

⑫ 特許公報 (B2) 平2-14714

⑬ Int. Cl.<sup>5</sup>  
G 09 G 1/00

識別記号 庁内整理番号  
8121-5C

⑭公告 平成2年(1990)4月9日

発明の数 1 (全11頁)

⑮発明の名称 画像制御装置

審判 昭61-11678 ⑯特願 昭55-82484 ⑰公開 昭57-8584  
⑱出願 昭55(1980)6月18日 ⑲昭57(1982)1月16日

⑳発明者 小口 哲司 東京都港区芝五丁目33番1号 日本電気株式会社内

㉑出願人 日本電気株式会社 東京都港区芝5丁目33番1号

㉒代理人 弁理士 内原 晋

審判の合議体 審判長 白樫 栄一 審判官 堤 隆人 審判官 石川 伸一

㉓参考文献 特開 昭53-71535 (JP, A) 特開 昭53-86122 (JP, A)

1

2

㉔特許請求の範囲

1 直交座標系でX軸及びY軸と交差する円の交点データを記憶する記憶部と、前記X軸と交差する円の交点データのうちそのY座標データを所定数ずつ変化させた時の円周上でのX座標データを  
5  $X = \sqrt{r^2 - Y^2}$  (rは半径) から算出する第1の演算部と、前記Y軸と交差する円の交点データのうちそのX座標データを所定数ずつ変化させた時の円周上でのY座標データを  
10  $Y = \sqrt{r^2 - X^2}$  から算出する第2の演算部と、前記第1及び第2の演算部での演算実行の切り換えを制御する制御部とを含み、前記制御部はX座標の変化に比べてY座標の変化が大きい範囲では前記第1の演算部を用いて演算を行ない、Y座標の変化に比べてX座標の変化が大きい範囲では前記第2の演算部を用いて演算を行なうように制御することを特徴とする画像制御装置。

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に表示すべき画像データ  
20 を作成する為の画像制御装置に関し、特に円図形もしくは円図形の一部(弧)の描画データを作成するための機構を有する画像制御装置に関する。

従来の画像データ作成用の制御装置は主として2種類に大別される。かかる2種類の制御装置には後述するように乗算機構や関数演算機構等複雑でかつ長時間の演算を要する機能が具備されてお

り、高速でデータを作成することができなかつた。しかしながら、このような機能は従来の制御方式には欠くべからざるもので、円図形等の画像データを高速で作成することは困難であつた。以下  
5 に、円画像データを作成する従来の制御方式を提示して上記欠点を明らかにする。

その第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

$$x_n = h + r, (h + r) - 1, (h + r) - 2, \dots$$

$$h, \dots, h - (r - 1), h - r \quad \dots \text{①}$$

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、

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

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座標いずれかを基準として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つて表現し、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。

中心座標(h, k)、半径rの円において、円周上の点(xn, yn)は、

$$xn = r \cdot \cos\theta n + h \quad \dots\dots(3)$$

$$yn = r \cdot \sin\theta n + k \quad \dots\dots(4)$$

の式によつて表現される。ここで、 $\theta n$ 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。

この描画方式では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つて、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のでハードウェア機構が複雑化してしまう。

この第2の描画方式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。従つて第1の従来方式と比較して描画品質は向上する。しかしながら、半径の値に従が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

なるにつれて描画ドット間隔が拡がりすぎるといふ不都合が生じる。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度としたものであつて、半径が1の小円であつても、半径が8の大円であつても、総描画ドット数はいずれも $360/15=24$ 点を必要とする。従つて、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。従つて、処理装置(例えばマイクロコンピュータ等)が大型化すると共に加減算に比べて数十倍以上も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つた。

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画像データを作成する画像制御装置を提供することで、特に円図形あるいは円図形の一部(円弧)を描画するに適した画像制御装置を提供する事にある。

本発明は、直交座標系でX軸及びY軸と交差する円の交点データを記憶する記憶部と、前記X軸と交差する円の交点データのうちそのY座標データを所定数づつ変化させた時の円周上でのX座標データを $X = \sqrt{r^2 - Y^2}$ (rは半径)から算出する第1の演算部と、前記Y軸と交差する円の交点データのうちそのX座標データを所定数づつ変化させた時の円周上でのY座標データを $Y = \sqrt{r^2 - X^2}$ から算出する第2の演算部と、前記第1及び第2の演算部での演算実行の切り換えを制御する制御部とを含み、前記制御部はX座標の変化に比べてY座標の変化が大きい範囲では前記第1の演算部を用いて演算を行ない、Y座標の変化に比べてX座標の変化が大きい範囲では前記第2の演算部を用いて演算を行なうように制御することを特徴とする。

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 $0^\circ \sim 45^\circ$ の第1の範囲、 $45^\circ \sim 135^\circ$ 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)を記憶部から読み出し、  
 そのY座標を所定の値づつ変化せしめ、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)データを記憶部から読み出して、そのX座標  
 を所定の値づ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  
 記憶部から読み出し、そのY座標を所定の値づつ  
 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)のデータを記憶部から読  
 み出し、そのX座標を所定の値づつ変化せしめて  
 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)座標を所定の値づつ変化  
 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整数  
 値データに規格化すれば処理が簡単になる。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。このようにして作成された画像  
 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對  
 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図に示すような半径の違  
 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)への多重表示を行なつた  
 りするような画像データを作成したりするような  
 ことは無く、等間隔で無駄のない円もしくは円弧  
 の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。従つて演算回路の簡略化ができ高速で  
 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つ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。

第3図に示す中心(h, k)、半径rの円に對  
 して、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

$$(x-h)^2+(y-k)^2=r^2 \quad \dots\dots\textcircled{5}$$

この1/8円弧描画の場合は前述したとうり、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

$$x=\sqrt{r^2-(y-k)^2}+h \quad \dots\dots\textcircled{6}$$

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  
 整数部を“I”、小数部を“F”で表現すると、

$$x=\sqrt{r^2-(y-k)^2}+h=I-F \quad \dots\dots\textcircled{7}$$

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。従つて、本実施例では⑦式の小数部分Fにのみ注目し、これによつ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つ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”には変更がなく、Fが1/2以上の場合、整数部“I”は1だけ減算された値となるように設定する。即ち、Fが1/2以上となつた場合には、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。従つて⑦式に四捨五入の条件を加味すると、

$$F = (I - h) - \sqrt{r^2 - (y - k)^2} \geq 1/2 \quad \dots\dots⑧$$

となり、これを变形して

$$\{r^2 - (y - k)^2\} - \{(I - h)^2 - (I - h) + 1/4\} \leq 0 \quad \dots\dots⑨$$

とすることができる。この範囲(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=k、I=r+hを代入すると⑨式は

$$r - 1/4 \leq 0 \quad \dots\dots⑩$$

となる。整数演算を行なう為1/4の値を1に切上げててもrが整数値であるので支障は無い。従つて、描画開始点では⑨式は

$$r - 1 \leq 0 \quad \dots\dots⑪$$

となり、これは描画開始点における小数部演算の中間結果Dの初期値となる。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。

次に、小数部の演算を実行して次描画点のX座

標を求める必要があるが、⑧式に示す小数部Fの値を直接求めると平方根演算を必要とし処理が大変である。しかしながら、実際の描画ではY座標が+1づつ変化する毎に、X座標が変化しないか、あるいは-1だけ変化するかのいずれかである。そしてこの変化の有無は前記中間結果Dが正となるか負となるかによつて決定されるわけである。故に、次の描画点での中間結果Dの変位を求めればよい。即ち、⑨式の左辺の変位を求めるとによつて次の中間結果を求めることができる。

ここで、⑨式の左辺を直接演算すると多数回の実数乗除算が必要となるので、演算時間が長くなる。従つて、⑨式の左辺が0以上のとき、X座標は変化せずY座標のみが1だけ加算されればよい。故に、次描画点の座標データAに対しては

$$\{r^2 - (y - k + 1)^2\} - \{(I - h - N)^2 - (I - h - N) + 1/4\} \quad \dots\dots⑫'$$

となり、前描画点Bに対しては

$$\{r^2 - (y - k)^2\} - \{(I - h - N)^2 - (I - h - N) + 1/4\} \quad \dots\dots⑫''$$

となるから、中間結果Dの変位D1はその差をとつて、

$$\begin{aligned} & \{r^2 - (y - k + 1)^2\} - \{(I - h - N)^2 - (I - h - N) + 1/4\} \\ & - \{r^2 - (y - k)^2\} + \{(I - h - N)^2 - (I - h - N) + 1/4\} \\ & = -\{2(y - k) + 1\} \quad \dots\dots⑬ \end{aligned}$$

となる。

一方、⑨式の左辺が0未満のときは四捨五入条件を満足するので、次描画点での中間結果はY座標が+1、X座標が-1されるため

$$\{r^2 - (y - k + 1)^2\} - \{(I - h - N - 1)^2 - (I - h - N - 1) + 1/4\} \quad \dots\dots⑬'''$$

となる。故に、中間結果Dのこの時の変位は⑨'''-⑨''で算出されるので、

$$\begin{aligned} & \{r^2 - (y - k + 1)^2\} - \{(I - h - N - 1)^2 - (I - h - N - 1) + 1/4\} \\ & - \{r^2 - (y - k)^2\} + \{(I - h - N)^2 - (I - h - N) + 1/4\} \\ & = 2\{(I - h) - N\} - \{2(y - k) + 1\} - 2 \quad \dots\dots⑭ \end{aligned}$$

となる。ここで、-\{2(y - k) + 1\}は変位D1と同値であるため、これを除く式、すなわち2\{(I - h) - N\} - 2を変位D2として算出し、そ

の結果からD 1分を引けば解が求まる。

なお、NはX座標の変化回路数を示す。

ここで、第4図に本発明の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座標を所定数づつ変化させた時の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座標を所定数づつ変化させた時の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に格納された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、①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②式において、Yがk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③式で示される小数部演算変位式の一部である $2\{(I-h)-N\}-2$ 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、 $D_2$ として $2\{(r-h)-1\}$ が設定される。X座標の方向変化が生ずるたびに、前記 $2\{(I-h)-N\}-2$ 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。

第5図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、 $D_1$ 、 $D_2$ 、X、Y、DCに設定される。ここで、レジスタD、 $D_1$ 、 $D_2$ には前述した中間結果D、変位 $D_1$ 、 $D_2$ が夫々格納される。さらにレジスタX、Yには夫々描画開始座標が格納され、レジスタ

DCには描画範囲に描画すべきドット数が格納される。

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、演算数が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は演算器であつて、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によつ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”に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E1~E7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、アンドゲート41にその出力は接続される。アンドゲート41の一方の入力には描画タイミング信号E7が接続されており、入力が共に“1”となったときアンドゲート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E2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 4 2 の出力 DRAW—  
 INGが“1”となり描画タイミング信号発生器  
 4 4 から描画タイミング信号がE 1 からE 7 の順  
 に発生し、タイミングE 7 においてDの値が全て  
 “0”でなければ、アンドゲート 4 1 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、描画が続行される。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 4 2 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+D1→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 4 5 a 及び 4 5 g の出力が共に  
 “1”となり、切換ゲート群 2 0 a, 2 0 b, 2  
 0 d が活性化され、バス 3 5 にはレジスタDの内  
 容が、バス 3 6 にはレジスタD1の内容が出力され、  
 各々ALU 5 0 に入力される。ALU 5 0 の出力、  
 即ち演算結果“D+D1”はバス 3 7、及び  
 切換ゲート 2 0 b 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  
 E 1 時の説明と類似しているので省略する。

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 1 0 0 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  
 後の、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。

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 $a_0$ を示す画像データとして画像制御回路 6  
 0 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ALU 5 0 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づ  
 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  
 順は前述した過程を繰り返して実行すればよい。  
 即ち、各タイミング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。

0°~45°の範囲に描画される点の値を第 8 図の  
 流れ図に沿って算出すると下に示す表 1 のような  
 結果が得られる。

表 1

E1	D	7	6	3	-2	3	-6	-7
E2	D1	-1	-3	-5	-7	-9	-11	-13
E3	D2	14	14	14	12	12	10	8
E4	X	8	8	8	7	7	6	5
E5	D	7	6	3	10	3	4	1
E6	Y	0	1	2	3	4	5	6
E7	DC	6	5	4	5	2	1	0

なお、表 1 は原点を中心とする半径 8 の円につ  
 いて求めたものであるが、中心 (h, k) の円に  
 ついてはX座標の各値にhを加算すればよいこと  
 は明白である。

かかる処理を逐次実行することにより、第 3 図  
 に示す点 $a_0 \sim a_6$ 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 6 0 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 $Y-1 \rightarrow Y,$   
 $X+1 \rightarrow X$ 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 6 0 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で、描画すべき円又は円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ため、平方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円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円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、マスク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。

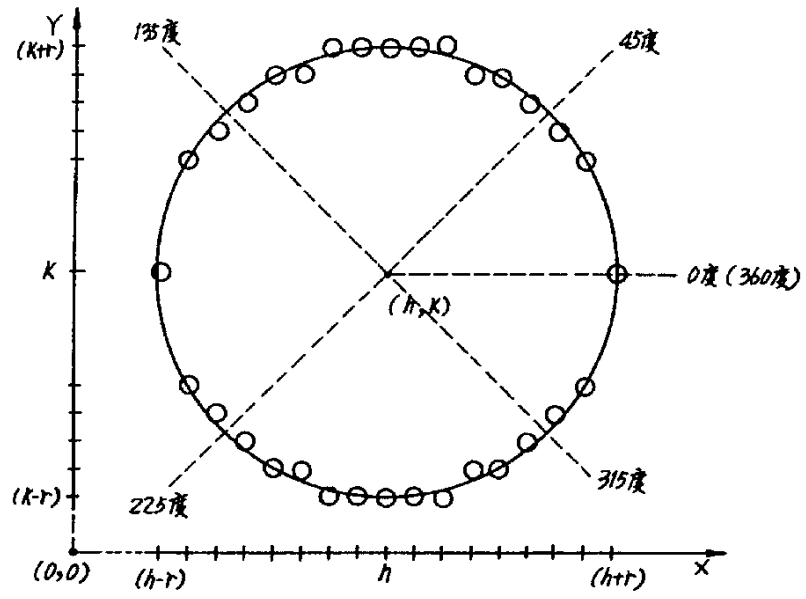
以上、0度から45度に至る1/8円弧描画の場合を例に取って説明したが、他の角度領域における円弧についても、第4図におけるX、Yの演算位置の交換、又はXY加減算の変更、描画開始点の設定変更によつて描画可能である。又、上記例ではX、Y座標値の算出を例に取っているが、画像記憶器のドット情報格納番地の算出を行なう場合

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。この様に、本発明によれば描画範囲に応じて $X = \sqrt{r^2 - Y^2}$ および $Y = \sqrt{r^2 - X^2}$ のいずれかを指定して描画ドット位置を算出するようにしているので、第1図のように描画間隔にムラを出すことはなく、また、第2図に示すようにドット間隔が広がりすぎることもない。さらに、同一座標への多重描画もなくなるという利点もある。とくに描画される円が上下左右対象になるようなドットデータを生成することができるので、従来のように不自然さを出すことなく、自然円に近い円を描画することができる。

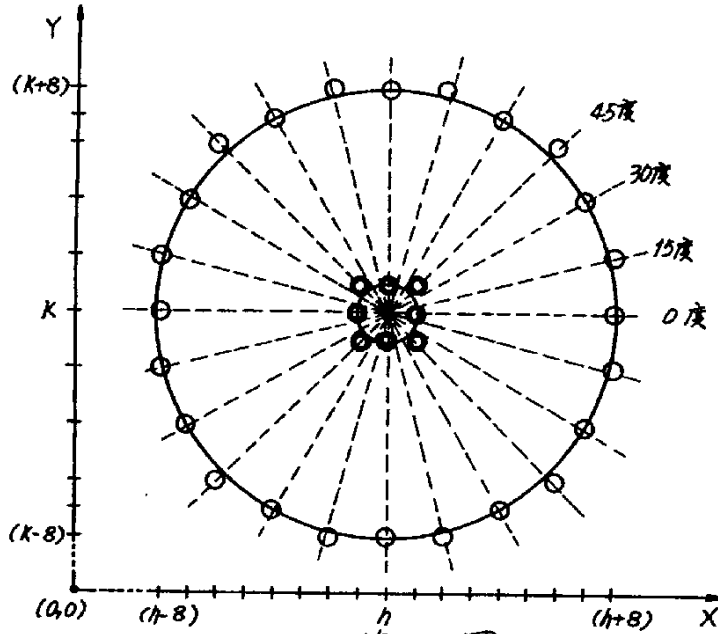
#### 図面の簡単な説明

第1図、第2図は夫々従来の描画方式によつ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式によつて描画された円図形図、第4図は本実施例描画制御装置の概略図で、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、 $a_1 \sim a_8 \dots 0^\circ \sim 45^\circ$ 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D<sub>1</sub>、3…レジスタD<sub>2</sub>、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。

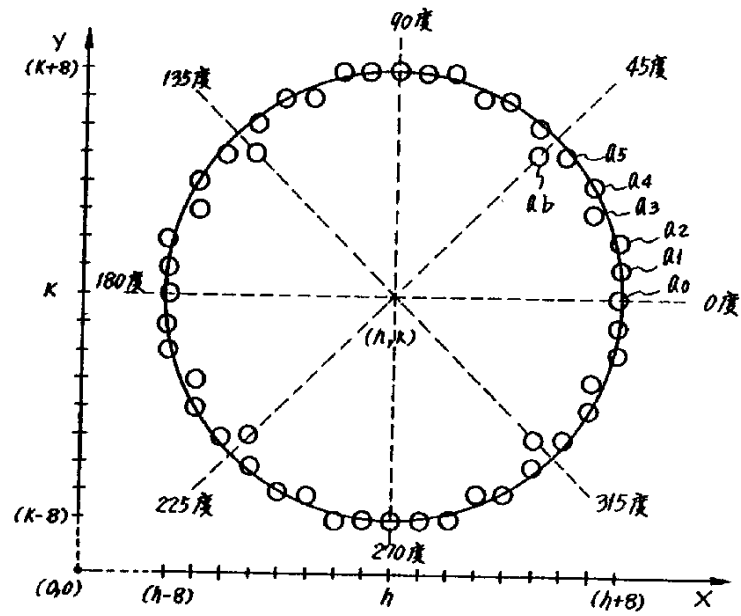


第 1 图

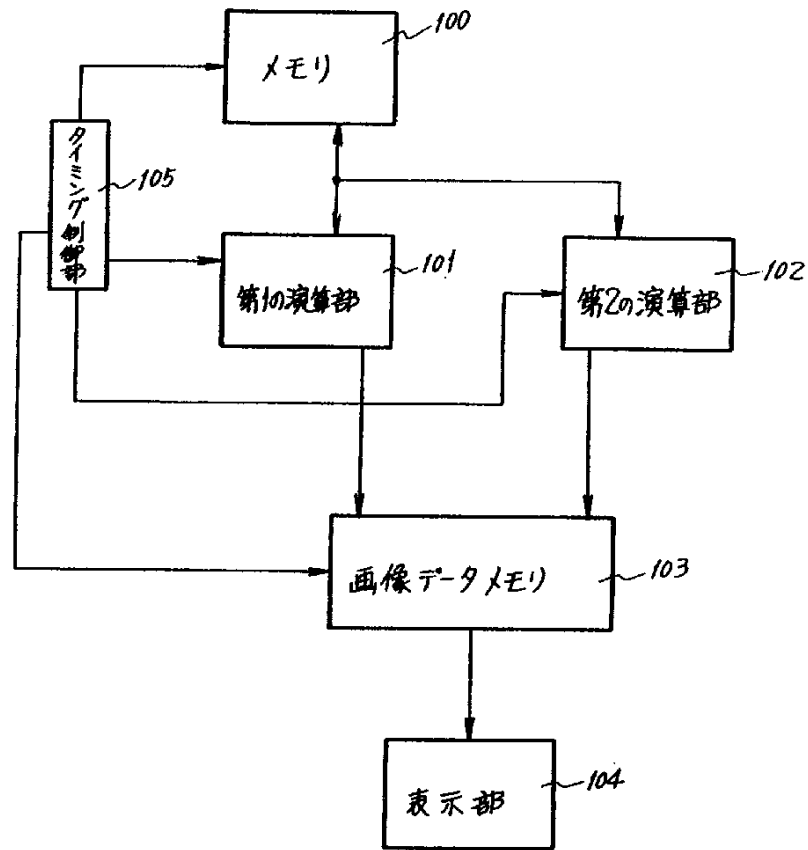


第 2 图

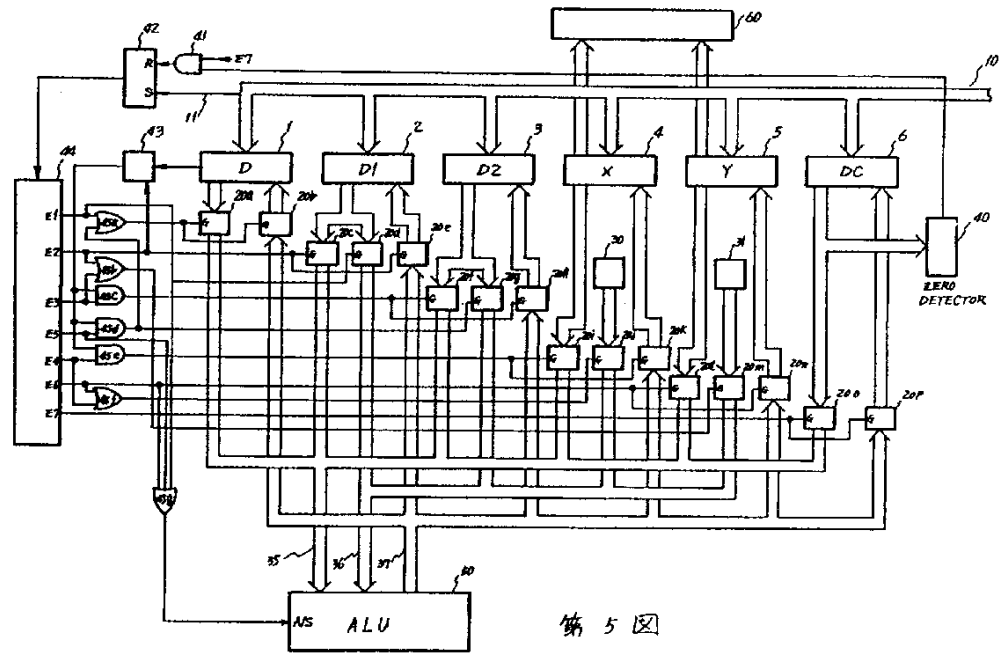




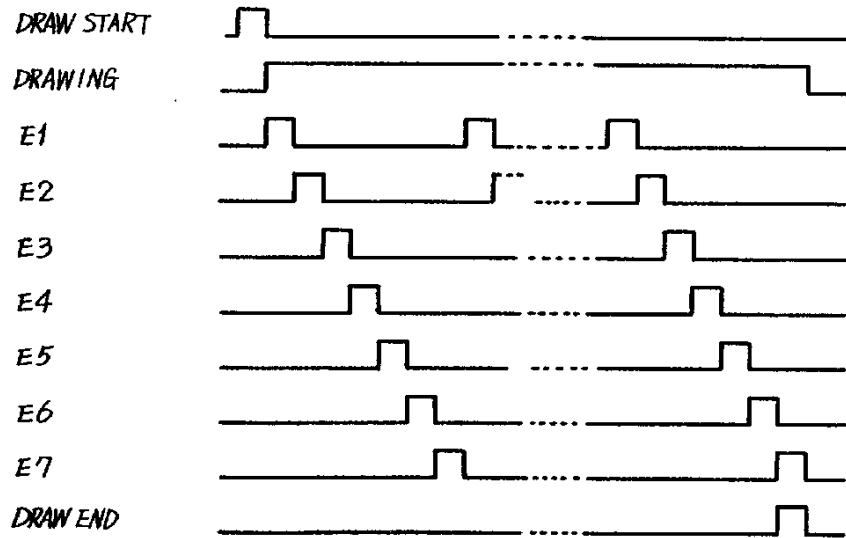
第 3 図



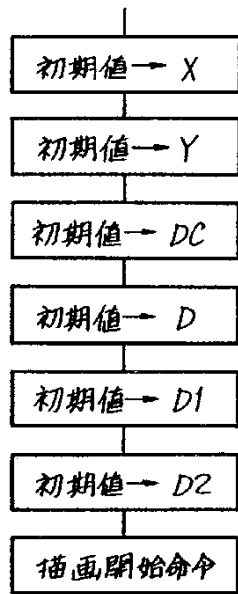
第 4 図



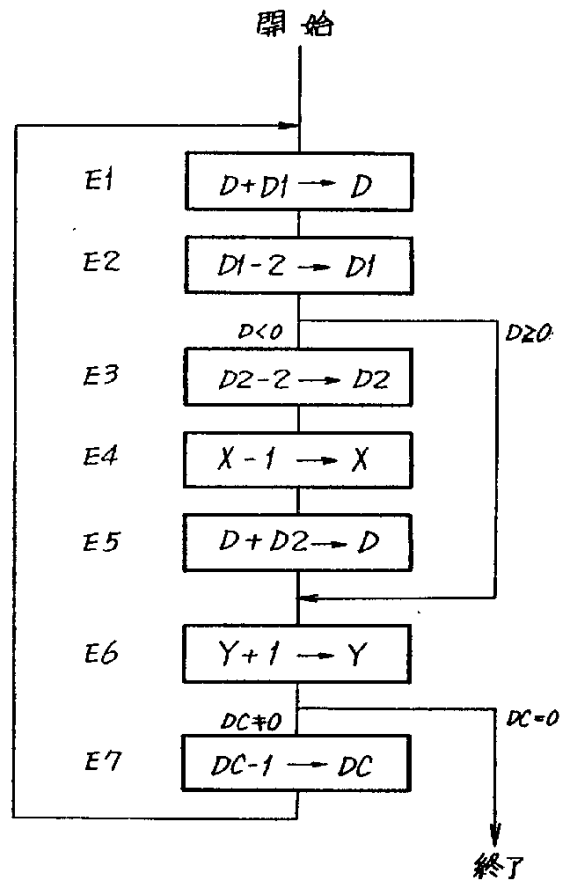
第 5 图



第 6 图



第 7 図



第 8 図