

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第5393168号  
(P5393168)

(45) 発行日 平成26年1月22日 (2014. 1. 22)

(24) 登録日 平成25年10月25日 (2013. 10. 25)

(51) Int. Cl.	F I
<b>B 4 1 J 5/30 (2006. 01)</b>	B 4 1 J 5/30 Z
<b>H O 4 N 1/00 (2006. 01)</b>	H O 4 N 1/00 C
<b>B 4 1 J 29/38 (2006. 01)</b>	B 4 1 J 29/38 Z

請求項の数 7 (全 21 頁)

(21) 出願番号	特願2009-6113 (P2009-6113)	(73) 特許権者	000001007
(22) 出願日	平成21年1月14日 (2009. 1. 14)		キヤノン株式会社
(65) 公開番号	特開2010-162745 (P2010-162745A)		東京都大田区下丸子3丁目30番2号
(43) 公開日	平成22年7月29日 (2010. 7. 29)	(74) 代理人	100076428
審査請求日	平成24年1月16日 (2012. 1. 16)		弁理士 大塚 康德
		(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】 画像形成装置及びその制御方法

(57) 【特許請求の範囲】

【請求項 1】

複数の処理ユニットを有する画像形成装置であって、

ブロック化された画像形成処理の対象となる対象データにおいて、ブロックの属性および属性毎の処理負荷パラメータおよび前記処理ユニットの数に基づいて、各処理ユニットにおいて処理対象の属性を決定する決定手段と、

前記決定手段によって決定された前記各処理ユニットにおいて処理対象の属性に応じた処理プログラムを、前記各処理ユニットがそれぞれ持つローカルメモリへロードするロード手段と、

前記ブロック化されたデータに対して、前記ローカルメモリにロードされた処理プログラムを実行して処理するよう制御する手段と、  
を備えることを特徴とする画像形成装置。

【請求項 2】

前記ロード手段は、前記属性に応じた処理プログラムに加えて、前記対象データを記述したページ記述言語の種類に応じた処理プログラムをロードすることを特徴とする請求項 1 に記載の画像形成装置。

【請求項 3】

前記属性には、テキストとグラフィックとテキスト及びグラフィックの混合とが含まれることを特徴とする請求項 1 又は 2 に記載の画像形成装置。

【請求項 4】

10

20

前記処理プログラムは、対象データをレンダリングするレンダラモジュールまたは対象データを画像処理する画像処理モジュール、またはレンダラモジュールと画像処理モジュールとを含むことを特徴とする請求項 1 乃至 3 のいずれか 1 項に記載の画像形成装置。

【請求項 5】

さらに、前記属性に応じて前記処理ユニットにロードされた前記処理プログラムを実行することにより処理されたブロックの処理済みデータを結合する結合手段と、

前記結合手段により結合された前記処理済みデータに基づいて、印刷媒体に画像を形成する画像形成手段とを備えることを特徴とする請求項 1 乃至 4 のいずれか 1 項に記載の画像形成装置。

【請求項 6】

複数の処理ユニットを有する画像形成装置の制御方法であって、

前記画像形成装置の重み付け手段が、ブロック化された画像形成処理の対象となる対象データにおいて、ブロックの属性および属性毎の処理負荷パラメータおよび前記処理ユニットの数に基づいて、各処理ユニットにおいて処理対象の属性を決定する決定工程と、

前記画像形成装置のロード手段が、前記決定工程によって決定された前記各処理ユニットにおいて処理対象の属性に応じた処理プログラムを、前記各処理ユニットがそれぞれ持つローカルメモリへロードするロード工程と、

前記画像形成装置の制御手段が、前記ブロック化されたデータに対して、前記ローカルメモリにロードされた前記処理プログラムを実行して処理するよう制御する工程と、を有することを特徴とする画像形成装置の制御方法。

【請求項 7】

請求項 1 乃至 5 のいずれか一項に記載の画像形成装置の各手段としてコンピュータを機能させるためのプログラム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、画像形成装置及びその制御方法に関し、特に、画像形成のための画像処理ならびにレンダリングに関する。

【背景技術】

【0002】

従来の画像形成装置において、レンダリングや画像処理（両者をまとめて画像処理と呼ぶ）を短時間でを行うために、CPUやASIC等専用のハードウェア（描画処理部と呼ぶ）を用意し、ホストコンピュータから受信したPDLデータを描画処理部により処理するものがあった（図2参照）。

【0003】

さらに、描画処理部を複数用意して、画像処理を並列に実行するものも提案されている。その際には、スキャンラインごとの処理負荷の推定値を得て、その推定値に基づいて複数の描画処理部へ画像処理が割り当てられる（たとえば特許文献1等参照）。これによって各描画処理部による処理負荷をならしている。

【0004】

この場合の処理の分散を効率的に行うために、処理の種類毎に処理負荷を推定できる情報を用意し、この情報に基づき複数ある処理装置への処理の割り当て方法を決定する提案もされている。この処理によって、複数ある処理装置の処理時間をおおむね平準化することが出来る（たとえば特許文献2等参照）。

【0005】

処理されるデータに基づいて処理を分散しても、描画処理部の性能差により生じる処理時間のばらつきは生じ得る。このような課題に対して、処理装置の処理方法を切り替えるという手法がある（例えば特許文献3等参照）。すなわち、命令をロードして実行可能な処理を切り替えられる処理装置を描画処理部に用い、ロードする命令を切り替え、データに適切な処理装置を実現する。ロードする命令はデータの特性に関連付けて決定される。

10

20

30

40

50

【特許文献１】特開平１０－３０７９２４号公報

【特許文献２】特開２００７－８１７９５号公報

【特許文献３】特開平１１－１６５４３４号公報

【発明の開示】

【発明が解決しようとする課題】

【０００６】

上述したように、処理可能な命令を切り替え可能な処理装置を用いることで、画像形成に係る処理の高速化が実現される。しかし、処理装置が処理可能な命令を切り替える遅延時間が生じるという課題があった。この処理装置の切り替えは、データが分割されているためにデータに合わせて複数回発生する可能性があり、その遅延時間はさらに拡大して、結局並列化によってもたらされる高速化の効果を損ね、処理速度の低下を招くおそれがあった。

10

【０００７】

本発明は、上記の従来例に鑑みてなされたもので、上述した課題を解決することを目的とする。即ち、処理装置における処理可能な命令の切り替えの回数を減少させることで高速化を実現する画像形成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】

【０００８】

上記目的を達成するために本発明は以下の構成を備える。

【０００９】

20

複数の処理ユニットを有する画像形成装置であって、

ブロック化された画像形成処理の対象となる対象データにおいて、ブロックの属性および属性毎の処理負荷パラメータおよび前記処理ユニットの数に基づいて、各処理ユニットにおいて処理対象の属性を決定する決定手段と、

前記決定手段によって決定された前記各処理ユニットにおいて処理対象の属性に応じた処理プログラムを、前記各処理ユニットがそれぞれ持つローカルメモリへロードするロード手段と、

前記ブロック化されたデータに対して、前記ローカルメモリにロードされた処理プログラムを実行して処理するよう制御する手段とを備える。

【発明の効果】

30

【００１０】

本発明によれば、複数の処理ユニットを用いて分散処理を行う際の処理を高速化する効果がある。

【発明を実施するための最良の形態】

【００１１】

以下、本発明の発明を実施するための最良の形態について図面を用いて説明する。

【００１２】

[画像形成処理]

図１は本発明の一実施形態に係る画像形成装置１２と画像形成装置に接続された情報処理装置１１の構成を示すブロック図である。図１において、情報処理装置１１は、パーソナルコンピュータなどの情報処理装置である。情報処理装置１１は同装置上でＰＤＬ（ページ記述言語）データ１００を生成することが出来る。さらに情報処理装置１１はネットワークやＵＳＢに代表されるインターフェースを経由してＰＤＬデータ１００を含むデータを情報処理装置１１の外部の装置、たとえば画像形成装置１２に転送することが出来る。

40

【００１３】

ＰＤＬデータ１００は、図４に示すように、描画データと印刷制御データを含むことが出来る。描画データは、画像形成装置１２が解釈し、可視的な描画を行うことが出来るデータである。また印刷制御データは、ＰＤＬデータ１００をどのような方法で画像形成すればよいかを含む情報である。例えば印刷制御データは、紙のサイズや、紙の種類を指定

50

する情報を示すことが出来る。また、印刷制御データ100-2が持つことが出来る情報は、製本処理の方法などを含む、画像形成を行った後の加工処理のための情報を示すことが出来る。

#### 【0014】

次に、画像形成装置12について説明する。画像形成装置12はプリンタや複合機などであり、PDLデータ100に代表されるデータをもとに、紙に代表されるシート媒体に可視像を形成することが出来る。次に、画像形成装置12がシート媒体に可視像を形成するための、画像形成に係る処理を説明する。図3は画像形成装置12が行う画像形成に係る処理のフロー図である。

#### 【0015】

10

図3において、まず、s10でネットワーク部12-1が情報処理装置11からPDLデータを受信する。ネットワーク部12-1はIEEE802.3に代表されるプロトコルで他装置と通信を行うことで、PDLデータを含むデータを得ることが出来る。また、このs10の処理では、USB部12-2を用いて他装置と通信を行い、PDLデータを含むデータを得てもよい。このほかにも画像形成装置12をIEEE1394aやRS-232C乃至IEEE1284に準拠したインターフェースを介して他装置と接続することで、PDLデータ含むデータを得ても本発明は実施可能である。

#### 【0016】

画像形成装置12が情報処理装置11に代表される他装置から得たPDLデータは、次に画像形成装置12内の画像形成処理部12-3が受け取る(s12)。次に画像形成処理部12-3は、s14で得たPDLデータをもとに印刷データを生成する。s14の詳細は図7を参照して後で説明する。次にs16で、印刷エンジン12-4がこの印刷データ102を得る。印刷データ102は、図5が示すように、ラストイメージデータ102-1と、印刷エンジン制御データ102-2を持つ。印刷エンジンは得た印刷データ102をもとに画像形成を行い、紙などに代表される印刷媒体上に可視像を形成する(s18)。ラストイメージデータ102-1は、複数のブロック(あるいはタイル)102-1-1に分割されてブロック化され、各ブロック毎に処理された後で再結合されてラストイメージデータ102-1が再生される。印刷エンジン12-4による画像形成に用いられるデータは、この再生されたラストイメージデータ102-1である。

20

#### 【0017】

30

##### [ 画像形成処理部による処理 ]

ここまでで画像形成装置12が行う画像形成について説明した。次にその画像形成の中でも、画像形成処理部12-3内で行われる図3のs14で示した処理についてより詳細に説明する。

#### 【0018】

図6は画像形成装置12が持つ画像形成処理部12-3の構成を図示している。画像形成処理部12-3は、CPUに代表される汎用プロセッサ12-3-1を持つ。この汎用プロセッサ12-3-1は、例えば処理ユニットやバス、I/Oコントローラ等、画像形成処理部12-3内の他の構成部分を制御する制御部として機能することが出来る。以下、その画像形成処理部12-3内のほかの装置について述べる。

40

#### 【0019】

複数の処理ユニット1(12-3-2)、処理ユニット2(12-3-3)、処理ユニット3(12-3-4)及び処理ユニット4(12-3-5)はそれぞれ、汎用プロセッサ12-3-1から制御可能な処理ユニットである。なおこれら処理ユニットを、以下、まとめて「処理ユニット(12-3-2、12-3-3、12-3-4、12-3-5)」と略称する。処理ユニットは汎用プロセッサの制御のもとで、PDLデータ100の処理や、後述する中間データ101を生成するレンダリング処理及び生成したラストイメージデータ102-1の画像処理を行うことが出来る。

#### 【0020】

さらに、各処理ユニット(12-3-2、12-3-3、12-3-4、12-3-5

50

)、それぞれローカルメモリ 1 ( 1 2 - 3 - 2 - 1 )、ローカルメモリ 2 ( 1 2 - 3 - 3 - 2 )、ローカルメモリ 3 ( 1 2 - 3 - 3 - 3 ) 及びローカルメモリ 4 ( 1 2 - 3 - 3 - 4 ) ( 以下まとめて「ローカルメモリ ( 1 2 - 3 - 2 - 1、1 2 - 3 - 3 - 2、1 2 - 3 - 3 - 3、1 2 - 3 - 3 - 4 ) 」と称する ) を持つ。これらのローカルメモリ ( 1 2 - 3 - 2 - 1、1 2 - 3 - 3 - 2、1 2 - 3 - 3 - 3、1 2 - 3 - 3 - 4 ) は少なくとも、それぞれの処理ユニット ( 1 2 - 3 - 2、1 2 - 3 - 3、1 2 - 3 - 4、1 2 - 3 - 5 ) が処理可能な処理形態を決定するための命令群を保存することが出来る。例えば各ローカルメモリには複数の命令群が格納されている。汎用プロセッサ 1 2 - 3 - 1 はそれぞれの処理ユニット ( 1 2 - 3 - 2、1 2 - 3 - 3、1 2 - 3 - 4、1 2 - 3 - 5 ) を制御し、これらローカルメモリ ( 1 2 - 3 - 2 - 1、1 2 - 3 - 3 - 2、1 2 - 3 - 3 - 3、1 2 - 3 - 3 - 4 ) に保存される命令群を選択することが出来る。これによって、汎用プロセッサ 1 2 - 3 - 1 は、それぞれの処理ユニット ( 1 2 - 3 - 2、1 2 - 3 - 3、1 2 - 3 - 4、1 2 - 3 - 5 ) が可能な処理を決定することが出来る。たとえば、ひとつの命令群は、テキストの展開に最適なプログラムコード列であり、また他の命令群はグラフィックデータの展開に最適なプログラムコード列である。プログラムをこれら処理別に分けることで、汎用プログラムを用いた場合の冗長処理 ( たとえばオブジェクトの種類の判定など ) が不要となり、コードを最適化 ( あるいは効率化 ) することができる。このように相異なる複数のプログラムコード列を選択的にロード可能であれば、データ種類毎に最も効率よく処理可能なプログラムコード列を実行してデータを処理することが可能となる。

【 0 0 2 1 】

メモリコントローラ 1 2 - 3 - 5 はメインメモリ 1 2 - 3 - 6 及び R O M 1 2 - 3 - 9 へのインターフェースである。即ち、メモリコントローラ 1 2 - 3 - 5 は、メインメモリ 1 2 - 3 - 6 へのデータの読み込み及び書き込みを可能ならしめる。また、メモリコントローラ 1 2 - 3 - 5 は R O M 1 2 - 3 - 9 へのデータの読み込みを可能ならしめる。

【 0 0 2 2 】

I O コントローラ 1 2 - 3 - 8 は画像形成処理部 1 2 - 3 外の構成部分とのインターフェースである。即ち、I O コントローラ 1 2 - 3 - 8 は、画像形成処理部 1 2 - 3 外の装置とのデータの交換を可能ならしめる。

【 0 0 2 3 】

ローカルバス 1 2 - 3 - 7 は、汎用プロセッサ 1 2 - 3 - 1 と、各処理ユニット ( 1 2 - 3 - 2、1 2 - 3 - 3、1 2 - 3 - 4、1 2 - 3 - 5 ) と、メモリコントローラ 1 2 - 3 - 5 と、I O コントローラ 1 2 - 3 - 8 とを接続する。即ち、ローカルバス 1 2 - 3 - 7 はこれら接続した装置間でデータの交換を可能ならしめる。

【 0 0 2 4 】

以上説明した構成で画像形成処理装置 1 2 - 3 は P D L データを元に印刷データを生成する。次に中間データ 1 0 1 の構成を図 8 で、印刷データ生成に係る処理ステップ ( 図 3 の s 1 4 ) を図 7 を用いながら説明する。

【 0 0 2 5 】

< 中間データ >

図 8 は中間データ 1 0 1 を説明した図である。中間データ 1 0 1 は複数の分割された中間データ 1 0 1 - 1 に分割される。また、中間データ 1 0 1 は印刷制御中間データ 1 0 1 - 2 を持つ。分割された中間データ 1 0 1 - 1 はさらに、グラフィックデータ 1 0 1 - 1 - 1、テキストデータ 1 0 1 - 1 - 2 及び領域情報 1 0 1 - 1 - 3 を持つ。

【 0 0 2 6 】

ここで中間データ 1 0 1 は、例えば図 9 のような構造を持つ。分割された中間データ 1 0 1 - 1 は、中間データ 1 0 1 を空間的に分割した各ブロックのデータを持つ。もちろん、中間データ 1 0 1 の分割の仕方は図 9 に例示された分割方法に限らず、本発明を実施できる。しかしながら、負荷分散のための負荷の予測等のためには、各ブロックは同一のサイズであることが望ましい。

【 0 0 2 7 】

10

20

30

40

50

さらにその分割された中間データ101-1はそれぞれ、グラフィックデータ101-1-1とテキストデータ101-1-2とを持つことが出来る。グラフィックデータとは、本実施形態ではラスタイメージとベクタイメージとを含むデータをさす。また、本実施形態ではテキストデータとは、描画すべき文字データをさす。たとえば文字コードとサイズや書体等の就職情報とを含む。

#### 【0028】

領域情報101-1-3は、分割された中間データ101-1それぞれの領域情報を持つ。図10は図9の中間データ101が持つ領域情報を例示した図である。本実施例では領域情報101-1-3は4種類の情報を持つことが出来る。即ち、領域情報101-1-0はその領域に描画すべきデータがないことを示している。領域情報101-3-1はその領域がグラフィックデータを描画すべきであることを示す。領域情報101-3-2はその領域がテキストを描画すべきであることを示す。領域情報103-3-3はその領域がテキストとグラフィックを描画すべきであることを示す。

10

#### 【0029】

印刷制御中間データ101-2は、印刷制御データ100-2をもとに生成されるデータである。この印刷制御データ100-2は少なくとも、画像形成すべき媒体の大きさや（例えば用紙サイズと呼ばれるもの）、製本処理の方法を示す情報などを持つことが出来る。

#### 【0030】

以上、汎用プロセッサ12-3-1によって生成される中間データ101を説明した。次に印刷データ生成処理を図7を参照して説明する。

20

#### 【0031】

##### <印刷データ生成処理>

図7において、画像生成処理部12-3の汎用プロセッサ12-3-1はPDLデータ100の解釈処理を行い、中間データ101を生成する(s14-0)。このPDLデータ100は複数の種類がある。例えばアドビシステムズ社のPDFやポストスクリプト（登録商標）、キヤノン株式会社のLIPSなどが知られている。汎用プロセッサ12-3-1は、これら異なるデータフォーマットが存在するPDLデータ100から、共通のデータ形式である中間データ101を生成する。ただし、汎用プロセッサ12-3-1は異なる全てのデータフォーマットに対応する必要はなく、少なくともいずれか一つに対応しているだけであっても良い。本例ではこの中間データがブロック化された処理対象のデータすなわち対象データである。しかし、PDLデータを処理対象と呼ぶこともできる。

30

#### 【0032】

次に汎用プロセッサ12-3-1は、処理ユニットへ命令（プログラムモジュール）をロードする(s14-2)。この処理s14-2を図11を用いながら説明する。汎用プロセッサ12-3-1は生成された中間データを解析する。少なくともその解析をもとに処理ユニット(12-3-2、12-3-3、12-3-4、12-3-5)のローカルメモリ(12-3-2-1、12-3-3-2、12-3-3-3、12-3-3-4)へそれぞれ転送する命令群を決定する。この汎用プロセッサ12-3-1が行う解析と命令のロード(s14-2)はより詳細には下記のように行われる。

40

#### 【0033】

まず汎用プロセッサ12-3-1は、s14-2-0で、少なくとも中間データに含まれる属性情報101-1-3が示す情報を分析する。より詳細には、汎用プロセッサ12-3-1は、中間データ101が持つ各分割された中間データ101-1に対応した領域情報101-1-3から、"None"101-1-3-0,"Text"101-1-3-2,"Graphic"101-1-3-1,"Text+Graphic"101-1-3-3の属性それぞれに分類される中間データ101-1（すなわちブロック或いはタイル）がいくつあるかを数える。以下、この属性を示す分割された中間データ101-1の数を、その属性のタイル数という。

#### 【0034】

50

次に汎用プロセッサ 12 - 3 - 1 は s 14 - 2 - 2 で、少なくとも s 14 - 2 - 0 で得られた情報をもとに、各属性に対応する重み付けパラメータを計算する。より詳細には、各重み付けパラメータは、少なくとも各属性のタイル数と、あらかじめ用意された各属性の処理負荷パラメータとを乗算して得られる。このあらかじめ用意された各属性の処理負荷パラメータは、印刷データ 102 を得るために、属性ごとに、分割された中間データ 101 - 1 を処理するのに必要な処理負荷の比率を示す。本実施形態では、例えば図 16 のような値が与えられる。即ち、"Text" 101 - 1 - 3 - 2 の属性を持つ分割された中間データ 101 - 1 は、"Graphic" 101 - 1 - 3 - 1 の属性を持つ分割された中間データ 101 - 1 に対して、5 / 3 (おおよそ 1.7 倍) の処理時間がかかることを示す。図 16 では、データ無し (NONE) を基準として示されている。本実施形態では、このあらかじめ用意された各属性の処理負荷パラメータは ROM 12 - 3 - 9 に保存されている。汎用プロセッサ 12 - 3 - 1 はメモリコントローラ 12 - 3 - 5 を介して ROM 12 - 3 - 9 に保存された各属性の処理負荷パラメータを得ることが出来る。

10

#### 【0035】

また本実施形態では、少なくとも s 14 - 2 - 0 で得られた中間データをもとに、属性に対応する重み付けパラメータを計算を下記のような計算式で行う。

$wc(none) = Tile(none) \times PL(none)$

$wc(image) = Tile(image) \times PL(image)$

$wc(text) = Tile(text) \times PL(text)$

$wc(image + text) = Tile(image + text) \times PL(image + text) \dots$  (数式 1)

20

ここで、 $wc(attr)$  は各属性 ( $attr$ ) の重み付けパラメータを、 $Tile(attr)$  は各属性 ( $attr$ ) のタイル数 (ブロック数) を、 $PL(attr)$  は各属性 ( $attr$ ) の処理負荷パラメータを示す。タイル数はデータ量の指標値として利用されている。タイル数は処理単位毎であってよく、たとえばバンディングを行っている場合には、バンド内で数えられる。このほかページを単位として数えても良いし、ジョブを単位として数えても良い。処理負荷パラメータはたとえば実験的に予め測定するなどして決定しておくことができる。

#### 【0036】

計算結果である  $wc(attr)$  は、中間データ 101 を処理し印刷データ 102 を得るのに必要とする属性毎の処理の重みを示す。すなわち、ある印刷単位 (ジョブやページ或いはバンドなど) を処理する場合の、属性毎の処理負荷の指標値である。例えばある属性の  $wc(attr)$  が他の属性の  $wc(attr)$  より大きいということは以下のことを意味する。即ち、中間データ 101 から印刷データ 102 を得るために、前者の属性の  $wc(attr)$  を示す分割された中間データ 101 - 1 の処理の総時間が、後者の属性を示す中間データ 101 - 1 の処理の総時間より大きい。

30

#### 【0037】

次に、この計算結果から汎用プロセッサ 12 - 3 - 1 は s 14 - 2 - 4 でそれぞれの処理ユニット (12 - 3 - 2、12 - 3 - 3、12 - 3 - 4、12 - 3 - 5) へロードすべき命令を決定する。

40

#### 【0038】

この s 14 - 2 - 4 では、汎用プロセッサ 12 - 3 - 1 は、数式 1 でえられた  $wc(attr)$  の値にもとづき、ROM 12 - 3 - 9 乃至 12 - 3 - 6 に保存された命令を選択する。図 12 は ROM 12 - 3 - 9 乃至メインメモリ 12 - 3 - 6 に保存された命令を図解している。

#### 【0039】

命令群 1000 は、s 14 - 4 で処理ユニット (12 - 3 - 2、12 - 3 - 3、12 - 3 - 4、12 - 3 - 5) にロードすることのできる命令群である。以下に、その命令群に含まれる命令の種類について説明する。

#### 【0040】

50

< 提供し得るプログラムモジュール >

基本プログラム用命令群 1 0 0 0 - 1 は、主にローカルメモリ ( 1 2 - 3 - 2 - 1、1 2 - 3 - 3 - 2、1 2 - 3 - 3 - 3、1 2 - 3 - 3 - 4 ) のメモリ管理や以下に示される他命令群の管理を行う。

【 0 0 4 1 】

レンダラ命令群 1 0 0 0 - 2 は、レンダリング処理プログラムである。ここで言うレンダリング処理は、中間データ 1 0 1 中の分割された中間データ 1 0 1 - 1 すなわちタイルをレンダリング処理し、ラスタイメージデータを生成する処理である。レンダラ命令群 1 0 0 0 - 2 はさらに、P D L - A 用レンダラモジュール 1 0 0 0 - 2 - 0、P D L - B 用レンダラモジュール 1 0 0 0 - 2 - 1 及び共通レンダラモジュール 1 0 0 0 - 2 - 3 を持つ。それぞれのレンダラモジュールは、互いに異なる P D L に基づいて生成された中間データを解釈してレンダリングするために適したプログラムモジュールである。

10

【 0 0 4 2 】

画像処理命令群 1 0 0 0 - 3 は、レンダリング処理で得られたラスタイメージデータに画像処理をするためのプログラムである。画像処理命令群 1 0 0 0 - 3 には、文字画像処理モジュール 1 0 0 0 - 3 - 0、グラフィック画像処理モジュール 1 0 0 0 - 3 - 1、共通画像処理モジュール 1 0 0 0 - 3 - 2 が含まれる。

【 0 0 4 3 】

C M S ( カラーマネジメントシステム ) 命令群 1 0 0 0 - 4 は、色調整を行うためのプログラムである。C M S 命令群 1 0 0 0 - 4 はさらに、P D L - A 用 C M S モジュール 1 0 0 0 - 4 - 0、P D L - B 用 C M S モジュール 1 0 0 0 - 4 - 1、共通 C M S モジュール 1 0 0 0 - 4 - 2 を持つ。

20

【 0 0 4 4 】

次にこれら、命令群すなわちプログラムが、P D L - A 用と P D L - B 用や文字画像処理とグラフィック処理といったように複数種類用意されている理由について述べる。上述したように、P D L データ 1 0 0 は複数の種類のデータフォーマットが存在する。繰り返しになるが、P D L には、例えばアドビ社の P D F やポストスクリプト ( 登録商標 )、さらには L I P S など存在する。汎用プロセッサ 1 2 - 3 - 1 は、これら複数種類ある P D L データ 1 0 0 を単一のデータフォーマットである中間データ 1 0 1 に変換する。しかし、中間データ 1 0 1 のデータフォーマットは統一されているものの、P D L データ 1 0 0 の種類に依存してその中間データ 1 0 1 が持つデータの傾向は異なり得る。

30

【 0 0 4 5 】

例えば、アドビ P D F は、その P D L データ 1 0 0 に透過情報を持つことが出来る。このため、P D F で記述した P D L データ 1 0 0 から生成された中間データ 1 0 1 は透過情報を含むか、それに依存した情報を持つ。一方でポストスクリプト ( 登録商標 ) や L I P S はその P D L データ 1 0 0 に透過情報を持つことが出来ない。このため、ポストスクリプト ( 登録商標 ) や L I P S の P D L データから生成された中間データ 1 0 1 は透過情報及び、それに依存した情報を持つことはない。

【 0 0 4 6 】

このように P D L データ 1 0 0 の種類の違いから中間データ 1 0 1 で記述されうる情報が異なるため、中間データ 1 0 1 からラスタイメージデータを生成する際に必要となる処理も異なる。即ち、例えばアドビ P D F で記述した P D L データから生成された中間データ 1 0 1 は透過情報乃至それに依存した情報を処理してラスタイメージデータを生成する必要がある。しかしながら、例えばポストスクリプト ( 登録商標 ) や L I P S で記述した P D L データから生成された中間データ 1 0 1 は、透過情報乃至それに依存した情報を処理してラスタイメージデータを生成する必要がない。

40

【 0 0 4 7 】

本実施形態ではこれらの必要な処理の違いを考慮し、R O M 1 2 - 3 - 9 乃至メインメモリ 1 2 - 3 - 6 に複数種類の命令 ( プログラムモジュール ) が記録されている。例えば、P D L - A 用レンダラモジュールは、ポストスクリプト ( 登録商標 ) 用であり、透過情

50



報を処理するための機能と、また、ラスタデータをCMYK表色系で出力する機能とを有する。一方、PDL-B用レンダラモジュールは、他のPDL用であり、透過情報を処理するための機能は持たず、また、ラスタデータをRGB表色系で出力する機能を有する。もちろんこれは一例に過ぎない。

#### 【0048】

次に、画像処理命令群1000-3に含まれる文字画像処理モジュール1000-3-0とグラフィック画像処理モジュール1000-3-1を説明する。文字画像処理モジュール1000-3-0は、文字部分のエッジ強調など、文字に特有な画像処理を行うことが出来る。対してグラフィック画像処理モジュール1000-3-1は、エッジ強調処理など文字に特有な処理機能を持たない代わりに、写真の赤目補正や肌色補正処理などの処理機能を持つ。

10

#### 【0049】

上述したように、中間データ101が持つ分割された中間データ101-1は属性情報101-1-3を持つことが出来る。この属性情報101-1-3はその分割された中間データ101-1が持つ描画データの特性を示している。然るに、この属性情報101-1-3から、汎用プロセッサ12-3-1或いは処理ユニット(12-3-2、12-3-3、12-3-4、12-3-5)は、ある分割された中間データ101-1が文字画像処理モジュール1000-3-0が持つ処理を必要としているのか、或いはグラフィック画像処理モジュール1000-3-1が持つ処理を必要としているのかを判断することが出来る。

20

#### 【0050】

本実施形態では、属性毎に必要な処理の違いを考慮し、ROM12-3-9乃至メインメモリ12-3-6に複数種類の命令が記録されている。

#### 【0051】

次に、CMS命令群1000-4に含まれるPDL-A用CMSモジュールとPDL-B用CMSモジュール1000-4-1を説明する。上述したように、PDLデータ100の種類は複数ある。このためPDLデータ100の種類に依存して、必要とされる色調整に係る処理も異なる。例えばポストスクリプト(登録商標)は、その言語仕様のため、レンダリング処理をCMYK色空間で行う必要がある。このため本実施形態において汎用プロセッサ12-3-1及び処理ユニット(12-3-2、12-3-3、12-3-4、12-3-5)は、画像形成処理装置12-3に入力されたPDLデータがポストスクリプト(登録商標)である場合には、そのPDLデータ100、及び中間データ101の処理をCMYK色空間内で処理を行う。

30

#### 【0052】

これに対して、LIPSなどのPDLデータ100はCMYK色空間ではなくRGB色空間で色調整が施されてもよい。ところでCMYK色空間に対してRGB色空間は、色を表現するために必要な情報量が少ない。このためRGB色空間で色調整を行うと、CMYK色空間で色調整処理を行うよりも、処理の高速化及び必要とされるメモリ量の削減が可能である。このため、汎用プロセッサ12-3-1及び処理ユニット(12-3-2、12-3-3、12-3-4、12-3-5)は、RGB色空間で処理可能なLIPS等のPDLデータ100から生成された中間データ101をRGB色空間で処理した方が望ましい。処理速度の向上と処理に必要とされるメモリ量の削減が見込めるからである。

40

#### 【0053】

本実施形態では、これらの必要な処理の違いを考慮し、ROM12-3-9乃至メインメモリ12-3-6に複数種類の命令群すなわち複数種類のプログラムモジュールが記録されている。例えばPDL-A用CMSモジュールは、ポストスクリプト(登録商標)用であり、CMYK表色系を処理する。また、透過情報について処理する機能を持っていても良い。また例えばPDL-B用CMSモジュールは、その他のPDL用であり、RGB表色系を処理する。また、透過情報について処理する機能は不要である。

#### 【0054】

50

以上、重み付けパラメータの計算結果から汎用プロセッサ 12-3-1 が s 14-2-4 で決定すべき命令群（プログラムモジュール）について説明した。

#### 【0055】

<プログラムモジュールの選択処理>

次に、この s 14-2-4 で汎用プロセッサ 12-3-1 が処理ユニット（12-3-2、12-3-3、12-3-4、12-3-5）にロードすべき命令を選択する処理を説明する。図 13 はその処理 s 14-2-4 をより詳細を図解している。

#### 【0056】

汎用プロセッサ 12-3-1 はまず、s 14-2-4-0 で各属性に対応するプログラムモジュールの数を決定する。この処理は、複数ある処理ユニット（12-3-2、12-3-3、12-3-4、12-3-5）にロードすべきプログラムモジュールの種類の数を決定する。

#### 【0057】

本実施形態では処理ユニット（12-3-2、12-3-3、12-3-4、12-3-5）にロードされる命令群（プログラムモジュール）は少なくとも属性情報 101-1-3 に関連付けられている。即ち、汎用プロセッサ 12-3-1 は少なくとも処理ユニット（12-3-2、12-3-3、12-3-4、12-3-5）にロードすべき命令（プログラムモジュール）として、少なくとも以下のものを選択できる。

- 1．属性情報 101-1-3 が "Text" 101-1-3-2 を示す分割された中間データ 101-1 に関連付けられたプログラムモジュール
- 2．属性情報 101-1-3 が "Graphic" 101-1-3-1 を示す分割された中間データ 101-1 に関連付けられたプログラムモジュール
- 3．属性情報 101-1-3 が "Text + Graphic" 101-1-3-3 を示す分割された中間データ 101-1 に関連付けられたプログラムモジュール。

#### 【0058】

汎用プロセッサ 12-3-1 は少なくとも数式 1 で与えられる重み付けパラメータから、これらの命令（プログラムモジュール）をそれぞれいくつ処理ユニット（12-3-2、12-3-3、12-3-4、12-3-5）にロードすべきかを決定する。

#### 【0059】

例えば、

$$w c ( t e x t ) = 30.0$$

$$w c ( g r a p h i c ) = 42.0$$

$$w c ( t e x t + g r a p h i c ) = 102.0$$

という計算結果が得られたとする。

#### 【0060】

ここで本実施形態では、図 6 に示したように処理ユニット（12-3-2、12-3-3、12-3-4、12-3-5）は 4 つある。このことから、汎用プロセッサ 12-3-1 は重み付け係数  $w c ( a t t r )$  を、それぞれの比が示す値の合計値が 4 になるように正規化する。すなわち重み付け係数は、その合計値が、レンダリング等の処理が分散される処理ユニットの数と等しくなるように正規化される。画像形成処理部 12-3 の汎用プロセッサ 12-3-1 はこの正規化を行って上記重み付け係数をおおむね以下のような値に変換する。ここで「normal」は正規化済みの値であることを示す。

$$n o r m a l ( w c ( t e x t ) ) = 0.69$$

$$n o r m a l ( w c ( g r a p h i c ) ) = 0.97$$

$$n o r m a l ( w c ( t e x t + g r a p h i c ) ) = 2.35。$$

#### 【0061】

これらの重み付け係数値は、属性情報に関連付けられた処理の負荷の比を示す。因って、得られた重み付け係数値にもとづき、汎用プロセッサ 12-3-1 は属性情報に関連付けられた命令群（プログラムモジュール）を、それぞれいくつ処理ユニット（12-3-2、12-3-3、12-3-4、12-3-5）にロードすべきかを判断できる。各処

理ユニットは1個未満に分割することが出来ないので、汎用プロセッサ12-3-1は以下のように処理を行う。

1. 重み付け係数が1未満の場合には、1に切り上げる。すなわち、該当する属性の中間データのタイルを処理するために、1つの処理ユニットを割り当てる。
2. 割り当て済みの処理ユニットの数を、重み付け係数の合計値（すなわち全処理ユニット数）から差し引く。
3. 差し引かれた値を、残りの属性に対応する重み付け係数値の比で分割する。もちろん整数となるように分割される。
4. 以上の手順でもしも処理ユニット数の合計値に過不足があった場合には、たとえば最も多くの処理ユニットが割り当てられている属性について、その処理ユニット数で調整するなどすればよい。こうして、おおむね各属性の重み付け係数の比率で、各属性に割り当てる処理ユニットの数を決定することができる。このとき数だけでなく、各属性に割り当てる処理ユニットを決定しても良い。

#### 【0062】

本実施形態では具体例として以下になる。

1. 汎用プロセッサ12-3-1は、属性情報101-1-3が"Text"101-1-3-2を示す分割された中間データ101-1に関連付けられた命令を処理ユニット(12-3-2、12-3-3、12-3-4、12-3-5)に1つロードすることを決定する。これは重み付け係数が1未満であるためである。
2. 汎用プロセッサ12-3-1は、属性情報101-1-3が"Graphic"101-1-3-1を示す分割された中間データ101-1に関連付けられた命令を処理ユニット(12-3-2、12-3-3、12-3-4、12-3-5)に1つロードすることを決定する。これは重み付け係数が1未満であるためである。
3. 汎用プロセッサ12-3-1は、属性情報101-1-3が"Text+Graphic"101-1-3-3を示す分割された中間データ101-1に関連付けられた命令を処理ユニット(12-3-2、12-3-3、12-3-4、12-3-5)に2つロードすることを決定する。これは残された処理ユニットを全て割り当てられるためである。

#### 【0063】

以上、汎用プロセッサ12-3-1は属性情報101-1-3に関連付けられた命令群を、それぞれいくつ処理ユニット(12-3-2、12-3-3、12-3-4、12-3-5)にロードすべきかを判断した。次に、汎用プロセッサ12-3-1はs14-2-4-2で上記それぞれの命令群を生成するために、命令の選択処理を行う。尚、この命令の選択処理は、汎用プロセッサ12-3-1によってまずそれぞれの処理ユニットに属性を対応付けておく。そして、属性情報101-1-3が示す属性ごとに図14の処理が行われる。図14の処理の結果、属性に対応してプログラムモジュールが選択される。選択されたプログラムモジュールは、当該属性に対して割り当てられた処理ユニットのローカルメモリにロードされる。一属性に対して複数の処理ユニットが割り当てられている場合には、該当する処理ユニット全てに共通のプログラムモジュールがロードされる。

#### 【0064】

さて、図14はその命令選択処理s14-2-4-2をより詳細に説明する図である。以下、図14を用いながら命令選択処理s14-2-4-2のより詳細な説明を行う。

#### 【0065】

汎用プロセッサ12-3-1は、まずs14-2-4-2-0で中間データ101の生成に使われたPDLデータ100がROPや透過処理といった処理を記述可能な種類かを判断する。汎用プロセッサ12-3-1は、中間データ101が若しROPや透過処理といった処理を記述可能な種類のPDLデータ100から生成されていれば、s14-2-4-2-2に進む。汎用プロセッサ12-3-1は、中間データ101が若しROPや透過処理といった処理を記述可能な種類のPDLデータ100から生成されていなければ、s14-2-4-2-4に進む。

## 【 0 0 6 6 】

汎用プロセッサ 1 2 - 3 - 1 は s 1 4 - 2 - 4 - 2 - 2 では、処理ユニット ( 1 2 - 3 - 2、1 2 - 3 - 3、1 2 - 3 - 4、1 2 - 3 - 5 ) にロードすべき命令として上述した P D L - A 用レンダラモジュール 1 0 0 0 - 2 - 0 を選択する。汎用プロセッサ 1 2 - 3 - 1 は、一方で s 1 4 - 2 - 4 - 2 - 4 では、処理ユニット ( 1 2 - 3 - 2、1 2 - 3 - 3、1 2 - 3 - 4、1 2 - 3 - 5 ) にロードすべき命令として上述した P D L - B 用レンダラモジュール 1 0 0 0 - 2 - 1 を選択する。

## 【 0 0 6 7 】

次に汎用プロセッサ 1 2 - 3 - 1 は s 1 4 - 2 - 4 - 2 - 6 で、処理ユニット ( 1 2 - 3 - 2、1 2 - 3 - 3、1 2 - 3 - 4、1 2 - 3 - 5 ) にロードすべき命令として上述した共通レンダラモジュール 1 0 0 0 - 2 - 3 を選択する。

10

## 【 0 0 6 8 】

以上の処理で、汎用プロセッサ 1 2 - 3 - 1 によるレンダラ命令群 1 0 0 0 - 2 中の命令の選択が完了する。

## 【 0 0 6 9 】

次に汎用プロセッサ 1 2 - 3 - 1 は s 1 4 - 2 - 4 - 2 - 8 で、処理の対象となっている属性情報 1 0 0 - 1 - 3 が示す属性が " T e x t " かを調べる。汎用プロセッサ 1 2 - 3 - 1 は、若し属性が " T e x t " であれば s 1 4 - 2 - 4 - 2 - 1 0 に進む。一方、汎用プロセッサ 1 2 - 3 - 1 は、若し属性が " T e x t " でなければ s 1 4 - 2 - 4 - 2 - 1 2 に進む。汎用プロセッサ 1 2 - 3 - 1 は、s 1 4 - 2 - 4 - 2 - 1 0 では、処理ユニット ( 1 2 - 3 - 2、1 2 - 3 - 3、1 2 - 3 - 4、1 2 - 3 - 5 ) にロードすべき命令として上述した文字画像処理モジュール 1 0 0 0 - 3 - 0 を選択する。

20

## 【 0 0 7 0 】

また、汎用プロセッサ 1 2 - 3 - 1 は s 1 4 - 2 - 4 - 2 - 1 2 では、処理の対象となっている属性情報 1 0 0 - 1 - 3 が示す属性が " G r a p h i c " かを調べる。汎用プロセッサ 1 2 - 3 - 1 は、若し属性が " G r a p h i c " であれば s 1 4 - 2 - 4 - 2 - 1 4 に進む。一方、汎用プロセッサ 1 2 - 3 - 1 は、若し属性が " G r a p h i c " でなければ s 1 4 - 2 - 4 - 2 - 1 6 に進む。汎用プロセッサ 1 2 - 3 - 1 は、s 1 4 - 2 - 4 - 2 - 1 4 では、処理ユニット ( 1 2 - 3 - 2、1 2 - 3 - 3、1 2 - 3 - 4、1 2 - 3 - 5 ) にロードすべき命令として上述したグラフィック画像処理モジュール 1 0 0 0 - 3 - 1 を選択する。

30

## 【 0 0 7 1 】

また、汎用プロセッサ 1 2 - 3 - 1 は s 1 4 - 2 - 4 - 2 - 1 6 では、処理の対象となっている属性情報 1 0 0 - 1 - 3 が示す属性が " T e x t + G r a p h i c " かを調べる。汎用プロセッサ 1 2 - 3 - 1 は、若し属性が " T e x t + G r a p h i c " であれば s 1 4 - 2 - 4 - 2 - 1 8 に進む。一方、汎用プロセッサ 1 2 - 3 - 1 は、若し属性が " T e x t + G r a p h i c " でなければ s 1 4 - 2 - 4 - 2 - 2 2 に進む。汎用プロセッサ 1 2 - 3 - 1 は、s 1 4 - 2 - 4 - 2 - 1 8 では、処理ユニット ( 1 2 - 3 - 2、1 2 - 3 - 3、1 2 - 3 - 4、1 2 - 3 - 5 ) にロードすべき命令として上述した文字画像処理モジュール 1 0 0 0 - 3 - 0 を選択する。さらに、汎用プロセッサ 1 2 - 3 - 1 は、s 1 4 - 2 - 4 - 2 - 2 0 では、処理ユニット ( 1 2 - 3 - 2、1 2 - 3 - 3、1 2 - 3 - 4、1 2 - 3 - 5 ) にロードすべき命令として上述したグラフィック画像処理モジュール 1 0 0 0 - 3 - 1 を選択する。

40

## 【 0 0 7 2 】

上記処理を行った上で、汎用プロセッサ 1 2 - 3 - 1 は s 1 4 - 2 - 4 - 2 - 2 2 で、処理ユニット ( 1 2 - 3 - 2、1 2 - 3 - 3、1 2 - 3 - 4、1 2 - 3 - 5 ) にロードすべき命令として上述した共通画像処理モジュール 1 0 0 0 - 3 - 2 を選択する。

## 【 0 0 7 3 】

以上の処理で、汎用プロセッサ 1 2 - 3 - 1 による画像処理命令群 1 0 0 0 - 3 中の命令の選択が完了する。

50

## 【 0 0 7 4 】

次に汎用プロセッサ 1 2 - 3 - 1 は s 1 4 - 2 - 4 - 2 - 2 4 で、中間データ 1 0 1 の生成に使われた P D L データ 1 0 0 が C M Y K 色空間で処理が必要な種類の P D L データ 1 0 0 かを調べる。若し C M Y K 色空間で処理が必要な種類の P D L データ 1 0 0 であれば、汎用プロセッサ 1 2 - 3 - 1 は s 1 4 - 2 - 4 - 2 - 2 6 にすすむ。一方、汎用プロセッサ 1 2 - 3 - 1 は、若し C M Y K 色空間で処理が必要な種類の P D L データ 1 0 0 でなければ s 1 4 - 2 - 4 - 2 - 2 8 に進む。汎用プロセッサ 1 2 - 3 - 1 は、s 1 4 - 2 - 4 - 2 - 2 6 では、処理ユニット ( 1 2 - 3 - 2、1 2 - 3 - 3、1 2 - 3 - 4、1 2 - 3 - 5 ) にロードすべき命令として上述した P D L - A 用 C M S モジュール 1 0 0 0 - 4 - 0 を選択する。また、汎用プロセッサ 1 2 - 3 - 1 は、s 1 4 - 2 - 4 - 2 - 2 8 では、処理ユニット ( 1 2 - 3 - 2、1 2 - 3 - 3、1 2 - 3 - 4、1 2 - 3 - 5 ) にロードすべき命令として上述した P D L - B 用 C M S モジュール 1 0 0 0 - 4 - 1 を選択する。

10

## 【 0 0 7 5 】

上記処理を行った上で、汎用プロセッサ 1 2 - 3 - 1 は s 1 4 - 2 - 4 - 2 - 3 0 で、処理ユニット ( 1 2 - 3 - 2、1 2 - 3 - 3、1 2 - 3 - 4、1 2 - 3 - 5 ) にロードすべき命令として上述した共通 C M S モジュール 1 0 0 0 - 4 - 2 を選択する。

## 【 0 0 7 6 】

以上、この命令の選択処理 s 1 4 - 2 - 4 - 2 は、汎用プロセッサ 1 2 - 3 - 1 を用いて、属性情報 1 0 1 - 1 - 3 が示す属性の数だけ行う。汎用プロセッサ 1 2 - 3 - 1 はこの繰り返し処理によって、属性情報 1 0 1 - 1 - 3 が示す属性に関連付けられた命令群を生成する。

20

## 【 0 0 7 7 】

もちろんこのような処理を行わずに、予め P D L の種類と属性とプログラムモジュールの識別子とを関連づけたテーブルを R O M 等に用意しておき、P D L の種類と属性とに関連づけられたプログラムモジュールを選択するようにしても良い。その場合の関連づけは、図 1 4 にしめした要領で行われる。テーブル化した方が、プログラムモジュールの選択を依り迅速に行うことができる。

## 【 0 0 7 8 】

< プログラムモジュールのロード >

30

以上 s 1 4 - 2 - 4 を詳細に説明したので、以下に図 1 1 に示されている s 1 4 - 2 - 6 を説明する。汎用プロセッサ 1 2 - 3 - 1 は s 1 4 - 2 - 8 で、s 1 4 - 2 - 6 で選択された命令群を、それぞれ処理ユニット ( 1 2 - 3 - 2、1 2 - 3 - 3、1 2 - 3 - 4、1 2 - 3 - 5 ) へロードする。このロード処理 s 1 4 - 2 - 8 によって、各処理ユニット ( 1 2 - 3 - 2、1 2 - 3 - 3、1 2 - 3 - 4、1 2 - 3 - 5 ) がもつローカルメモリ ( 1 2 - 3 - 2 - 1、1 2 - 3 - 3 - 2、1 2 - 3 - 3 - 3、1 2 - 3 - 3 - 4 ) に命令群 ( プログラム ) がロードされる。このローカルメモリ ( 1 2 - 3 - 2 - 1、1 2 - 3 - 3 - 2、1 2 - 3 - 3 - 3、1 2 - 3 - 3 - 4 ) に命令群がロードされることによって、処理ユニット ( 1 2 - 3 - 2、1 2 - 3 - 3、1 2 - 3 - 4、1 2 - 3 - 5 ) は中間データ 1 0 1 を処理して印刷データ 1 0 2 を生成する処理 s 1 4 - 4 を可能となる。

40

## 【 0 0 7 9 】

以上、図 1 1 に示された処理ユニット ( 1 2 - 3 - 2、1 2 - 3 - 3、1 2 - 3 - 4、1 2 - 3 - 5 ) へ命令をロードする処理 s 1 4 - 2 を説明した。

## 【 0 0 8 0 】

< 印刷データ生成処理 >

次に、図 1 5 を用いて、中間データ 1 0 1 を処理し印刷データ 1 0 2 を生成する処理 s 1 4 - 4 を説明する。

## 【 0 0 8 1 】

先ず、汎用プロセッサ 1 2 - 3 - 1 は、s 1 4 - 4 - 0 で未処理の分割された中間データ 1 0 1 - 1 があるかを調べる。此处で言う未処理の分割された中間データとは、未だ印

50

刷データ102を生成するための処理が終わっていない中間データ101の部分の意味する。

【0082】

若し未処理の分割された中間データ101-1があれば、汎用プロセッサ12-3-1はs14-4-2に進む。一方、未処理の分割された中間データ101-1がなければ、汎用プロセッサ12-3-1はs14-4の処理を終了する。

【0083】

次に、汎用プロセッサ12-3-1はs14-4-2で次の未処理の分割された中間データ101-1を読み込む。この処理によって、汎用プロセッサ12-3-1は、未処理の分割された中間データ101-1中の属性情報101-1-3が示す属性を調べることが出来る。

10

【0084】

次に、汎用プロセッサ12-3-1はs14-4-4で処理をしていない処理ユニット(12-3-2、12-3-3、12-3-4、12-3-5)の中から、未処理の分割された中間データ101-1が示す属性情報101-1-3に対応する処理ユニット(12-3-2、12-3-3、12-3-4、12-3-5)があるかを調べる。もし処理をしていない処理ユニット(12-3-2、12-3-3、12-3-4、12-3-5)の中から、処理対象である未処理の分割された中間データ101-1が示す属性情報101-1-3に対応する処理ユニット(12-3-2、12-3-3、12-3-4、12-3-5)があれば、汎用プロセッサ12-3-1はs14-4-6に進む。また、この判断によって、処理をしていない処理ユニット(12-3-2、12-3-3、12-3-4、12-3-5)がないか、処理をしていない処理ユニット(12-3-2、12-3-3、12-3-4、12-3-5)のうち未処理の分割された中間データ101-1が示す属性情報101-1-3に対応する処理ユニット(12-3-2、12-3-3、12-3-4、12-3-5)が無かった場合には、汎用プロセッサ12-3-1はs14-4-8に進む。

20

【0085】

次に汎用プロセッサ12-3-1はs14-4-8で次の未処理の分割された中間データ101-1を探す処理を行う。この処理によって、もしs14-4-0で処理の分割された中間データ101-1が見つければ、次のs14-4-2で示された処理で汎用プロセッサ12-3-1はs14-4-8で見つかられた未処理の分割された中間データ101-1に注目して処理を行う。

30

【0086】

なおs14-4-8では、すべての処理ユニットがビジーである場合には、いずれかの処理ユニットにおける処理が終了するまで待つことが望ましい。すべての処理ユニットがビジーであると新たな未処理の分割された中間データを見付けても処理できないためである。その場合、いずれかの処理ユニットにおける処理が終了したなら、改めてs14-4-8から処理を再開する。

【0087】

一方s14-4-6で汎用プロセッサ12-3-1は、s14-4-4で見つけた処理ユニット(12-3-2、12-3-3、12-3-4、12-3-5)に対して、上記未処理の分割された中間データ101-1を通知する。すなわち処理対象のデータをブロック谷で通知する。通知されるのは、処理対象を特定するための情報であり、たとえばそのアドレス及びサイズである。また中間データそのものであっても良い。これによって、対象となった処理ユニット(12-3-2、12-3-3、12-3-4、12-3-5)は、分割された中間データ101-1をもとに、分割されたラストイメージデータ102-1-1を生成する。

40

【0088】

次に、上記s14-4-6を受けて行われる処理ユニット(12-3-2、12-3-3、12-3-4、12-3-5)が行う処理について以下に説明する。

50

## 【 0 0 8 9 】

先ず処理ユニット( 1 2 - 3 - 2、1 2 - 3 - 3、1 2 - 3 - 4、1 2 - 3 - 5 )は s 1 4 - 4 - 1 0 で、汎用プロセッサ 1 2 - 3 - 1 によって示された未処理の分割された中間データ 1 0 1 - 1 を読み込む。本実施形態では、この分割された中間データ 1 0 1 - 1 はメインメモリ 1 2 - 3 - 6 に保存されている。この処理 s 1 4 - 4 - 1 0 によって処理ユニット( 1 2 - 3 - 2、1 2 - 3 - 3、1 2 - 3 - 4、1 2 - 3 - 5 )はローカルメモリ( 1 2 - 3 - 2 - 1、1 2 - 3 - 3 - 2、1 2 - 3 - 3 - 3、1 2 - 3 - 3 - 4 )に未処理の分割された中間データ 1 0 1 - 1 をロードすることが出来る。

## 【 0 0 9 0 】

次に処理ユニット( 1 2 - 3 - 2、1 2 - 3 - 3、1 2 - 3 - 4、1 2 - 3 - 5 )は s 1 4 - 4 - 1 2 で分割された中間データを処理して、分割されたラストイメージデータ 1 0 2 - 1 - 1 を生成する。この処理 s 1 4 - 4 - 1 2 は処理ユニット( 1 2 - 3 - 2、1 2 - 3 - 3、1 2 - 3 - 4、1 2 - 3 - 5 )が、汎用プロセッサ 1 2 - 3 - 1 が s 1 4 - 2 - 4 - 2 で生成した命令群を使って実行する。また、本実施形態では、処理ユニット( 1 2 - 3 - 2、1 2 - 3 - 3、1 2 - 3 - 4、1 2 - 3 - 5 )は分割されたラストイメージデータ 1 0 2 - 1 - 1 をローカルメモリ( 1 2 - 3 - 2 - 1、1 2 - 3 - 3 - 2、1 2 - 3 - 3 - 3、1 2 - 3 - 3 - 4 )上に保存する。

## 【 0 0 9 1 】

次に処理ユニット( 1 2 - 3 - 2、1 2 - 3 - 3、1 2 - 3 - 4、1 2 - 3 - 5 )は s 1 4 - 4 - 1 4 で、ローカルメモリ( 1 2 - 3 - 2 - 1、1 2 - 3 - 3 - 2、1 2 - 3 - 3 - 3、1 2 - 3 - 3 - 4 )に生成された上記分割されたラストイメージデータ 1 0 2 - 1 - 1 をメインメモリ 1 2 - 3 - 6 に転送する。

## 【 0 0 9 2 】

次に処理ユニット( 1 2 - 3 - 2、1 2 - 3 - 3、1 2 - 3 - 4、1 2 - 3 - 5 )は s 1 4 - 4 - 1 6 で、転送した上記分割されたラストイメージデータ 1 0 2 - 1 - 1 のメインメモリ 1 2 - 3 - 6 上の位置を汎用プロセッサ 1 2 - 3 - 1 に通知する。

## 【 0 0 9 3 】

以上の s 1 4 - 4 - 1 0 から s 1 4 - 4 - 1 6 までの処理によって、処理ユニット( 1 2 - 3 - 2、1 2 - 3 - 3、1 2 - 3 - 4、1 2 - 3 - 5 )は未処理の分割された中間データ 1 0 1 - 1 から分割されたラストイメージデータ 1 0 2 - 1 - 1 をメインメモリ上に生成することが出来る。メインメモリには、分割された処理済みのラストイメージデータが分割前のように統合あるいは結合されて格納される。

## 【 0 0 9 4 】

即ち、汎用プロセッサ 1 2 - 3 - 1 と処理ユニット( 1 2 - 3 - 2、1 2 - 3 - 3、1 2 - 3 - 4、1 2 - 3 - 5 )は図 1 4 に示したように、以下のように処理する。未処理の分割された中間データ 1 0 1 - 1 がある限り以下の処理を繰り返す。

1 . 汎用プロセッサ 1 2 - 3 - 1 は、処理ユニット( 1 2 - 3 - 2、1 2 - 3 - 3、1 2 - 3 - 4、1 2 - 3 - 5 )に対して分割されたラストイメージデータ 1 0 2 - 1 の生成処理( s 1 4 - 4 - 1 0 から s 1 4 - 4 - 1 6 まで)を指示する。

2 . 処理ユニット( 1 2 - 3 - 2、1 2 - 3 - 3、1 2 - 3 - 4、1 2 - 3 - 5 )は、印刷エンジン 1 2 - 4 が画像形成 s 1 8 に必要なラストイメージデータ 1 0 2 - 1 を生成する。

## 【 0 0 9 5 】

次に、s 1 4 - 4 - 1 8 で汎用プロセッサ 1 2 - 3 - 1 は、印刷制御中間データ 1 0 1 - 2 をもとに印刷エンジン制御データ 1 0 2 - 2 を生成する。

## 【 0 0 9 6 】

以上、汎用プロセッサ 1 2 - 3 - 1 及び処理ユニット( 1 2 - 3 - 2、1 2 - 3 - 3、1 2 - 3 - 4、1 2 - 3 - 5 )は、s 1 4 の処理によって、中間データ 1 0 1 を元に印刷エンジン 1 2 - 4 が画像形成のために使う印刷データ 1 0 2 を生成する。この処理 s 1 4 は、印刷エンジンが s 1 6 で印刷データ 1 0 2 を得ることを可能ならしめる。

10

20

30

40

50

## 【0097】

以上説明してきたように、本実施例では、二つ以上の処理ユニット（12-3-2、12-3-3、12-3-4、12-3-5）を持つ画像生成処理ユニット（12-3-2、12-3-3、12-3-4、12-3-5）12-3が以下の処理を行う。

1．画像形成処理装置12-3が、分割されたベクタデータを含むデータを持つ中間データ101を得る。

2．汎用プロセッサ12-3-1が、この中間データ101中の属性情報101-1-3と、処理負荷パラメータ（図16参照）から処理負荷パラメータ（数式1参照）を算出する。

3．汎用プロセッサ12-3-1が、上記処理負荷パラメータとPDLの種類の情報をもとに、上記処理ユニット（12-3-2、12-3-3、12-3-4、12-3-5）が処理可能な処理の構成を決定するプログラムモジュールを決定する。

4．汎用プロセッサ12-3-1が、決定されたプログラムモジュールを処理ユニット（12-3-2、12-3-3、12-3-4、12-3-5）にロードする。こうすることで処理ユニット（12-3-2、12-3-3、12-3-4、12-3-5）が処理可能な構成を変更する。

5．処理ユニット（12-3-2、12-3-3、12-3-4、12-3-5）が上記分割されたベクタデータを含むデータ101-1を元にラストイメージデータ102-1を持つ印刷データ102を生成する。

## 【0098】

<本実施形態の効果>

以上の処理によって、処理の切り替えが複数回起こり、複数命令の切り替えることを回避可能たらしめる。即ち、分割されたベクタデータを含むデータのレンダリング処理及び画像処理において以下のような効果がある。すなわち、画像形成処理装置12-3は、あらかじめ処理負荷に応じて処理ユニット（12-3-2、12-3-3、12-3-4、12-3-5）が可能な処理の構成を決定した上でレンダリング処理及び画像処理を開始できる。そのため、同処理中に処理ユニット（12-3-2、12-3-3、12-3-4、12-3-5）構成を変更することなしに高速に処理可能である。

## 【0099】

なお、本実施形態では属性をテキスト、グラフィックス、テキストとグラフィックスの混合と分類したが、他の分類でも良い。たとえば、塗り潰し処理、ラストイメージ処理、その混合などと分類することもできる。

## 【0100】

なお本発明は、複数の機器（例えばホストコンピュータ、インタフェイス機器、リーダ、プリンタなど）から構成されるシステムに適用しても、一つの機器からなる装置（例えば、複写機、ファクシミリ装置など）に適用してもよい。また本発明の目的は、前述の実施形態の機能を実現するプログラムを記録した記録媒体を、システムあるいは装置に供給し、そのシステムあるいは装置のコンピュータが記憶媒体に格納されたプログラムを読み出し実行することによっても達成される。この場合、記憶媒体から読み出されたプログラム自体が前述した実施形態の機能を実現することになり、そのプログラム自体およびプログラムを記憶した記憶媒体は本発明を構成することになる。

## 【0101】

また、本発明には、プログラムの指示に基づき、コンピュータ上で稼働しているオペレーティングシステム(OS)などが実際の処理の一部または全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれる。さらに、記憶媒体から読み出されたプログラムが、コンピュータに挿入された機能拡張カードやコンピュータに接続された機能拡張ユニットに備わるメモリに書込まれた場合についても、本発明は適用される。その場合、書き込まれたプログラムの指示に基づき、その機能拡張カードや機能拡張ユニットに備わるCPUなどが実際の処理の一部または全部を行い、その処理によって前述した実施形態の機能が実現される。



## 【図面の簡単な説明】

【 0 1 0 2 】

【図 1】本発明の実施例における画像形成装置及び情報処理装置の構成図である。

【図 2】従来の画像形成装置及び情報処理装置の構成図である。

【図 3】本発明の実施例の処理フローを示す図である。

【図 4】PDL データを示す図である。

【図 5】印刷データを示す図である。

【図 6】本発明の実施例における画像形成処理装置の構成を示す図である。

【図 7】本発明の実施例における画像形成処理を示す図である。

【図 8】中間データを示す図である。

【図 9】本発明の実施例における中間データ例を示す図である。

【図 10】本発明の実施例における中間データと属性情報例を示す図である。

【図 11】画像処理装置の画像形成に係る処理を示す図である。

【図 12】処理装置にかかる命令を示す図である。

【図 13】処理装置への命令ロード処理を示す図である。

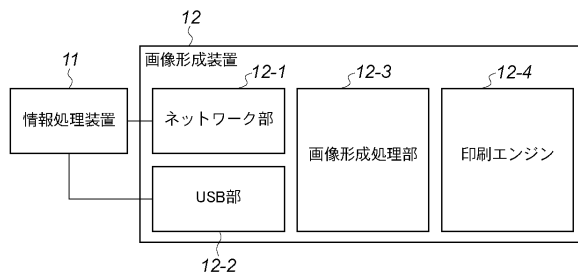
【図 14】本発明の実施例における命令群の選択処理を示す図である。

【図 15】本発明の実施例における印刷データ生成処理を示す図である。

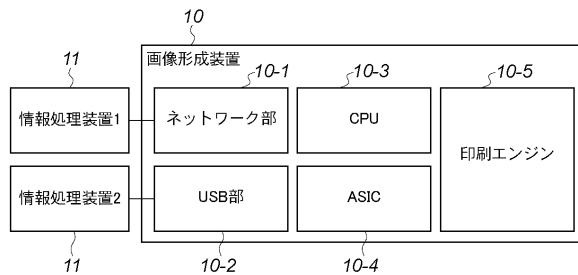
【図 16】本発明の実施例における重み付けパラメータのテーブルを示す図である。

10

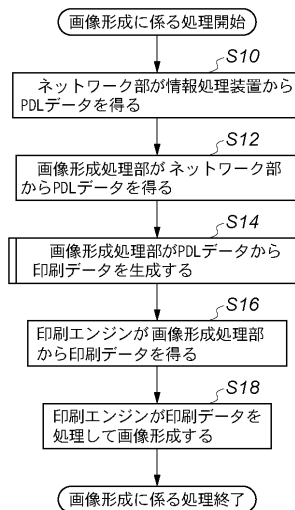
【図 1】



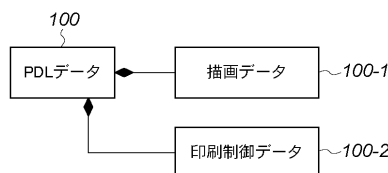
【図 2】



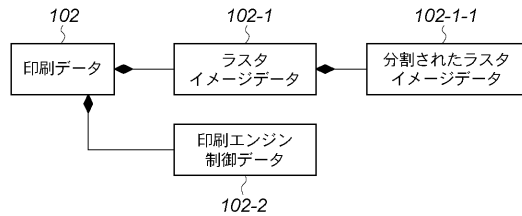
【図 3】



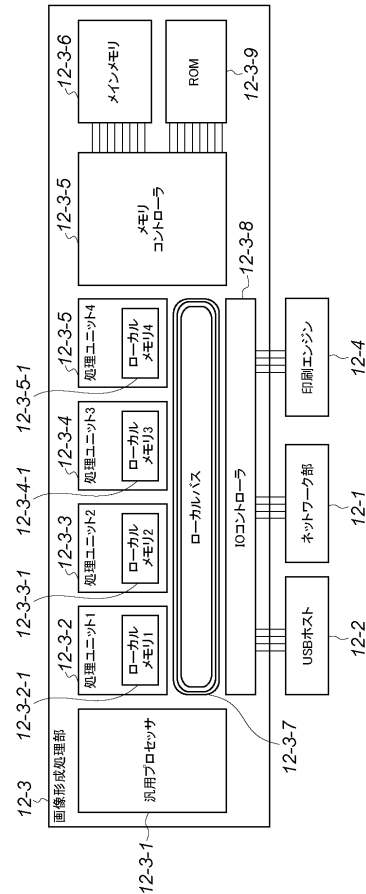
【図 4】



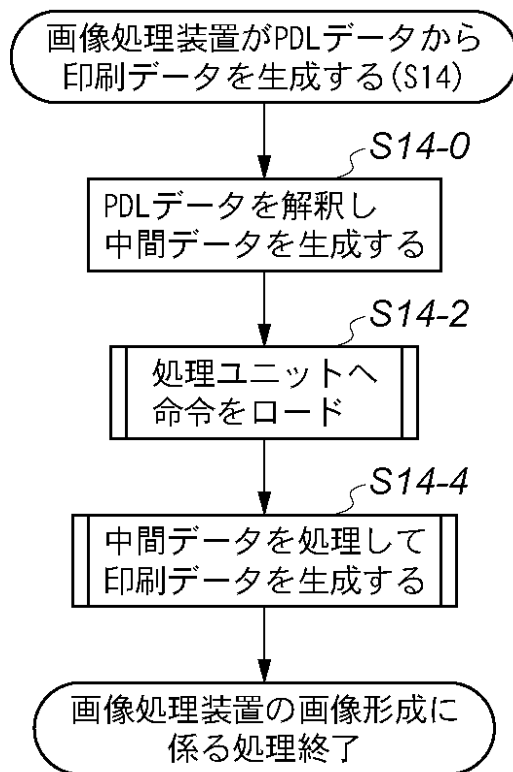
【図 5】



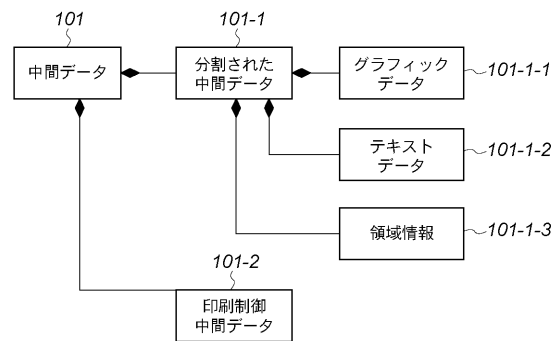
【図 6】



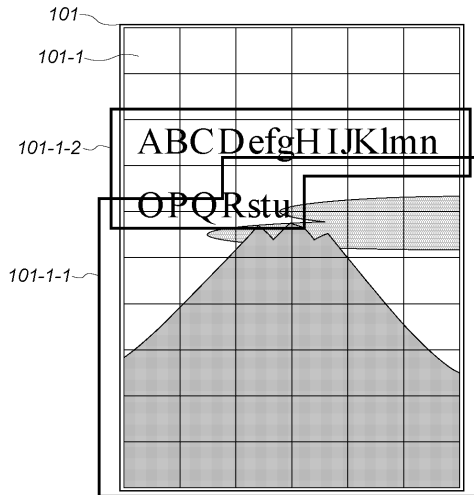
【図 7】



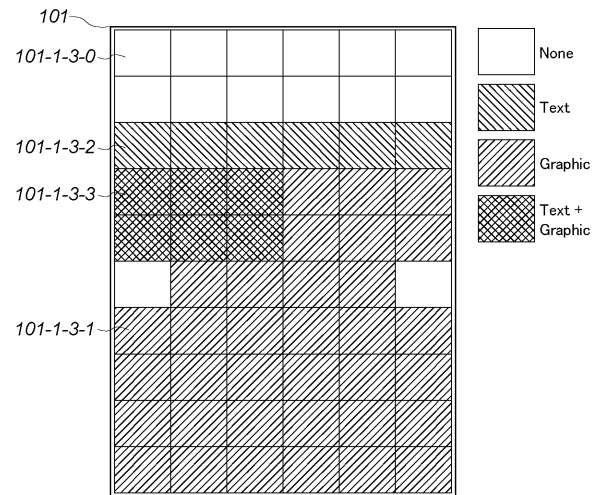
【図 8】



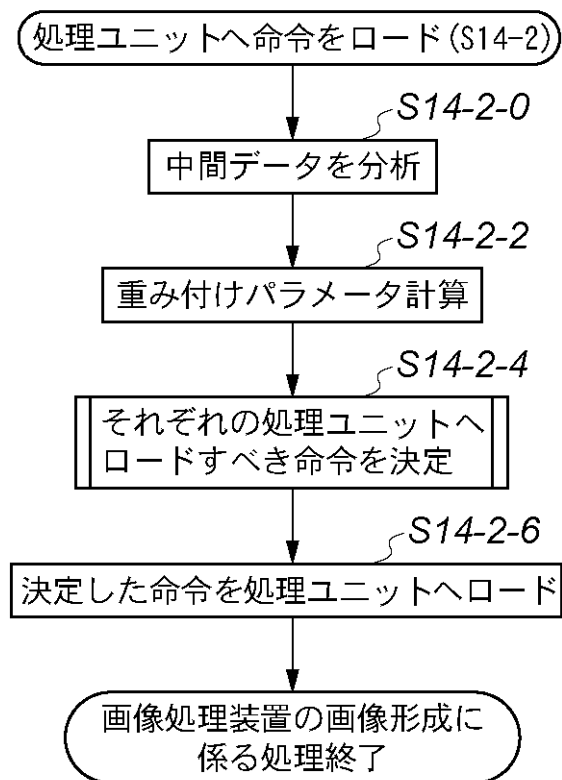
【図 9】



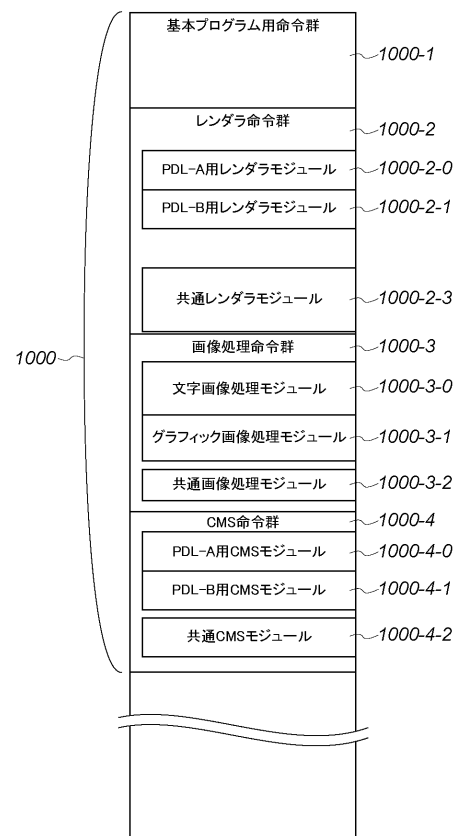
【図 10】



【図 11】



【図 12】





---

フロントページの続き

(72)発明者 武石 大樹  
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

審査官 立澤 正樹

(56)参考文献 特許第3125863(JP, B2)  
特開2007-081795(JP, A)

(58)調査した分野(Int.Cl., DB名)  
B41J 5/30  
B41J 29/38  
H04N 1/00