

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第3559618号

(P3559618)

(45) 発行日 平成16年9月2日(2004.9.2)

(24) 登録日 平成16年5月28日(2004.5.28)

(51) Int.Cl.⁷G06F 3/12
// H04N 1/00

F I

G06F 3/12 B
G06F 3/12 T
H04N 1/00 I O 7 A

請求項の数 10 (全 11 頁)

(21) 出願番号	特願平7-171106	(73) 特許権者	000001007
(22) 出願日	平成7年7月6日(1995.7.6)		キヤノン株式会社
(65) 公開番号	特開平9-22334		東京都大田区下丸子3丁目30番2号
(43) 公開日	平成9年1月21日(1997.1.21)	(74) 代理人	100076428
審査請求日	平成13年12月13日(2001.12.13)		弁理士 大塚 康德
		(74) 代理人	100093908
			弁理士 松本 研一
		(72) 発明者	相場 司
			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	近藤 聡
		(56) 参考文献	特開平07-334318(JP, A)
			特開平8-234937(JP, A)
			最終頁に続く

(54) 【発明の名称】 印刷システムとその制御方法及び印刷制御装置

(57) 【特許請求の範囲】

【請求項1】

プリンタに図形データを送信すると共に該図形データをビットマップ化するビットマップ化手段と、

プリンタより処理が高速か否かを判断する判断手段と、

前記判断手段によりプリンタより処理が高速であると判断された場合、前記ビットマップ化手段によりビットマップ化されたビットマップデータをプリンタに送信する手段とを有することを特徴とする印刷制御装置。

【請求項2】

図形データをプリンタで展開できるか調べる手段と、

図形データをプリンタで展開可能な場合、プリンタに図形データを送信すると共に該図形データをビットマップ化し、図形データをプリンタで展開可能でない場合、図形データをビットマップ化するビットマップ化手段と、

プリンタより処理が高速か否かを判断する判断手段と、

前記判断手段によりプリンタより処理が高速であると判断された場合、前記ビットマップ化手段によりビットマップ化されたビットマップデータをプリンタに送信する手段とを有することを特徴とする印刷制御装置。

【請求項3】

プリンタ状況を受信する手段をさらに有し、

前記判断手段は、受信した情報を基にプリンタより処理が高速か否かを判断することを特

10

20

徴とする請求項 1 または 2 記載の印刷制御装置。

【請求項 4】

プリンタに展開終了を送信する手段をさらに有することを特徴とする請求項 3 記載の印刷制御装置。

【請求項 5】

コンピュータとプリンタとを有する印刷システムであって、

前記コンピュータは、プリンタに図形データを送信すると共にコンピュータ内部で図形データをビットマップ化する第 1 ビットマップ化手段を有し、

前記プリンタは、コンピュータから受信した図形データをビットマップ化する第 2 ビットマップ化手段と、コンピュータより処理が高速でない場合、前記第 1 ビットマップ化手段でビットマップ化されたコンピュータから受信したデータの印刷を行い、コンピュータより処理が高速である場合、前記第 2 ビットマップ化手段によりビットマップ化されたデータの印刷を行なう手段とを有することを特徴とする印刷システム。

10

【請求項 6】

プリンタに図形データを送信すると共に該図形データをビットマップ化するビットマップ化ステップと、

プリンタより処理が高速か否かを判断する判断ステップと、

前記判断ステップによりプリンタより処理が高速であると判断された場合、前記ビットマップ化ステップによりビットマップ化されたビットマップデータをプリンタに送信するステップと

20

を有することを特徴とする印刷制御方法。

【請求項 7】

図形データをプリンタで展開できるか調べるステップと、

図形データをプリンタで展開可能な場合、プリンタに図形データを送信すると共に該図形データをビットマップ化し、図形データをプリンタで展開可能でない場合、図形データをビットマップ化するビットマップ化ステップと、

プリンタより処理が高速か否かを判断する判断ステップと、

前記判断ステップによりプリンタより処理が高速であると判断された場合、前記ビットマップ化ステップによりビットマップ化されたビットマップデータをプリンタに送信するステップと

30

を有することを特徴とする印刷制御方法。

【請求項 8】

プリンタ状況を受信するステップをさらに有し、

前記判断ステップは、受信した情報を基にプリンタより処理が高速か否かを判断することを特徴とする請求項 6 または 7 記載の印刷制御方法。

【請求項 9】

プリンタに展開終了を送信するステップをさらに有することを特徴とする請求項 8 記載の印刷制御方法。

【請求項 10】

コンピュータとプリンタとを有する印刷システムにおける印刷方法であって、

40

前記コンピュータにおいて、プリンタに図形データを送信すると共にコンピュータ内部で図形データをビットマップ化し、

前記プリンタにおいて、コンピュータより処理が高速でない場合、図形データに基づきコンピュータでビットマップ化されコンピュータから受信したデータの印刷を行い、コンピュータより処理が高速である場合、コンピュータから受信した図形データに基づきプリンタでビットマップ化されたデータの印刷を行なうことを特徴とする印刷方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、例えばホストコンピュータ装置のプリンタドライバなどからの印刷データを受

50

信して印刷を行う印刷システムとその制御方法及び印刷制御装置に関するものである。

【０００２】

【従来の技術】

従来、図形データを印刷する場合、ホストコンピュータ側に図形展開能力を持ち、その図形を展開したビットマップデータをプリンタへ送信している。プリンタ側ではその印刷データを受信し、その印刷データのまま印刷している。

【０００３】

また、ホストコンピュータ側のプリンタドライバにて、図形をプリンタが判別可能な図形描画データに置き換えて送信するといったことも行われている。プリンタ側ではその図形描画データを受信し、その図形描画データをプリンタに内蔵されている図形展開機能から、ビットマップに展開して印刷している。

10

【０００４】

図形展開機能は、ホストコンピュータ側には必ず用意されており、プリンタには持つ場合と持たないのどちらもある。図形展開機能はハードウェアもしくはソフトウェアにて用意されている。ホストコンピュータは一般的に、ユーザの作成したファイルを、ビットマップデータ以外、図形データとして保存し、それを表示する場合、その図形データを展開してビットマップ化し、表示させる。プリンタ側が図形展開機能を持つ場合、プリンタはホストコンピュータより図形データを受け取って展開を行い、ビットマップデータ化して印刷を行う。

【０００５】

20

【発明が解決しようとする課題】

ところが、従来例では図形データのビットマップ化が、ホストコンピュータまたはプリンタの性能に依存しているため、ホストコンピュータでの印刷開始からプリンタでの印刷終了までの時間がそれぞれの装置に依存し、特に速度が要求される場合でも印刷が速くならないという問題があった。

【０００６】

本発明は、この問題点を解決するためになされたもので、印刷開始から終了までの時間を短縮することを第１の目的とする。

【０００７】

又、ホストコンピュータとプリンタのそれぞれが図形展開能力を持つ場合、ホストコンピュータ側のディスプレイ表示と、プリンタから印刷される図形のデザインやレイアウトが違ふという問題があった。

30

【０００８】

本発明は、この問題点を解決するためになされたもので、必ずホストコンピュータ側のディスプレイ表示とプリンタからの印刷結果が一致させることを第２の目的とする。

【０００９】

又、従来ではホストコンピュータ側から送った図形データがプリンタ側で展開できないという場合があり、プリンタからユーザの指定とは異なった図形が出力されたり、印刷されないという問題があった。

【００１０】

40

本発明は、この問題点を解決するためになされたもので、プリンタ側にユーザが指定した図形の展開能力がない場合でも、指定した図形を印刷させることを第３の目的とする。

【００１１】

【課題を解決するための手段】

上記目的を達成する本発明は次のような構成から成る。すなわち、印刷制御装置であって、プリンタに図形データを送信すると共に該図形データをビットマップ化するビットマップ化手段と、プリンタより処理が高速か否かを判断する判断手段と、前記判断手段によりプリンタより処理が高速であると判断された場合、前記ビットマップ化手段によりビットマップ化されたビットマップデータをプリンタに送信する手段とを有する。

【００１２】

50

あるいは、印刷制御装置であって、図形データをプリンタで展開できるか調べる手段と、図形データをプリンタで展開可能な場合、プリンタに図形データを送信すると共に該図形データをビットマップ化し、図形データをプリンタで展開可能でない場合、図形データをビットマップ化するビットマップ化手段と、プリンタより処理が高速か否かを判断する判断手段と、前記判断手段によりプリンタより処理が高速であると判断された場合、前記ビットマップ化手段によりビットマップ化されたビットマップデータをプリンタに送信する手段とを有する。

【 0 0 1 3 】

また好ましくは、上記印刷制御装置は、プリンタ状況を受信する手段をさらに有し、前記判断手段は、受信したプリンタ状況を基にプリンタより処理が高速か否かを判断する。

10

【 0 0 1 4 】

また好ましくは、上記印刷制御装置はプリンタに展開終了を送信する手段をさらに有する。

【 0 0 1 5 】

あるいは、コンピュータとプリンタとを有する印刷システムであって、前記コンピュータは、プリンタに図形データを送信すると共にコンピュータ内部で図形データをビットマップ化する第1ビットマップ化手段を有し、前記プリンタは、コンピュータから受信した図形データをビットマップ化する第2ビットマップ化手段と、コンピュータより処理が高速でない場合、前記第1ビットマップ化手段でビットマップ化されたコンピュータから受信したデータの印刷を行い、コンピュータより処理が高速である場合、前記第2ビットマップ化手段によりビットマップ化されたデータの印刷を行なう手段とを有する。

20

【 0 0 1 6 】

【 発明の実施の形態 】

[実施形態 1]

次に、本発明の好適な実施形態について図面を参照して詳細に説明する。

【 0 0 1 7 】

図1は本発明の一実施形態であるホストコンピュータ装置とプリンタ装置のブロック図である。なお、本発明の機能が実行されるのであれば、単体の機器であっても、複数の機器からなるシステムであっても、LAN等のネットワークを介して処理が行われるシステムであっても本発明を適用出来ることは言うまでもない。

30

【 0 0 1 8 】

図1において、100はホストコンピュータ装置を示す。オペレーティングシステム（以下、OS）104がCPU101により所定のプログラム（後述するフローチャートに係るプログラムも含む）を実行することで動作して、ホストコンピュータ100を司る。OS104はシステムバス105を介して動作を実現し、更に接続されているディスプレイ111等の各種デバイスも制御する。ユーザが、RAM102や外部メモリ112に記憶されたアプリケーションソフトで、キーボード110を通して作成した文書ファイルは外部メモリ112に蓄えられる。文書ファイルは文字や図形によって構成されている。図形はイメージだけでなく、表計算等の罫線やドロー系アプリケーションソフトウェアで描かれる円や矩形等の図形データも含まれる。

40

【 0 0 1 9 】

ユーザがそのファイルを印刷する場合、RAM102にプリンタドライバがロードされていない場合は、外部メモリ112よりプリンタドライバを読み出し、RAM102に常駐させる。OS104によりこれらの制御を行い、更にRAM102上のプリンタドライバプログラムをCPU101により動作させる。プリンタドライバは印刷する文書ファイルを読み出し、前述した図形データが使用されている場合は、プリンタドライバが、プリンタ300にてこの図形を展開出来る／出来ないを判断し、それぞれの場合での印刷データを生成し、プリンタコントローラ109へ送り出す。

【 0 0 2 0 】

プリンタコントローラ109を経由した印刷データは、外部ケーブル200（双方向セン

50

トロ、シリアル、LAN等)を通して、プリンタ装置300へ送られる。プリンタ300内の入出力部206にて受信した印刷データは、一旦、RAM302に蓄えられる。蓄えられたデータは、プログラムROM304内のプログラム(後述するフローチャートに係るプログラムも含む)をCPU301が実行することで処理される。その処理におけるホストコンピュータ100に対する情報伝達は、入出力部306を介して行われる。印刷データ内に図形データがある場合は、プログラムROM304に格納されたプログラムにより、図形データは展開されてビットマップ化されて、エンジン部インタフェース307を介して、インクジェット方式や電子写真方式等のプリントエンジン部308に送られ、実際に印刷される。これらの処理は、CPU301が司り、システムバス305を介して行う。

10

【0021】

特に、図形について詳しく説明する。図形には、円、扇形、円弧、楕円、四分楕円、弓形、矩形、角丸矩形、折れ線(直線;線分を含む)、曲線(ベジェ等)、交互塗りつぶしの多角形、全て塗りつぶしの多角形等があり、それぞれの図形において、境界線と内部の塗りが定義可能である。線は、実線や点線及びパターン線等のスタイルを持ち、更に幅や色の定義が出来るものとする。塗りつぶしは、定型パターンやユーザ定義のビットマップ等のスタイルを持ち、指定されているハッチのスタイルや色の定義が出来るものとする。この様な方式で、ユーザ作成ファイルを生成するアプリケーションソフトウェアは既に数多く存在する。ファイル内格納データを図形として保持する理由は、展開されたビットマップデータより、メモリ効率が良く、また解像度に依存しないで出来るためである。

20

【0022】

ホストコンピュータ100側にはディスプレイ表示を行うため、必ずこの図形の展開能力を持つ。例えば、OS104やディスプレイコントローラ107がこの図形展開を行う。プリンタ装置300でも、この図形展開能力をもつものがある。例えば、PostScript(アドビシステムズ社)等のページ記述言語に対応したページプリンタである。図形展開は、プログラムROM304内のソフトウェア等をCPU302により実行することで、RAM302上で行われる。

【0023】

ファイル内の一部又は全てがビットマップデータの場合は、データをホストコンピュータ100内のプリンタドライバにてプリンタ命令に変換し(圧縮する場合もある)、プリンタ装置300に転送するため、基本的に図形展開機能を必要としない。

30

【0024】

ここで、印刷結果における文字デザインの一致を図るために、ホストコンピュータ装置100内での図形展開機能と、プリンタ装置300内での図形展開機能は同等とする。即ち、図形展開をしてビットマップ化したデータがホストコンピュータとプリンタとで同じ結果であるとする。

【0025】

以下、フローチャート等を用いて本発明の一実施形態について詳しく述べる。

【0026】

図2は、本発明を実施するホストコンピュータ100内のプリンタドライバの一連の手順を示すフローチャートである。このフローチャートは、RAM102に記憶されたプリンタドライバのプログラムをCPU101により実行することで実現される。

40

【0027】

ユーザがファイル印刷を開始すると、プリンタドライバは最初、ステップS2.1においてプリンタ300に対し印刷開始データを送信する。ステップS2.2でプリンタ内処理(図4で説明)をまった後、ステップS2.3で、プリンタ300よりプリンタ内の図形展開機能情報を受信する。ステップS2.4では、印刷データ内の図形データをプリンタ内で展開できるか調べ、プリンタが図形展開可能な場合は、ステップS2.5でプリンタへ図形データを送出する。但し、実際には、図形データだけではなく、その図形の大きさ、位置、修飾情報等も同時に送付する。この展開可能か否かの判定は、図形データがプリ

50

ンタで正しく扱える記述となっているかということに基づいて判定される。

【0028】

ステップS2.6でホストコンピュータ100内部の図形展開処理(図3で説明)を行う。ステップS2.6で図形のビットマップ化が終了した後、ステップS2.7でプリンタにビットマップ化終了を送信する。ステップS2.8ではプリンタ内処理(図5で説明)を待ち、ステップS2.9でプリンタ300よりプリンタ側処理状況を受信する。プリンタ300より受信した情報を基に、プリンタ側の処理具合をステップS2.10で判断する。プリンタ側が高速(図形のビットマップ化があるレベル以上終了している)の場合には、プリンタドライバの処理を終了する。ステップS2.10でホストコンピュータ側が高速と判断した場合には、ステップS2.12では、プリンタに対して、ステップS2.6でビットマップに展開された図形を送信する。

10

【0029】

一方、ステップS2.4にてプリンタ側に図形展開機能がないと判断した図形データの場合は、次にステップS2.11で図形をホストコンピュータ内部で展開してビットマップ化し(ステップS2.6と同じ処理)、ステップS2.12でプリンタへビットマップデータを送信する。

【0030】

ステップS2.10での判断において、プリンタへのビットマップデータの転送時間は考慮に入れることが可能である。この場合、プリンタよりもホストコンピュータにおける展開処理時間に、展開されたビットマップデータの転送に要する時間を加えて、いずれが高速か判定する。

20

【0031】

図3は、図2で説明した中での、プリンタドライバの印刷データ処理を示すフローチャートである。最初のステップS3.1でユーザが印刷したいファイルのデータを読み出す。その読み出したデータを解析し、ステップS3.2においてデータ中に使われている図形の展開が必要か否かを調べる。図形展開が必要である場合、ステップS3.3へ進み、展開のために必要なデータ等を取り出す。その後、ステップS3.4で図形の展開を行う。これは、プリンタドライバ内部で行っても、ホストコンピュータのOSやディスプレイコントローラに行なわせても構わない。図形の展開によりビットマップデータが得られる。ステップS3.2で展開が必要でないと判断された図形、例えばビットマップデータ等の場合は、ステップS3.5の印刷用データに変換する作業のみを行う。即ち、圧縮処理やプリンタへの描画命令に変換すること等である。

30

【0032】

以上の手順により、ホストコンピュータかプリンタのうち、図形を展開して迅速に印刷に供することができることができる方によって展開した図形データをプリンタにより印刷出力する。

【0033】

図4は、図2で説明した中での、ステップS2.1においてプリンタ側で行われる処理を示すフローチャートである。

【0034】

40

最初のステップS4.1は、ホストコンピュータがステップS2.1で送信した印刷開始データを受信することを示す。ステップS4.2は図形展開能力が存在する/しないの判断で、図形展開能力が存在する(図形の一部機能の展開能力があるものを含む)場合、ステップS4.3で図形展開能力存在の詳細(展開可能な図形を列挙、もしくは図形全てを展開可能等)をホストコンピュータへ送信し、ステップS4.4でホストコンピュータよりの図形描画データ等を受信する。そして、ステップS4.5においてプリンタ内部で図形展開を行い、図形描画データをビットマップ化させる。

【0035】

一方、ステップS4.2でプリンタに図形展開能力がないと判断した場合は、ホストコンピュータへ図形展開機能なしの旨を送信する。

50

【 0 0 3 6 】

なお、ステップ S 4 . 3 及び S 4 . 6 で送信したデータはステップ S 2 . 3 でホストが受信する。また、ステップ S 4 . 4 で受信するデータは、ホストがステップ S 2 . 5 で送信したデータである。

【 0 0 3 7 】

図 5 は、図 2 で説明した中での、ステップ S 2 . 8 の間にプリンタ側で行われる処理を示すフローチャートである。

【 0 0 3 8 】

最初のステップ S 5 . 1 では、ホストコンピュータがステップ S 2 . 7 で送信した図形展開終了を受信する。ステップ S 5 . 2 ではプリンタの図形展開状況と照らし合わせてプリンタ側が高速か否か判断する。この場合、ホストにおける図形の展開は終了しているため、この時点でプリンタにおいて行うべき処理に要する時間と、ホストコンピュータからビットマップデータの転送に要する時間とを比較する。転送時間の方が短いならばプリンタ側が遅いと判断し、ステップ S 5 . 3 でホストコンピュータへビットマップデータを要求する。その後、ステップ S 5 . 4 のホストコンピュータよりビットマップデータを受信し、ステップ S 5 . 5 で印刷を行う。

10

【 0 0 3 9 】

一方、ステップ S 5 . 2 にてプリンタ側が高速と判断した場合、例えばプリンタにおける展開処理が既に終了しているような場合には、ステップ S 5 . 6 でホストコンピュータへ印刷終了を送信し、ホストコンピュータ側の処理を終了させる。ステップ S 5 . 7 で、必要があれば展開処理を続けて図形を全てビットマップ化した後、そのデータをステップ S 5 . 5 にて印刷させる。

20

【 0 0 4 0 】

なお、ステップ S 5 . 3 及びステップ S 5 . 6 で送信したデータは、ステップ S 2 . 9 によりホストで受信される。また、ステップ S 5 . 4 で受信するデータは、ホストがステップ S 2 . 1 2 で送信したデータである。

【 0 0 4 1 】

このような処理により、ホストコンピュータとプリンタのいずれかに図形をビットマップデータへと展開する機能が備えられていれば、図形データを扱うことができ、また、両方にその機能がある場合には、より高速な方を展開に使用することができる。また、プリンタに図形展開機能がなくともホストコンピュータがそれを代替することができるため、ユーザが期待する画像を出力することができる。

30

【 0 0 4 2 】

〔 実施形態 2 〕

実施形態 1 では、図 2 のフローチャートにおいて、処理分散の考え方を、単純にホストコンピュータとプリンタどちらかで処理を分けていた。即ち、図形の展開（ビットマップ化）は、どちらかのみで行うことを示した。

【 0 0 4 3 】

本実施形態では、どちらか一方でビットマップ化するのではなく、ホストコンピュータとプリンタの両方で図形のビットマップ化を行う考え方を示す。

40

【 0 0 4 4 】

例えば、1 ページ内で、上半分がプリンタ担当、下半分はホストコンピュータ担当というように処理を分散する。処理分散を 5 0 % 対 5 0 とは限らず、双方の処理能力や処理状況によりその割合を変えても構わない。即ち、4 0 % 対 6 0 % や 8 0 % 対 2 0 % の様にある。この場合には、図 2 のステップ S 2 . 1 0 や図 5 のステップ S 5 . 2 の判定は不用であり、必ずホストで展開したビットマップデータをプリンタに転送するようにする。それを受信したプリンタは、自身で展開したビットマップデータと合わせて印刷出力する。

【 0 0 4 5 】

又、複数ページの印刷においては、ページ単位の処理分散でも可能である。例えば、1 ページ目をプリンタ側がビットマップ化し、2 ページ目をホストコンピュータ側でビットマ

50

ップ化するという方法である。この考え方から、奇数ページ対偶数ページや印刷する全ページ数の最初の半分対後半等という方法も可能である。

【 0 0 4 6 】

更に、プリンタがホストコンピュータ側で持っている図形機能の全てを持たない場合も考えられる。この場合において、プリンタ側で展開出来る図形だけをプリンタ展開とし、それ以外は全てホストコンピュータ展開とさせる方法も考えられる。

【 0 0 4 7 】

このようにすることで、ホストとプリンタの処理能力を生かすことができ、より高速な出力が可能となる。

【 0 0 4 8 】

10

【 実施形態 3 】

図 1 において、ホストコンピュータとプリンタの接続は、双方向通信と限定したが、片方向通信（例えばセントロニクスなど）の場合でも、ホストコンピュータ側の設定によって、処理分散割合をあらかじめ決めておく方法も可能である。この場合には、プリンタから応答はないため、予め決めておいた割合で図形を展開して 1 つの画像を形成する。

【 0 0 4 9 】

但し、片方向通信の場合には、プリンタ側の図形展開能力を、ホストコンピュータ側のプリンタドライバに予め設定させておく必要がある。

【 0 0 5 0 】

このようにすれば、片方向通信が用いられている場合でも、ホストコンピュータとプリンタ双方の処理能力をできるだけ生かした印刷処理が可能となる。

【 0 0 5 1 】

【 発明の効果 】

以上説明したように、本発明の印刷システムとその制御方法及び印刷制御装置は、印刷開始から終了までの時間を短縮することができる効果を奏する。

【 0 0 5 2 】

又、プリンタに図形展開の能力がなくともユーザが期待する出力結果を得ることができるという効果を奏する。

【 0 0 5 3 】

【 図面の簡単な説明 】

30

【 図 1 】 実施形態を示すホストコンピュータ装置とプリンタ装置の構成を説明するブロック図である。

【 図 2 】 ホストコンピュータにおける一連のプリンタドライバ制御手順を示すフローチャートである。

【 図 3 】 ホストコンピュータにおける図形データ処理手順を示すフローチャートである。

【 図 4 】 プリンタにおける印刷開始時の処理手順を示すフローチャートである。

【 図 5 】 プリンタにおけるプリンタ内部処理の手順を示すフローチャートである。

【 符号の説明 】

1 0 0 ホストコンピュータ装置

1 0 1 C P U

40

1 0 2 R A M

1 0 3 フォントメモリ

1 0 4 O S

1 0 5 システムバス

1 0 6 キーボードコントローラ

1 0 7 ディスプレイコントローラ

1 0 8 ディスクコントローラ

1 0 9 プリンタコントローラ

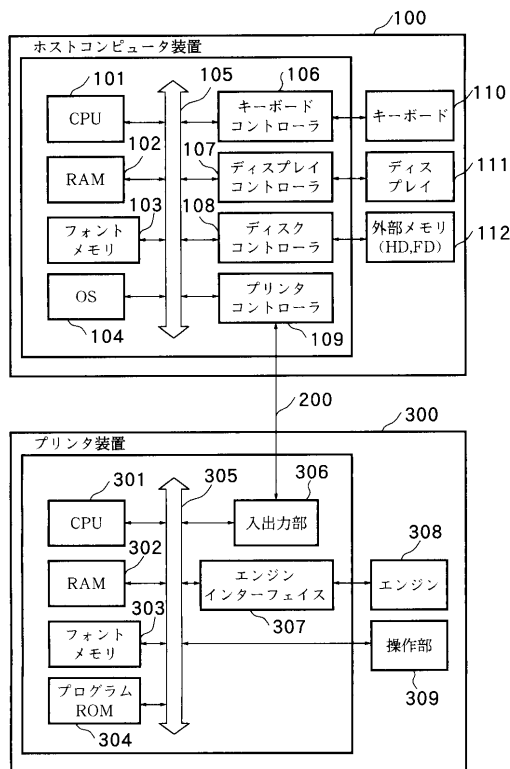
1 1 0 キーボード

1 1 1 ディスプレイ

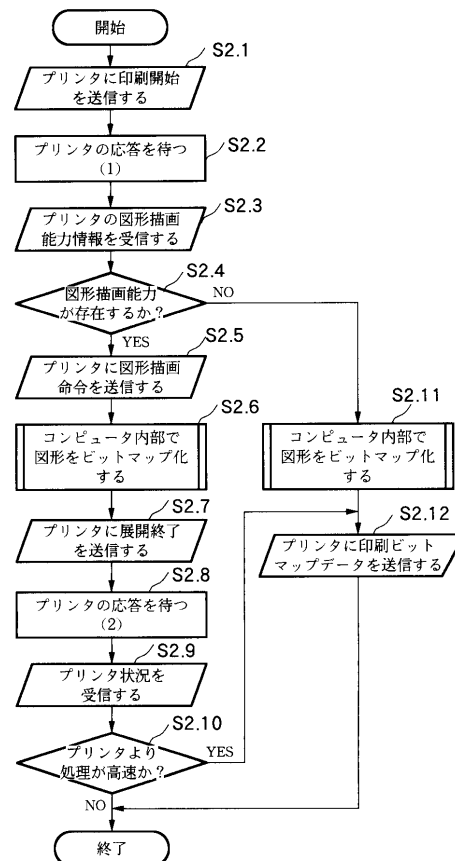
50

- 1 1 2 メモリ (H D , F D)
- 2 0 0 双方向インタフェース
- 3 0 0 プリンタ装置
- 3 0 1 C P U
- 3 0 2 R A M
- 3 0 3 フォントメモリ
- 3 0 4 プログラム R O M
- 3 0 5 システムバス
- 3 0 6 入出力部
- 3 0 7 エンジンインタフェース
- 3 0 8 エンジン
- 3 0 9 操作部

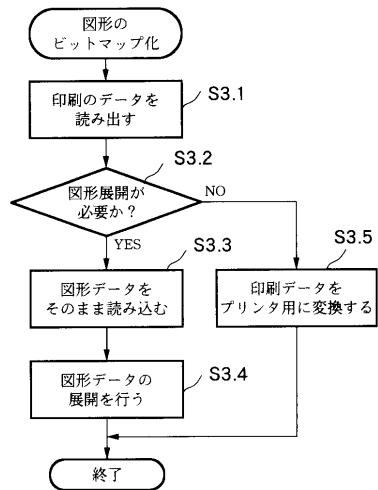
【 図 1 】



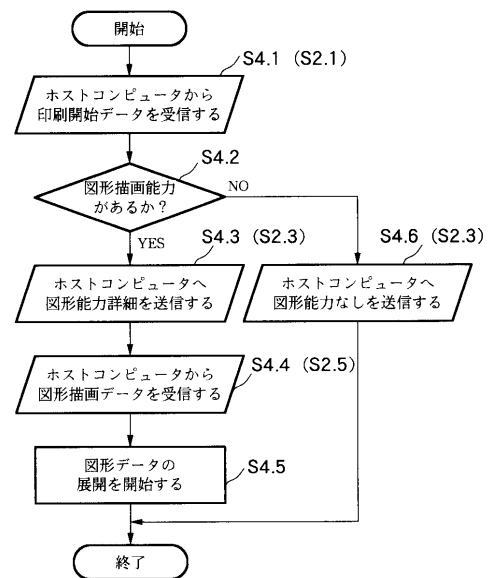
【 図 2 】



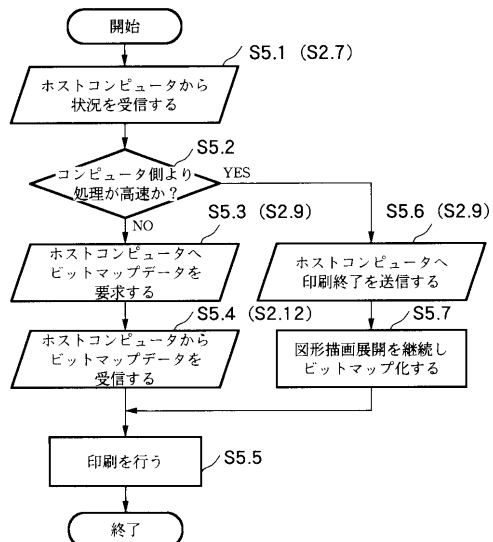
【図 3】



【図 4】



【図 5】



フロントページの続き

(58)調査した分野(Int.Cl.⁷, D B 名)

G06F 3/12

H04N 1/00

B41J 29/00