

⑫ 公開特許公報 (A)

昭62-123576

⑮ Int. Cl.⁴
G 06 F 15/72

識別記号 庁内整理番号
6615-5B

⑬ 公開 昭和62年(1987)6月4日

審査請求 有 発明の数 1 (全9頁)

⑭ 発明の名称 円描画方法

⑰ 特 願 昭61-163276
⑱ 出 願 昭55(1980)6月18日
⑲ 特 願 昭55-82484の分割

⑲ 発 明 者 小 口 哲 司 東京都港区芝5丁目33番1号 日本電気株式会社内
⑳ 出 願 人 日本電気株式会社 東京都港区芝5丁目33番1号
㉑ 代 理 人 弁理士 内 原 晋

明 細 書

1. 発明の名称 円描画方法

2. 特許請求の範囲

X-Y座標系に円を描画する方法において、0~45度、135~225度および315~360度の範囲に描画される円についてはX軸上の点を開始点とし、45度~135度および225度~315度の範囲に描画される円についてはY軸上の点を開始点とし、 $X = \sqrt{r^2 - Y^2} = I - F$ および $Y = \sqrt{r^2 - X^2} = I - F$ (rは半径、Iは正整数、Fは小数点) に基いて各描画位置を求め、YおよびXを所定数ずつ増加もしくは減少した時の小数点Fを四捨五入して整数Iを求め、これをX軸およびY軸の座標データとすることにより、加減算のみで円描画データを得ることを特徴とする円描画方法。

3. 発明の詳細な説明

本発明はCRT(カソード・レイ・チューブ)、

プリンター等の表示装置へ画像データを表示する方法に関し、特に円図形もしくは円図形の一部(弧)を描画する方法に関する。

従来の円画像描画方法は、後述するように乗算や関数演算等複雑でかつ長時間の演算を要していた。以下に、従来の円画像データの描画方法を説明しその欠点をのべる。

第1は、円の方程式 $x^2 + y^2 = r^2$ を基にしてX座標、又はY座標を1座標単位(1ドット単位)で変更しながら描画すべき円上の点を表示装置のドット位置として算出していく方式である。より具体的な説明を加えると以下の如くとなる。

中心座標(h, k)、半径rの円を点(h+r, k)より描画を開始し、X座標方向に1座標単位で座標を変更しながらY座標を算出し、円描画を行なう場合、

X座標(xn)は
$$x_n = h + r, (h + r) - 1, (h + r) - 2, \dots, h, \dots, h - (r - 1), h - r \dots \textcircled{1}$$

のように(h+r)から(h-r)の座標まで

1ずつ減算され、その都度Y座標が、

$$y_n = \pm \sqrt{r^2 - (x-h)^2 + k} \quad \dots\dots ②$$

の式から算出される。

この第1の円描画方式では、②式から明らかなように表示座標1ドットの算出毎に2回の整数乗算、1回の実数平方根演算、3回の整数加減算を必要とする。その為、乗算回路、平方根演算回路、加減算回路を用いた長時間の演算を必要とし描画が遅れるという欠点がある。更にX座標もしくはY座標いづれかを基準として1座標単位で変更しながら、その時のY座標(X座標)を求めている為、第1図に示すように0度から45度、135度から225度、315度から360度(0度)の間の円周上に描画されるドットの間隔が拡がり、描画精度及び品質が劣る等の欠点もある。

第2の従来より実施されている円描画方式は、円周上の点を三角関数を含む数式によって表現し、1単位角度毎に描画すべきドットのX座標、Y座標を算出、変更する方式であり、より具体的な説明を加えると以下の如くとなる。

のでハードウェア機構が複雑化してしまう。

一方、第2の描画方法では角度をドット描画位置変更の単位としているので、円周上に描画されるドットの間隔は、その半径が一定であれば等間隔となる。従って第1の従来方法と比較して描画品質は向上する。しかしながら、半径の値に従って角度変更の単位を最適に変更する手段を用意しないと、半径の値がどのような円であろうと、例えば半径が1であろうと1000であろうと同数の描画ドット数となり、半径が小さい場合には同一座標への無駄なドット多重描画、又は半径が大きくなるにつれて描画ドット間隔が拡がりすぎるという不都合が生じる。第2図は、この欠点を説明するための描画像図である。この図の例では角度変更の単位を15度としたものであって、半径が1の小円であっても、半径が8の大円であっても、総描画ドット数はいずれも $360/15 = 24$ 点を必要とする。従って、半径が1の小円では同一座標に多数回のドット描画が実行され、又半径が8の円では、描画ドットの間隔が拡がり表示品質

中心座標(h, k)、半径rの円において、円周上の点(x_n, y_n)は、

$$x_n = r \cdot \cos \theta_n + h \quad \dots\dots ③$$

$$y_n = r \cdot \sin \theta_n + k \quad \dots\dots ④$$

の式によって表現される。ここで、 θ_n を例えば1度毎に0度から360度まで360回変更し、角度変更毎に上式からX, Y座標を算出する。

この描画方法では1ドットの描画毎に、2回の実数三角関数演算、2回の実数乗算、2回の整数加算を必要とする為、これも描画時間が長くなるという欠点を除去することはできない。一方描画時間短縮を目的として、あらかじめ三角関数テーブルをメモリに用意しておき、そのテーブルを参照する事によって、上記した2回の実数三角関数演算を省略することも考えられるが、関数テーブルとして大容量メモリを必要とし、一般性、拡張性に乏しくなるという欠点が生じてくる。しかも、いずれにせよ少なくとも2回の実数乗算が実行されなければならないので演算時間は長く、かつそのための乗算機構などを用意しなければならない

が劣化してしまう。

尚、特に図示していないが、これら2種の従来方式では表示画面への表示位置(ドット位置)を算出するのに、乗算演算回路や平方根演算回路等複雑なハードウェア機構を必要とする。従って、処理装置(例えばマイクロコンピュータ等)が大形化すると共に加減算に比べて数十倍以上の演算処理時間を費さねばならないということが、特に大きな欠点であった。

本発明の目的は、極めて簡単な演算処理だけを用いて短時間で円もしくは円弧を描画する方法を提供することである。

本発明によれば、座標(X, Y)を中心として半径Rの円もしくはその円の一部(円弧)を表示面上に描画すべき表示位置制御データ(画像データ)を作成する場合、X軸を始線として反時計回りに0°~45°の第1の範囲、45°~135°の第2の範囲、135°~225°の第3の範囲、225°~315°の第4の範囲、315°~360°の第5の範囲に分割し、前記円を第1及び第5の範囲に描く

時は描画データ作成開始点 (X + R, Y) を第1の記憶部から読み出し、そのY座標を所定の値つつ変化せしめ、その時のX座標を第1の演算部で求め、前記円を第2の範囲に描く時は描画データ作成開始点 (X, Y + R) データを第1の記憶部から読み出して、そのX座標を所定の値つつ変化せしめその時のY座標を第2の演算部で求め、前記円を第3の範囲に描く時は描画データ作成開始点 (X - R, Y) のデータを第1の記憶部から読み出し、そのY座標を所定の値つつ変化せしめその時のX座標を第1の演算部で求め、前記円を第4の範囲に描く時は描画データ作成開始点 (X, Y - R) のデータを第1の記憶部から読み出し、そのY座標を所定の値つつ変化せしめてその時のY座標を第2の演算部で求め、夫々の演算結果を画像データとして第2の記憶部に設定するように制御される。前記X (又はY) 座標を所定の値つつ変化せしめた時の前記Y (又はX) 座標は、円の方程式 $X^2 + Y^2 = R^2$ から得られる前記Y (又はX) 座標の値のうち小数部を四捨五入して得られ

る表示位置データを得ることができる。又、小数演算部を四捨五入して演算データを作成しているため乗算や平方根演算の複雑な演算回路は一切不要である。従って演算回路の簡略化ができ高速かつ表示品質のよい画像データを作成することができる。

以下に、図面を参照して本発明の一実施例を詳細に説明する。まず第3図に本発明で作成した画像データに基づいて表示面上に円を描画した様子を示す。これは中心座標 (h, k)、半径Rの円を描画したときの一実施例を示す描画図である。このとき全円は、45度毎に8個の1/8円弧に分割され、その1/8円弧を8回描画開始点及び描画方向を変更する事によって全円周の描画位置データが作成され、このデータが表示部に送られて全円周が描画される。315度から45度 (第1及び第5の範囲)、135度から225度 (第3の範囲) の1/8円弧描画に関しては、描画データ作成開始点を中心から方向0度及び180度の円周上の点に選択し、Y座標を1座標単位で移

る整数値データに規格化されることを特徴とする。ここで、座標とはX-Y直交座標系を意味し、その座標データ (X, Y) は表示面上で、n番目の主走査線がX、m番目の副走査線がYに対応するものと考えてもよい。更に、第2の記憶部に設定された円あるいは円弧を示す画像データで指示された表示面上の対応する位置 (画素) にドットパターンが描画されて所望の円もしくは円弧が表示される。

本発明によれば、第1、第3、第5の範囲に描画される円に対してはY座標を基準としてX座標を求め、第2及び第4の範囲に描画される円に対してはX座標を基準としてY座標を求めるように設定しているので、従来の第1図に示すようなドット表示間隔にムラを出すような画像データを作成したり、あるいは第2図に示すように半径の違いに応じてドット表示間隔が拡がりすぎたり又は同一座標 (同一ドット) への多重表示を行ったりするような画像データを作成したりすることは無く、等間隔で無駄のない円もしくは円弧

動させたときのX座標値を算出して描画を行ない、45度から135度 (第2の範囲)、225度から315度 (第4の範囲) の1/8円弧描画に関しては描画データ作成開始点を中心から方向90度及び270度の円周上の点に選択し、X座標を1座標単位で移動させたときのY座標値を算出して描画を行なう。

以下、本発明の一実施例における円描画位置データのX, Y座標算出の為のアルゴリズムについて詳細に説明する。

0度から45度に至る1/8円弧描画を例にあげると、円の方程式は

$$x^2 + y^2 = r^2 \quad \dots\dots(5)$$

この1/8円弧描画の場合は前述したとおり、Y座標の1座標変更毎に円周上の画像データを算出するので⑤式を変形して

$$x = \pm \sqrt{r^2 - y^2} \quad \dots\dots(6)$$

ここで座標は全て“正の整数”で定義されなければ、表示面上にドットパターンを表示することができないので⑥式で定まるxの値に対してその整

部を「I」、小数部を「F」で表現すると、

$$x = \sqrt{r^2 - y^2} = I - F \quad \dots\dots ⑦$$

ここで⑦式をそのままの形で演算し、X座標を直接算出すると2回の整数乗算、1回の整数減算、1回の実数平方根演算を必要とすることが容易に理解できる。しかしながら実数平方根や乗算演算は長時間の演算時間を必要とするので描画速度が遅くなり好ましくない。従って、本実施例では⑦式の小数部分Fにのみ注目し、これによって描画ドット位置のXあるいはY方向の変位分を検出して描画すべきドット位置の座標を決定するように工夫している。この方式によれば後述するように、わずかな整数加減算のみで描画ドット座標位置を指示する画像データを求めることができ、極めて高速度で描画データの作成を行なうことができる。

座標は一般に“正の整数”で定義されるので、得られる小数部分Fを四捨五入する事によって直線的に変化する円の軌跡を座標上のドットとして近似することができる。このため⑦式において、Fが1/2未満の場合は、整数部「I」には変更

中間結果となる。この式の正負を判断する事により、⑦式における四捨五入の判断をする事が可能となることは容易に理解できる。今r=8であるので⑨式は満足されず、小数部Fは0となる。一方、⑩式が満足されるときは、四捨五入の条件が満足されるので、その時のX座標の整数部Iを1だけ減算してX座標を-1だけ移動すればよい。

さらに、小数部演算についても⑨式を直接演算実行したのでは、多数回の実数乗除算を必要とし演算時間が長くなる。従って、⑩式が0以上のとき、小数部演算の変位は、X座標に変化が無く、Y座標のみが1だけ加算されればよいので次のように定義できる。

$$\begin{aligned} & \{r^2 - (y+1)^2 - ((I-N)^2 - (I-N)+1/4)\} \\ & - \{r^2 - y^2 - ((I-N)^2 - (I-N)+1/4)\} \\ & = -(2y+1) \quad \dots\dots ⑩ \end{aligned}$$

一方、⑩式が0未満のときは、X座標を1だけ減算すると共に、Y座標を1だけ加算すればよいので次のようになる。

がなく、Fが1/2以上の場合は、整数部「I」は1だけ減算された値となるように設定する。又、次回演算の為の準備として的小数部「F」は1だけ減算される。即ち、Fが1/2以上となった場合には、描画ドットデータのX座標を-1だけ移動させて各範囲での円の座標データを規格化する。従って⑦式に四捨五入の条件を加味すると、

$$F = I - \sqrt{r^2 - y^2} \geq 1/2 \quad \dots\dots ⑧$$

となり、これを変形して

$$(r^2 - y^2) - (I^2 - I + 1/4) \leq 0 \quad \dots\dots ⑨$$

とすることができる。この軌跡(0°~45°)では描画開始点が中心から0度の円周上の点(h+r, k)に定められるので、y=0, I=rを代入すると⑨式は

$$r - 1/4 \leq 0 \quad \dots\dots ⑪$$

となる。整数演算を行なう為I/4の値を1に切上げてもrが整数値であるので支障は無い。従って、描画開始点では⑩式は

$$r - 1 \leq 0 \quad \dots\dots ⑫$$

となり、これは描画開始点における小数部演算の

$$\begin{aligned} & \{r^2 - (y+1)^2 - ((I-N)^2 - (I-N)+1/4)\} \\ & - \{r^2 - y^2 - ((I-N)^2 - (I-N)+1/4)\} \\ & = 2(I-N) - (2y+1) - 2 \quad \dots\dots ⑬ \end{aligned}$$

⑬、⑭式において、NはX座標の変化の回数を示している。

ここで、第4図に本発明で使用する画像制御装置の概略を示すブロック図を提示する。第4図において、メモリ100は複数のレジスタ等で構成されたもので、CPUあるいは外部メモリから送られるX、Y座標と円との交点、即ち画像データ作成開始点が設定される。このメモリ100からのデータはY座標を所定数ずつ変化させた時のX座標を前述の方式に基いて算出する第1の演算部101と、X座標を所定数ずつ変化させた時のY座標を算出する第2の演算部102とに送られ、夫々の演算結果が画像データ格納メモリ103に出力される。尚、演算途中結果はメモリ100に一時格納するものとする。メモリ103から表示部104へ画像データが表示信号に変換されて転送されることにより所望の円あるいは円弧の表示

を可能ならしめる。105は各部のタイミングを制御する信号を出力する制御部である。

ここで、メモリ100に格納されるデータについてより詳しく説明する。分割された角度(第1～第5の範囲)のうち例えば第1の範囲に表示すべき画像データを作成する場合、初期値X、初期値Yとして描画開始点の座標 $X = h + 8$ 、 $Y = k$ が設定され、小数部演算の中間結果Dとして描画開始時においては、⑩式で与えられた値 $r - 1 = 7$ が設定される。更に小数部演算の変位 D_1 として描画開始時においては⑪式において、Yが0であるから -1 が設定される。さらに、小数部演算の結果が0未満となったときに追加実行される値 D_2 として⑫式で示される小数部演算変位式の一部である $2(I - N) - 2$ の計算結果が格納される。描画開始時において $N = 0$ 、 $I = r$ であるので、 D_2 に $2(r - 1)$ が設定される。X座標の方向変化が生ずるたびに、前記 $2(I - N) - 2$ のNの値が1だけ増加されるので、その結果 D_2 は2だけ減算される。

始命令が与えられたとき信号線11に発生する信号DRAW STARTによって出力が能動状態となり、ANDゲート41から供給されるリセット信号が「1」となったとき非能動状態となる。42の出力は描画タイミング発生回路44に供給され、第6図に示されるタイミング信号 $E_1 \sim E_7$ が順次発生される。40は零検出回路でありレジスタDCの内容が全ビット「0」となったとき出力は「1」となり、ANDゲート41にその出力は接続される。ANDゲート41の一方の入力には描画タイミング信号 E_7 が接続されており、入力が共に「1」となったときANDゲート41の出力であるDRAW END信号(第6図参照)が「1」となり描画指示回路42を非能動とする。フリップフロップ43は描画タイミング E_2 のときのレジスタDのデータ正負を記憶する。43の出力は45c, d, eに供給され、第5図に示す演算処理制御を可能とする。30及び31はデータ発生回路であり、各々データ「1」及びデータ「2」を発生する。60は映像制御回路であり本描画演算実

行回路は描画演算実行回路のブロック図でありCPUから送出される命令又はパラメータがデータ及び制御バス10を經由してレジスタD, D1, D2, X, Y, DCに設定され、さらに描画開始命令が与えられたとき描画指示信号発生回路(フリップフロップ)42を起動する為の信号線11がフリップフロップ42のセット端子に接続される。20aから20pは各々切換ゲート群であり、G入力が「1」となったとき、入力信号が出力に接続され、G入力が「0」のときは、出力には信号が出力されず開放状態となる。35, 36, 37は3バス形式を持った演算器のデータバスであり、被演算数が35へ、演算数36へ、ALU50の出力である演算結果が37へそれぞれ接続される。ALU50は演算器であって、オア・ゲート45gよりA/Sに供給される信号線が「1」のとき加算を「0」のとき減算を実行し、被演算数がバス35から、演算数がバス36からそれぞれ入力され、演算結果がバス37へ出力される。42は描画指示信号発生回路であり、CPUから描画開

行回路にて演算生成された描画位置座標X, Yが供給され実際に描画を実行する。

以下、第5図の回路動作について説明する。CPUよりパラメータ設定及び描画開始命令が与えられると信号線11にDRAW START信号が発生し描画指示信号発生器42の出力DRAW INDEXが「1」となり描画タイミング信号発生器44から描画タイミング信号が $E_1 \sim E_7$ の順に発生し、タイミング E_7 においてDCの値が全て「0」でなければ、ANDゲート41の出力信号であるDRAW ENDは「0」のままであり、引き続き E_1 から E_7 の描画タイミング信号が発生し、描画が継続される。タイミング E_7 においてDCの値が全て「0」であれば上記DRAW ENDは「1」となり、描画指示信号発生器42をリセットし、描画を終了する。

次に、描画タイミング E_1 において実行される「 $D + D_1 \rightarrow D$ 」の演算実行を例に取って回路動作を説明する。描画タイミング信号 E_1 が出力されるとオアゲート45a及び45gの出力が共に「1」

となり、切換ゲート群20a, 20b, 20dが活性化され、バス35にはレジスタDの内容バス36にはレジスタD₁の内容が出力され、各々ALU50に入力される。ALU50の出力、即ち演算結果「D+D₁」はバス37、及び切換ゲート20bを経由してレジスタDに読み込まれ「D+D₁→D」の演算が実行される。描画実行タイミングE₂以降の演算に関しては、上記E₁時の説明と類似しているため省略する。

この様な回路動作の下で実行される円図形の画像データ作成過程を第7図及び第8図の手順図を参照して以下に説明する。

第7図に示すように画像データ作成開始時に、メモリ100内のレジスタX, Y, DC, D, D₁, D₂に夫々前述したデータを設定する。この後、描画開始命令の入力により第8図に示す流れ図に沿って、タイミング制御の下で画像データ作成処理を実行していく。

第8図の流れ図に沿って0°～45°の範囲に描画される円図形の画像データを求める。

順は前述した過程を繰り返して実行すればよい。即ち、多タイミングE₁～E₇で第8図に示す加算を実行し、その演算結果特にレジスタDとDCとの内容を調べて、次に実行すべき処理を決定する。この処理は第8図に示す単純な加減算処理でよい。

かかる処理を逐次実行することにより、第3図に示す点a₀～a₆の各座標位置を示す画像データが作成され、画像制御回路60に順次格納される。一方、45°～135°, 225°～315°の範囲では、第8図でタイミングE₄及びE₆の処理が夫々Y-1→Y, X+1→Xに変更されるだけで他は同様の処理でよい。この様にして全円を示す画像データが全て画像制御回路60に格納され、表示タイミングに同期して表示部へ送られる。

本実施例は円を4分割して45°～135°, 135°～225°, 225°～315°, 315°～45°の各範囲で初期値をYあるいはX座標上に選定して、XあるいはYの座標を所定の値(実施例では1)ずつ変化させて、その時のYあるいはXの座標を第4図の小数部簡易演算方式を用いて算出するもの

まず、レジスタYに設定されている開始点データ、この場合にはX=h+8, Y=kが、第3図での点a₀を示す画像データとして画像制御回路60に転送される。この時、タイミングE₁でレジスタDとD₁との内容がALU50で加算される。即ちD=r-1=7, D₁=-(2Y+1)=-1(*: Y=0)だからD+D₁=7-1=6となり、データ6がレジスタDに設定される。更に、タイミングE₂でレジスタD₁の内容が(-2)加算され、-3がレジスタD₁に設定される。この時D=6>0だからタイミングE₆での処理にジャンプする。この期間はYの内容を+1変化させる処理を実行する期間で、Y=1がレジスタYに設定され、次の画像データ作成の準備をする。一方レジスタDCには0°～45°の範囲に描画すべきドット数7が設定されており、1ドット作成する毎に1ずつ減算される。例えばリングカウンタによりその値が6に設定される(タイミングE₇)。

次に再びタイミングE₁に戻り、Y座標が+1されたk+1でのX座標の算出が開始される。手

で、描画すべき円又は圆弧の半径が異なっても、一定の間隔で円周上のドットパターンを近似することができ、半径の小さい円での同一ドットの多重化及び半径の大きい円でのドット間の拡がりをなくした品質の良い円描画を実行することができる。又、第8図より明らかなように円のドット位置を作成する演算方式は独得の小数部簡易演算法を採用しているので、平行根演算や乗算演算等、長時間を要する演算を使うことなく単純な加減算だけでよいので、描画速度は従来に比して約100～1000倍程高速化できる。

尚、本実施例では円の全周及びその任意の一部の圆弧のいずれでも描画可能であるが、描画すべき円あるいは圆弧がどの範囲に相当するものであるかを予め規定しておく必要がある。しかしながら、この規定は極めて簡単で、例えば各範囲に表示されるべきドット数を規定しておき、第8図でドットデータを作成する度にその数を検出して所定の値になった時、演算を中止したりあるいは範囲の設定を変更したりすればよい。又、マスキレ

ジスタを付加して目的とするデータだけを画像制御回路60に出力するようにしてもよい。

以上、0度から45度に至る1/8円弧描画の場合を例に取って説明したが、他の角度領域における円弧についても、第4図におけるX、Yの演算位置の交換、又はXY加減算の変更、描画開始点の設定変更によって描画可能である。又、上記例ではX、Y座標値の算出を例に取っているが、画像記憶器のドット情報格納番地の算出を行なう場合に置換しても効果は同じである事は言うまでも無い。更に、基準となるXもしくはY座標の変化分を1ではなく2もしくは3等他の整数値に変更すれば、円周上で隣り合うドットの間隔を任意に変化させることができ楕円表示等も可能となる。

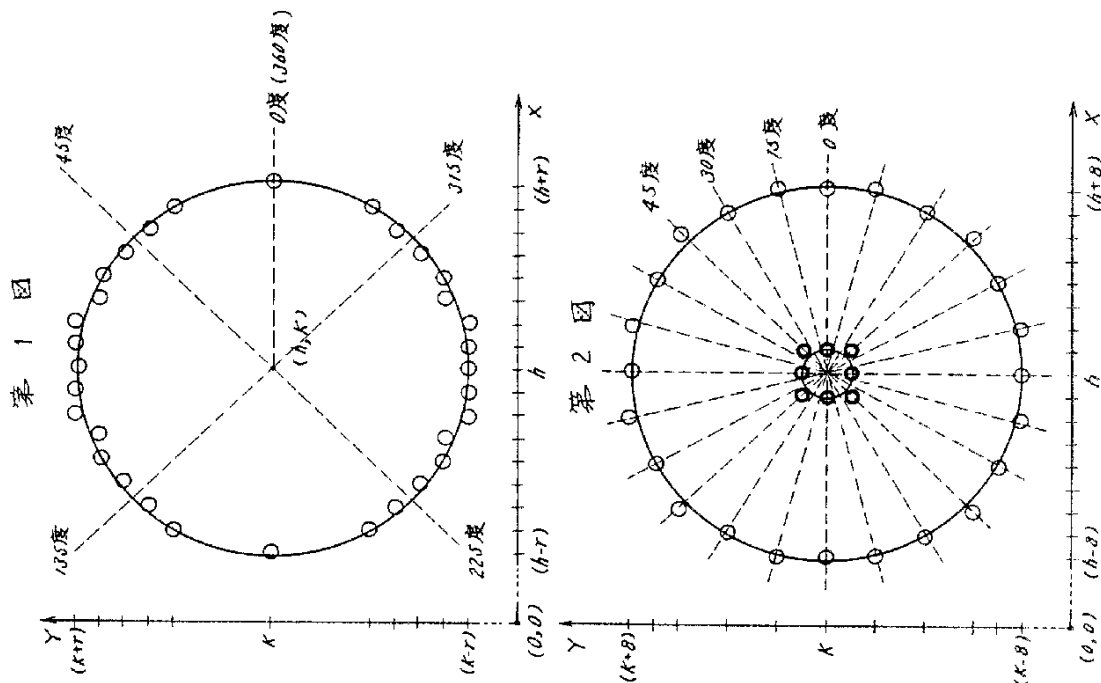
で、第5図は画像データ作成回路図、第6図はタイミング信号発生図、第7図、第8図は夫々動作処理の実行手順を示す流れ図である。

(h, k) ……円の中心座標、 $a_0 \sim a_6$ …… $0^\circ \sim 45^\circ$ の範囲に表示される半径8の円のドット位置、1 ……レジスタD、2 ……レジスタD₁、3 ……レジスタD₂、4 ……レジスタX、5 ……レジスタY、6 ……レジスタDC、10 ……バス、11 ……セット信号、20 ……ゲート、42 ……フリップフロップ、45 ……タイミング信号発生回路、50 ……ALU、60 ……画像制御回路、100、103 ……メモリ、101、102 ……演算部、104 ……表示部、105 ……タイミング制御部。

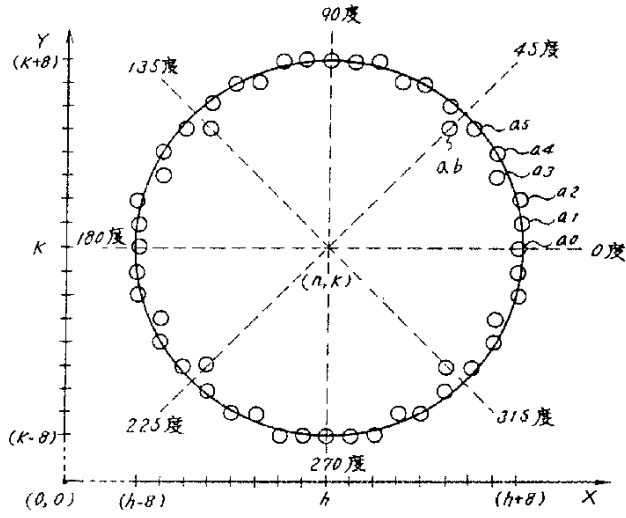
代理人 弁理士 内 原 晋

4. 図面の簡単な説明

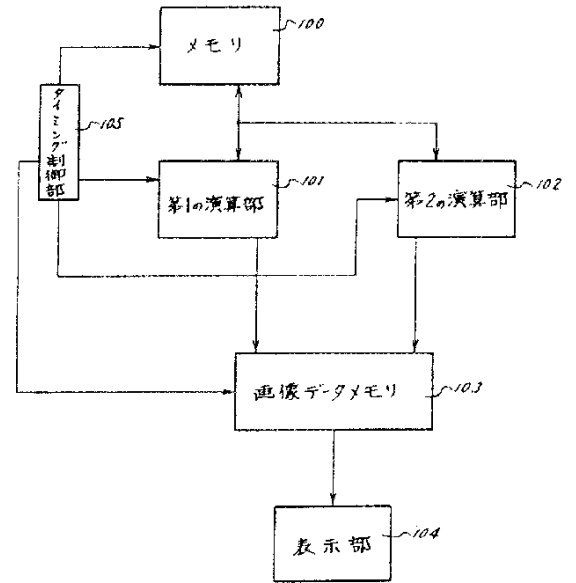
第1図、第2図は夫々従来の描画方式によって描画された円図形図、第3図は本発明の一実施例による描画方法によって描画された円図形図、第4図は本実施例で使用する描画制御装置の戦略図



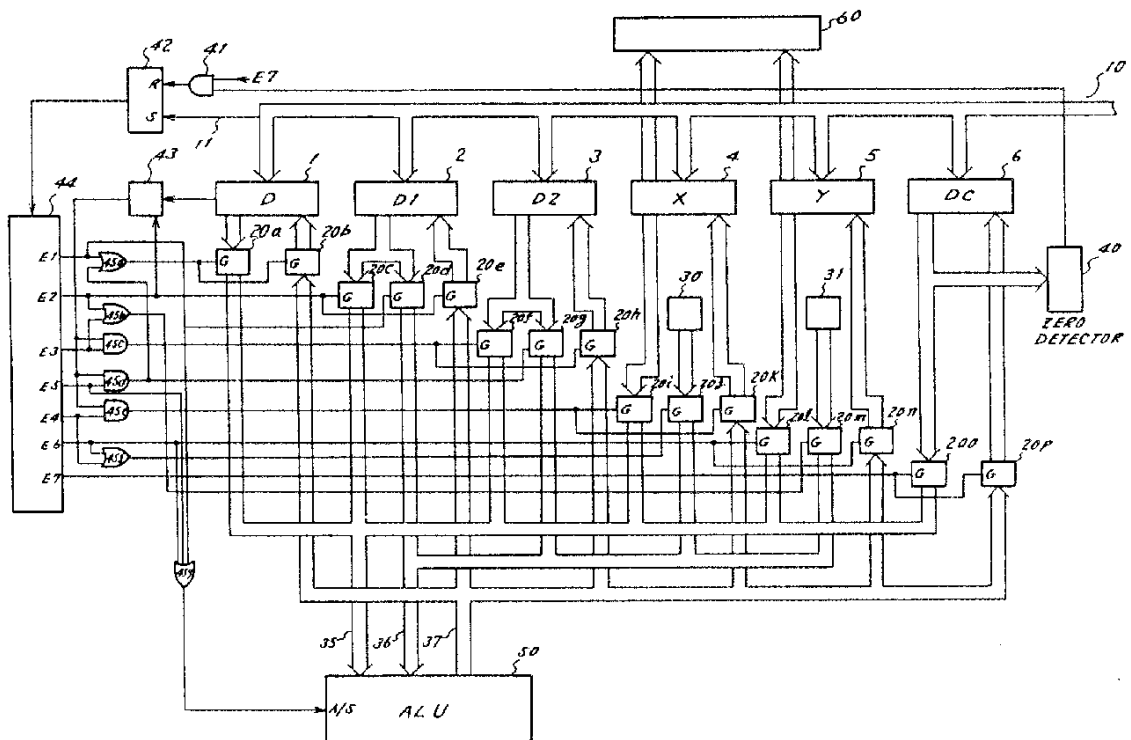
第3図



第4図



第5図



第7図

第8図

第6図

