

(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2007-240848

(P2007-240848A)

(43) 公開日 平成19年9月20日(2007.9.20)

(51) Int. Cl.

G09G 5/24 (2006.01)

F I

G09G 5/24 620L

テーマコード(参考)

5C082

審査請求 有 請求項の数 4 O L (全 12 頁)

(21) 出願番号 特願2006-62747(P2006-62747)
 (22) 出願日 平成18年3月8日(2006.3.8)

(71) 出願人 000002369
 セイコーエプソン株式会社
 東京都新宿区西新宿2丁目4番1号
 (74) 代理人 100066980
 弁理士 森 哲也
 (74) 代理人 100075579
 弁理士 内藤 嘉昭
 (74) 代理人 100103850
 弁理士 崔 秀▲てつ▼
 (72) 発明者 澤崎 高
 長野県諏訪市大和3丁目3番5号 セイコーエプソン株式会社内
 Fターム(参考) 5C082 AA01 BA02 BB34 DA11 DA31 DA42 MM02

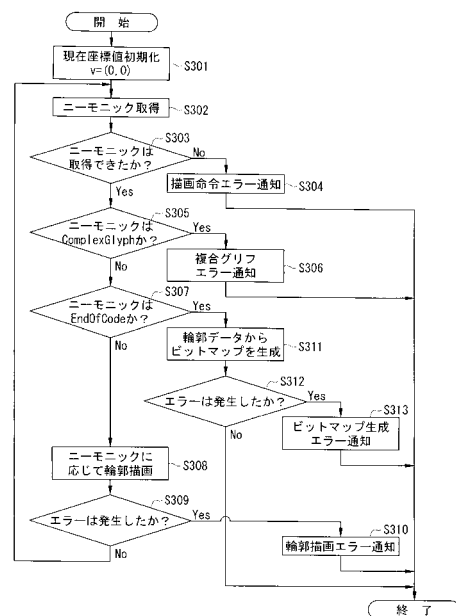
(54) 【発明の名称】 表示プログラム、データ構造及び表示装置

(57) 【要約】

【課題】 描画処理の処理速度を向上すること。

【解決手段】 取得した命令が、ComplexGlyphかそれ以外の命令(EndOfCode等)かを判定し(ステップS305、S307、S308)、ComplexGlyphであると判定された場合には、そのComplexGlyphに続いて取得されるグリフィンデックス、アフィンパラメータ、及びオフセット(パーツグリフに関する情報)に基づいてパーツグリフそれぞれを描画するようにした(ステップS306)。そのため、例えば、取得される全ての情報にパーツグリフに関する情報であるか否かを示すフラグを付加しておき、それら情報を取得するたびに前記フラグを読み出して当該情報がパーツグリフに関する情報であるか否かを判定する方法と異なり、パーツグリフに関する情報であるか否かを全ての情報について判定する必要がなく、処理速度を向上できる。

【選択図】 図6



【特許請求の範囲】**【請求項 1】**

文字を構成する各要素それぞれを描画するための命令を取得し、その命令に基づいて前記要素を描画して前記文字を表示する表示プログラムであって、

取得した命令が、前記文字のグリフとして複数のパーツグリフの組み合わせからなる複合グリフを用いることを示す複合グリフ命令かそれ以外の命令かを判定する判定機能と、前記複合グリフ命令であると判定された場合には、その複合グリフ命令に続いて取得される前記パーツグリフに関する情報に基づいて前記パーツグリフそれぞれを描画する描画機能と、をコンピュータに実行させることを特徴とする表示プログラム。

【請求項 2】

前記パーツグリフに関する情報は、前記パーツグリフのグリフインデックスを含み、前記描画機能は、前記パーツグリフのグリフインデックスが示す命令を取得するパーツグリフ取得機能と、その取得された命令に基づいて前記パーツグリフを描画する第 1 のパーツグリフ描画機能と、を備えたことを特徴とする請求項 1 に記載の表示プログラム。

【請求項 3】

文字を構成する各要素それぞれを描画するための命令を取得し、その命令に基づいて前記要素を描画して前記文字を表示する表示プログラムで用いられるデータ構造であって、前記文字のグリフとして複数のパーツグリフの組み合わせからなる複合グリフを用いることを示す複合グリフ命令を、そのパーツグリフに関する情報の前に配したことを特徴とするデータ構造。

【請求項 4】

文字を構成する各要素それぞれを描画するための命令を取得し、その命令に基づいて前記要素を描画して前記文字を表示する表示装置であって、

取得した命令が、前記文字のグリフとして複数のパーツグリフの組み合わせからなる複合グリフを用いることを示す複合グリフ命令かそれ以外の命令かを判定する判定手段と、前記複合グリフ命令であると判定された場合には、その複合グリフ命令に続いて取得される前記パーツグリフに関する情報に基づいて前記パーツグリフそれぞれを描画する描画手段と、を備えたことを特徴とする表示装置。

【発明の詳細な説明】**【技術分野】****【0001】**

本発明は、文字を構成する各要素それぞれを描画するための複数の情報を取得し、それら情報に基づいて前記要素を描画して前記文字を表示する表示プログラム、データ構造及び表示装置に関する。

【背景技術】**【0002】**

従来、この種の表示装置としては、例えば、文字を構成する各要素それぞれを描画するための複数の命令やグリフに関する情報等、各種情報を取得し、その情報に基づいて前記要素を描画して前記文字を表示するものがある（例えば、特許文献 1 参照）。

このような表示装置にあつては、一般に、取得される全ての情報にパーツグリフ（複合グリフを構成するグリフ）に関する情報であるか否かを示すフラグを付加しておき、それら情報を取得するたびに前記フラグを読み出して当該情報がパーツグリフに関する情報であるか否かを判定し、パーツグリフに関する情報である場合には、その情報に基づいてパーツグリフそれぞれを描画して前記複合グリフを描画するようになっている。

【特許文献 1】特開平 7 - 3 2 5 5 6 2 号公報

【発明の開示】**【発明が解決しようとする課題】****【0003】**

しかしながら、上記従来技術にあつては、全情報についてフラグの読み出しと判定とを行うようになっているため、処理量が増大し、処理速度が低下する原因となっていた。

10

20

30

40

50

本発明は、上記従来技術の未解決の課題を解決することを目的とするものであって、描画処理の処理速度を向上可能な表示プログラム、データ構造及び表示装置を提供することを課題とする。

【課題を解決するための手段】

【0004】

上記課題を解決するために、本発明の表示装置は、文字を構成する各要素それぞれを描画するための命令を取得し、その命令に基づいて前記要素を描画して前記文字を表示する表示プログラムであって、取得した命令が、前記文字のグリフとして複数のパーツグリフの組み合わせからなる複合グリフを用いることを示す複合グリフ命令かそれ以外の命令かを判定する判定機能と、前記複合グリフ命令であると判定された場合には、その複合グリフ命令に続いて取得される前記パーツグリフに関する情報に基づいて前記パーツグリフそれぞれを描画する描画機能と、をコンピュータに実行させることを特徴とする。

10

【0005】

また、前記パーツグリフに関する情報は、前記パーツグリフのグリフインデックスを含み、前記描画機能は、前記パーツグリフのグリフインデックスが示す命令を取得するパーツグリフ取得機能と、その取得された命令に基づいて前記パーツグリフを描画する第1のパーツグリフ描画機能と、を備えるようにしてもよい。

さらに、本発明のデータ構造は、文字を構成する各要素それぞれを描画するための命令を取得し、その命令に基づいて前記要素を描画して前記文字を表示する表示プログラムで用いられるデータ構造であって、前記文字のグリフとして複数のパーツグリフの組み合わせからなる複合グリフを用いることを示す複合グリフ命令を、そのパーツグリフに関する情報の前に配したことを特徴とする。

20

【0006】

また、本発明の表示装置は、文字を構成する各要素それぞれを描画するための命令を取得し、その命令に基づいて前記要素を描画して前記文字を表示する表示装置であって、取得した命令が、前記文字のグリフとして複数のパーツグリフの組み合わせからなる複合グリフを用いることを示す複合グリフ命令かそれ以外の命令かを判定する判定手段と、前記複合グリフ命令であると判定された場合には、その複合グリフ命令に続いて取得される前記パーツグリフに関する情報に基づいて前記パーツグリフそれぞれを描画する描画手段と、を備えたことを特徴とする。

30

【0007】

このような構成によれば、例えば、取得される全ての情報にパーツグリフに関する情報であるか否かを示すフラグを付加しておき、それら情報を取得するたびに前記フラグを読み出して当該情報がパーツグリフに関する情報であるか否かを判定する方法と異なり、全ての情報についてフラグの読み出しと判定とを行う必要はなく、構成を簡単なものすることができ、その結果、処理速度を向上することができる。

【発明を実施するための最良の形態】

【0008】

以下、本発明の表示装置の一実施形態を図面に基づいて説明する。

<表示装置の構成>

40

図1は、本実施形態の表示装置の内部構成を示すブロック図である。この図1に示すように、表示装置1は、入力手段2、入力手段コントローラ3、メモリ4、メモリコントローラ5、外部記憶6、外部記憶コントローラ7、CPU8、表示コントローラ9、表示体10、バスコントローラ11、及び電源コントローラ12を含んで構成される。

【0009】

入力手段2は、マウスやキーボード等、文字の描画を指示するときに操作させる機器である。そして、入力手段2は、文字の描画を指示するための操作が行われると、文字の描画指示を入力手段コントローラ3を介してCPU8に出力する。

入力手段コントローラ3は、入力手段2とCPU8との間のデータ転送を制御する。

メモリ4は、CPU8がプログラムを実行するときに各種プログラムを展開するワークエ

50

リアを形成するとともに、CPU 8により実行される各種プログラムに係るデータや、表示対象である情報のデータを記憶する記憶領域を形成する。

【0010】

メモリコントローラ5は、メモリ4とCPU 8との間のデータ転送を制御する。

外部記憶6は、CPU 8により実行される基本制御プログラムやアプリケーションプログラム等の各種プログラム及びこれら各種プログラムに係るデータを格納する。

外部記憶コントローラ7は、外部記憶6とCPU 8との間のデータ転送を制御する。

CPU 8は、外部記憶6に格納されている各種プログラムを読み込み、それらをメモリ4内に形成されるワークエリアに展開して各部2～12を制御する。

【0011】

具体的には、CPU 8は、入力手段2から文字の描画指示が出力されると、ビットマップ生成処理を実行する。そして、ビットマップ生成処理によって、その文字の文字コードに基づいて描画命令データ(前記文字を構成する各要素それぞれを描画するための複数のニーモニックからなるデータ、所定の順番でニーモニックが並べられたデータ)を取得し、その描画命令データからニーモニックを前記所定の順序に従って1つずつ取得し、そのニーモニックに基づいて、前記描画指示があった文字のビットマップを生成する。

10

【0012】

また、CPU 8は、複合グリフ(複数のグリフ(パーツグリフ)の組み合わせからなるグリフ)を示すニーモニック(ComplexGlyph)が取得された場合には、そのComplexGlyphの次に配されている前記パーツグリフのグリフインデックス、アフィンパラメータ、及びオフセットに基づいて前記パーツグリフそれぞれの輪郭データを生成し、それら輪郭データに基づいて、前記複合グリフで表される文字のビットマップを生成する。

20

【0013】

すなわち、描画命令データを構成するニーモニックは、ComplexGlyphの後に、そのComplexGlyphが示す複合グリフのパーツグリフに関する情報が配されている。

ここで、ComplexGlyph以外のニーモニックとしては、図2に示すように、MoveTo_S、MoveTo_M、MoveTo_L(処理対象の座標移動コード)、HorLineTo_S、HorLineTo_M、HorLineTo_L(水平線描画コード)、VerLineTo_S、VerLineTo_M、VerLineTo_L(垂直線描画コード)、LineTo_S、LineTo_M、LineTo_L(任意の直線描画)、ConicCuraveTo_S、ConicCuraveTo_M、ConicCuraveTo_L(2次ベジエ曲線描画コード)、CubicCuraveTo_S、CubicCuraveTo_M、及びCubicCuraveTo_L(3次ベジエ曲線描画コード)が挙げられる。

30

【0014】

また、ComplexGlyphは、複合グリフを構成するパーツグリフのグリフインデックス、アフィンパラメータ、及びオフセットが所定の個数続くことを示すコードである。

さらに、EndOfCodeは、描画命令データを構成する最後のニーモニックであることを示すコードである。

表示コントローラ9は、CPU 8でビットマップが生成されると、そのビットマップを表示体10に描画する。

【0015】

表示体10は、電力供給が停止されても表示内容を保持できる記憶性表示体(コレステリック液晶)を含んで構成される。そして、表示体10は、表示コントローラ9の描画動作によって表示内容が書き換えられる。

40

バスコントローラ11は、CPU 8と各部2～12との間のデータ転送を制御する。

電源コントローラ12は、CPU 8からの指令に従って、表示装置1の各部2～12への電源供給を制御する。

【0016】

< CPUの機能構成 >

図3は、CPU 8の機能構成を示すブロック図である。この図3に示すように、CPU 8は、上位関数13、文字列描画関数14、文字座標計算関数15、文字コード変換関数16、文字データ取得関数17、及び文字描画回路18を含んで構成される。

50

上位関数 1 3 は、入力手段 2 から文字列の描画指示が出力されると、その文字列の描画を要求する文字列描画要求を文字列描画関数 1 4 に出力する。

【 0 0 1 7 】

文字列描画関数 1 4 は、上位関数 1 3 から文字列描画要求が出力されると、その文字列描画要求に対応する文字列の文字コード群を文字座標計算関数 1 5 に出力する。

また、文字列描画関数 1 4 は、前記文字コード群の出力に対応して文字座標計算関数 1 5 から座標列が出力されると、その座標列を構成する座標と当該座標に対応する文字コードとを組み合わせると、1 つずつ文字コード変換関数 1 6 に出力する。

【 0 0 1 8 】

さらに、文字列描画関数 1 4 は、文字コード変換関数 1 6 から終了ステータスが出力されると、上位関数 1 4 に終了ステータスを出力する。

文字座標計算関数 1 5 は、文字列描画関数 1 4 から出力される文字コード群に含まれる文字コードそれぞれの描画位置の座標を算出し、その算出結果から構成される座標列を文字列描画関数 1 4 に出力する。

【 0 0 1 9 】

文字コード変換関数 1 6 は、文字列描画関数 1 4 から文字コードと座標との組が出力されると、その文字コードをグリフィンデックスに変換し、その変換結果と前記座標とを組み合わせると、文字データ取得関数 1 7 に出力する。

また、文字コード変換関数 1 6 は、文字データ取得関数 1 7 から終了ステータスが出力されると、文字列描画関数 1 4 に終了ステータスを出力する。

【 0 0 2 0 】

文字データ取得関数 1 7 は、文字コード変換関数 1 6 からグリフィンデックスと座標との組が出力されると、そのグリフィンデックスに基づいて描画命令データを取得し、その取得結果（グリフデータ）と前記座標とを組み合わせると、文字描画回路 1 8 に出力する。

また、文字データ取得関数 1 7 は、文字描画回路 1 8 から終了ステータスが出力されると、文字コード変換関数 1 6 に終了ステータスを出力する。

【 0 0 2 1 】

さらに、文字データ取得関数 1 7 は、文字描画回路 1 8 から出力された終了ステータスが複合グリフ用異常終了ステータス（ComplexGlyphを取得したことによって異常終了したことを示す終了ステータス）である場合には、そのComplexGlyphの次に配されているパーツグリフのグリフィンデックス、アフィンパラメータ、及びオフセットに基づいて前記パーツグリフのグリフデータを所定の順番に従って1 つずつ取得し、そのグリフデータと当該グリフデータに対応する座標とを文字描画回路に出力する。

【 0 0 2 2 】

文字描画回路 1 8 は、文字データ取得関数 1 7 からグリフデータと座標との組が出力されると、そのグリフデータからニーモニックを所定の順番に従って1 つずつ取得し、そのニーモニックに基づいて、当該グリフデータに対応する文字のビットマップを生成する。

また、文字描画回路 1 8 は、ビットマップを生成した後に、終了ステータスを文字データ取得関数 1 7 に出力する。なお、ComplexGlyphが取得された場合、ビットマップを生成せず、複合グリフ用異常終了ステータスを文字データ取得関数 1 7 に出力する。

【 0 0 2 3 】

< CPU の動作 >

次に、CPU 8 で実行されるビットマップ生成処理を図 4 のフローチャートに基づいて説明する。

このビットマップ生成処理は、文字の描画指示があると実行される処理であって、まず、そのステップ S 1 0 1 で、前記描画指示された文字の文字コードを取得し、その文字コードをフォントデータのグリフィンデックスに変換する。

次にステップ S 1 0 2 に移行して、前記ステップ S 1 0 1 で変換されたグリフィンデックスに基づいて、前記描画指示があった文字を表示体 1 0 に描画する第 1 の文字描画処理（後述）を実行してから、この演算処理を終了する。

10

20

30

40

50

【0024】

次に、前記ステップS102で実行される文字描画処理を図5のフローチャートに基づいて説明する。

このビットマップ生成処理は、まず、そのステップS201で、前記ステップS101で変換されたグリフィンデックスに基づいて描画命令データを取得する。

次にステップS202に移行して、前記ステップS201で取得された描画命令データにアフィン変換を適用する。

【0025】

次にステップS203に移行して、前記ステップS202でアフィン変換された描画命令データに基づいてビットマップを生成する第2の文字描画処理（後述）を実行する。

次にステップS204に移行して、前記ステップS203でエラーが発生したか否か、又は前記ステップS203はパーツグリフに関する処理であったか否かを判定する。そして、エラーが発生した場合又はパーツグリフに関する処理であった場合には（Yes）ステップS205に移行し、エラーが発生していない場合又はパーツグリフに関する処理ではない場合には（No）この演算処理を終了する。

【0026】

前記ステップS205では、前記ステップS204で発生したと判定されたエラーがComplexGlyphによるものか否かを判定する。そして、ComplexGlyphによるものである場合には（Yes）ステップS206に移行し、ComplexGlyphによるものでない場合には（No）ステップS209に移行する。

前記ステップS206では、後述するステップS302で取得されたComplexGlyphの次に配されているパーツグリフのグリフィンデックス、アフィンパラメータ、及びオフセットを取得する。

【0027】

前記ステップS207では、第1の文字描画処理を再帰的に実行する。

次にステップS208に移行して、前記ステップS206で取得されたグリフィンデックス等の次にパーツグリフのグリフィンデックス、アフィンパラメータ、及びオフセットがあるか否かを判定する。そして、次のパーツグリフのグリフィンデックス等がある場合には（Yes）前記ステップS206に移行し、次のパーツグリフのグリフィンデックス等がない場合には（No）この演算処理を終了する。

【0028】

一方、前記ステップS209では、前記ステップS204で発生したと判定されたエラーのエラー原因に応じた処理を実行する。

次にステップS210に移行して、エラーが発生したことを示すエラーフラグを解除状態としてから、この演算処理を終了する。

次に、前記ステップS203で実行される第2の描画処理を図6のフローチャートに基づいて説明する。

【0029】

この第2の描画処理は、まず、そのステップS301で、処理対象である画素を示す座標値（現在座標値） v を（0、0）とする。

次にステップS302に移行して、前記ステップS202でアフィン変換された描画命令データからニーモニックを所定の順番に従って1つ取得する。

次にステップS303に移行して、前記ステップS302でニーモニックを取得できたか否かを判定する。そして、ニーモニックを取得できた場合には（Yes）ステップS305に移行し、ニーモニックを取得できなかった場合には（No）ステップS304に移行する。

【0030】

前記ステップS304では、ニーモニックを取得できなかったことを通知する描画命令エラー通知を行ってから、この演算処理を終了する。

一方、前記ステップS305では、前記ステップS302で取得されたニーモニックが

10

20

30

40

50

ComplexGlyphであるか否かを判定する。そして、ComplexGlyphである場合には (Y e s) ステップ S 3 0 6 に移行し、ComplexGlyphでない場合には (N o) ステップ S 3 0 7 に移行する。

【 0 0 3 1 】

前記ステップ S 3 0 6 では、ComplexGlyphが取得されたことを報知する複合グリフエラー通知を行ってから、この演算処理を終了する。

一方、前記ステップ S 3 0 7 では、前記ステップ S 3 0 2 で取得されたニーモニックが EndOfCodeであるか否かを判定する。そして、EndOfCodeである場合には (Y e s) ステップ S 3 1 1 に移行し、EndOfCodeでない場合には (N o) ステップ S 3 0 8 に移行する。

【 0 0 3 2 】

前記ステップ S 3 0 8 では、前記ステップ S 3 0 2 で取得されたニーモニックに基づいて、前記描画指示があった文字の輪郭データを生成する。

次にステップ S 3 0 9 に移行して、前記ステップ S 3 0 8 でエラーが発生したか否かを判定する。そして、エラーが発生した場合には (Y e s) ステップ S 3 1 0 に移行し、エラーが発生していない場合には (N o) 前記ステップ S 3 0 2 に移行する。

前記ステップ S 3 1 0 では、輪郭データが生成できなかったことを報知する輪郭描画エラー通知を行ってから、この演算処理を終了する。

【 0 0 3 3 】

一方、前記ステップ S 3 1 1 では、前記ステップ S 3 0 8 で生成された輪郭データに基づいて、前記描画指示があった文字のビットマップを生成する。

次にステップ S 3 1 2 に移行して、前記ステップ S 3 1 1 でエラーが発生したか否かを判定する。そして、エラーが発生した場合には (Y e s) ステップ S 3 1 3 に移行し、エラーが発生していない場合には (N o) この演算処理を終了する。

前記ステップ S 3 1 3 では、ビットマップが生成できなかったことを報知するビットマップ生成エラー通知を行ってから、この演算処理を終了する。

【 0 0 3 4 】

< 表示装置の動作 >

次に、本実施形態の表示装置 1 の動作を具体的状況に基づいて説明する。

まず、利用者が文字の描画を指示する操作を行い、入力手段 2 によって、文字の描画指示が CPU 8 に出力されたとする。すると、CPU 8 によって、ビットマップ生成処理が実行され、図 4 に示すように、まず、そのステップ S 1 0 1 で、前記描画指示された文字の文字コードが取得され、その文字コードがフォントデータのグリフィンデックスに変換され、ステップ S 1 0 2 で、そのグリフィンデックスに基づいて第 1 の文字描画処理が実行される。

【 0 0 3 5 】

第 1 の文字描画処理が実行されると、図 5 に示すように、まず、そのステップ S 2 0 1 で、前記グリフィンデックスに基づいて描画命令データが取得され、ステップ S 2 0 2 で、その描画命令データにアフィン変換が適用され、ステップ S 2 0 3 で、そのアフィン変換された描画命令データに基づいて第 2 の描画処理が実行される。

【 0 0 3 6 】

第 2 の描画処理が実行されると、図 6 に示すように、まず、そのステップ S 3 0 1 で、処理対象である画素を示す座標値 (現在座標値) v が (0 , 0) とされ、ステップ S 3 0 2 で、前記アフィン変換された描画命令データからニーモニックが所定の順番に従って 1 つ取得され、そのニーモニックが ComplexGlyphであった場合、ステップ S 3 0 3 及び S 3 0 4 の判定が「 Y e s 」となり、ステップ S 3 0 6 で、複合グリフエラー通知が行われ、この演算処理を終了する。

【 0 0 3 7 】

そして、図 5 に示すように、ステップ S 2 0 4 及び S 2 0 5 の判定が「 Y e s 」となり、ステップ S 2 0 6 で、図 7 に示すように、複合グリフを構成するパーツグリフのグリフィンデックス (3 3)、アフィンパラメータ、及びオフセットが 1 組取得され、ステップ

10

20

30

40

50

S 2 0 7 で、そのグリフィンデックス等に基づいて、前記パーツグリフ (A) のビットマップを生成する第 1 の文字描画処理が再帰的に実行される。

【 0 0 3 8 】

第 1 の文字描画処理が実行されると、図 5 に示すように、前記ステップ S 2 0 1 で、前記グリフィンデックス等に基づいて描画命令データが取得され、前記ステップ S 2 0 2 で、その描画命令データにアフィン変換が適用され、前記ステップ S 2 0 3 で、そのアフィン変換された描画命令データに基づいて第 2 の描画処理が実行される。

【 0 0 3 9 】

第 2 の描画処理が実行されると、図 6 に示すように、前記ステップ S 3 0 1 ~ S 3 0 3 を経て、図 8 に示すように、前記パーツグリフのニーモニック (MoveTo_M) が取得され、前記ステップ S 3 0 5 を経て、前記 S 3 0 7 の判定が「 N o 」となり、ステップ S 3 0 8 で、前記取得されたニーモニックに応じて現在座標値 v が更新され、ステップ S 3 0 9 の判定が「 N o 」となり、前記ステップ S 3 0 2 から上記フローが繰り返し実行され、前記パーツグリフの輪郭データ (A) が生成される。

10

【 0 0 4 0 】

上記フローが繰り返されるうちに、EndOfCodeが取得されたとする。すると、前記ステップ S 3 0 7 の判定が「 Y e s 」となり、ステップ S 3 1 1 で、前記生成された輪郭データに基づいて前記パーツグリフ (A) のビットマップが生成され、ステップ S 3 1 2 の判定が「 N o 」となり、この演算処理を終了する。

そして、図 5 に示すように、ステップ S 2 0 8 の判定が「 Y e s 」となり、前記ステップ S 2 0 6 で、複合グリフを構成する 2 つ目のパーツグリフのグリフィンデックス (1 5 7)、アフィンパラメータ、及びオフセットが取得され、前記 S 2 0 7 で、第 1 の文字描画処理が再帰的に実行され、前記ステップ S 2 0 1 ~ S 2 0 7 を経て、図 9 に示すように、そのグリフィンデックス等に基づいて、前記パーツグリフ () のビットマップが生成された後、前記ステップ S 2 0 8 の判定が「 N o 」となり、この演算処理を終了する。

20

【 0 0 4 1 】

また、表示コントローラ 9 によって、それら生成されたビットマップ (A、) が表示体 1 0 に描画され、前記描画指示された文字 () が表示体 1 0 に表示される。

このように、本実施形態の表示装置にあっては、ComplexGlyphが取得された場合には、そのComplexGlyphに続いて取得されるグリフィンデックス、アフィンパラメータ、及びオフセット (パーツグリフに関する情報) に基づいてパーツグリフそれぞれを描画するようにした。そのため、例えば、取得される全ての情報にパーツグリフに関する情報であるか否かを示すフラグを付加しておき、それら情報を取得するたびに前記フラグを読み出して当該情報がパーツグリフに関する情報であるか否かを判定する方法と異なり、全ての情報についてフラグの読み出しと判定とを行う必要はなく、構成を簡単なものすることができる。その結果、処理速度を向上することができる。

30

【 0 0 4 2 】

また、例えば、文字を描画する上記機能をハードウェアで実現する場合にあっては、前記フラグを用いる方法と異なり、フラグの読み出しと判定とを行うハードウェア構成は必要なく、その結果、実装面積を小さくすることができる。

40

ちなみに、従来の表示装置にあっては、第 1 の描画処理で、図 1 1 に示すように、グリフィンデックスからグリフデータを取得し (ステップ S 4 0 1)、グリフデータから複合グリフフラグを取得し (ステップ S 4 0 2)、その複合グリフフラグが「 1 」である場合には (ステップ S 4 0 3 「 Y e s 」)、次に配されているパーツのグリフィンデックス等を取得し、文字描画処理が実行される (ステップ S 4 0 4 ~ 4 0 6)。なお、第 2 の描画処理にあっては、図 1 2 に示すように、ComplexGlyphの場合に複合グリフエラー通知を行うステップ (前記ステップ S 3 0 5、 S 3 0 6) は存在しない。

【 0 0 4 3 】

なお、上記実施形態では、図 6 のステップ S 3 0 5 が特許請求の範囲に記載の判定機能を構成し、以下同様に、図 5 のステップ S 2 0 4 ~ S 2 0 8 が描画機能を構成し、図 5 の

50

ステップS206がパーツグリフ取得機能を構成し、図5のステップS207が第1のパーツグリフ描画機能を構成し、図1のCPU8、図6のステップS305が判定手段を構成し、図1のCPU8、図5のステップS204~S208が描画手段を構成する。

【0044】

また、本発明の表示装置は、上記実施の形態の内容に限定されるものではなく、本発明の趣旨を逸脱しない範囲で適宜変更可能である。

上記実施形態では、文字「 」を描画する際、ComplexGlyphの後に、パーツグリフ「A」を定義するグリフィンデックス等と「 」を定義するグリフィンデックス等とが配された描画命令データを用いる例を示したが、これに限られるものではない。例えば、図10に示すように、パーツグリフ「A」を定義するグリフィンデックス等の後に、ComplexGlyphと「 」を定義するグリフィンデックスとが配された描画命令データを用いてもよい。

10

【図面の簡単な説明】

【0045】

【図1】本発明の表示装置の一実施形態の内部構成を示すブロック図である。

【図2】モニターを説明するための説明図である。

【図3】CPUの機能構成を説明するための説明図である。

【図4】ビットマップ生成処理のフローを示すフローチャートである。

【図5】第1の描画処理のフローを示すフローチャートである。

【図6】第2の描画処理のフローを示すフローチャートである。

【図7】表示装置の動作を説明するための説明図である。

20

【図8】表示装置の動作を説明するための説明図である。

【図9】表示装置の動作を説明するための説明図である。

【図10】本発明の変形例を説明するための説明図である。

【図11】従来が表示装置の動作を説明するための説明図である。

【図12】従来が表示装置の動作を説明するための説明図である。

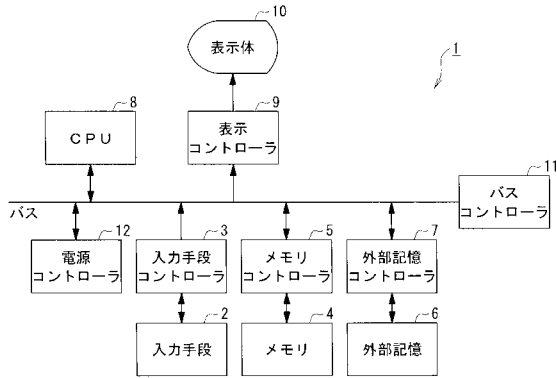
【符号の説明】

【0046】

1は表示装置、2は入力手段、3は入力手段コントローラ、4はメモリ、5はメモリコントローラ、6は外部記憶、7は外部記憶コントローラ、8はCPU、9は表示コントローラ、10は表示体、11はバスコントローラ、12は電源コントローラ

30

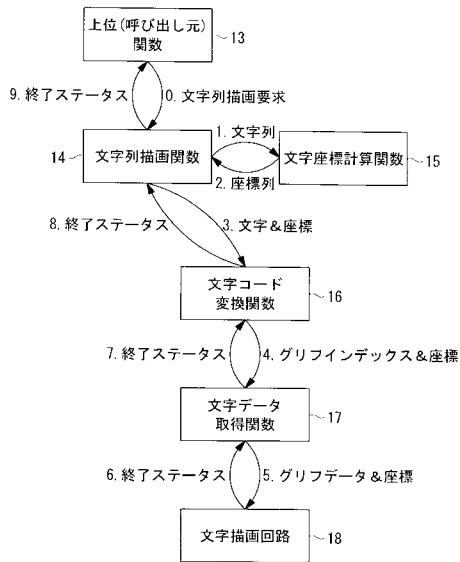
【 図 1 】



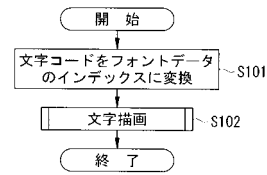
【 図 2 】

ニーモニック	命令ビット数	引数の数	X1	Y1	X2	Y2	X3	Y3	全ビット数
MoveTo_S	6	2	7bit	7bit	-	-	-	-	20
MoveTo_M	5	2	10bit	10bit	-	-	-	-	25
MoveTo_L	5	2	16bit	16bit	-	-	-	-	37
HorLineTo_S	4	1	7bit	-	-	-	-	-	11
HorLineTo_M	3	1	10bit	-	-	-	-	-	13
HorLineTo_L	5	1	16bit	-	-	-	-	-	21
VerLineTo_S	4	1	-	7bit	-	-	-	-	11
VerLineTo_M	3	1	-	10bit	-	-	-	-	13
VerLineTo_L	5	1	-	16bit	-	-	-	-	21
LineTo_S	4	2	7bit	7bit	-	-	-	-	18
LineTo_M	4	2	10bit	10bit	-	-	-	-	24
LineTo_L	8	2	16bit	16bit	-	-	-	-	40
ConicCurveTo_S	3	4	7bit	7bit	7bit	7bit	-	-	31
ConicCurveTo_M	4	4	10bit	10bit	10bit	10bit	-	-	44
ConicCurveTo_L	9	4	16bit	16bit	16bit	16bit	-	-	73
ComicCurveTo_S	5	6	7bit	7bit	7bit	7bit	7bit	7bit	47
ComicCurveTo_M	6	6	10bit	10bit	10bit	10bit	10bit	10bit	66
ComicCurveTo_L	9	6	16bit	16bit	16bit	16bit	16bit	16bit	105
ComGlyph	7	0	-	-	-	-	-	-	7
EndOfCode	6	0	-	-	-	-	-	-	6

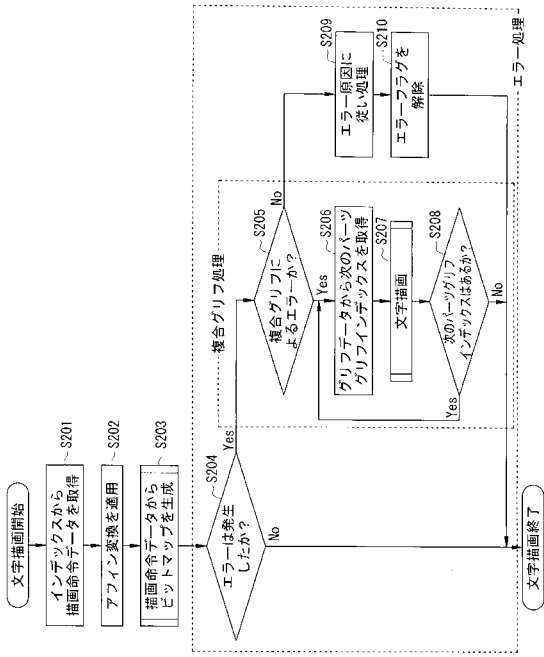
【 図 3 】



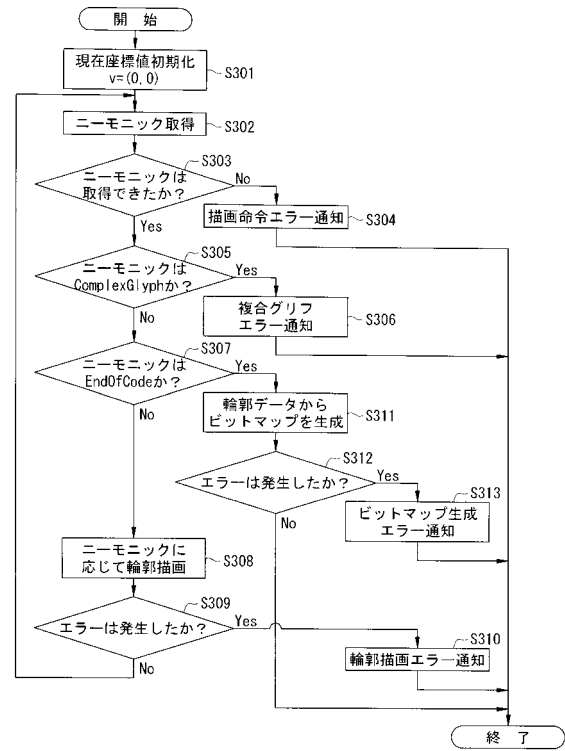
【 図 4 】



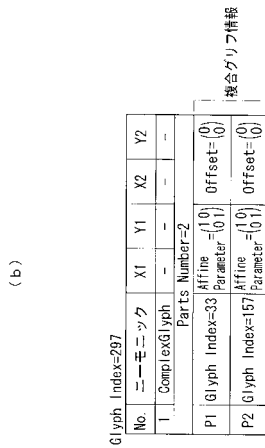
【 図 5 】



【 図 6 】



【 図 7 】



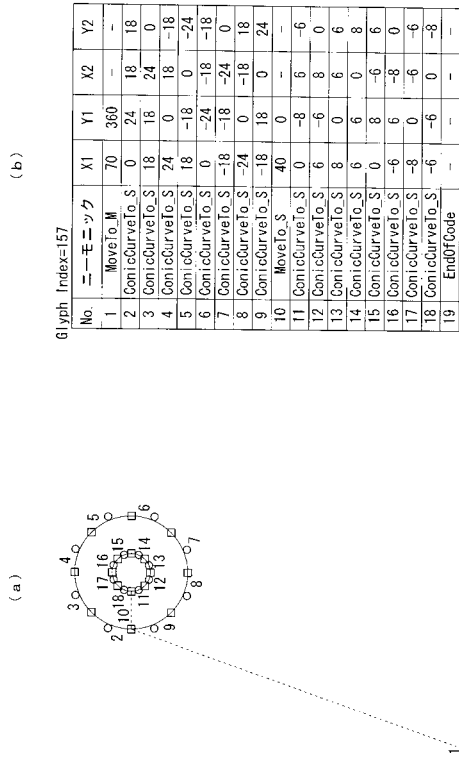
【 図 8 】

(a)

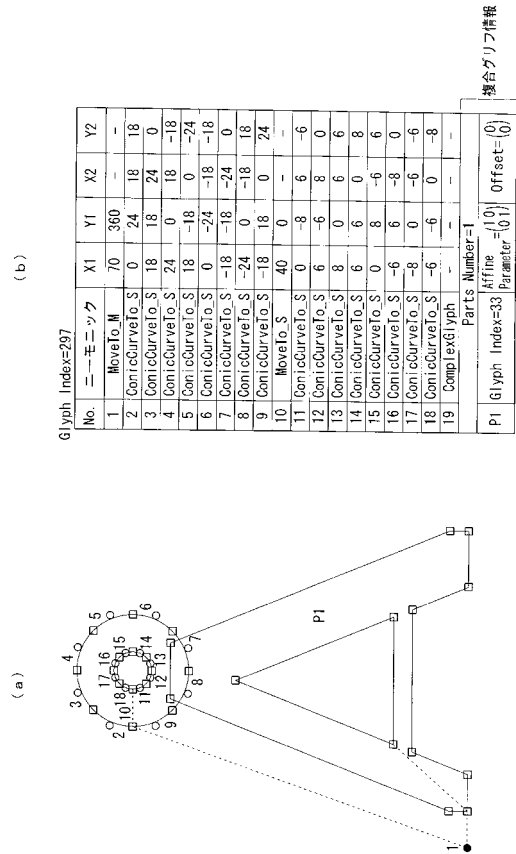
(b)

Glyph Index=33		X1	Y1	X2	Y2
No.	ニーモニック	40	0	-	-
1	MoveTo_S	120	300	-	-
2	VerLineTo_S	60	-	-	-
3	LineTo_M	120	-300	-	-
4	HorLineTo_M	-60	-	-	-
5	LineTo_M	-24	60	-	-
6	VerLineTo_S	-152	-	-	-
7	HorLineTo_S	-40	-	-	-
8	LineTo_M	72	80	-	-
9	HorLineTo_M	136	-	-	-
10	LineTo_M	-68	170	-	-
11	LineTo_M	-68	-170	-	-
12	LineTo_M	-	-	-	-
13	LineTo_M	-	-	-	-
14	LineTo_M	-	-	-	-
15	LineTo_M	-	-	-	-
16	EndOfCode	-	-	-	-

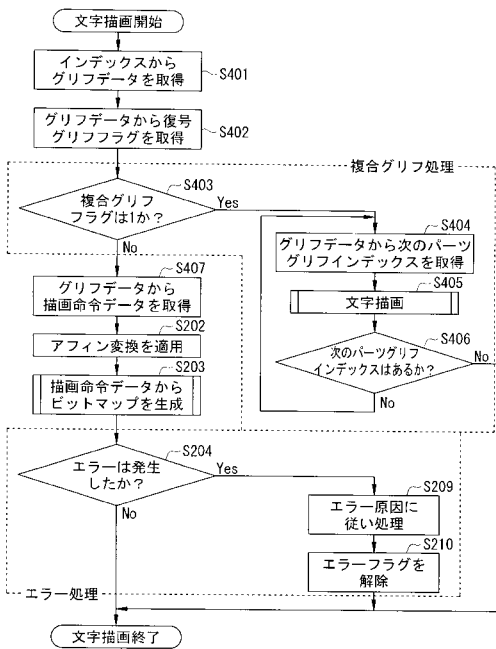
【 図 9 】



【 図 10 】



【 図 11 】



【 図 12 】

