(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) Publication number: 0 372 810 A2

(12) # EUROPEAN PATENT APPLICATION

(21) Application number: 89312409.9

(51) Int. Cl.⁵: B41J 2/17

(22) Date of filing: 29.11.89

(30) Priority: 02.12.88 US 279395
02.12.88 US 279483
08.12.88 US 279904
08.05.89 US 348842

(43) Date of publication of application:
13.06.90 Bulletin 90/24

(84) Designated Contracting States:
DE FR GB NL

(71) Applicant: SPECTRA-PHYSICS INC.
3333 N. First Street
San Jose California 95134(US)

(72) Inventor: Gatten, Ronald A.
2794 Calle Alegre
Pleasanton California(US)
Inventor: Joba, Lawrence R.
444 Western Drive
Santa Cruz California 95060(US)

(74) Representative: Jones, Ian et al
POLLAK MERCER & TENCH High Holborn
House 52-54 High Holborn
London WC1V 6RY(GB)

(54) **Inkjet printers.**

(57) In an ink jet printer, the time that has elapsed since the ink jet cartridge last printed is timed and after one hour has elapsed with no printing, all the nozzles of the cartridge are caused to print, to prevent clogging of the nozzles. Also, a line of constant average dot density along its length is plotted, regardless of the orientation of various segments making up the plotted line. For any one increment of chart movement, each point in the other dimension is plotted only once, to conserve ink and prevent heavy buildup of ink on the chart. When the printer is in plot mode, the nozzles are used in sequence, instead of always using the same nozzle, so as to lengthen the life of the cartridge.

EP 0 372 810 A2

INKJET PRINTERS

This invention relates to an ink jet printer and to a method of operating an ink jet printer.

Ink jet printers are well known and a typical printer 10 is shown in Figure 1 of the accompanying drawings, in which:

Figures 1 and 2 show printer systems consistent with the prior art,

Figures 3 and 4 show prior art printing, and

Figure 5 shows a prior art cartridge.

1. Priming.

Hewlett Packard manufactures the Hewlett Packard thermal Ink-jet print cartridge 12, which may be used as the printing component of the printer 10. The Hewlett Packard thermal Ink-jet cartridge 12, well known in the art, is a self-contained disposable printhead suitable for a wide variety of dot matrix printing and plotting on a medium such as paper 13. The cartridge 12 includes a pressure contact electrical interconnect for ease of replacement, and can be used in any orientation.

Ink capacity of over ten million dots (dots are the ink drops ejected by the cartridge so as to print on the medium) gives the twelve nozzles 14 etc. of the thermal ink-jet cartridge 12 the capacity of typically 500,000 characters (depending on the font) or extensive graphics. The thermal ink-jet cartridge prints on a variety of papers (i.e., media). The non-contact operation allows variation in distance to the media when printing.

As is well known in the art, however, such ink jet cartridges as described above are subject to the problem of viscous plug formation. A viscous plug is a plug of ink in the exit portion of a print cartridge nozzle 14 in which evaporation of the liquid ink in the cartridge causes a thickening of the ink near the nozzle exit, interfering with ink dot ejection. This plug can occur when the print cartridge is idle (i.e., not firing) for a period of time, normally several days, depending on the ambient temperature and humidity conditions. The viscous plug can often be removed by printing. Plug formation is worst in hot and dry conditions but the plugs are most difficult to clear in cold and dry conditions. Plugs cause poorly formed dots or no dots at all and so degrade print quality.

In the prior art, the printer 10 includes a blotter 16 located to the left of the left-most printing location 18. The printhead guide track 20 contains a ramp (not shown) to lift the cartridge 12 over the blotter 16. Upon system initialization, the cartridge 12 is moved to the blotter 16 and fires all nozzles 14 for 160 times in succession, to perform the prefiring algorithm.

A common application of ink jet printers is as a printer system for use with a personal computer such as is indicated at 20 which provides data to printer 10 on channel 21. Under these circumstances, the printer 10 is usually turned off when not in use for long periods of time, thus making any sort of periodic priming impossible. Automated periodic priming under power on conditions even where possible has the drawback in most personal computer applications that it might disconcert the operator, due to unanticipated movement of the print head during the priming operation.

2. Line Density.

Ink jet printers as shown in Figure 1 are readily adapted for use as plotters (see Figure 2), in which application the printer 110 is often used for example as a real time plotter to plot data provided by a host computer 112 or instrument. Ink jet plotters operate by expelling tiny dots of ink from an ink supply through orifices 113, etc. (called nozzles) onto a medium 114 such as a piece of paper. The ink supply and orifices are typically incorporated into a print cartridge 116, which is mounted on a carriage 118. One well known ink jet cartridge is the Hewlett Packard Thermal Ink-jet print cartridge. In a typical printer 110, the carriage 118 moves back and forth along a guide rail 120 under the control of a conventional stepper motor 122. The paper 114 is advanced through the printer by means of a conventional paper tractor typically driven by a second stepper motor 124.

The ejection of the ink droplets, the movement of the carriage, and the advancement of the paper are conventionally all under the control of a microcontroller 126 installed in the printer 110. The microcontroller 126 typically includes ROM 128 (read only memory) which stores a computer program for operation of the printer 110.

Use of such a dot type printer whose carriage and medium are moved in steps by a stepper motor is satisfactory for printing text, but poses problems when used for plotting charts, especially when the plotting is on a real time basis.

In a typical printer 110, the carriage stepper motor 122 moves the carriage 118 back and forth along a 7.25 inch (18.4 cm) length of the guide rail 120. The carriage stepper motor uses for example 2000 steps to move the carriage this length; 2000 steps ÷ 7.25 inch equals 276 steps per inch (2.54 cm) along the guide rail. The direction of the guide rail 120 is designated as the Y axis. However along the other axis, designated the X axis, at right angles to the guide rail 120, the printer prints for example 630 dots per inch (2.54 cm).

Figure 3 illustrates the resulting deficiency of the prior art. Line segment a-a, along the X axis (630 dots per inch), is more densely printed and thus appears darker than does line b-b which is more nearly parallel to the Y axis (276 dots per inch). Thus, this deficiency of the prior art results in plotted lines with ink densities differing from one line segment of the plot to another, depending on the angle relative to the axes of each segment. This is undesirable, especially since other kinds of plotters are available that do not use ink dots and stepper motors and so do not have these deficiencies.

3. Over Inking.

Plotters of various kinds are well known in the art. One kind of plotter is used to record data as output from scientific and industrial instrumentation. Typically as shown schematically in Figure 2, this kind of printer-plotter 110 operates as a strip chart recorder, with a roll of paper 114 (or other media) fed through the plotter 110, and a carriage 118 including a printing device such as a print cartridge 116 moving back and forth on a guide rail 120 along a Y axis as shown relative to the medium 114. Thus the X-axis as shown typically represents the first variable, time, and the second axis is the Y-axis, as defined by the direction of carriage 118 movement, and typically represents a second variable such as a signal.

The printer-plotter 110, as is typical of this type of device, advances the medium 114 (e.g., paper) in the X direction through the printer-plotter 110 in steps, not continually. The paper 114 is advanced typically by means of a stepper motor 122. When doing real time data plotting, typically the data points are sent from the system CPU 112 (i.e., the instrumentation or computer) to the plotter 110 every tenth of a second, or 600 times per minute. The chart speeds (i.e., the speed of the medium 114 through the plotter) available range typically from one tenth of a centimeter per minute to 20 centimeters per minute. Thus the medium 114 advances for example at 630 steps to the inch on the time (X direction) axis or 248 steps per centimeter. Since data input points are provided at, for example, 600 times per minute, (600 points/minute) ~ - (248 steps/cm) is equal to 2.42 centimeters per minute. This means that for this example any time the medium 114 is moving faster than 2.42 centimeters per minute, the medium 114 will take at least one step in the time axis for each data point. When moving slower than that speed there will be two or more data point times for some time steps. If the printer 110 is advancing the medium 114 at its slowest speed, 0.1 centimeter per minute in the example, there will be twenty-four data points plotted for one step in the time axis. Thus one step in the time axis will have twenty-four data points plotted on one time increment. Low level noise motion on the signal axis can cause repeated excursions of the carriage 118 over the same line even if no new data is plotted, providing undesirably twenty-four times as much ink on the medium 114 as needed. This gives a very dark plot line, the ink of which tends to run and so undesirably produces an uneven and smeared plot line as shown in Figure 4.

4. Ink Jet Failure.

The above-described Hewlett Packard thermal ink-jet cartridge is a disposable unit 220 (see Figure 5) which contains both ink supply and ink ejection means. The cartridge 220 consists of a liquid ink supply in a bladder 221, twelve ink-ejecting nozzles (i.e., jets) 222a, 222b, etc., and twelve corresponding thin film resistors (not shown) for applying ink selectively to the nozzles. The resistors are located directly below each jet 222a, 222b, etc. An ink-drop ejection process begins by heating the resistor of a selected ink-jet with a short electrical pulse. Within a few microseconds, the ink above the resistor is vaporized. The vapor bubble grows rapidly and imparts momentum to nonvaporized ink above the bubble. Some of this nonvaporized ink is ejected through the jet orifice 222a at velocities exceeding ten meters per second. The jet 222a is then automatically refilled with ink by capillary action.

The ink supply is contained in a synthetic rubber bladder 221 located immediately behind a printhead substrate 2. The bladder 221 is designed to maintain a relatively constant back pressure at the jets 222a, 222b, 222c, etc. which is high enough to refill the jets after firing but low enough so ink is only expelled when desired.

The cartridge 220 also includes locating pins 224, cover 225, resistor array electrical contacts 226a, 226b, etc., and body 227.

When conventionally used for plotting graphs, charts, etc., in scientific, industrial, and similar applications, the printer is operated in "plot mode" and in this plot mode only one of the twelve jets is used.

This heavy use of one jet in plot mode puts a disproportionate amount of "stress" on that one jet. Such stressing of one jet can result in higher failure rates of the cartridge. It is believed that there is more than one failure mechanism at work in what is termed "stress". First, there is the cycling of the resistor associated with the one jet used in plot mode which vaporizes the ink. The thermal cycling provides a higher than desired failure rate due to fractures in the resistor which can occur by way of the thermal expansion and contraction of the resistor.

Second, there is a mechanism described in the Hewlett-Packard Journal of May 1985 (page 32) as "kogation". Kogation is "the plaque buildup found on the resistors of a thermal ink-jet head after several firings of the system. It can cause the head to fail by insulating the resistor from the ink supply, which reduces bubble generation. The major source of kogation is the ink. By modifying the ink material appropriately, kogation can be varied from rapid buildup to excessive erosion. In the ThinkJet ink, formulas have been optimized to provide a very slow buildup of plaque."

Thus the prior art method of plotting tends to cause failure of the one jet used for plotting, due to thermal cycling and kogation on that one jet. This failure of one jet renders the entire cartridge useless for printing or plotting well before all the ink in the cartridge has been used. Thus the prior art method of plotting causes cartridge failure resulting in extra expense and possible loss of valuable plot data due to cartridge failure.

SUMMARY OF THE INVENTION

In one embodiment the present invention is a method to prevent viscous plugs from clogging the nozzles of an ink jet print cartridge when the printer is idle (i.e., not printing). The method preferably involves using a microcontroller (which is conventionally provided in many ink jet printers) as a timer to keep track of the amount of time that has elapsed since the print cartridge last printed. After a predetermined amount of time of such idleness, the microcontroller directs the printer to print one or more ink dots from each nozzle, thus priming all the nozzles. The preferred predetermined amount of time allowed to elapse is about one hour. The priming of the print cartridge, in accordance with this embodiment, is infrequent enough so that it uses only a small amount of the ink supply in the cartridge. This is so even if the cartridge sits idle for an extended period, such as a month, while periodically priming.

An object of a second embodiment of the present invention is to avoid the prior art method of printing one ink dot for each step in either axis direction. The present invention controls the printer so as to eliminate the above described prior art line density differences. In this embodiment, the method of the invention involves modifications to the conventional computer program in the printer microcontroller. The present invention achieves its object by providing a substantially constant average spacing in dots per inch along each line segment, regardless of the orientation of the line segment relative to the axes. Therefore a fractional value is computed for the current line segment being plotted based on the angle of the line segment and the dot density requested. For each step taken on the major axis (i.e., that axis having the greater number of steps for the current line segment), the fraction is added to an accumulator. When the accumulator overflows to a positive value, a dot is printed and the accumulator is set back to a -1 fractional value. The fractional value is based on the dot density divided by the cosine of the angle of the line segment. Line segments printed in accordance with the present invention are shown in Figure 6. Note that both line segments c-c and d-d are of the same density, i.e., have equal constant average dot spacing.

In accordance with a third embodiment of the invention, by turning off the ink supply to the plotter when plotting over previously plotted points, it is possible to save ink and to prevent a heavy ink buildup on the medium. Furthermore, by not moving the plotter carriage (i.e., the print cartridge) when plotting over previously plotted points, the plotting may be speeded up significantly during a replot when the data are recorded and then plotted at a faster than real time rate. This reduces the overlapping of multiple plot points by plotting a particular point only once. This is accomplished by computer software, preferably resident in the plotter microcontroller, which turns off the ink supply when receiving data to plot a particular point the second (or third or fourth etc.) time, and also does not even move the carriage when receiving data from the instrumentation or computer to plot a particular point the second (or third or fourth etc.) time. Figure 7 shows a plot line produced in accordance with the present invention in contrast to the prior art plot line of Figure 4.

In accordance with a fourth embodiment of the present invention, the computer program for controlling an ink-jet printer includes the capability, when the printer is in plot mode and so is printing from only one jet at a time, to alternate jets (preferably sequentially) upon the occurrence of a predetermined event. Preferably that event is the completion of plotting of the current graph, but the event could be another occurrence such as completion of a line of print. Where the printer-plotter prints by means of a commercially available ink-jet cartridge having twelve jets, ten of which are commonly used for printing ten-dot high characters, only those ten jets are included in the sequence of changing jets. The present invention thus distributes the stress of plotting over several of the jets, largely eliminating the problem of cartridge failure due to stress.

The invention is further described below, by way of example, with reference to the further figures of the accompanying drawings, in which:

Figures 6 and 7 show printing in accordance with the invention, and

Figures 8, 9, 10 and 11 show computer programs.

1. Priming.

In accordance with the present invention, in the preferred embodiment an ink jet printer 10 (see Figure 1) similar to that of the prior art having an eight bit microcontroller 22 such as the Intel 8052 microcontroller (of the 8031 microcontroller family) has a program conventionally installed in microcontroller ROM (read only memory) 24 to control the printhead 12 via channel 26.

This embodiment requires that the printer 10 be powered up (i.e., turned on) at all times, so that the microcontroller 20 has power and so can function as a timer, and also so that the printer 10 can periodically be primed by printing ink dots. Thus this embodiment is perhaps most suitable for industrial or scientific applications where the printer is normally left on for long periods. Instrumentation other than a personal computer might be substituted for computer 20 shown in Figure 1. The prevention of clogged nozzles is also of great importance in applications where the printer 10 is being used as a printer-plotter to record real time events, and the printer output may constitute the only record of the events. In this case a failure of the cartridge due to clogged nozzles would result in a loss of all data.

With reference to the flow chart shown in Figure 8, this embodiment operates as follows. The conventional prior art power-up sequence for the printer begins at START at 330, which is followed by the conventional System Initialization at 332, and then the prior art priming of the nozzles upon start-up by setting PRIMECOUNT to 160 (in one variation recommended by Hewlett Packard) at 334, followed by normal processing and plotting of data at 336, 338. The one difference in this sequence over the conventional printer control program is the introduction of the DOPRIME flag variable, which is turned on at 334.

The next column of the flowchart begins with the conventional NEXTDATA subroutine at 340, 342, which gets data from the printer buffer and translates the data so as to print the next data segment. If the DOPRIME flag is on at 344, then the PRIME subroutine at 346 is called.

The PRIME subroutine at 346 actually directs the printhead to move to the blotter and to fire the nozzles to prime them as described below. The PRIME subroutine at 346 is repeated as long as the DOPRIME flag at 344 is on. After priming is completed, the next data is input conventionally at NEXTDATA at 348. If there is more data, the PRIMECOUNT variable is checked at 350. If PRIMECOUNT is greater than zero, the path to the left is taken and PRIME is called again at 354, after the carriage position is saved. After priming again at 354, the carriage is returned at 356 to the position saved at 352.

The purpose of the loop 350, 352, 354, 356 is that if data is output to the printer before one hour has elapsed, but after three minutes have elapsed, there will be a number in PRIMECOUNT at 350 of 1 to 19. Then the PRIME subroutine will be called at 354, and PRIME executes the number of cycles equal to the number in PRIMECOUNT.

If PRIMECOUNT is zero at 350, then conventionally the data is set up and output to be printed at 358.

The third column of the flowchart shows the details of the PRIME subroutine at 360. First, the carriage is moved over to the blotter position without any firing of the nozzles at 362. Then a delay time of 800 μ seconds between nozzle firings is set up at 364. This is the reciprocal of the conventional 1250 firings per second. Then the PRIMECOUNT value is checked at 366. If the value is greater than zero, PRIMECOUNT is decremented by one and all twelve nozzles are fired once at 368. Then a step timer (for the priming cycle) is enabled at 370. This loop 366, 368, 370 is repeated for PRIMECOUNT number of times. After that, the loop is exited at 360 and the DOPRIME flag is cleared at 372. Then the carriage is returned to its prior position.

The last part of the flowchart shows the data timer (including subroutines DATHTLIM, DATHL3, and DATHL4), as implemented in the Data Timer Interrupt at 374. After a time equal to three minutes without data being input to the printer at 376, the prime counter PRIMECOUNT variable is incremented at 378 and the variable DATAHITIME, which times up to three minutes, is reset to zero at 378. When PRIMECOUNT reaches twenty (one hour) at 380, the DOPRIME flag is turned on at 382 which (as described above) causes priming followed by a return at 384. If the PRIMECOUNT is not yet 20, at 380, there is also a return at 384.

The assembly language program for the preferred embodiment of the invention thus adds subroutines and code to the conventional printer control program. As described above, the elements added to the conventional program include: (1) those which implement the date timer, subroutines

DATHTLIM, DATHL3, and DATHL4; (2) that which implements priming, subroutine PRIME; (3) addition of the DOPRIME flag to system initialization; (4) modifications (so as to call PRIME and restore the carriage to its position after priming) to the conventional subroutine NEXTDATA.

2. Line Density Control.

In accordance with this embodiment, the program installed in the microcontroller of the ink jet printer-plotter includes additional subroutines to control line density. The process in this embodiment is performed in several steps for each line segment to be plotted. First, it is necessary to determine for each line segment to be plotted in which axis (X or Y) the lesser number of steps are to be taken. Then a ratio is calculated of the number of steps to be taken in the axis with the lesser number of steps to the number of steps to be taken in the other axis. This value is stored in a variable called RATIO, as the numerator less one of a fraction having 256 as the denominator. Thus a RATIO value of 255 means that the fraction is one; a RATIO value of zero means that the fraction is 1/256.

The tangent of the angle of the line segment relative to the major axis is then calculated as being equal to (RATIO + 1/256) * (steps/inch major axis)/(steps/inch minor axis). The major axis is the axis in which the plotter takes more steps in plotting the particular line segment, and the minor axis is the other axis.

The dot flow rate to obtain the maximum line density, (i.e., the number of dots per inch) for the line segment is then calculated as being 1/cosine of the angle whose tangent was calculated above. Therefore the line density is equal to ((steps/inch for minor axis/(steps/inch of axis having a greater number of steps/inch))/cosine (arctangent ((RATIO + 1/256) * (steps/inch primary axis)/(steps/inch secondary axis)))). In order to simplify the calculations, the trigonometric values are obtained by table lookup. This line density value is the ratio of the space between steps on the major axis to the average space between dots along the line segment vector. This line density value preferably is then multiplied by a value called DDENSE (dot density). Dot density is a value provided so as to plot darker or lighter lines. In the preferred embodiment, a choice of eight line densities is provided.

The product of line density and DDENSE is called the dot flow rate (DFLOW). This DFLOW value is added to a DOT-RATIO accumulator to determine output (i.e., printing), of individual dots; a dot is printed whenever the DOT-RATIO accumulator overflows.

The above-described method is illustrated in the flowchart shown in Figures 9A and 9B. First, for each line segment, in a conventional subroutine called GONOW for setting the next line segment motion of the carriage and medium at 450, the value of RATIO is calculated as seen in Figure 9A. First the program determines for a particular line segment whether that line segment has more steps in the X axis direction or in the Y axis direction at 452. If there are more steps in the X axis direction, the X axis is designated the major axis, and the flag variable NFASTAX is assigned the value of one at 454. If there are more steps in the Y axis direction, then Y is the major axis and NFASTAX is assigned the value of zero at 456. The value of RATIO is then computed at 458 instead of 460.

These two values--NFASTAX and RATIO--are then provided to the subroutine GETDF at 462, which calculates the dot flow versus step ratio, DF.

GETDF first checks that the system is in plot mode at 464 and that the next pen is on (meaning that ink output is requested by the host for the next line segment) at 466. If the major axis is the Y axis at 468, then the number for the full plot density is obtained from table YDFTABLE at 470. If the X axis is the major axis, then the number is obtained from table XDFTABLE at 472. The value AB obtained from table YDFTABLE or XDFTABLE is then multiplied by a number obtained from a third table, DDTABLE at 474, which represents the dot density as specified externally.

The resulting product is divided by two and is the value of NDFLOW at 474. The program then checks to see if the pen (i.e., ink supply) is off at 476; if not, because plotting is still in progress, the program exits; if yes, then a new series of continuous line segments is being initiated and so DOT-RATIO is set equal to -(NDFLOW + 1/2) at 478 so as always to overflow the accumulator on the first cycle.

Tables YDFTABLE and XDFTABLE are lookup tables that save calculations of the relevant trigonometric functions. For each table, the independent variable is the value of RATIO. For YDFTABLE, the dependent variable is, in the preferred embodiment, equal to:

128/cos (arc tan (((RATIO + 1)/256) * HSTIN/VSTIN))) where HSTIN is the number of horizontal steps per inch taken by the stepper motor moving the paper and VSTIN is the number of steps per inch for the stepper motor moving the carriage. The value of 128 is chosen because it is one half of the maximum value of RATIO.

For table XDFTABLE, the dependent variable is equal to:

128 * (HSTIN/VSTIN)/COS(ARCTAN (((RATIO + 1)-/256) * VSTIN/HSTIN))).

To give an example of the results of the calculations, a value of 128 for the dot flow will result in one dot of ink plotted for each step taken on the major axis. Since the steps in the example given above are closer together on the minor axis, only 56 dots are plotted on the minor axis for each 128 steps taken on the minor axis in order to obtain maximum ink density. The calculation is: 128 steps * ((276 steps/inch)/(630 steps/inch)) equals 56. Thus the average dot spacing on both the major and minor axes will be equal.

The plotting of dots is controlled by the program as shown in the second part of the flowchart by the subroutine GETDOTS whose purpose is to set up the ink dot pattern (i.e., determining which nozzles on the print cartridge will print at a particular step.) GETDOTS is called by another subroutine, NEXTPLOT, which is a conventional plotting subroutine for one step of the carriage and/or paper motion and inking.

In GETDOTS at 482, as seen in Figure 9B, first the variable NEXTDOTS is cleared (i.e., set to equal zero) at 484. Then the program checks to see that pen is on at 486. If the pen is off, GETDOTS is exited at RETURN at 506; otherwise, the value of DFLOW (dot flow) is added to the value of DOT-RATIO at 488. Note that DOT-RATIO is an input variable provided by the previous subroutine GETDF. If there is no overflow at 490 (i.e., no carry) in DOT-RATIO, then the subroutine is exited at 506. If there is an overflow, then 128 is subtracted from DOT-RATIO at 492. Then the pattern for the dots to be printed is put into variable NEXTDOTS, as follows.

If the double dot flag (DDENSE.3) is on at 496, then subroutine ONEDOT is called at 496 and ONEDOT puts the dot pattern for dot number DOT-SELECT-1 in the high byte of the variable NEXTDOTS at 498. ONEDOT is a conventional subroutine for determining the dot pattern, which means determining what signals will be provided to the print cartridge to fire a particular nozzle. Then ONEDOT is called at 500 and ONEDOT puts the pattern for dot number DOT-SELECT in the low byte of the NEXTDOTS at 502.

Then subroutine ADDDOTS at 504 is (optionally) called to add the number of dots to be plotted to the dot total kept in ADDDOTS.

The actual plotting is then performed conventionally using variable NEXTDOTS as determined above.

3. Ink Saving Method

This embodiment to save ink and prevent ink build-up is illustrated in flowchart form in Fig. 10.

The subroutine GONOW is part of the conventional printer control program and is normally called for output of a line segment, output of a printed character, or a non-output motion of the printer carriage. This embodiment establishes an exception to that motion and output when the requested motion makes no change in the X-axis (time) and simultaneously the Y-axis (signal) motion overlays an already plotted line segment.

The inputs to GONOW are conventionally (1) the new end point defined by the parameters NEXTX and NEXTY, for the new values on respectively the X and Y axes; (2) a plot mode flag (PLOT) which is off for printing and on for plotting; (3) a pen status flag NEXTPEN which indicates if the pen (i.e., the ink supply for actual plotting) is on or off; (4) XTARG and YTARG which define the end point for previous line segment on respectively the X and Y axes.

Thus the GONOW subroutine is conventionally called by specifying the new end point (NEXTX, NEXTY), a plot mode flag (PLOT), and a pen status flag (NEXTPEN). The origin of the line segment is the end point of the previous line segment (XTARG, YTARG).

In order to establish the required exception logic made in accordance with the present invention, four variables are used. These variables are the two end points of the previously inked line, a variable for saving the location where the instrumentation or computer thinks the carriage is, and a flag. Thus the four variables are:

YMIN - The minimum Y excursion of the previously inked line.

YMAX - The maximum Y excursion of the previously inked line.

SAVEYPOS - The current logical Y location of the line.

TWOSTAGE - A flag to indicate activation of the mode of the present invention.

The most direct way through the flowchart of Figure 10 goes straight from the beginning at INKSGO 520 down to the return 532. On that path there is no new YMIN (i.e., YMINIMUM) at 528 and no new YMAX (i.e., YMAXIMUM) at 530 and no motion in the X-axis direction at 524. If there is a motion in the X direction at 524 or if the pen (i.e, ink supply) is off or the plotter is not plotting at 522, the YMIN and YMAX are reestablished at 534 and are set to be the current value of Y. In other words, the previous bounds on the line segment are removed.

Going to the point where there is a new YMIN at 528, 540 or a new YMAX at 530, 542, the program then checks and if NEXTY (the next value of Y) is equal to the YTARG at 546 (which is the physical carriage location on the Y-axis), that NEXTY value is probably the last end point so the program sets NEXTY equal to SAVEYPOS at 552 which initially was set equal to NEXTY at the beginning 520, so normally SAVEYPOS will be equal to NEXTY on this path. On the escape path SAVEYPOS will have another value.

Then GONOW at 554 is called to draw the line segment. If NEXTY was not equal to YTARG at 546 then NEXTPEN (the pen status) is saved and NEXTX is saved at 548 and GONOW 550 is executed. That will step the print carriage to the point that had been the old Y maximum or minimum, which is the end point of the line segment. That end point is where the pen (i.e., the ink supply) last turned off. Then the program gets NEXTY, and gets NEXTPEN at 544, and sets NEXTY equal to SAVEYPOS at 552 which was saved up at the beginning 520. Now the program goes from what is the end point of the line that was drawn to the new minimum or maximum value to make the line longer. That new end point location is the next YTARG. Thus the carriage is at the end of the new longer line, and the program returns.

The only other possibility is either a user turns the pen off or the plotter stops plotting at 522, or the plotter moves along the X-axis at 524. All three of those cases cause the program to go to INKSG6 where the program will set new minimums and maximums equal to the present location at 534. If the TWOSTAGE flag at 536 is not on, that is the normal path when there is always an X motion for each Y motion, so the program goes to GONOW and draws the line segment at 554.

Thus the program is in the ink saver mode (i.e., TWOSTAGE flag on at 536) only if there is plotting and no X motion. So if there is X motion, or if the pen is turned off, or if there is printing (and not plotting), the program takes the path direct to GONOW at 554. Otherwise if the TWOSTAGE flag is on at 536, the program clears the flag and exchanges the NEXTY with SAVEYPOS at 538 which is the NEXTY from the last motion, or from the last time there was no next motion, which either could be an end point of a plot line established previously or else it is a point skipped over and not plotted because there was no new ink plotted. So the program now begins at the plot point in the middle of that line from which it is desired to draw the new line.

Then the program goes through the path beginning at 546. If NEXTY is not equal to YTARG at 546, the program goes through and will move the carriage to the point in the middle of that line with the pen turned off, and then draw the new line segment from that point in the middle to the location at the new X position and continue on from there.

Thus the program ensures that if the carriage is instructed to move up and down on the Y-axis it will not even move unless the movement will make the inked line longer. Then when the carriage leaves that inked line and goes back to wherever it was logically at the last motion, even if the carriage did not make the last motion, the carriage goes back to that point and draws the line to the new X location.

4. Ink Jet Rotation.

In accordance with this embodiment (see Figure 2), the conventional printer-plotter 110 includes a conventional ink-jet cartridge 116 and is controlled by a microcontroller 126 preferably of the commercially available (such as from Intel) eight-bit 8052 type. Medium 114 is conventionally movable in direction X (perpendicular to normal printing direction Y) by means of a platen 124. (In this example, "X" and "Y" are shown turned 90° from their usual geometric orientations, because graphs are often printed in this orientation.) Printer-plotter 220 prints the output of host computer system 112. Ink-jet cartridge 116 conventionally is movable in direction Y by carriage 118.

When the printer-plotter 110 is used in plot mode (i.e., to draw graphs), in accordance with the invention (see Figure 5) one jet such as 222a in the cartridge 220 is preferably used continuously under control of a computer program (not shown) installed (see Figure 2) in ROM (Read Only Memory) 128 in the microcontroller 126 until the graph is complete. When the next graph is initiated, a different jet such as 222b (see Figure 5) is used, and similarly, when the second graph is completed and another initiated, a third jet such as 222c is used. A graph (or graphic image) is typically one chart, or one picture. In other embodiments, the jets may be rotated more or less frequently such as at each line, half-line, after two graphs, or after a certain elapsed plotting time.

In the preferred embodiment of the invention, this alternating or "rotation" of jets 222a, 222b, 222c, etc. continues through the tenth jet (not shown) whereupon the first jet 222a is used again. Although the conventional cartridge 220 has twelve jets 222a, 222b, etc., available, preferably only the first ten jets are used in the rotation. This is because the lower two jets are typically only rarely used when printing text using the most commonly used character fonts. If a particular jet has clogged or otherwise failed, it is desirable to be able to diagnose the failure by observation of printed characters which typically are printed after the plotting of each graph using the ten main jets. If one of the two lower jets had failed, the fact of the failure would not be obvious by observing the character text, because the letters would be well formed. Thus it is preferred in the present invention not to use these two lower jets at all for plotting.

In the preferred embodiment, information as to which jet was used to plot the graph last completed is stored (see Figure 2) in the microcontroller 126. The information as to the last jet used is lost whenever the printer-plotter 110 is powered down in a conventional printer-plotter, and power must remain on (or nonvolatile memory provided) for the jets to be rotated for plotting of sequential graphs. Each time the printer-plotter 110 is powered up or the microcontroller 126 is reset, the rotation of jets preferably begins again with the same first jet 222a. In another embodiment, any of the jets may be the first jet.

The preferred embodiment of the present invention is described in more detail below with reference to Figure 11, which depicts in a flowchart the computer program conventionally installed in the microcontroller which carries out the ink-jet rotation. The following explains the computer program.

The actual change of ink-jet being used for plotting takes place when the printer-plotter has completed printing character text and is about to begin plotting, i.e., is going into plot mode. This is so as not to lose track of the location in the X direction (i.e., an index position) relative to the medium of the particular ink-jet being used for plotting. In print mode when all ink-jets are in use, the index position is always the same, while this is not the case in plot mode where only one jet is used for any one plot.

As is conventional, in the flow chart in Figure 11 the ovals depict the beginning and end of subroutines. The diamonds are decision points. The rectangles are calculations. The hexagons are calling of other subroutines. The text inside each shape describes the activity taking place.

Subroutine BACKLASH 660 is thus called by the conventional printer-plotter software for each line segment or character to be plotted. (Note that conventionally characters may be "plotted" in plot mode or "printed" in print mode.) The purpose of BACKLASH 660 is to correct the position of the medium relative to the jet to be used, and to account for any corrections needed in the vertical (X direction) movement of the medium due to change in direction of movement (i.e., moving up and down) of the platen or due to rotation (i.e., changing) of ink-jets during transitions between printing and plotting.

DOT_SHIFT is an input variable representing the total motion to be made by the platen to correct for ink-jet rotation, so that while plotting with different ink-jets the registration of the plotted line of ink dots in plot mode is maintained with respect to printed text in print mode.

Subroutine BACKLASH checks the value of variable DOT_SHIFT at 661, and if DOT_SHIFT is not equal to zero, then the next X direction is saved at 662. The next X direction is set equal to the sign of DOT_SHIFT at 663. If this is a change in the X direction, then subroutine ADDBACK is called at 670 to add or subtract backlash.

Then subroutine BLSUB 640 is called at 671 to update the X direction and move the platen a distance equal to the value of DOT_SHIFT plus the backlash distance. Then the original next X direction is restored at 679.

If there is a change in X direction at 672, then subroutine ADDBACK is called at 673 to add or subtract backlash. BLSUB 640 is called at 674 to update the X direction and move the platen the backlash direction without changing the position count.

As explained above, subroutine BLSUB 640 is called by the BACKLASH routine at 671 and 674. The purpose of BLSUB 640 is to cause the stepper motor (not shown) which controls the platen to move the number of steps in the accumulator (ACC) 641 in direction X without changing the position count at 643.

Subroutine DOT_ADJ 730 is called by the conventional printer-plotter software whenever new data to be plotted is received by the microcontroller. The purpose of DOT_ADJ 730 is to get the adjustment for dot position (i.e., ink-jet) and script (superscripts and subscripts) into variable DOT_SHIFT. DOT_ADJ determines if the printer-plotter has been requested to be in plot mode at 731 and checks to see if the printer-plotter is currently in plot mode at 732. If the printer-plotter is undergoing a transition between print mode and plot mode, a value is set in the accumulator at 733 to adjust for the printing of superscripts or subscripts. Subroutine SCRIP ADJ 700 is then called at 734.

The purpose of SCRIP ADJ 700 is to get the offset value for variable DOT_SHIFT to adjust for subscript or superscript printing. SCRIP ADJ checks for a subscript flag at 701 and a superscript flag at 702, then returns to subroutine DOT_ADJ at 734.

At this point, in DOT ADJ at 735 register B is set equal to the value of variable NEXT DOT, and the accumulator (ACC) is set equal to 12, if a plot to print transition is in effect. If a print to plot transition is in effect, the accumulator is set equal to the value of variable NEXT DOT and register B is set equal to 12 at 736, and then subroutine ALIGN DOT 690 is called at 737.

ALIGN DOT 690 looks up an actual dot count by calling subroutine GETALD 680 at 691 and 692. GETALD 680 looks up at 681 in a table ALDTABLE the dot alignment values for transitions between printing and plotting, based on the number of vertical steps per inch made by the platen in the X direction of movement of the medium, and on the spacing of the dots per inch as they are printed by the cartridge. A table such as ALDTABLE is used in the preferred embodiment, in place of a calculation of dot alignment, because there is not an integral number of steps in the X-direction (see Figure 2) taken by the platen 124 between the relative positions with regard to the medium 114 of adjacent ink-jets. Therefore a look-up table such as ALDTABLE is provided with pre-calculated dot alignment (i.e., relative spacing) data so as to properly index a particular jet to the medium 114.

ALIGN DOT 690 (see Figure 11) thus provides values for register B and for the accumulator ACC at 691, 692 and sets variable DOT SHIFT equal to the previous value of DOT SHIFT plus the total number of steps that have to be moved by the platen to take into account changes in dot position due to transitions between printing and plotting at 693.

Subroutine INCDOT 740 is called by the conventional printer-plotter software when the printer-plotter executes its first carriage return after a period of being in plot mode. (Conventionally, carriage returns are only executed in print mode.) INCDOT 740 at 741 increments the value of NEXT DOT by one, thus telling the printer plotter to plot the next plot (i.e., graph) using the next ink-jet.

Subroutine XPOSNORM 710 is only called in event of a power failure of the printer-plotter or a soft reset of the host system. In the event of a power failure, the printer-plotter will send its vertical position on the medium back to the host system. In the case of a soft reset, the printer-plotter will save that value. The purpose of XPOSNORM therefore is to ensure that when the printer-plotter is reset that it does not lose track of the vertical index, labelled X POSITION in 711.

The above description of the invention is descriptive and not limiting; further modifications to the described embodiments will be apparent to one of ordinary skill in the art in light of this disclosure and the appended claims.

Claims

1. A method of priming an ink jet printer system having timing means for measuring elapsed time, printing means with at least one nozzle for printing dots, and control means for controlling the printing means, characterised by the steps of:
measuring the amount of elapsed time since the printing means last printed; and
directing the printing means to print at least one dot from each nozzle after a predetermined time has elapsed without printing.

2. A device for priming an ink jet printer system comprising:
printing means for printing dots with at least one nozzle;
control means for controlling the printing means; and
timing means for measuring the amount of elapsed time since the printing means last printed;
characterised in that the control means comprises means for directing and printing means to print at least one dot from each nozzle after a predetermined time has elapsed.

3. A method of maintaining substantially constant line density of a line of dots plotted by a plotter on a medium, the plotter having:
printing means for plotting the dots, the printing means being moveable in steps of a substantially constant length in at least two directions perpendicular to each other relative to the medium; and
control means for controlling the printing means;
characterised by the step of plotting at least one segment of the line so as to have a substantially constant average spacing of the dots along such a line segment for any orientation of the line segment relative to the two directions.

4. A method as claimed in claim 3 comprising the step, prior to the step of plotting, of computing a ratio of the number of steps of the printing means to plot the line segment in the first direction to the number of steps of the printing means to plot the line segment in the second direction.

5. A plotter comprising printing means for plotting a line of dots on a medium, the printing means being movable in steps of a substantially constant length in at least two directions perpendicular to each other relative to the medium; and control means for controlling the printing means to plot at least one segment of the line so as to have a substantially constant average spacing of the dots along such a line segment for any orientation of the line segment relative to the two direction.

6. A method of plotting points on a medium for a plotter having ink supply means for supplying ink for plotting; a carriage moveable relative to the medium; and control means for moving the carriage and turning the ink supply means on and off;
characterised in that the control means turns off the ink supply means when plotting a point that has been previously plotted on the medium.

7. A method as claimed in claim 6 wherein the control means keeps the carriage stationary relative to the medium so as not to plot a point that has been previously plotted on the medium.

8. A plotter for plotting points on a medium comprising in supply means for supplying ink for plotting; a carriage movable relative to the medium; and control means for moving the carriage and turning the ink supply means on and off;
characterised in that the control means turns off the

ink supply means when plotting a point that has been previously plotted on the medium.

9. A method of using a plotter having a plotting means with a plurality of jets for plotting and a control means for controlling the plotting means, characterised by the step of alternating between two or more jets used for plotting upon the occurrence of a predetermined condition.

10. A method as claimed in claim 9 having the step of alternating the jets in a predetermined order.

11. A plotter having a plotting means with a plurality of jets for plotting and a control means for controlling the plotting means, characterised in that the control means effects alternation between two or more jets used for plotting upon the occurrence of a predetermined condition.

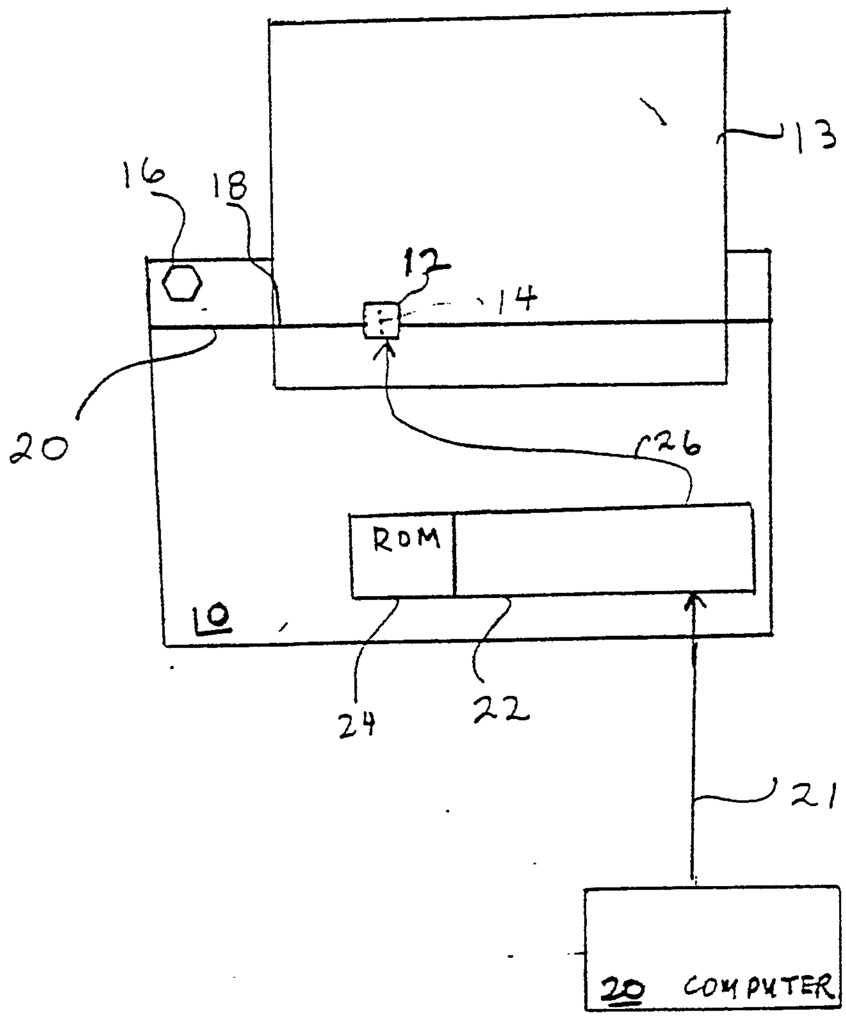

Figure 1

Prior Art

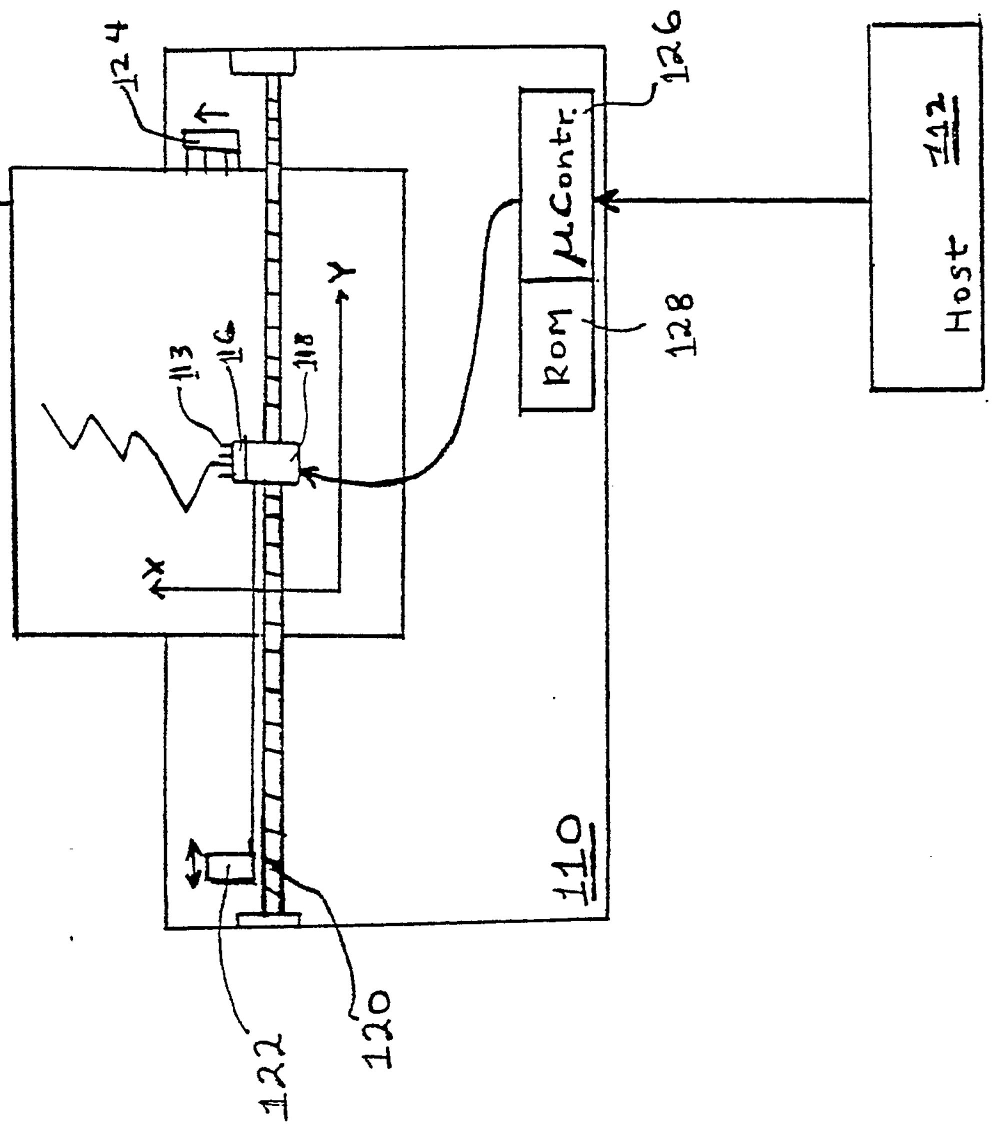

Figure 2
Prior Art

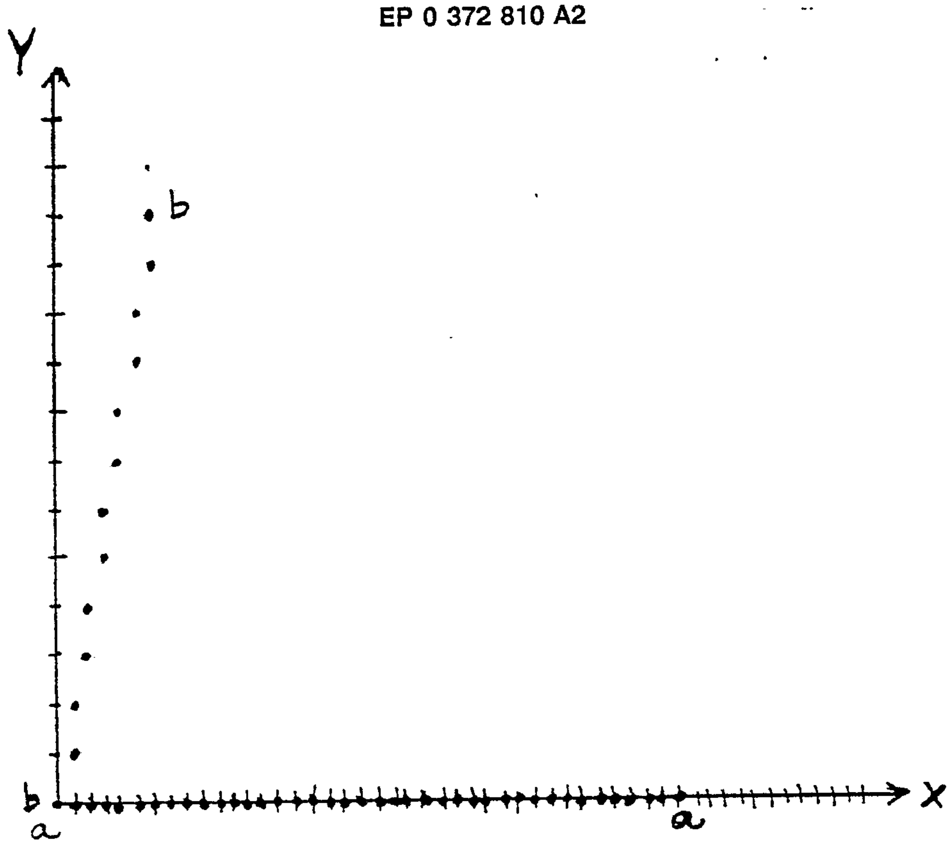

Figure3
Prior Art

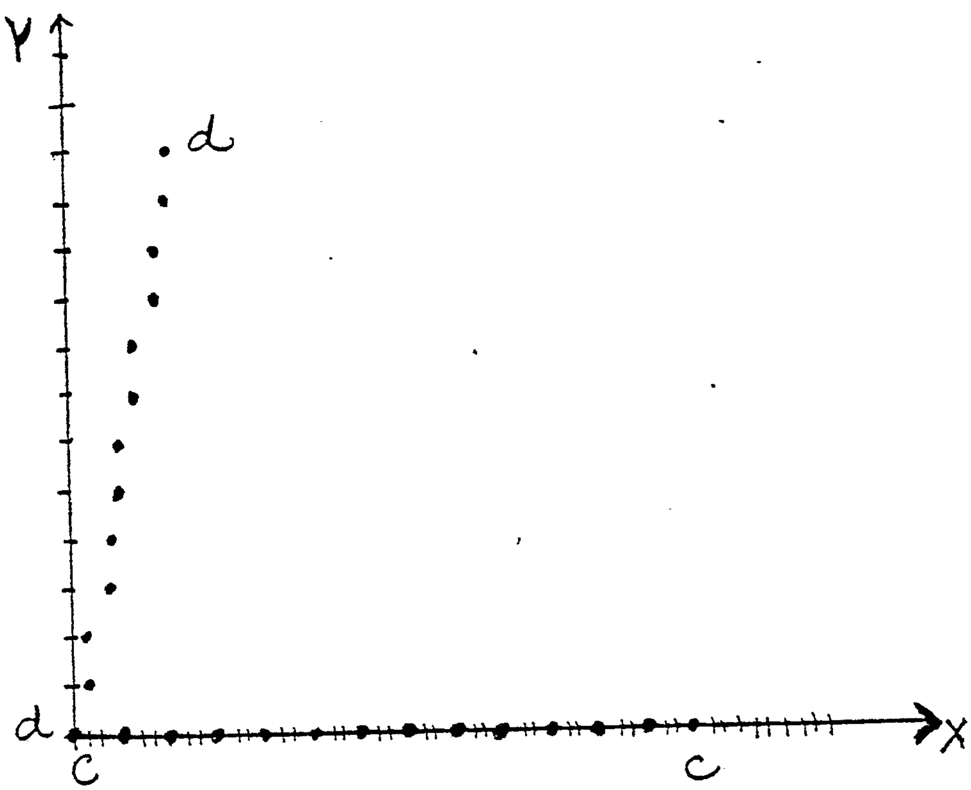

Figure 6

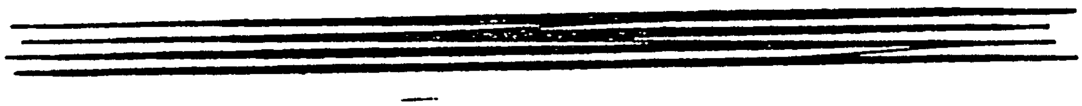

Figure 4
Prior Art

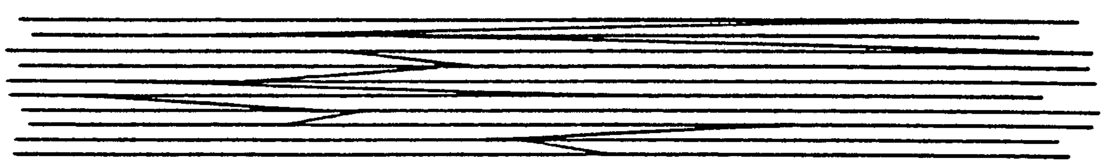

Figure 7

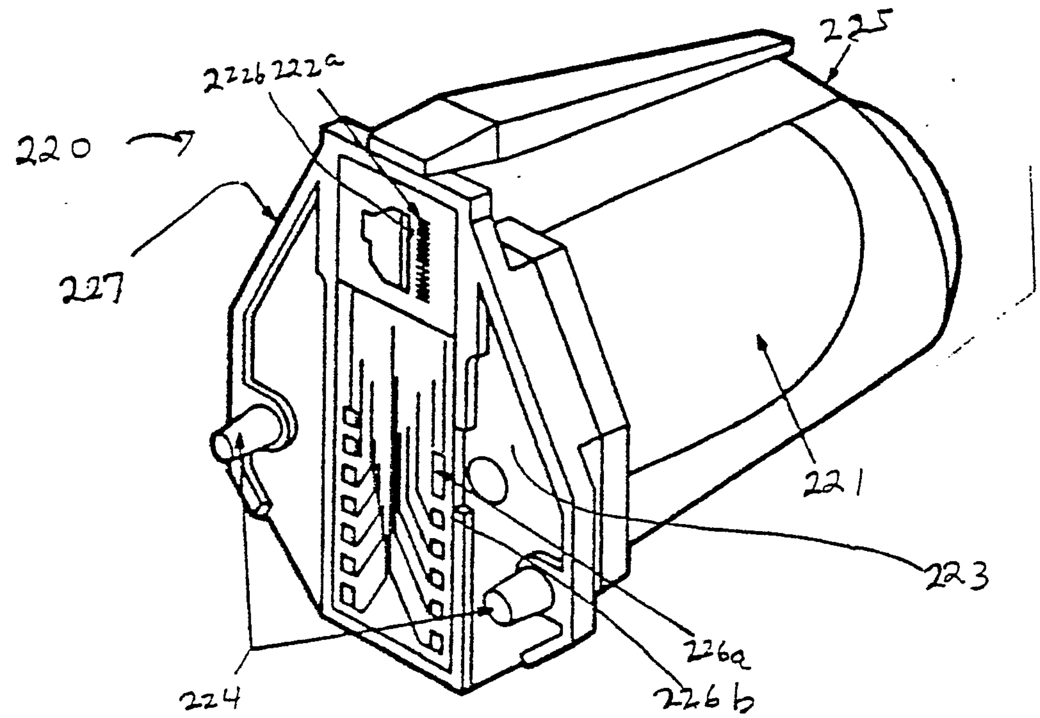

Fig. 5

Prior Art

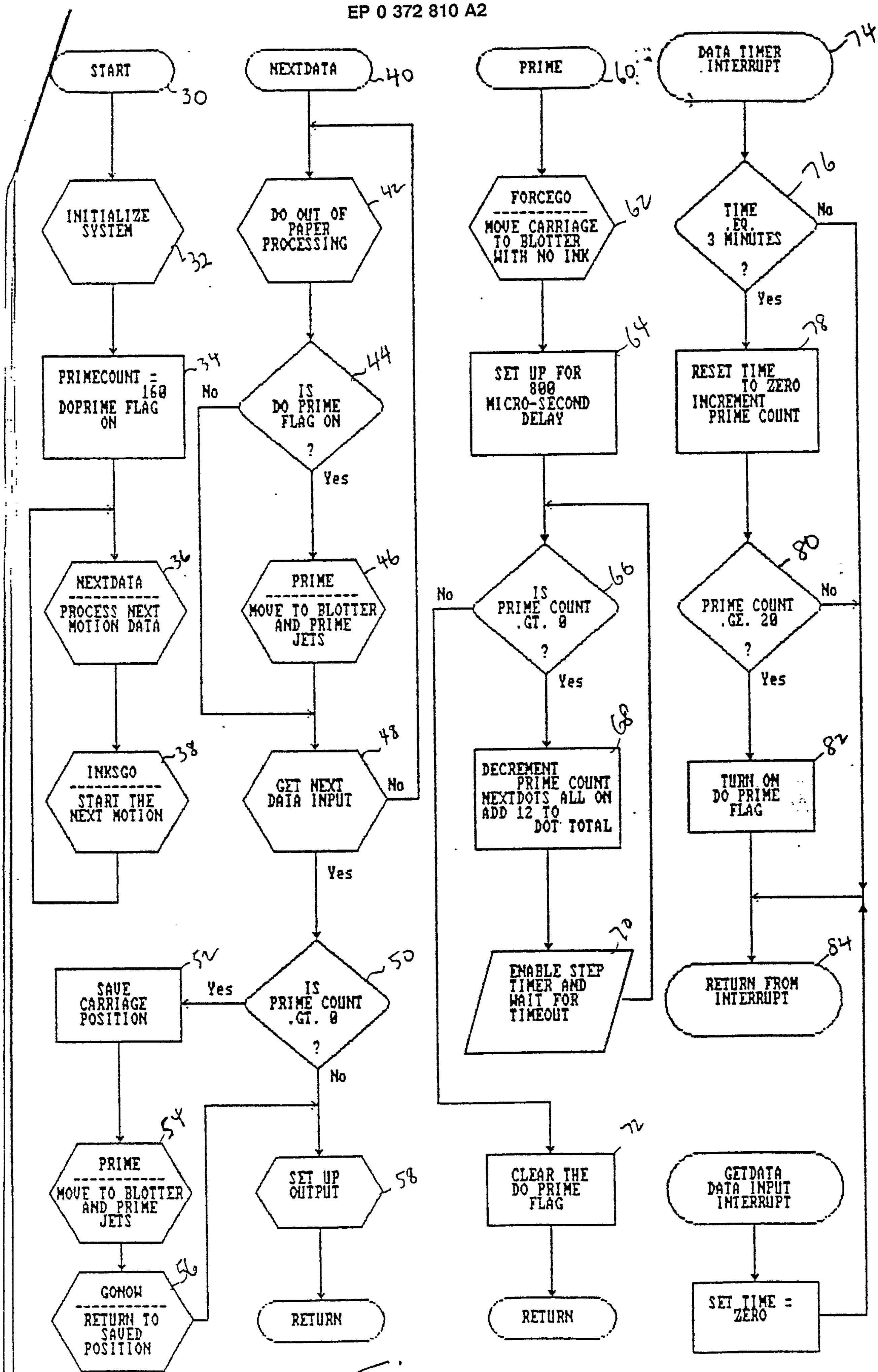
START
30
INITIALIZE SYSTEM
32
PRIMECOUNT = 160 DOPRIME FLAG ON
34
NEXTDATA
PROCESS NEXT MOTION DATA
36
INKSGO
START THE NEXT MOTION
38
NEXTDATA
40
DO OUT OF PAPER PROCESSING
42
IS DO PRIME FLAG ON ?
44
No
Yes
PRIME
MOVE TO BLOTTER AND PRIME JETS
46
GET NEXT DATA INPUT
48
No
Yes
IS PRIME COUNT .GT. 0 ?
50
Yes
No
SAVE CARRIAGE POSITION
52
PRIME
MOVE TO BLOTTER AND PRIME JETS
54
GONOW
RETURN TO SAVED POSITION
56
SET UP OUTPUT
58
RETURN
PRIME
60
FORCEGO
MOVE CARRIAGE TO BLOTTER WITH NO INK
62
SET UP FOR 800 MICRO-SECOND DELAY
64
IS PRIME COUNT .GT. 0 ?
66
No
Yes
DECREMENT PRIME COUNT NEXTDOTS ALL ON ADD 12 TO DOT TOTAL
68
ENABLE STEP TIMER AND WAIT FOR TIMEOUT
70
CLEAR THE DO PRIME FLAG
72
RETURN
DATA TIMER INTERRUPT
74
TIME .EQ. 3 MINUTES ?
76
No
Yes
RESET TIME TO ZERO INCREMENT PRIME COUNT
78
PRIME COUNT .GE. 20 ?
80
No
Yes
TURN ON DO PRIME FLAG
82
RETURN FROM INTERRUPT
84
GETDATA DATA INPUT INTERRUPT
SET TIME = ZERO

Figure 8

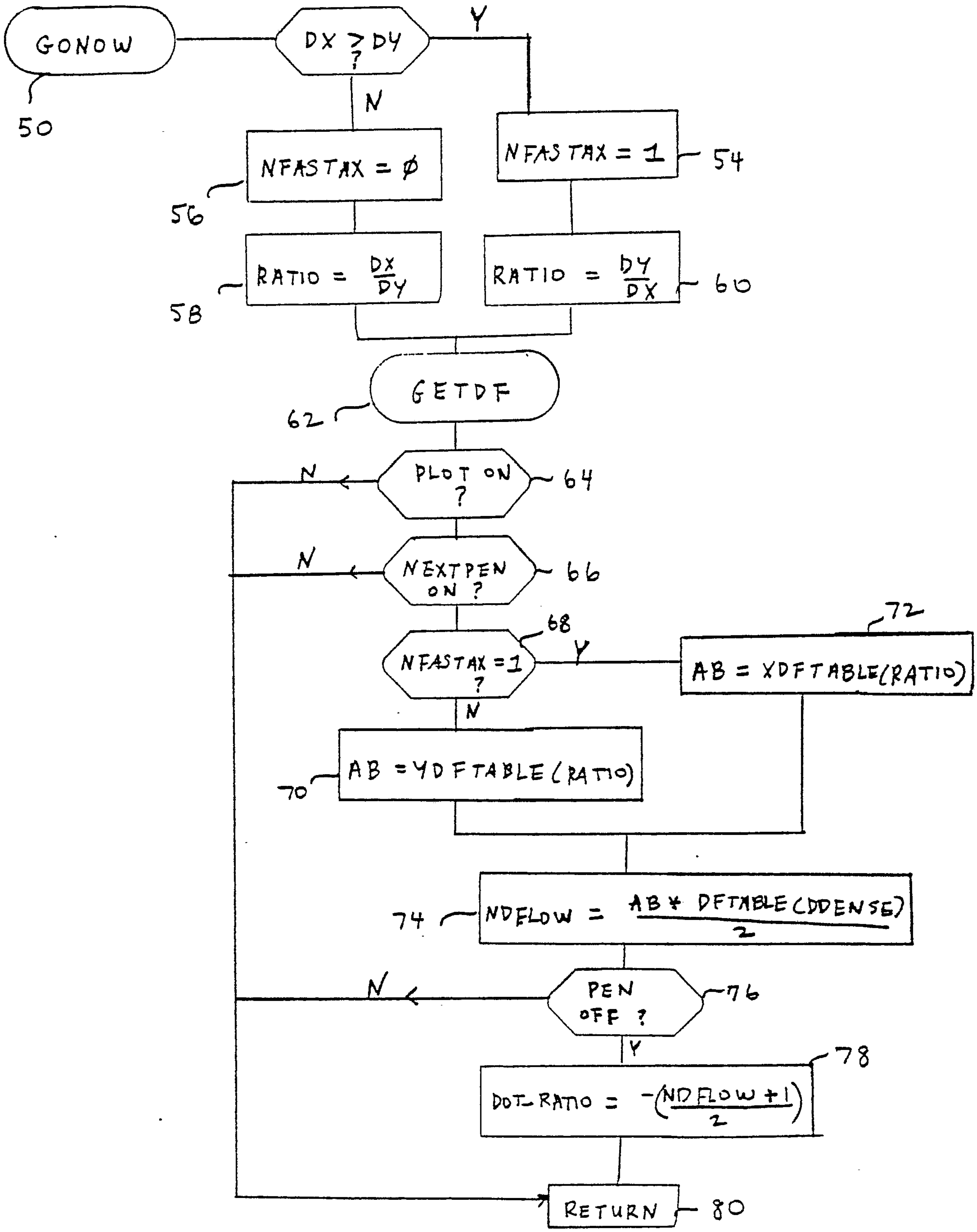
GONOW
50
DX > DY ?
Y
N
NFASTAX = Ø
56
NFASTAX = 1
54
RATIO = DX/DY
58
RATIO = DY/DX
60
GETDF
62
PLOT ON ?
64
N
NEXTPEN ON ?
66
N
68
NFASTAX = 1 ?
Y
72
AB = XDFTABLE(RATIO)
N
AB = YDFTABLE(RATIO)
70
NDFLOW = AB * DFTABLE(DDENSE) / 2
74
PEN OFF ?
76
N
Y
78
DOT-RATIO = -(NDFLOW + 1 / 2)
RETURN
80

Figure 7A

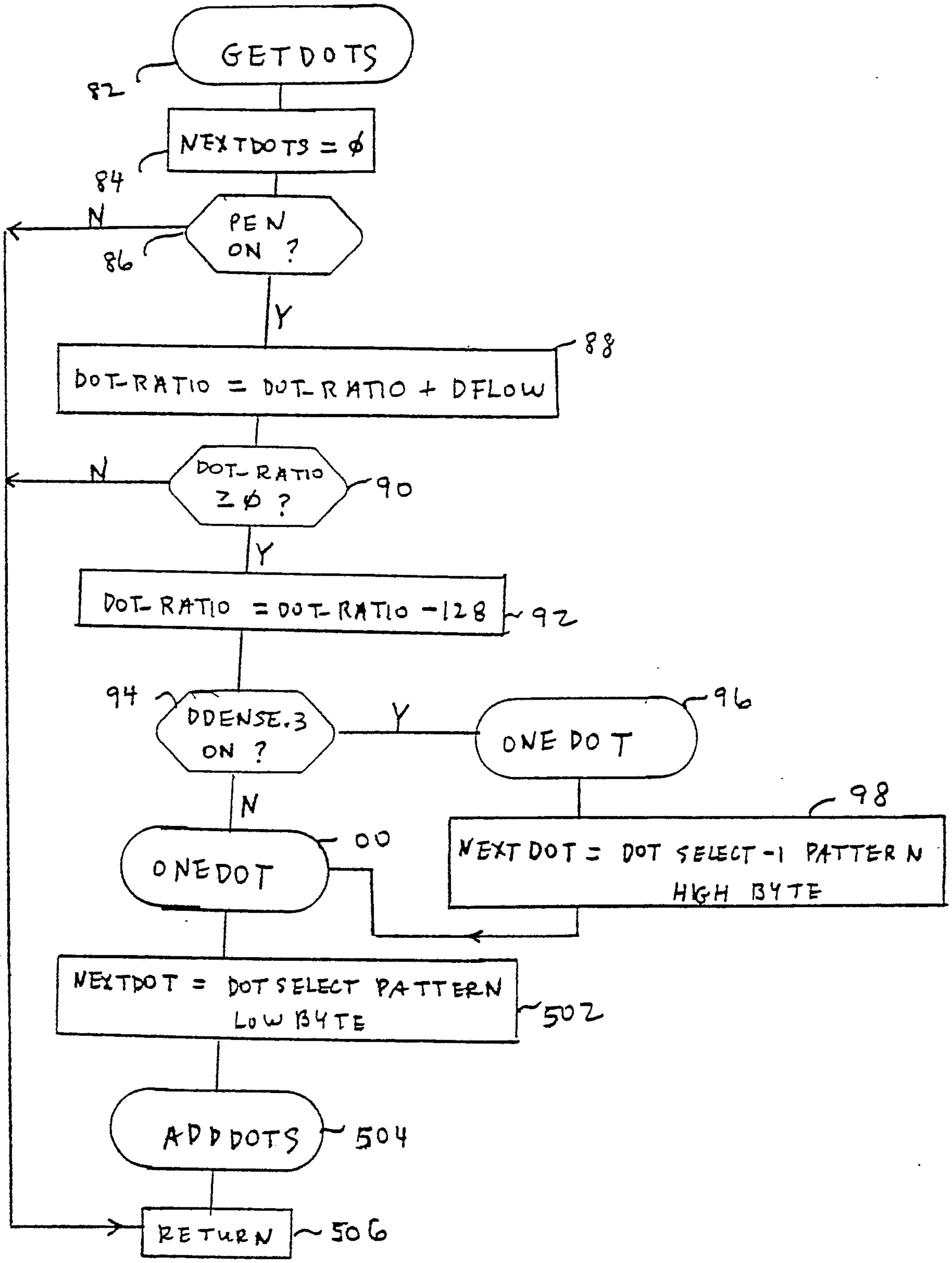
GETDOTS
82
NEXTDOTS = ∅
84
PEN ON ?
86
N
Y
DOT-RATIO = DOT-RATIO + DFLOW
88
DOT-RATIO ≥ ∅ ?
90
N
Y
DOT-RATIO = DOT-RATIO -128
92
DDENSE.3 ON ?
94
Y
N
ONE DOT
96
NEXT DOT = DOT SELECT-1 PATTERN HIGH BYTE
98
ONEDOT
00
NEXTDOT = DOT SELECT PATTERN LOW BYTE
502
ADDDOTS
504
RETURN
506

Figure 9 B

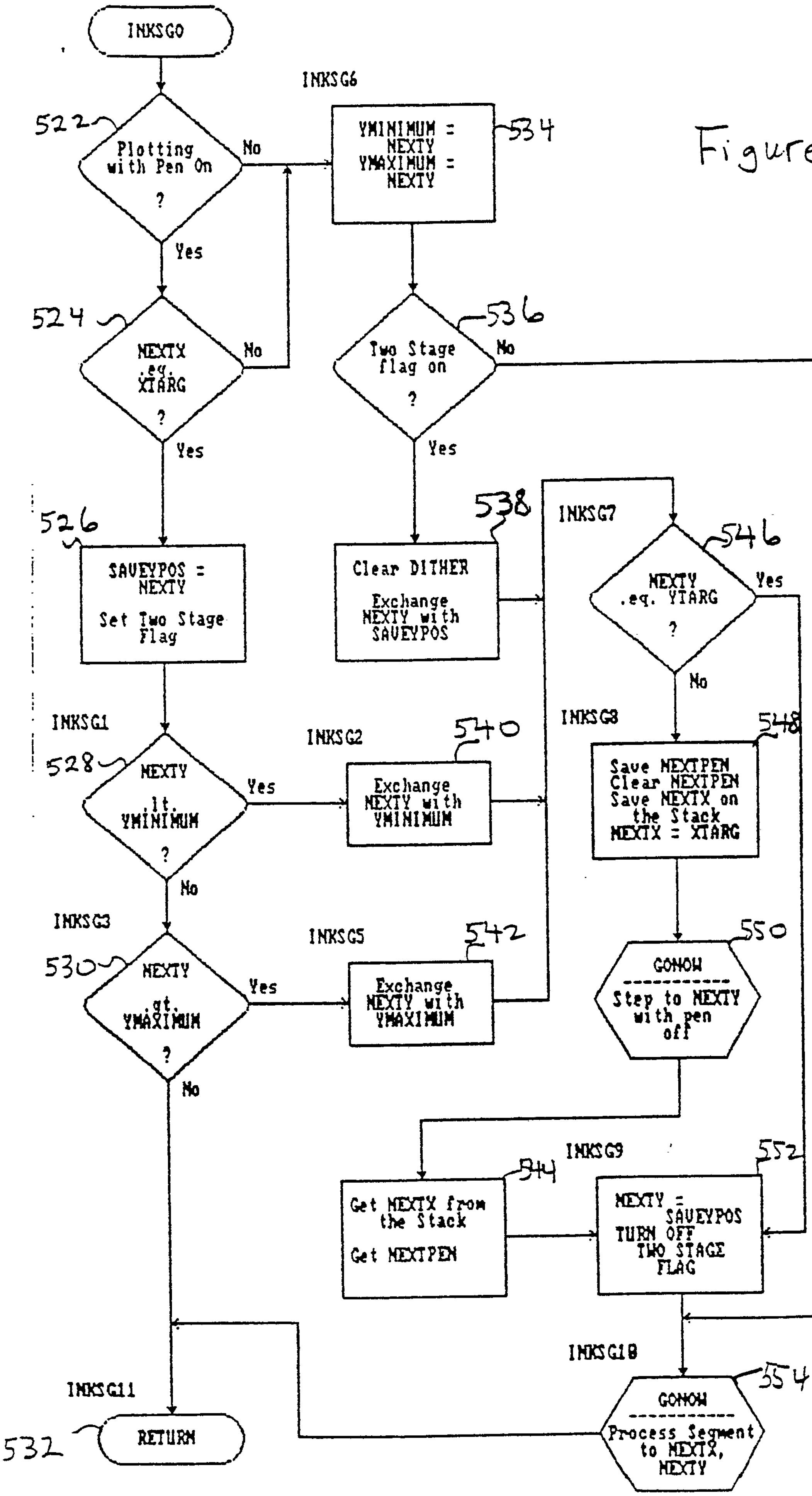
INKSG0
522
Plotting with Pen On ?
No
Yes
INKSG6
YMINIMUM = NEXTY
YMAXIMUM = NEXTY
534
524
NEXTX .eq. XTARG ?
No
Yes
536
Two Stage flag on ?
No
Yes
526
SAVEYPOS = NEXTY
Set Two Stage Flag
538
Clear DITHER
Exchange NEXTY with SAVEYPOS
INKSG7
546
NEXTY .eq. YTARG ?
Yes
No
INKSG1
528
NEXTY .lt. YMINIMUM ?
Yes
No
INKSG2
540
Exchange NEXTY with YMINIMUM
INKSG8
548
Save NEXTPEN
Clear NEXTPEN
Save NEXTX on the Stack
NEXTX = XTARG
INKSG3
530
NEXTY .gt. YMAXIMUM ?
Yes
No
INKSG5
542
Exchange NEXTY with YMAXIMUM
550
GONOW
Step to NEXTY with pen off
544
Get NEXTX from the Stack
Get NEXTPEN
INKSG9
552
NEXTY = SAVEYPOS
TURN OFF TWO STAGE FLAG
INKSG10
554
GONOW
Process Segment to NEXTX, NEXTY
INKSG11
532
RETURN

Figure 10

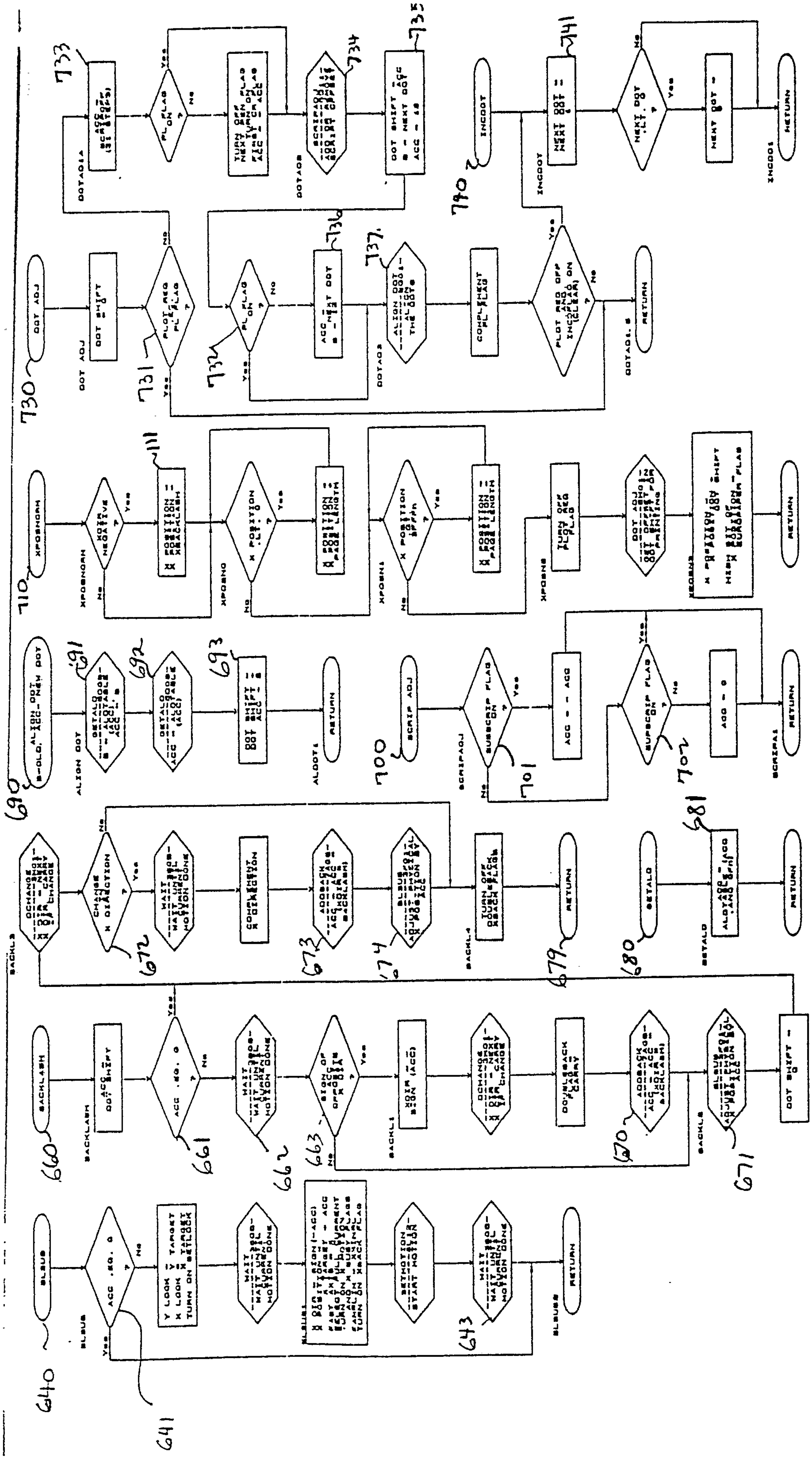

Figure 11