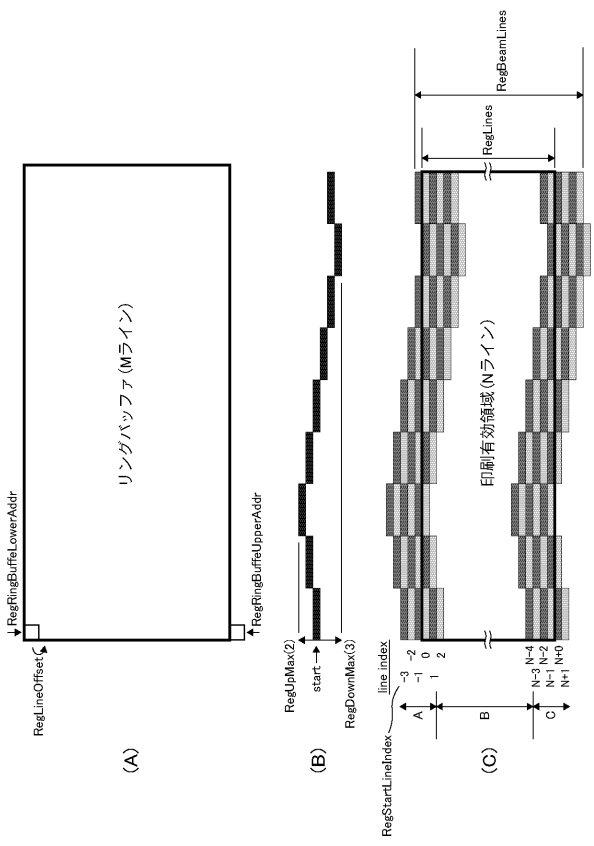
【図２１】



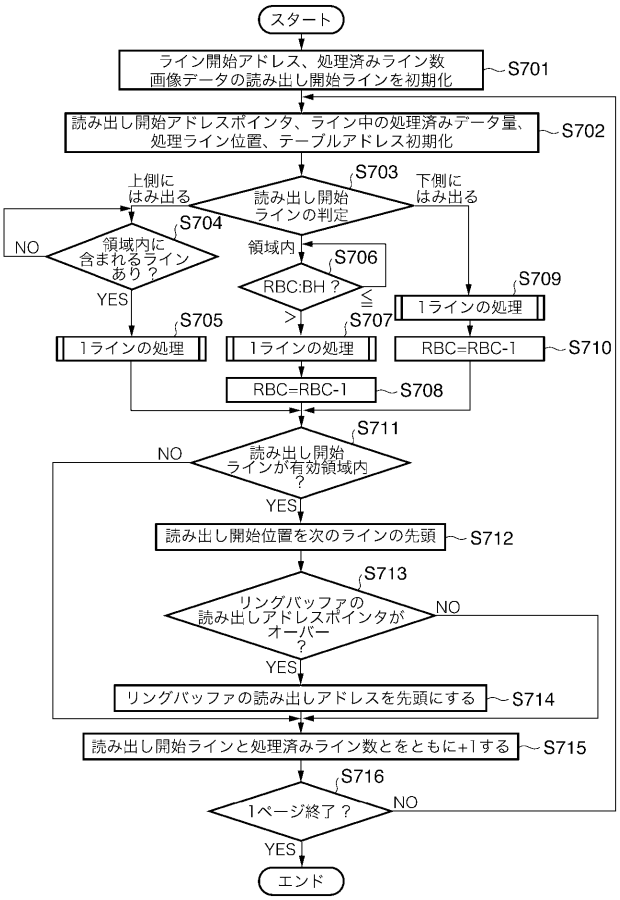
【図２２】



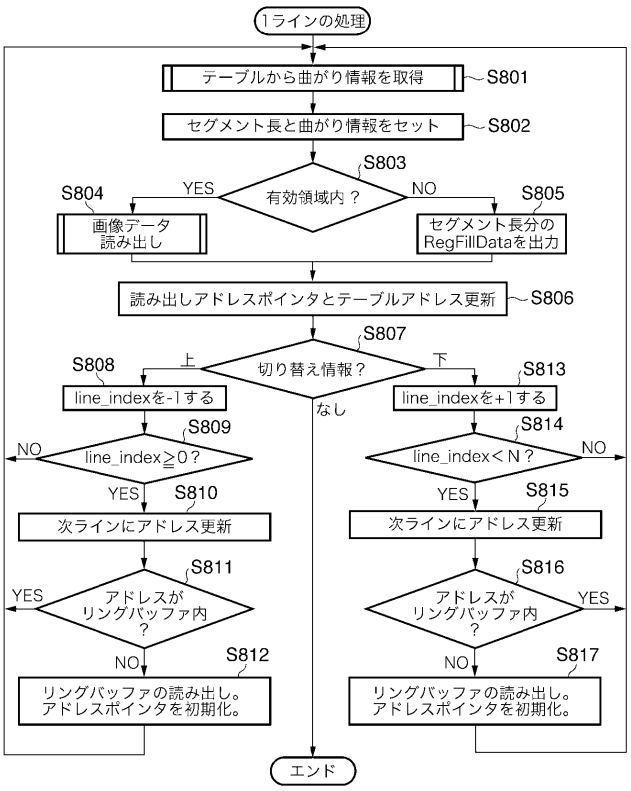
【図２３】



【図２４】



【図２５】


1ラインの処理
テーブルから曲がり情報を取得
S801
セグメント長と曲がり情報をセット
S802
S803
有効領域内？
YES
NO
S804
画像データ
読み出し
S805
セグメント長分の
RegFillDataを出力
読み出しアドレスポインタとテーブルアドレス更新
S806
S807
切り替え情報？
上
下
なし
S808
line_indexを-1する
S813
line_indexを+1する
S809
line_index≧0？
NO
YES
S810
次ラインにアドレス更新
S814
line_index＜N？
NO
YES
S815
次ラインにアドレス更新
S811
アドレスが
リングバッファ内
？
YES
NO
S812
リングバッファの読み出し。
アドレスポインタを初期化。
S816
アドレスが
リングバッファ内
？
YES
NO
S817
リングバッファの読み出し。
アドレスポインタを初期化。
エンド

フロントページの続き

(72)発明者　森田　靖二郎
東京都大田区下丸子３丁目３０番２号　キヤノン株式会社内

Ｆターム(参考) 2C187 AC07 AE07 BF14 BF44 BG03 FB06 FC03 FC08 FD01
2C362 BB05 BB15 CB75 CB77 CB78
2H027 DA23 DB01 DE07 DE09 EC06 ED04 ED06 EE07 EE08 HB05
HB06 HB16 HB17 ZA07
5C076 AA32 AA40 BA02 BA03 BA04 BA08