

- 3,058,657 10/1962 Tripp.....235/154

A numerical path control system including a time shared digital computer generating for each of a plurality of axes regularly recurring numerical excursion commands. A two axis system is shown to be capable of deriving from a small number of data words successive straight line path or circular arc path command signals at a rate of 50 times a second and of converting these into intermediate or secondary command signals which are applied to the servos of the machine axes at a rate of 500 times a second.

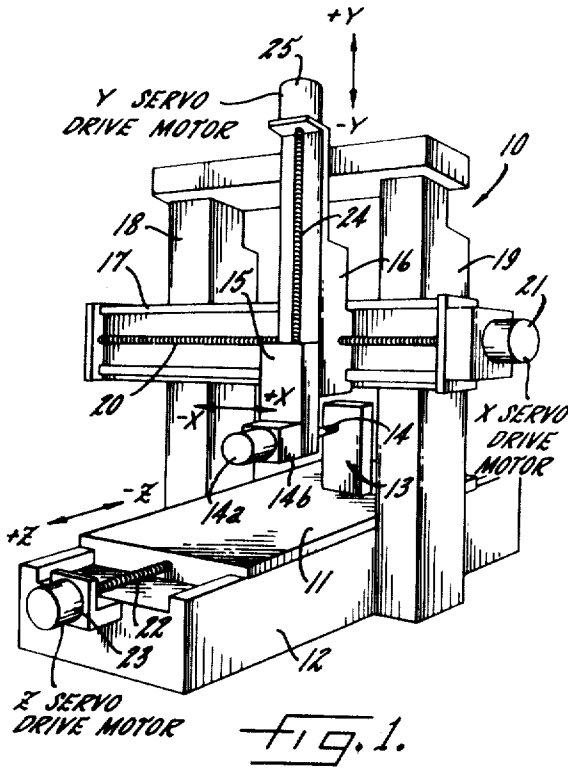


Fig. 1.

BLOCK 1
" 2
" 3
" 4
" 5
" 6
" 7
" 8
" 9
" 10
BLOCK 11

Fig. 3.

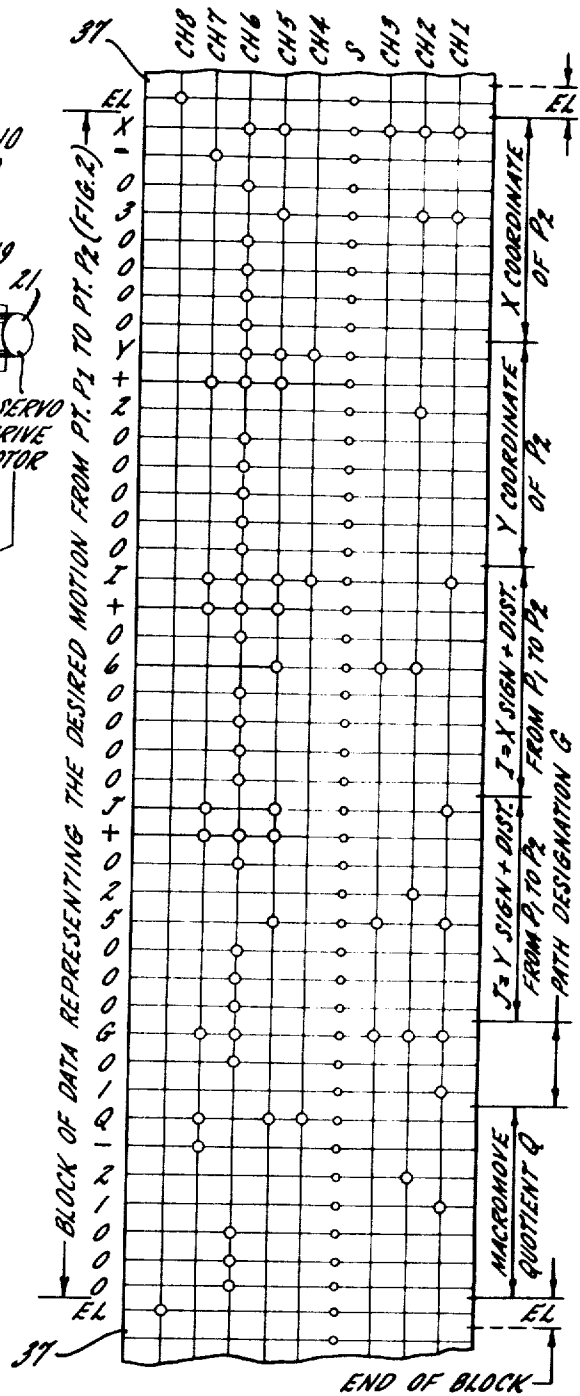
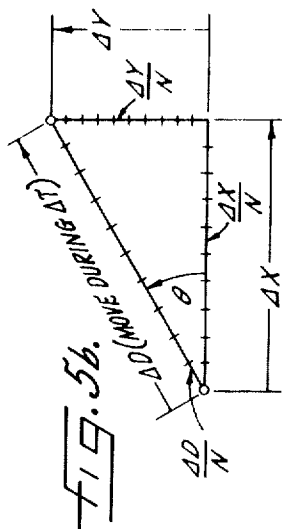
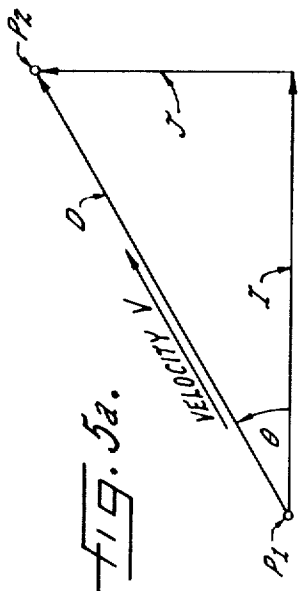


Fig. 4.

INVENTOR.
JOHN K. MCGEE
BY *Wolfe, Hubbard, Vont & Orourke*
ATTORNEYS.



$$\Delta D = V \Delta T \quad (1)$$

$$\Delta X = I \cdot \frac{V \cdot \Delta T}{\sqrt{I^2 + J^2}} \quad (6)$$

$$\Delta Y = J \cdot \frac{V \cdot \Delta T}{\sqrt{I^2 + J^2}} \quad (7)$$

$$\text{IF } \frac{V \cdot \Delta T}{\sqrt{I^2 + J^2}} = Q \text{ THEN } (8)$$

$$\Delta X = I \cdot Q \quad (9)$$

$$\Delta Y = J \cdot Q \quad (10)$$

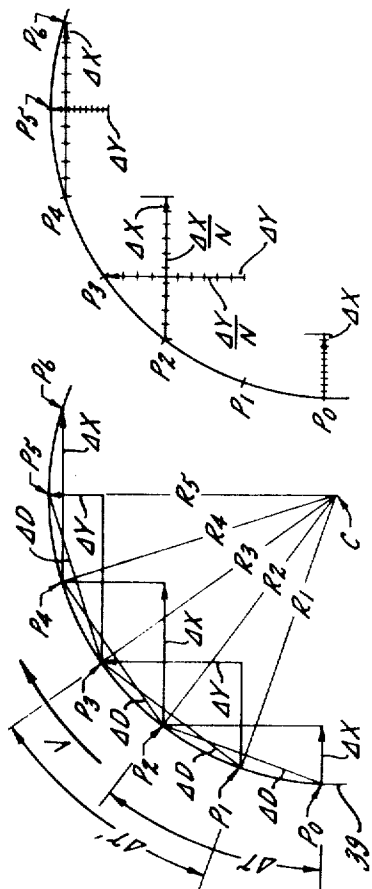


FIG. 6a.

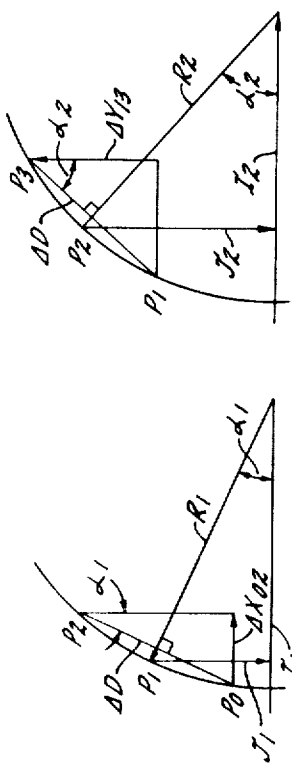


FIG. 6b.

$$\Delta X_{02} = -J_1 \frac{V \cdot \Delta T}{\sqrt{I_1^2 + J_1^2}} \quad (15)$$

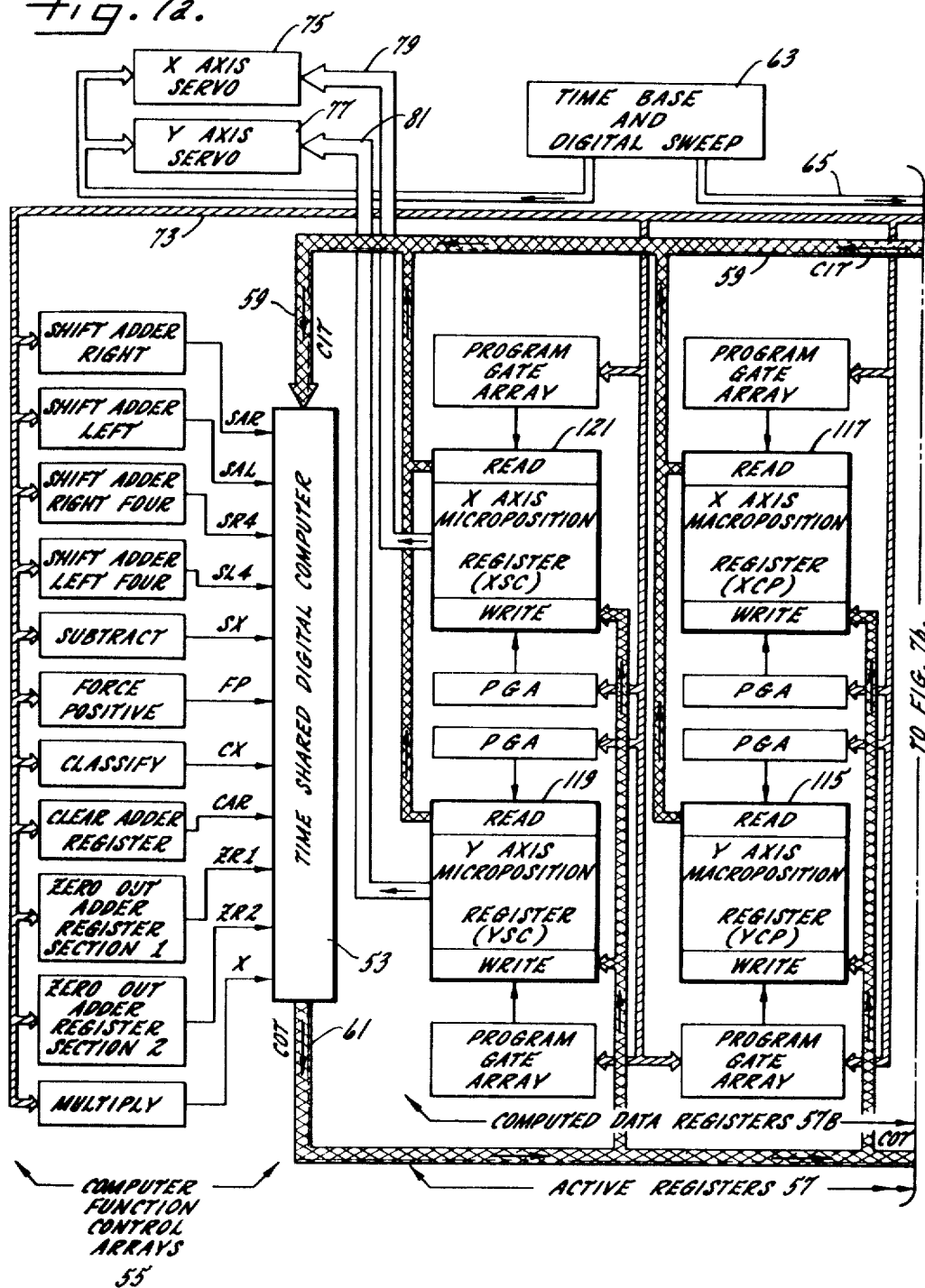
$$\text{IF } \frac{V \cdot \Delta T}{\sqrt{I_1^2 + J_1^2}} = Q = \frac{V \cdot \Delta T'}{\sqrt{I_2^2 + J_2^2}}, \text{ THEN } (82)$$

$$\Delta X_{02} = -J_1 \cdot Q \quad (19) \text{ AND } \Delta Y_{02} = +I_2 \cdot Q \quad (19) \text{ FOR CW}$$

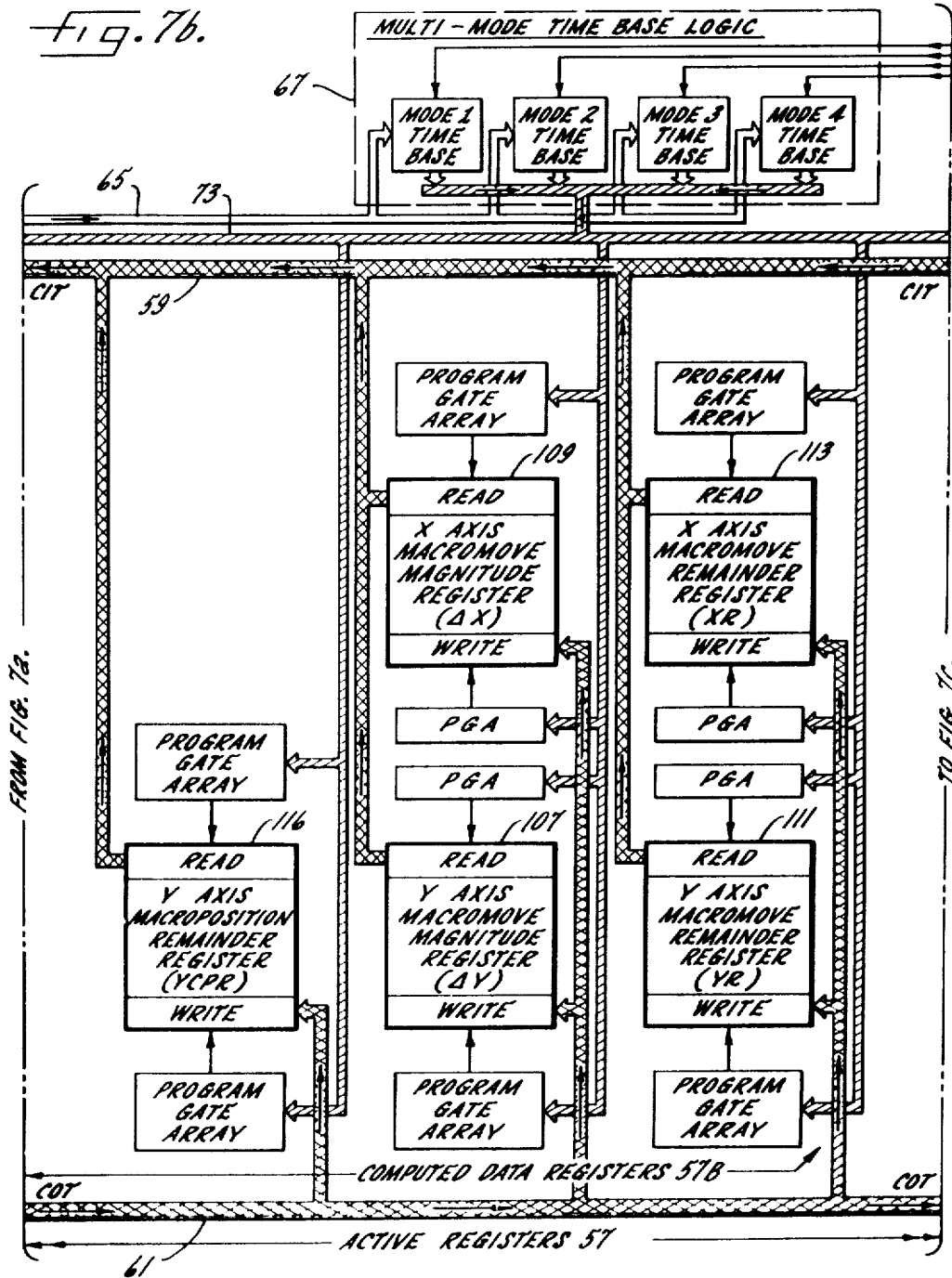
(THE SIGNS CHANGE IN (18) AND (19) FOR CCW)

INVENTOR.
JOHN K. MCGEE
BY *Wolfe, Hubbard, Vort, Orann*
ATTORNEYS.

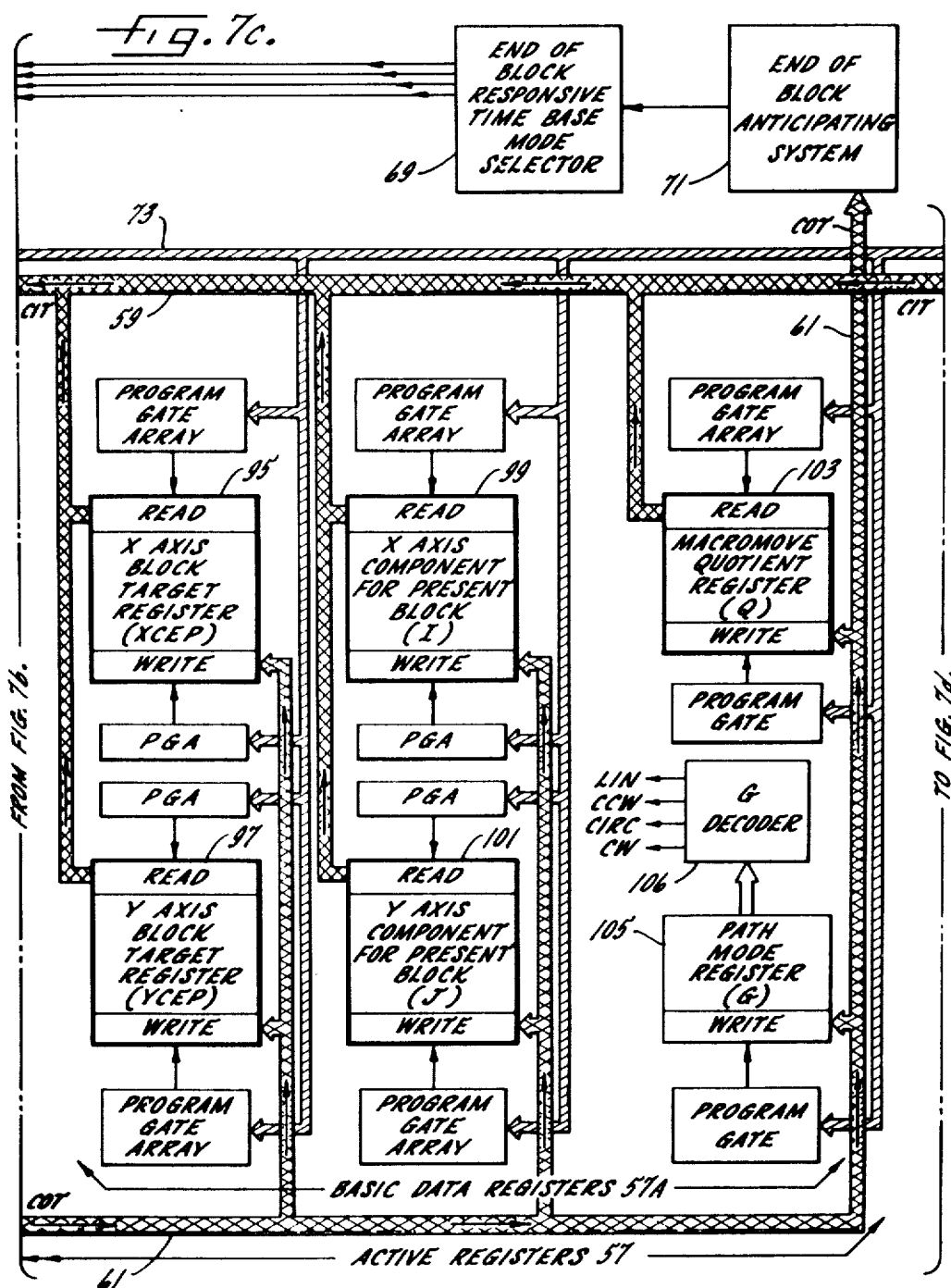
Fig. 7a.



INVENTOR.
JOHN K. MCGEE
BY *Wolfe, Hubbard, Vort & O'Connell*
ATTORNEYS.

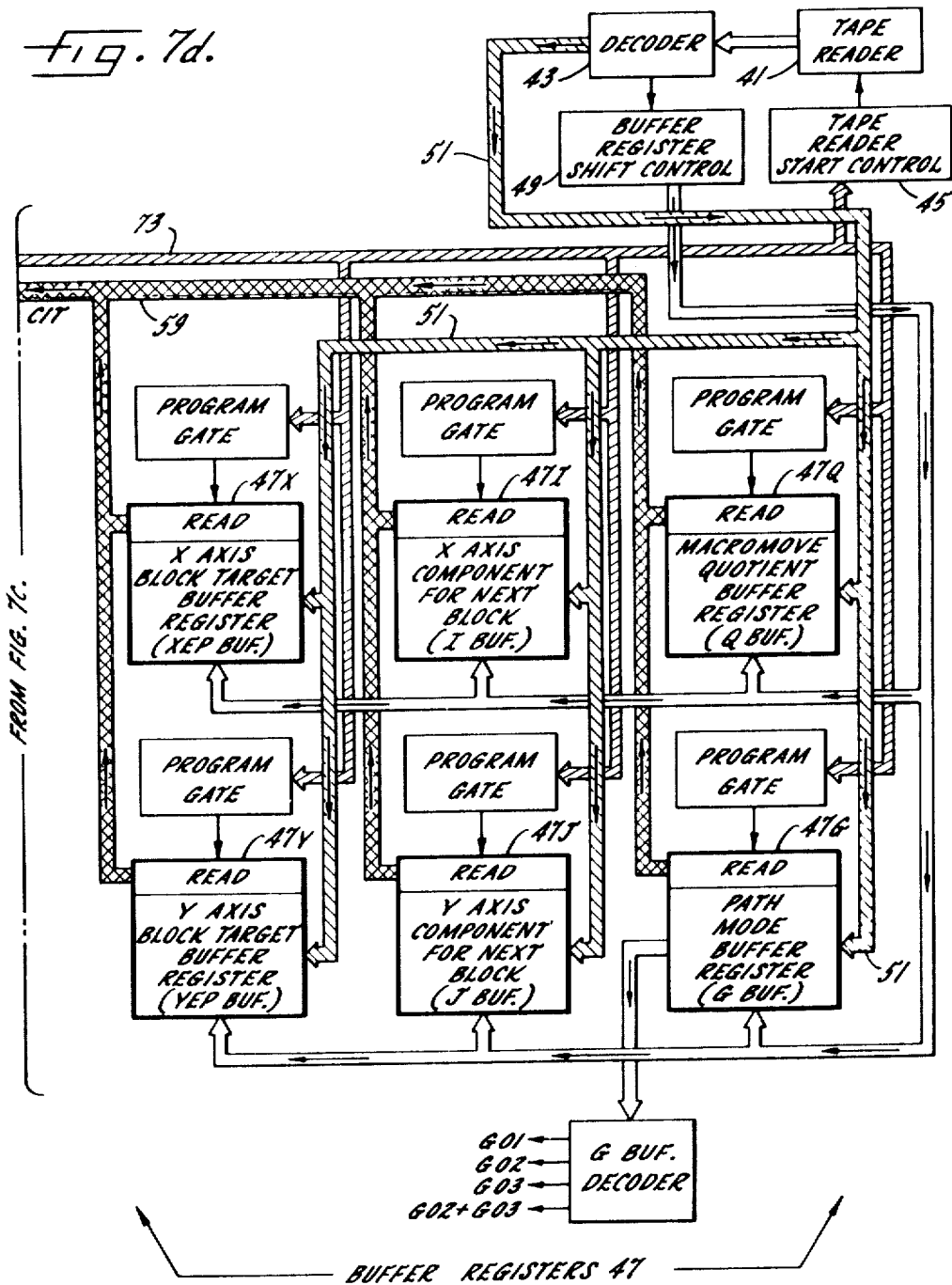


INVENTOR.
JOHN K. MCGEE
BY *Wolfe, Hubbard, Vout & Oran*
ATTORNEYS.

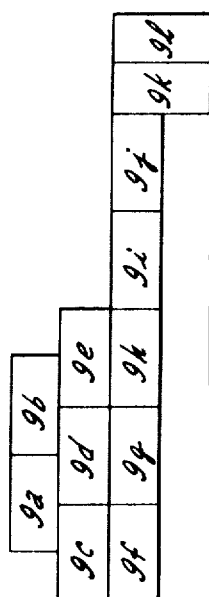
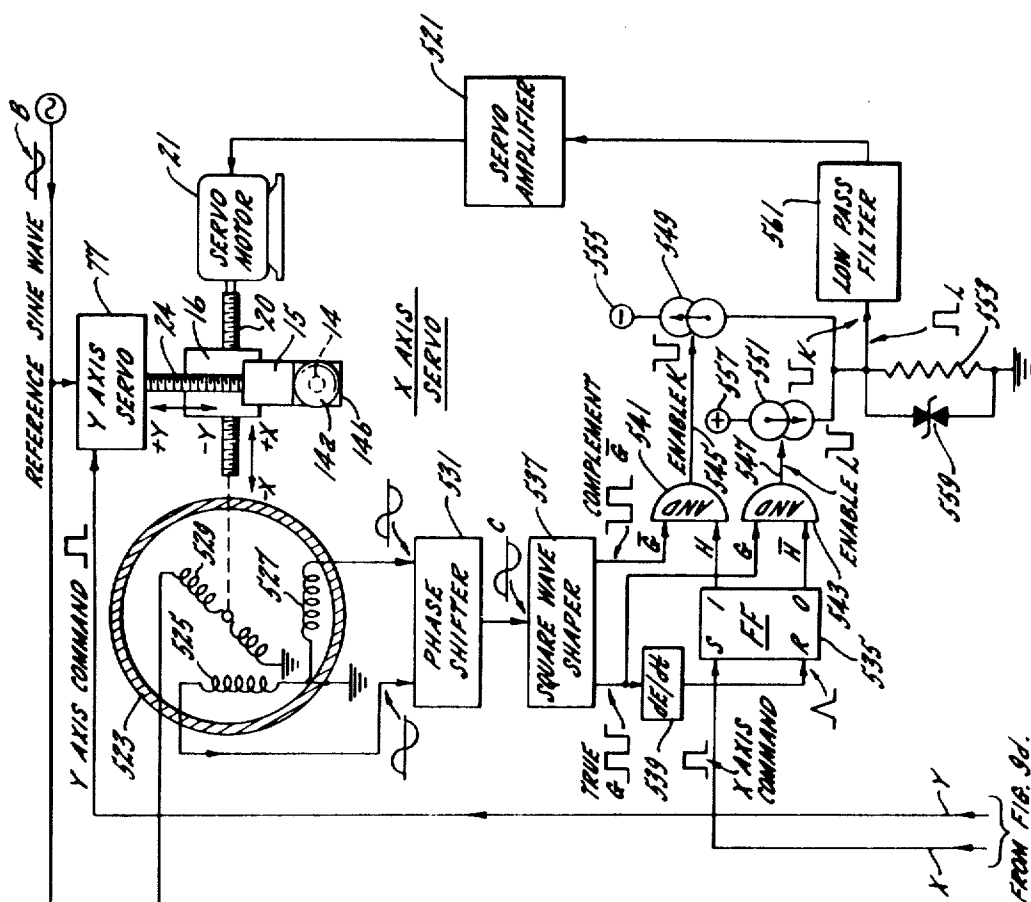


INVENTOR.
JOHN K. MCGEE
BY *Wolfe, Hubbard, Voigt, Osann*
ATTORNEYS.

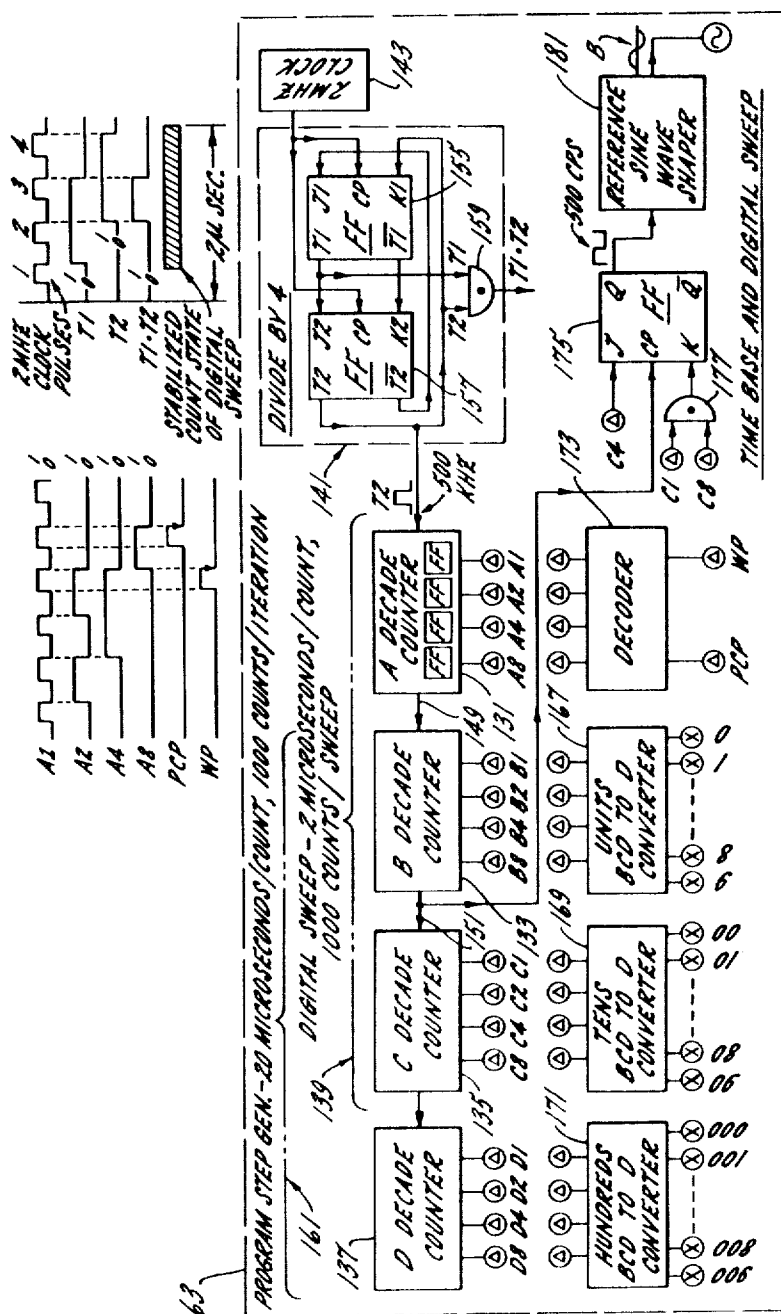
Fig. 7d.



INVENTOR.
JOHN K. MCGEE
BY *Wolfe, Hubbard, Voigt & Osann*
ATTORNEYS.

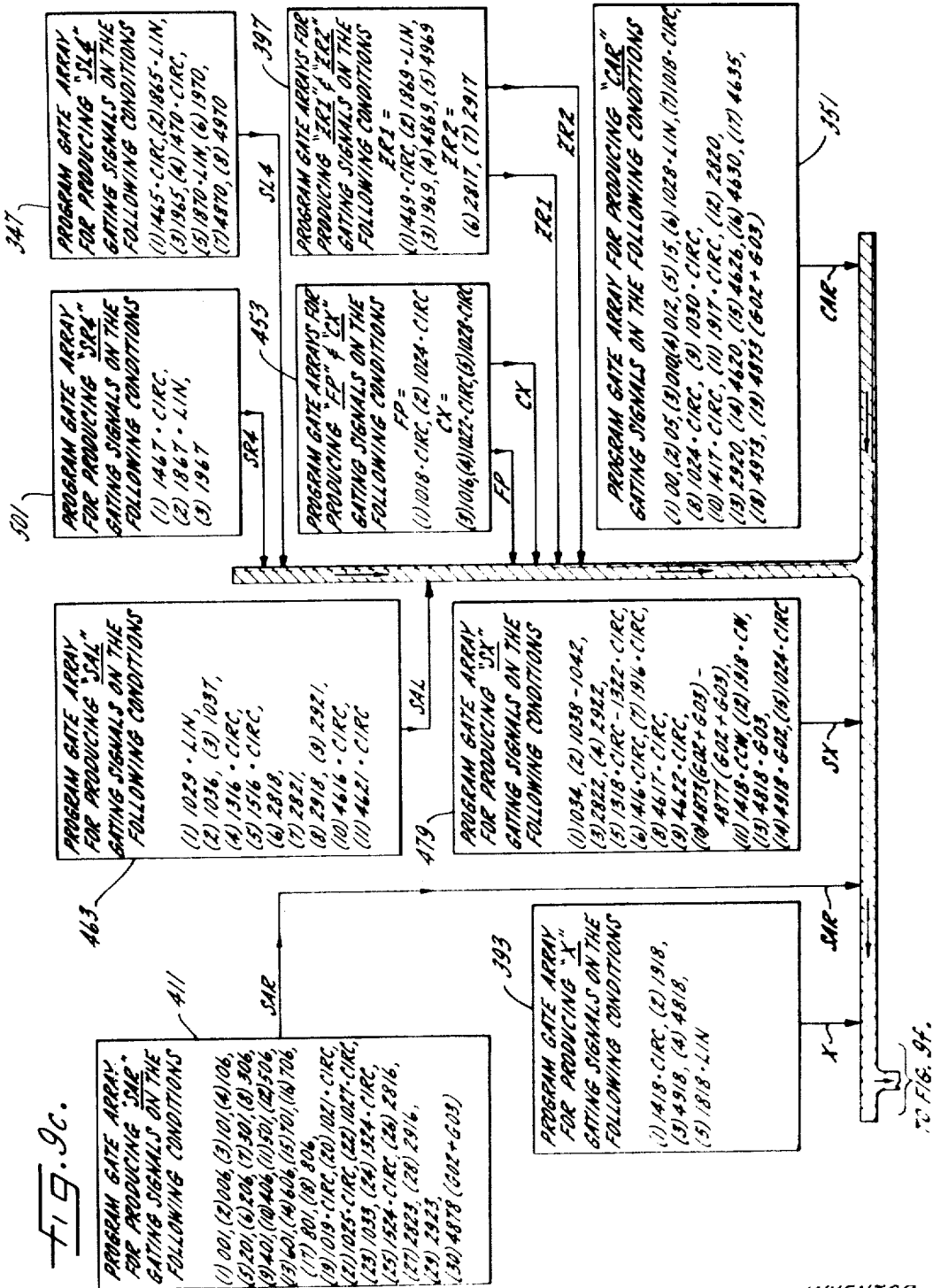


INVENTOR.
JOHN K. MCGEE
BY *Wolfe, Hubbard, Voitz & Osann*
ATTORNEYS.

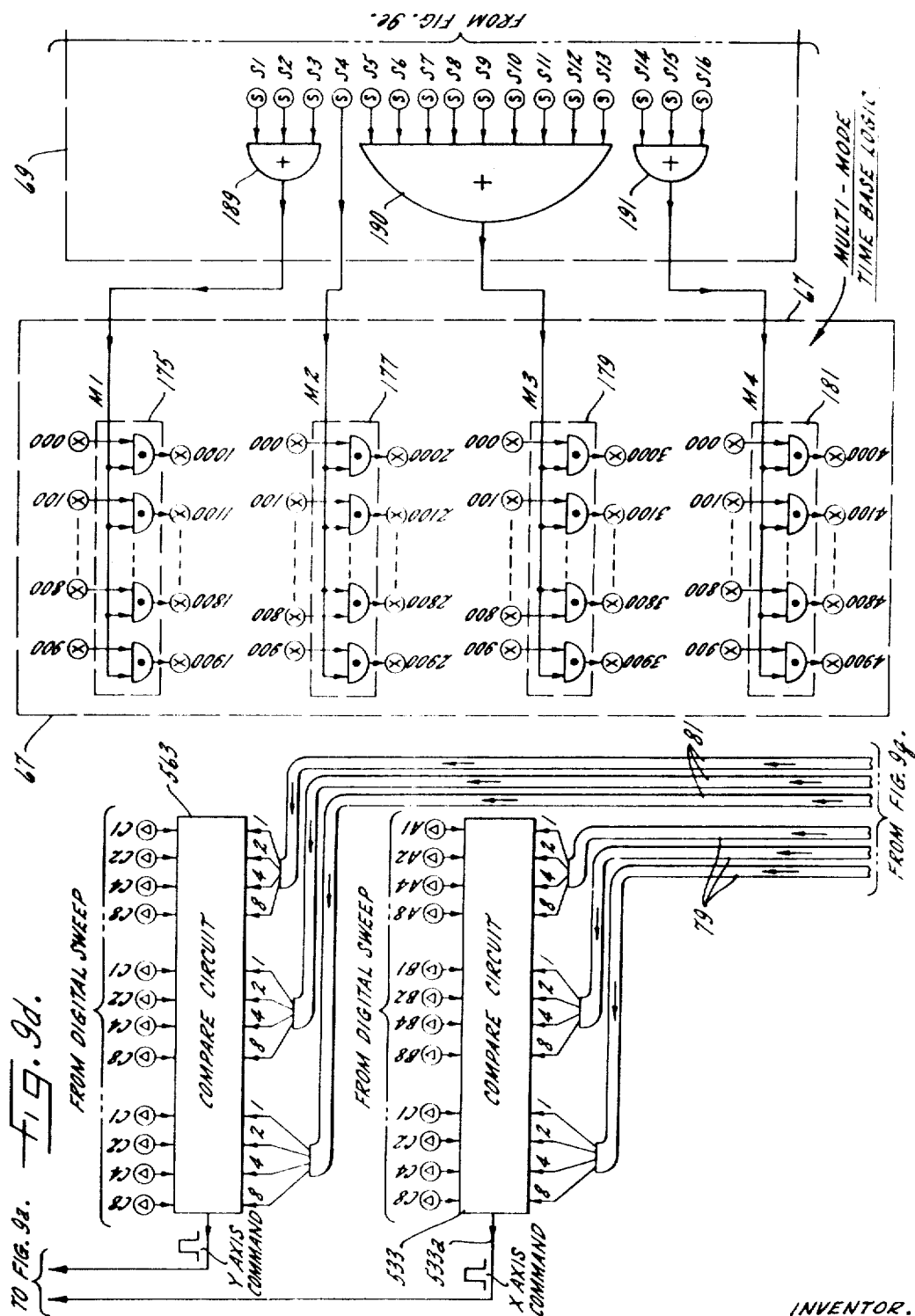


96.54

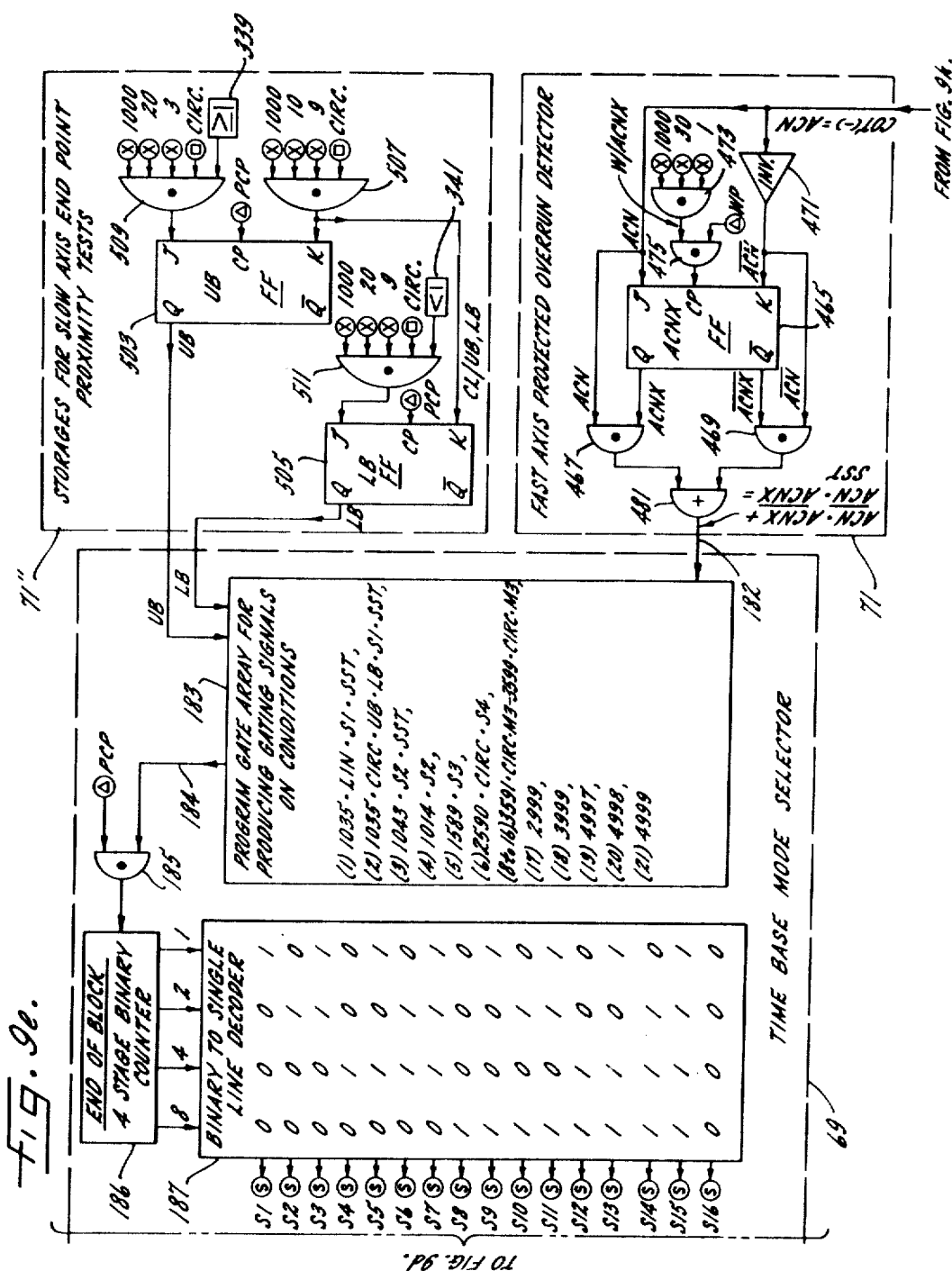
INVENTOR.
JOHN K MCGEE
BY *Wolfe, Hubbard, Voit & Osann*
ATTORNEYS.



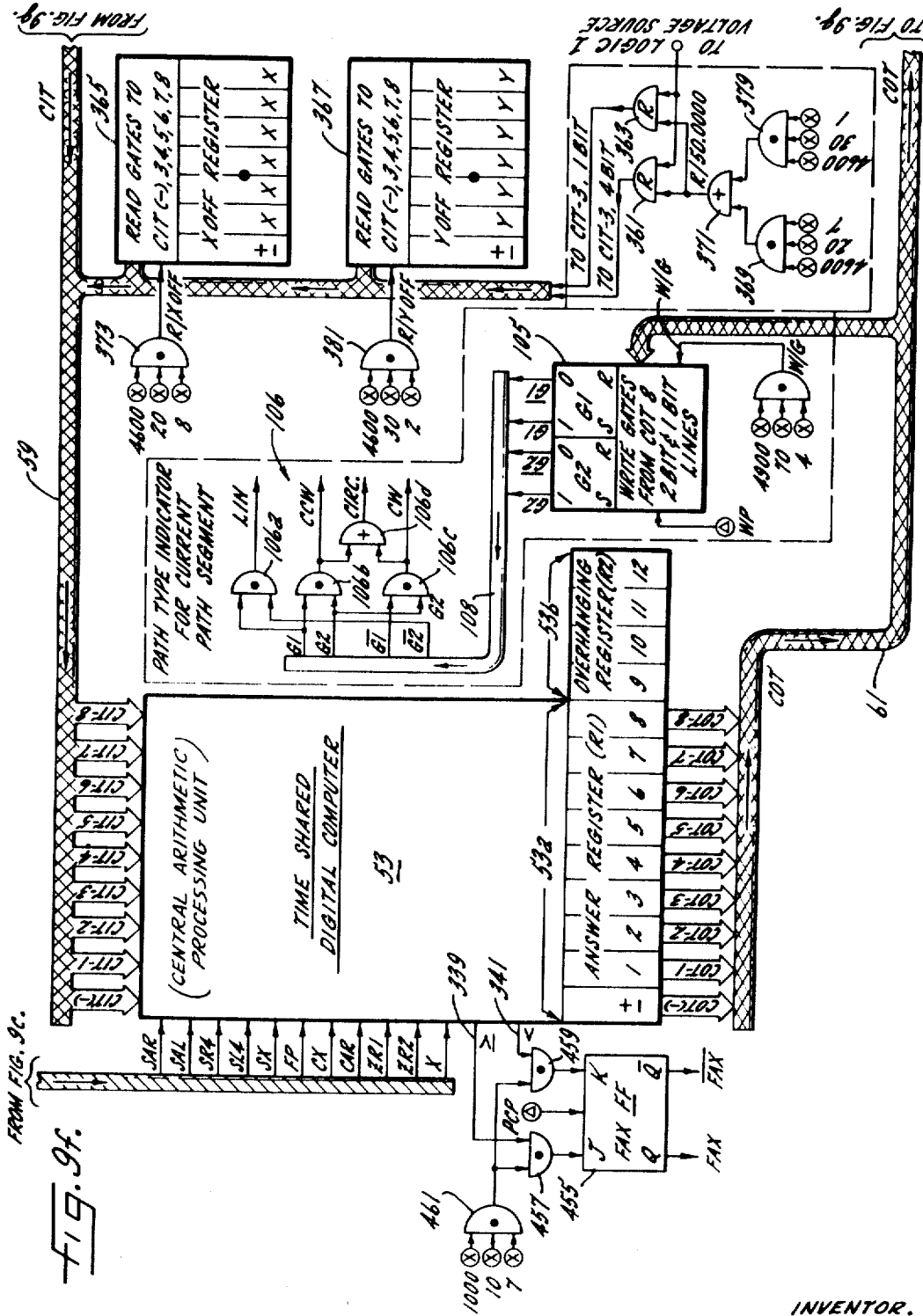
INVENTOR.
 JOHN K. MCGEE
 BY *Wolfe, Hubbard, Voigt & Osann*
 ATTORNEYS.



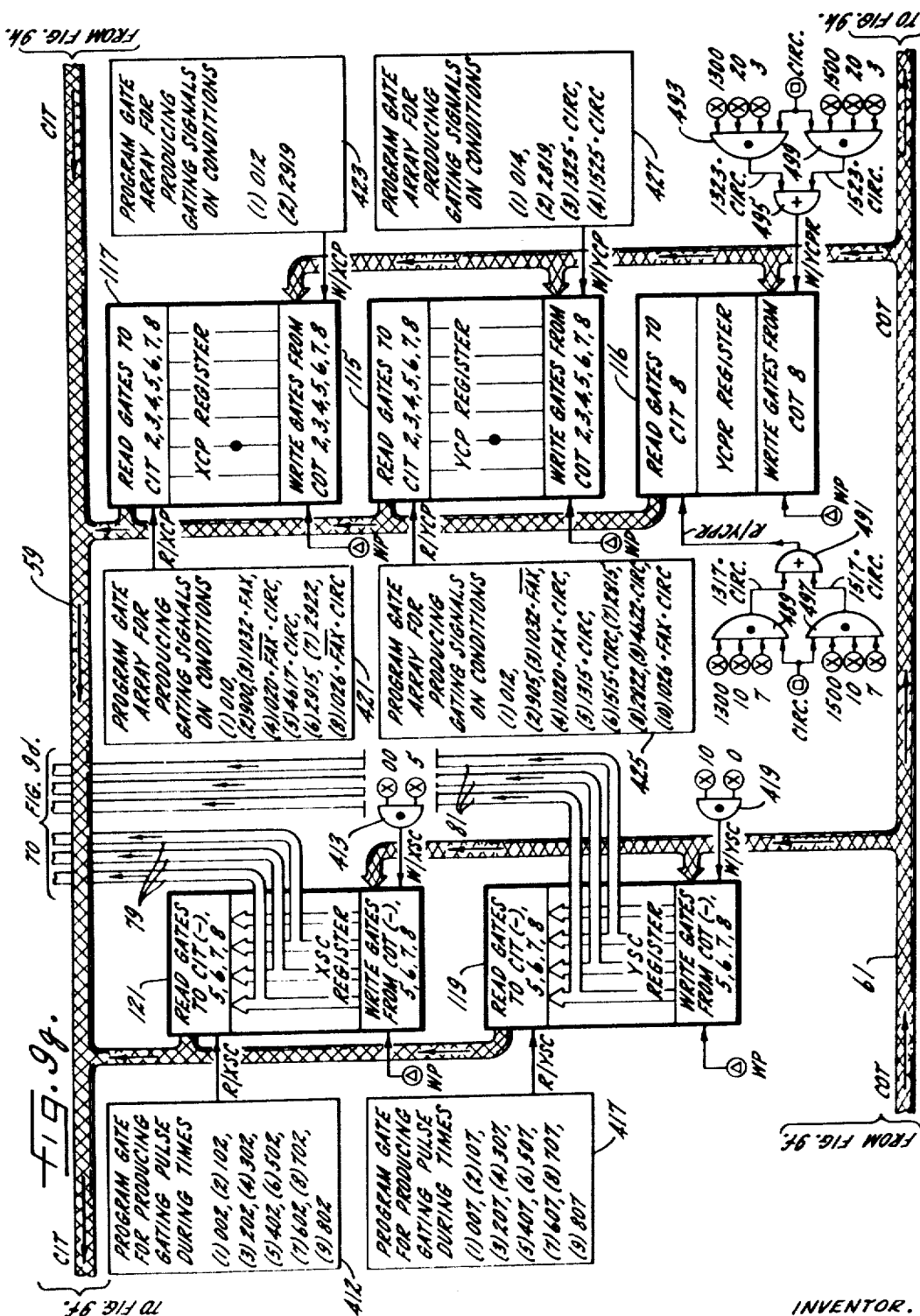
INVENTOR.
JOHN K. MCGEE
BY *Wolfe, Hubbard, Voigt & Osann*
ATTORNEYS.



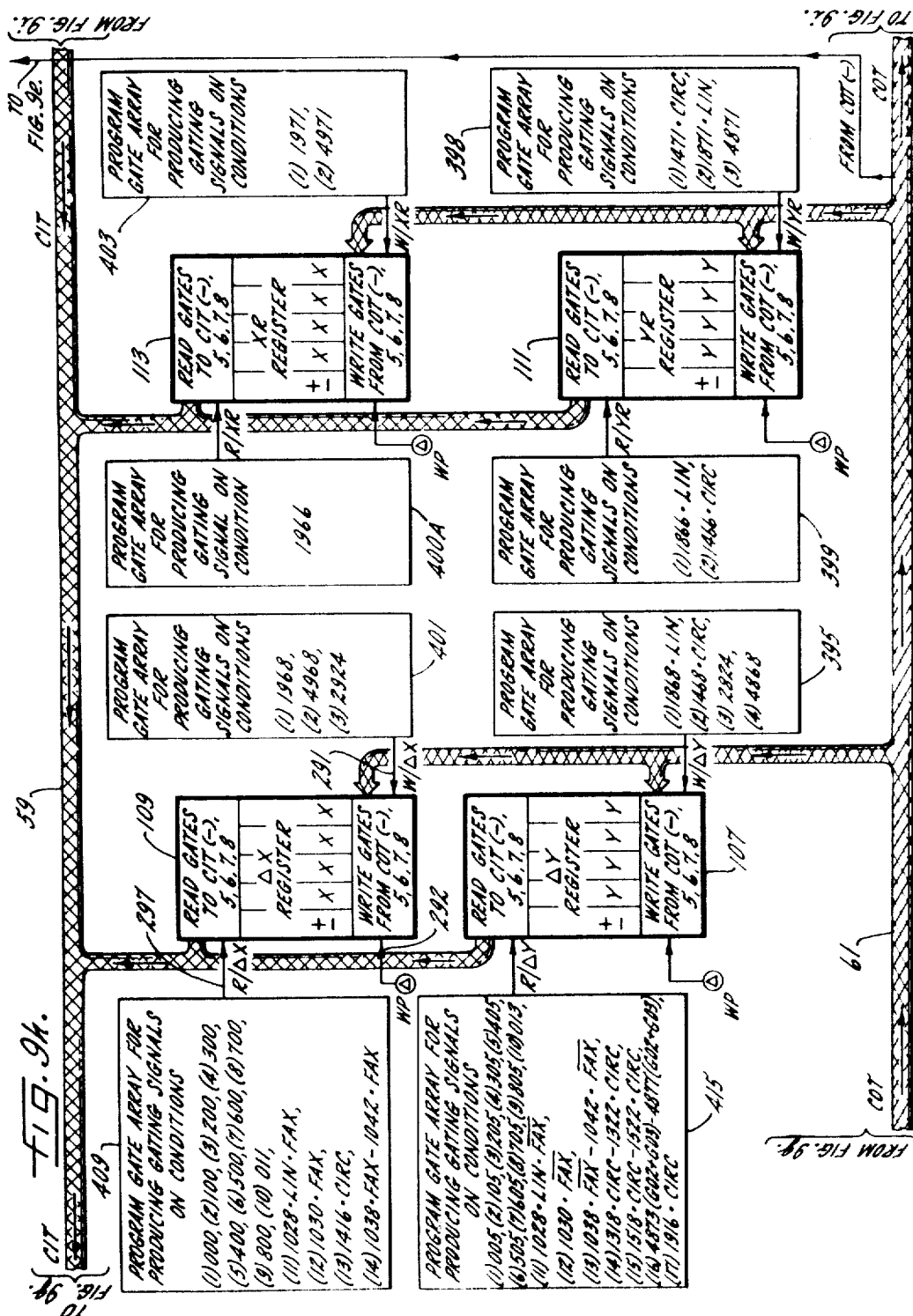
INVENTOR.
JOHN K. MCGEE
BY *Wolfe, Hubbard, Voigt & Osann*
ATTORNEYS.



INVENTOR.
JOHN K. MCGEE
BY
Wolfe, Hubbard, Vort & Osann
ATTORNEYS.

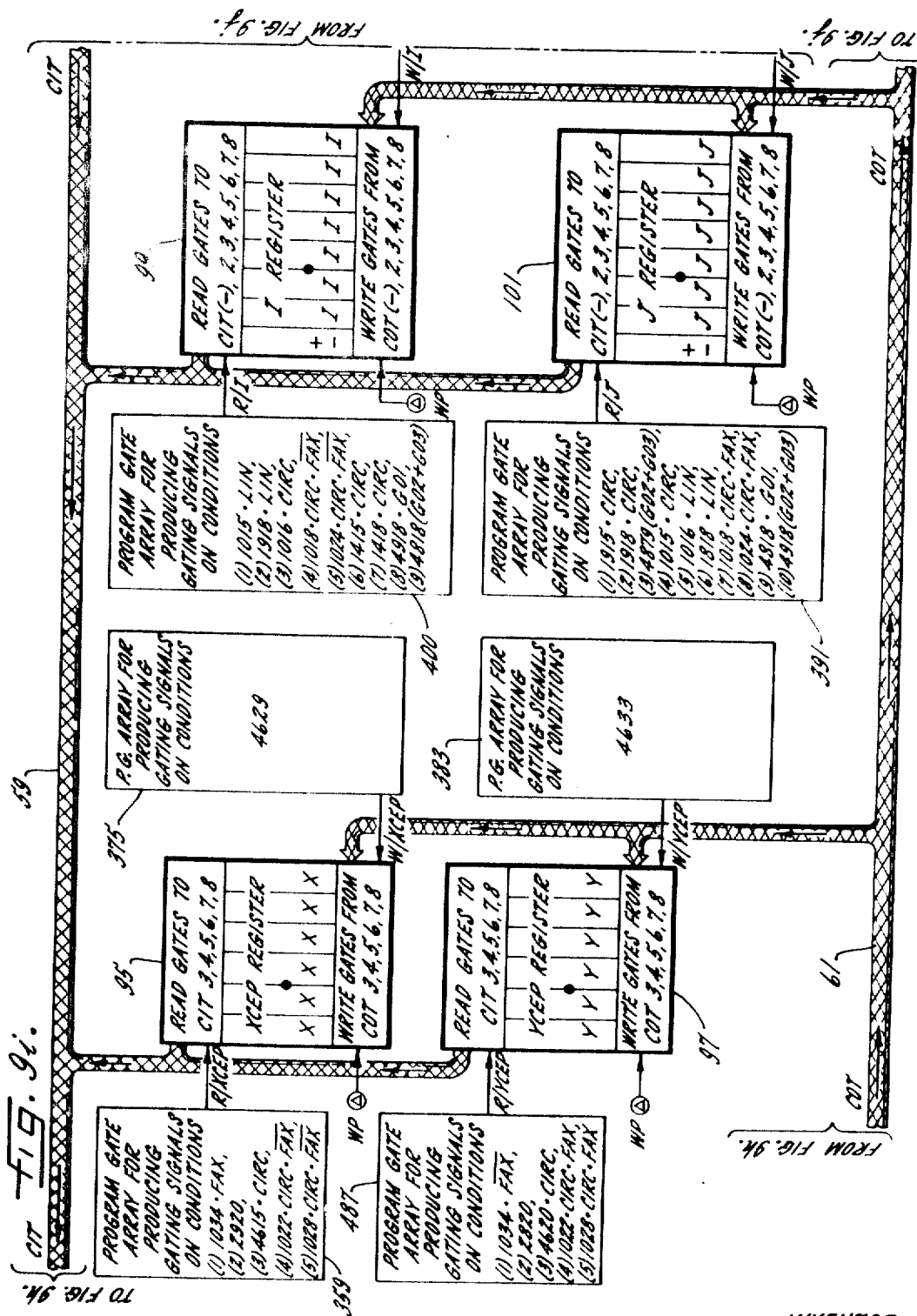


INVENTOR.
JOHN K. MCGEE
BY *Wolfe, Hubbard, Vout, Orann*
ATTORNEYS.

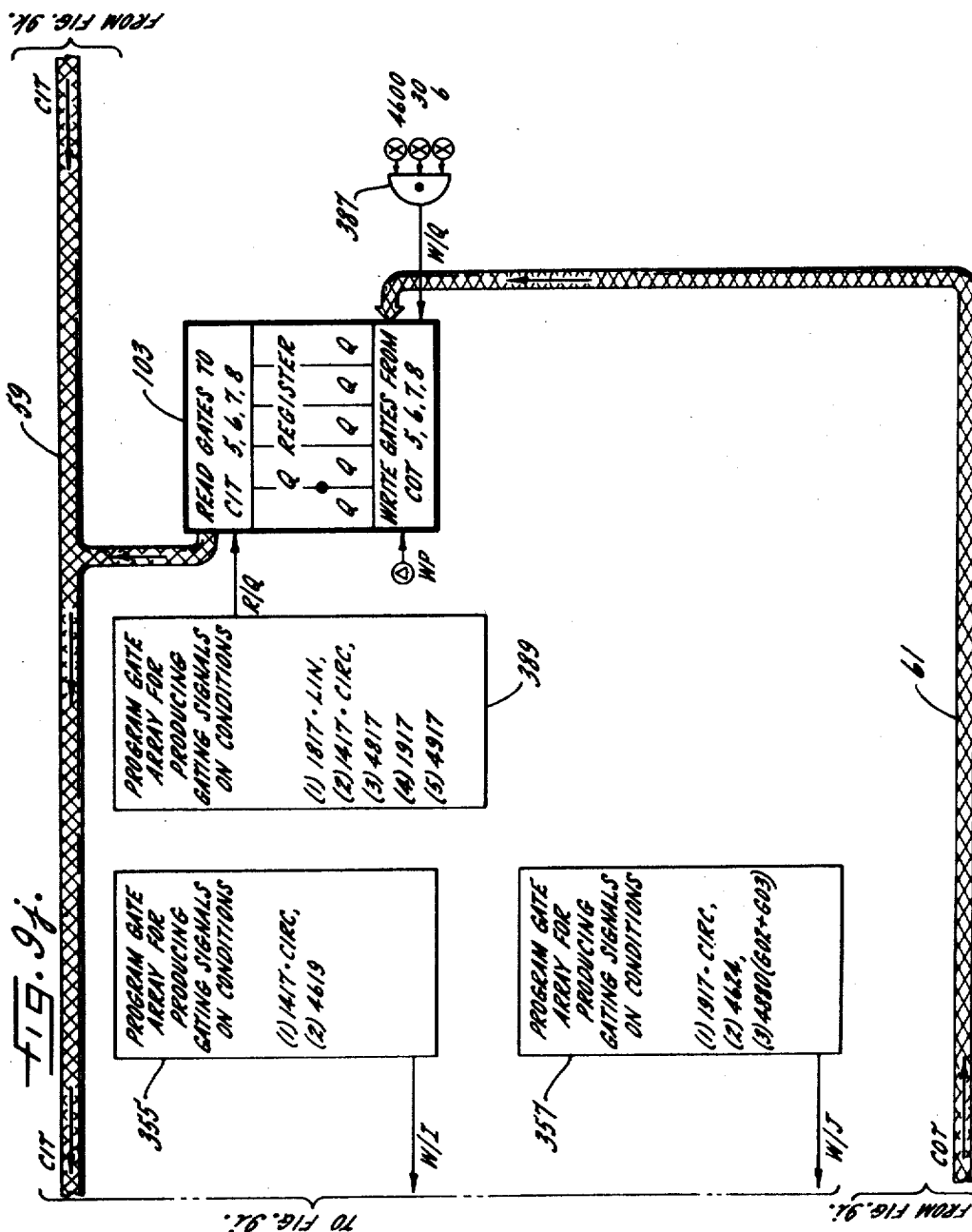


INVENTOR.
JOHN K. MCGEE

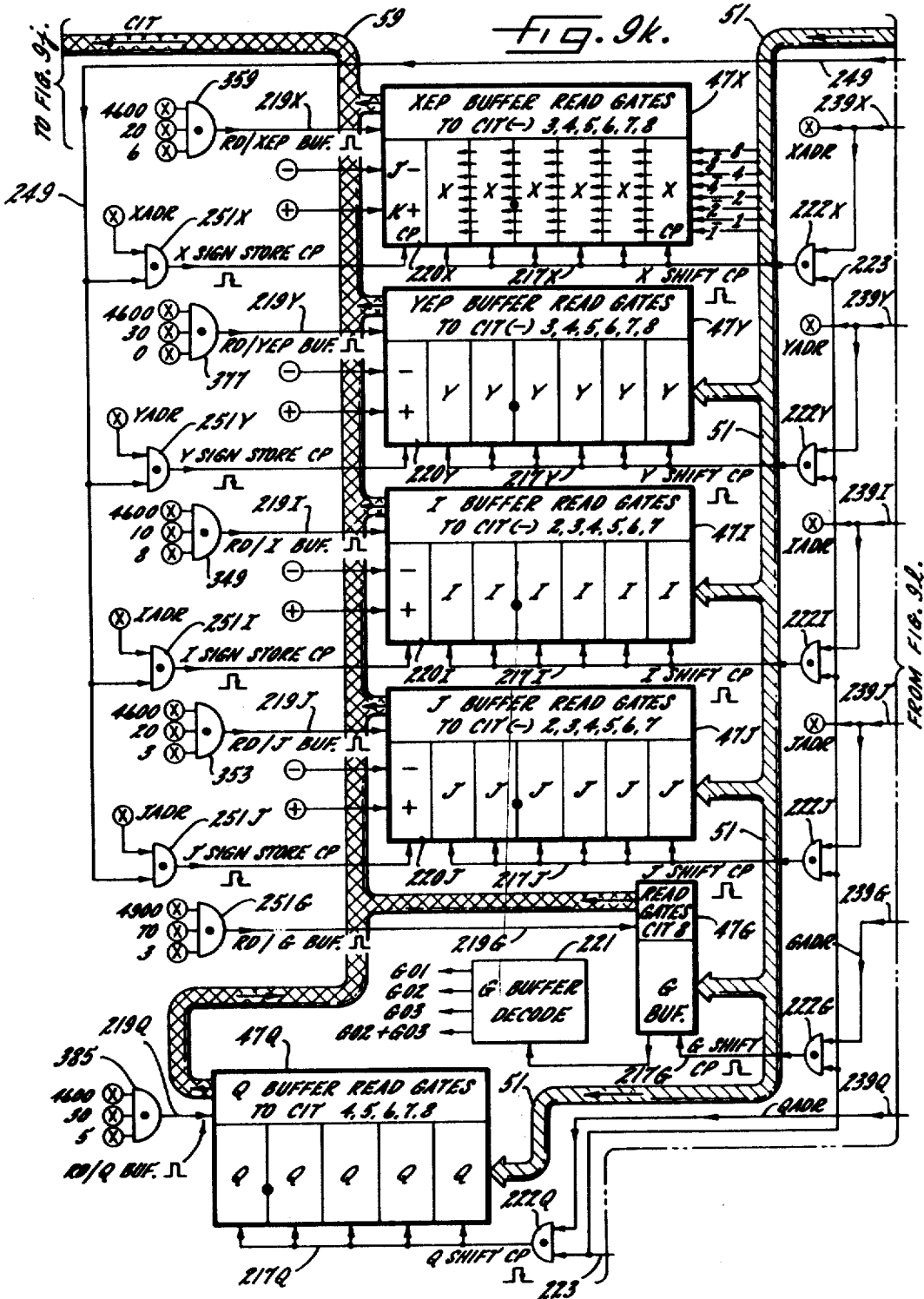
BY
Wolfe, Hubbard, Voigt & Osann
ATTORNEYS.



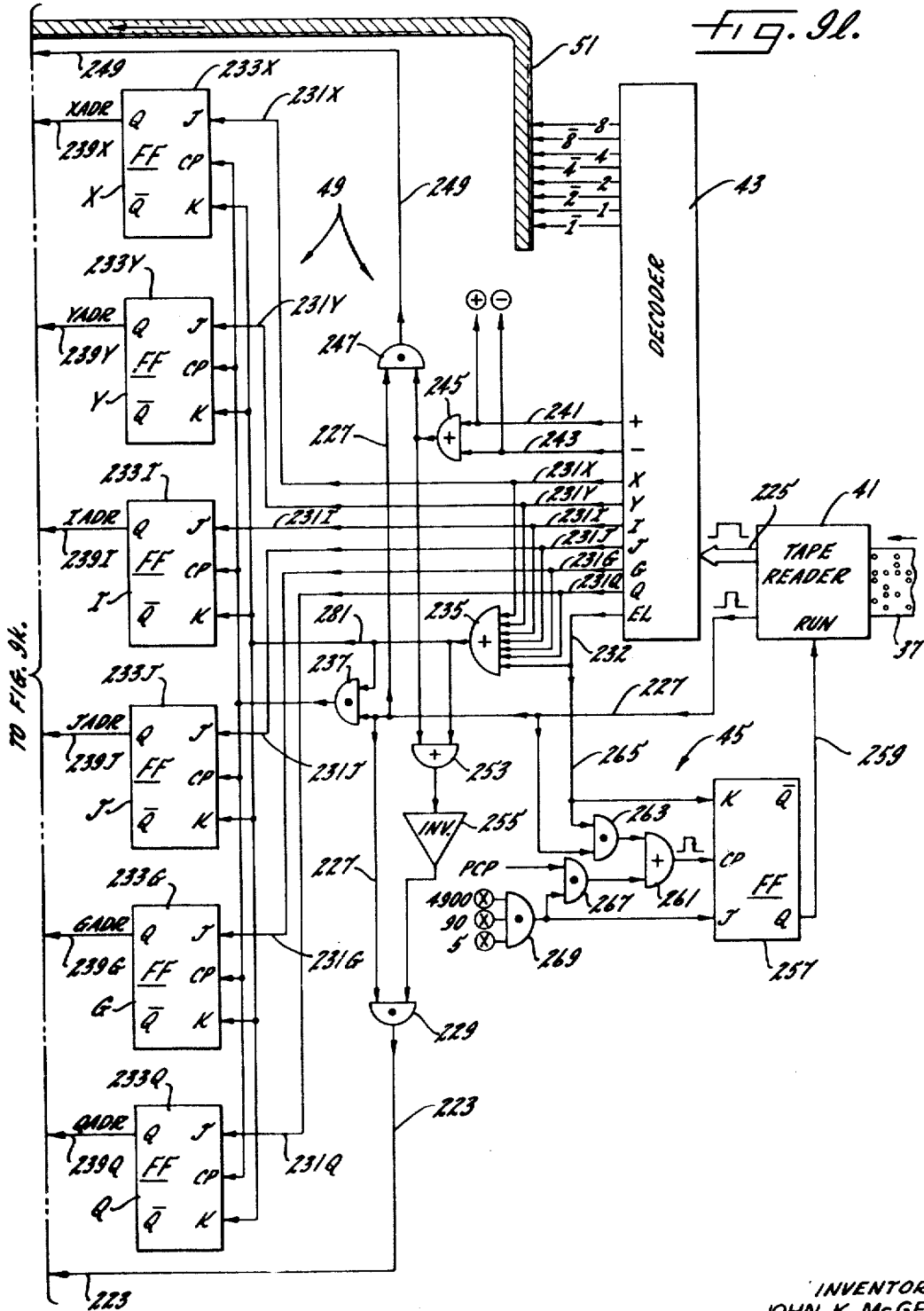
INVENTOR.
JOHN K. MCGEE
BY *Wolfe, Hubbard, Voigt, Osann*
ATTORNEYS.

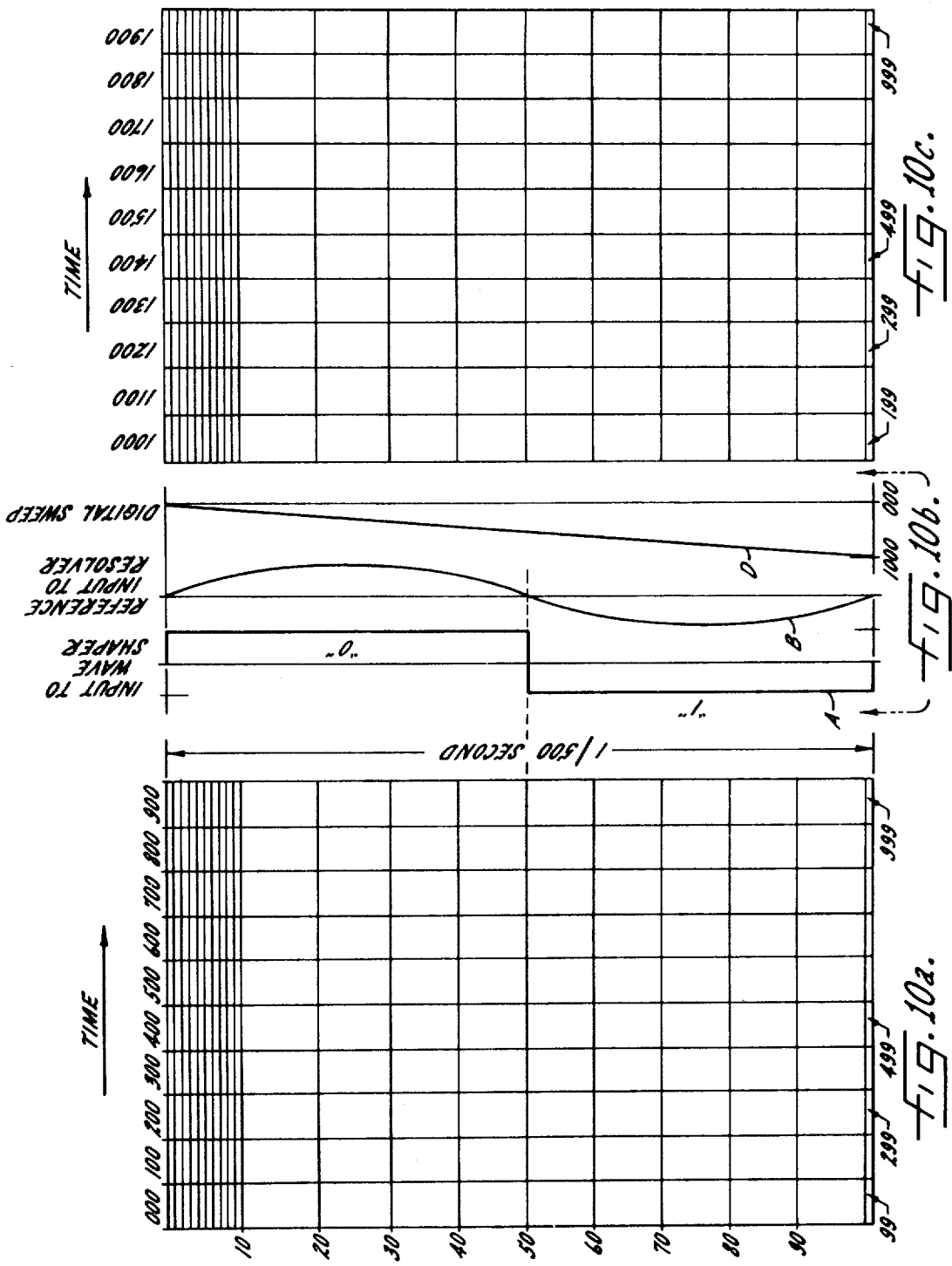


INVENTOR.
JOHN K. MCGEE
BY *Wolfe, Hubbard, Voigt, Osann*
ATTORNEYS.

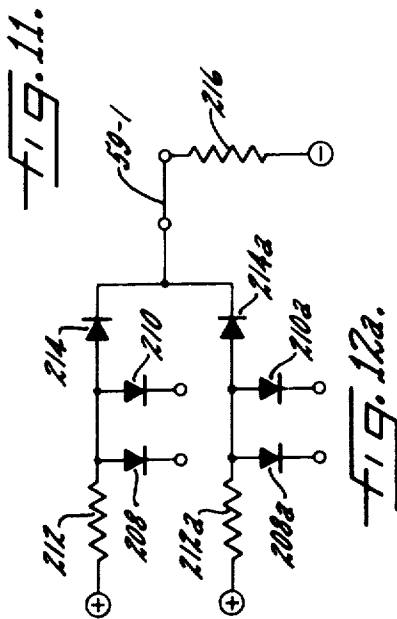
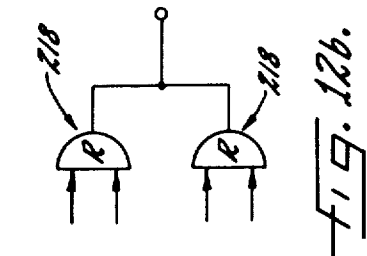
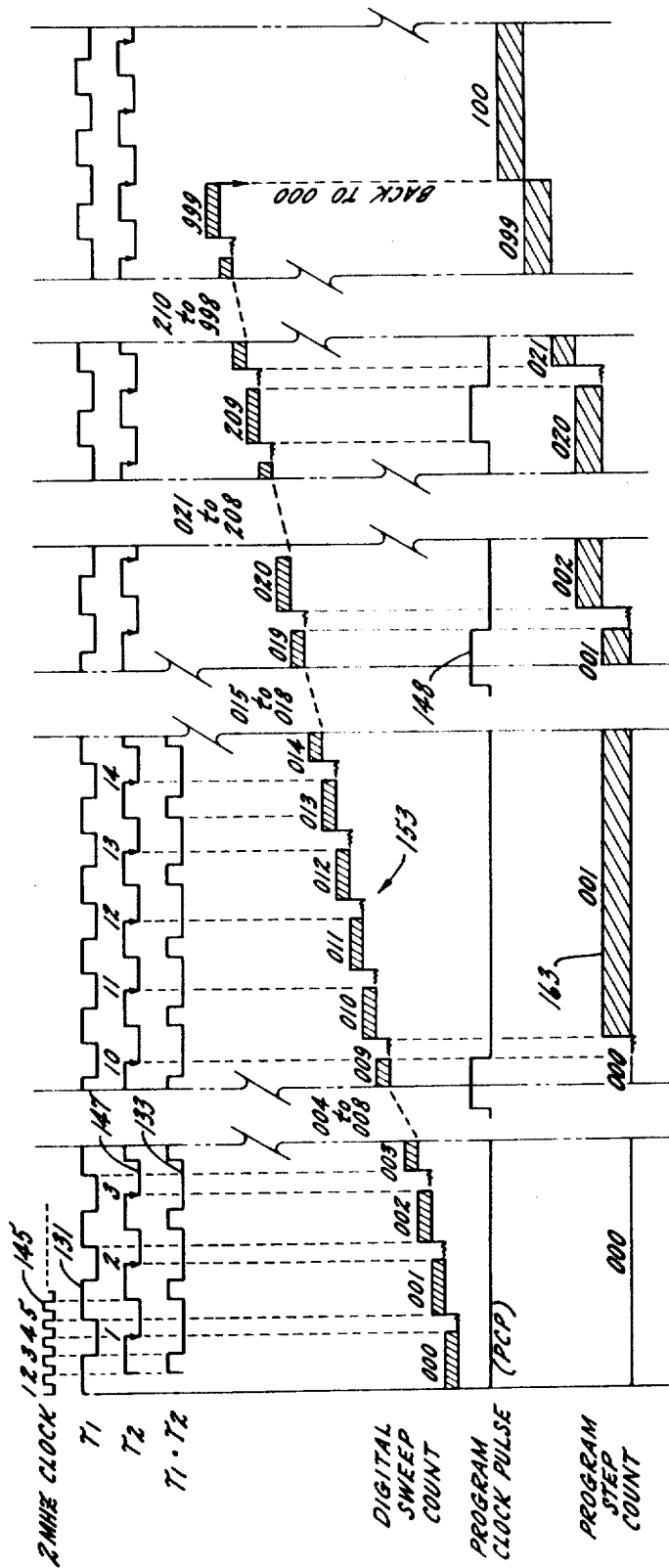


INVENTOR.
JOHN K. MCGEE
BY *Wolfe, Hubbard, Vont & Osann*
ATTORNEYS.





INVENTOR.
JOHN K. MCGEE
BY *Wolfe, Hubbard, Voigt, Osann*
ATTORNEYS.



INVENTOR.
JOHN K. MCGEE
BY *Wolfe, Haddad, Voigt, Osann*
ATTORNEYS.

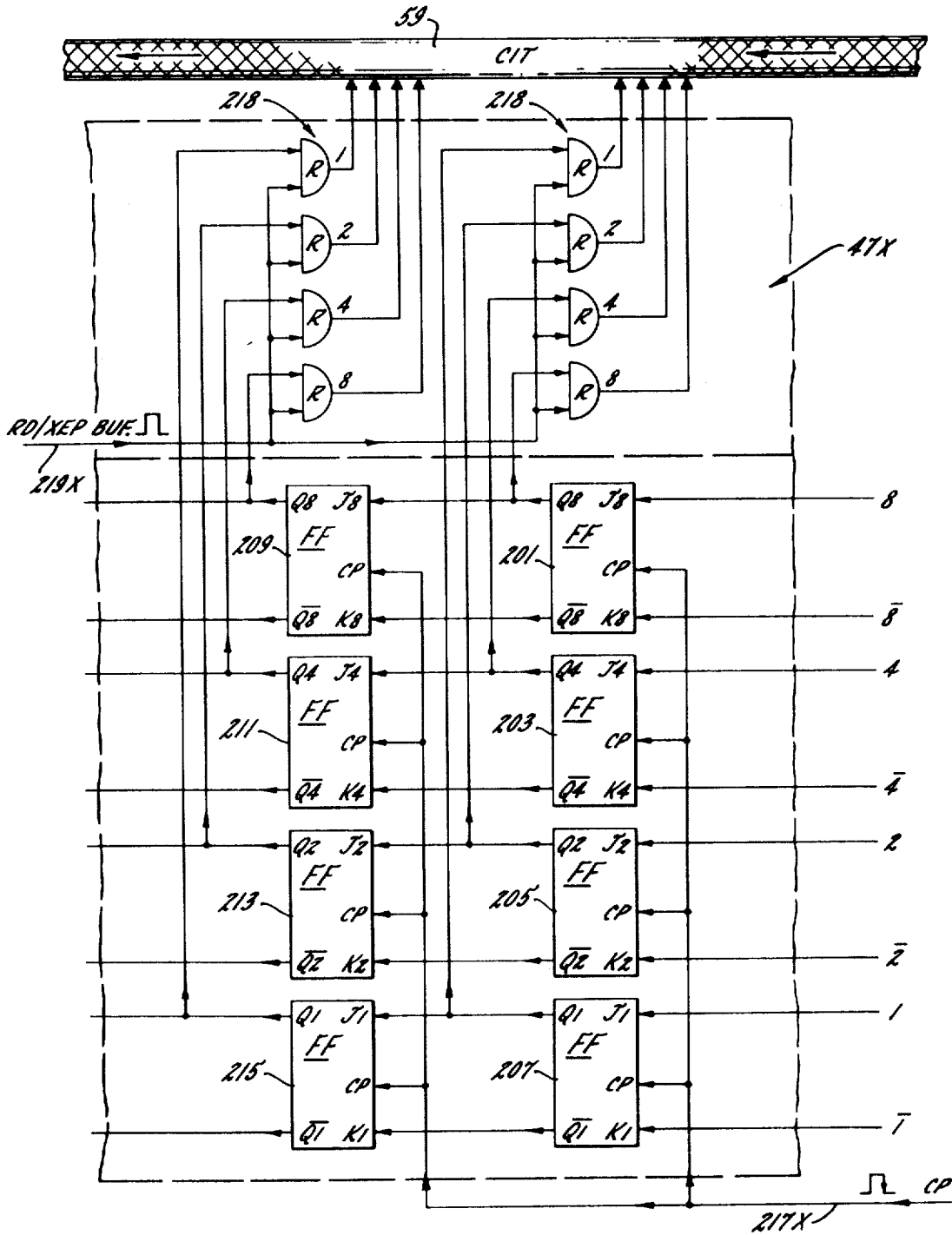
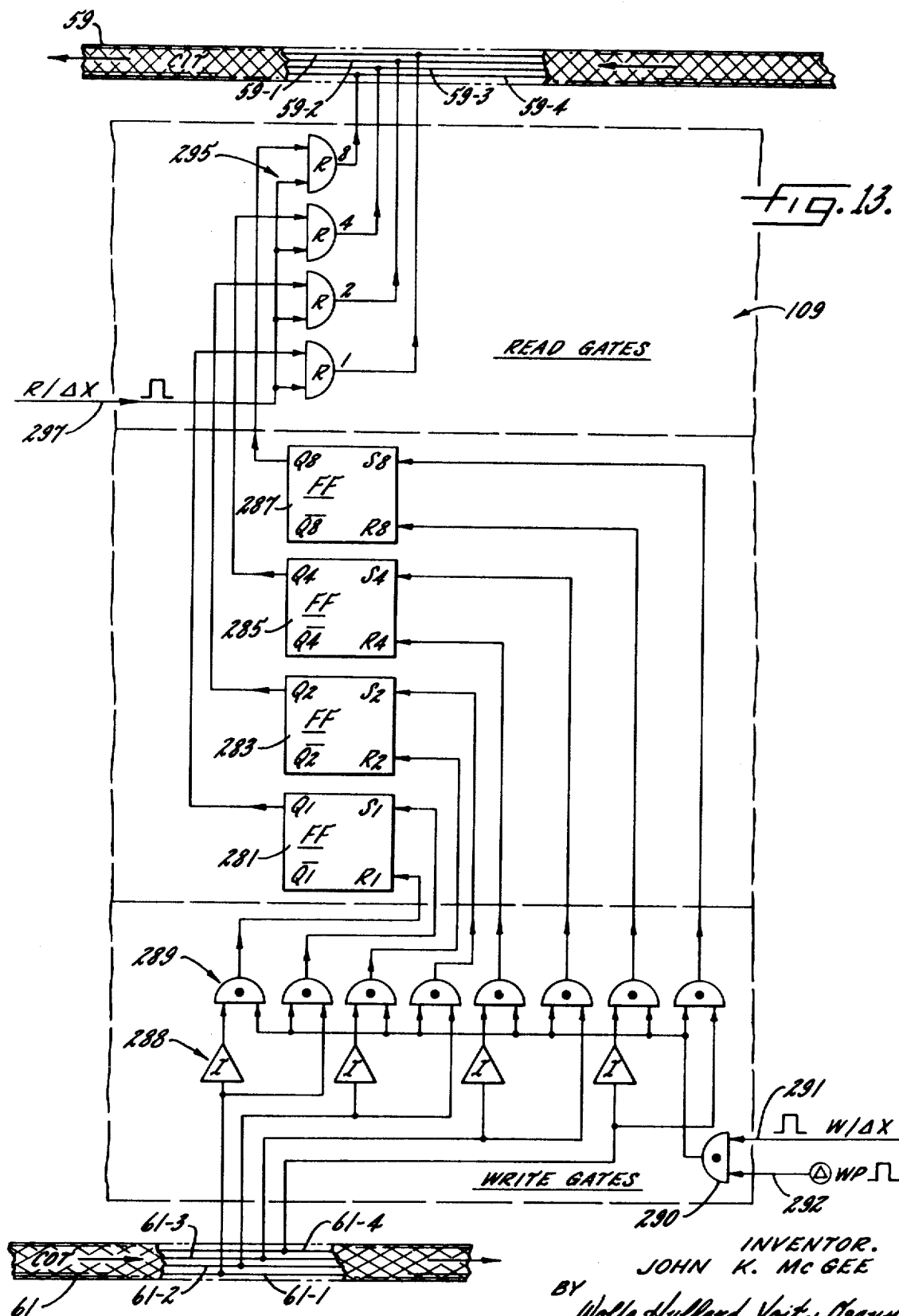


Fig. 12.

INVENTOR.
JOHN K. MCGEE
BY *Wolfe, Hubbard, Votz, Osann*
ATTORNEYS.



INVENTOR.
JOHN K. MCGEE
BY *Wolfe, Hubbard, Vort & Osann*
ATTORNEYS.

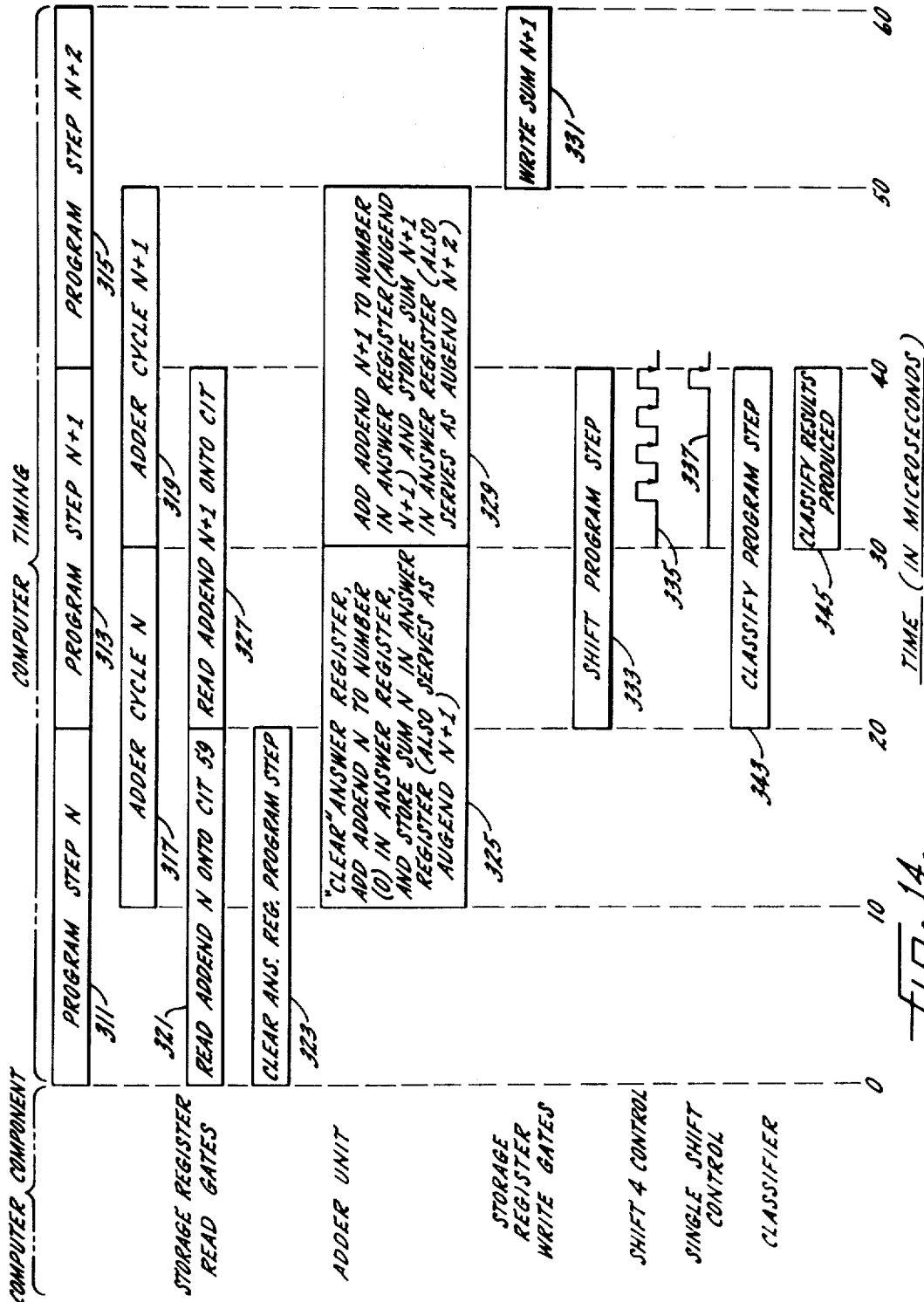


FIG. 14.

INVENTOR.
JOHN K MCGEE
BY *Wolfe, Hubbard, Vint & O'Connell*
ATTORNEYS.

	CIT-0	CIT-1	CIT-2	CIT-3	CIT-4	CIT-5	CIT-6	CIT-7	CIT-8
XEP BUF.	±			X	X	X	X	X	X
XCEP				X	X	X	X	X	X
XCP			X	X	X	X	X	X	X
XSC				X	X	X	X	X	X
XFO				5					
XOFF	±		X	X	X	X	X	X	X
ΔX	±				X	X	X	X	X
XR	±				X	X	X	X	X
YEP BUF.	±			Y	Y	Y	Y	Y	Y
YCEP				Y	Y	Y	Y	Y	Y
YCP			Y	Y	Y	Y	Y	Y	Y
YSC				Y	Y	Y	Y	Y	Y
YCPR									Y
YFO				5					
YOFF	±		Y	Y	Y	Y	Y	Y	Y
ΔY	±				Y	Y	Y	Y	Y
YR	±				Y	Y	Y	Y	Y
I BUF.	±		I	I	I	I	I	I	I
I	±		I	I	I	I	I	I	I
J BUF.	±		J	J	J	J	J	J	J
J	±		J	J	J	J	J	J	J
G BUF.									G
G									G
Q BUF.									Q
Q									Q

Fig. 15.

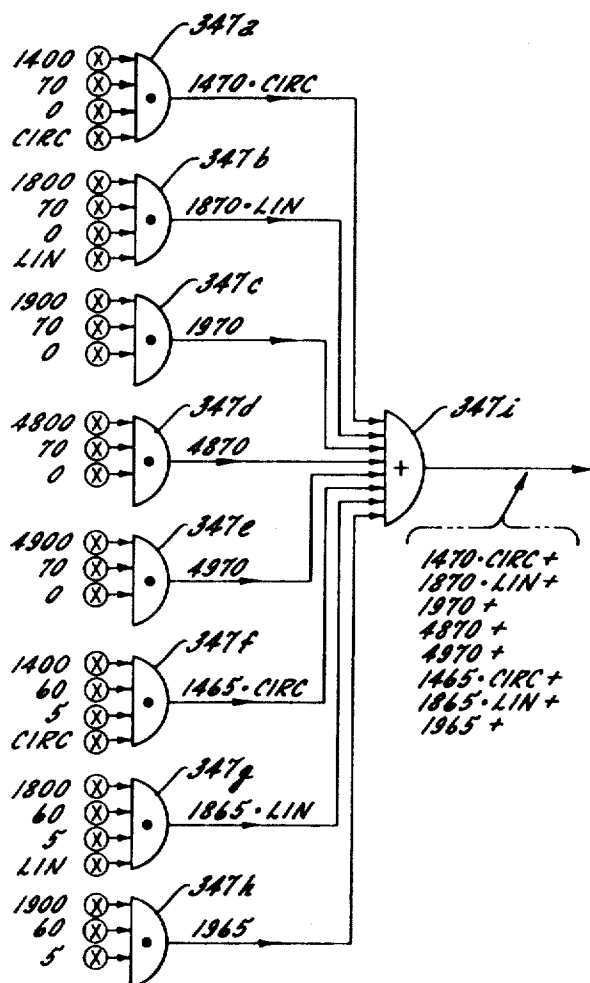


Fig. 16.

INVENTOR.
JOHN K. MCGEE
BY *Wolfe, Hubbard, Vint, & Orann*
ATTORNEYS.

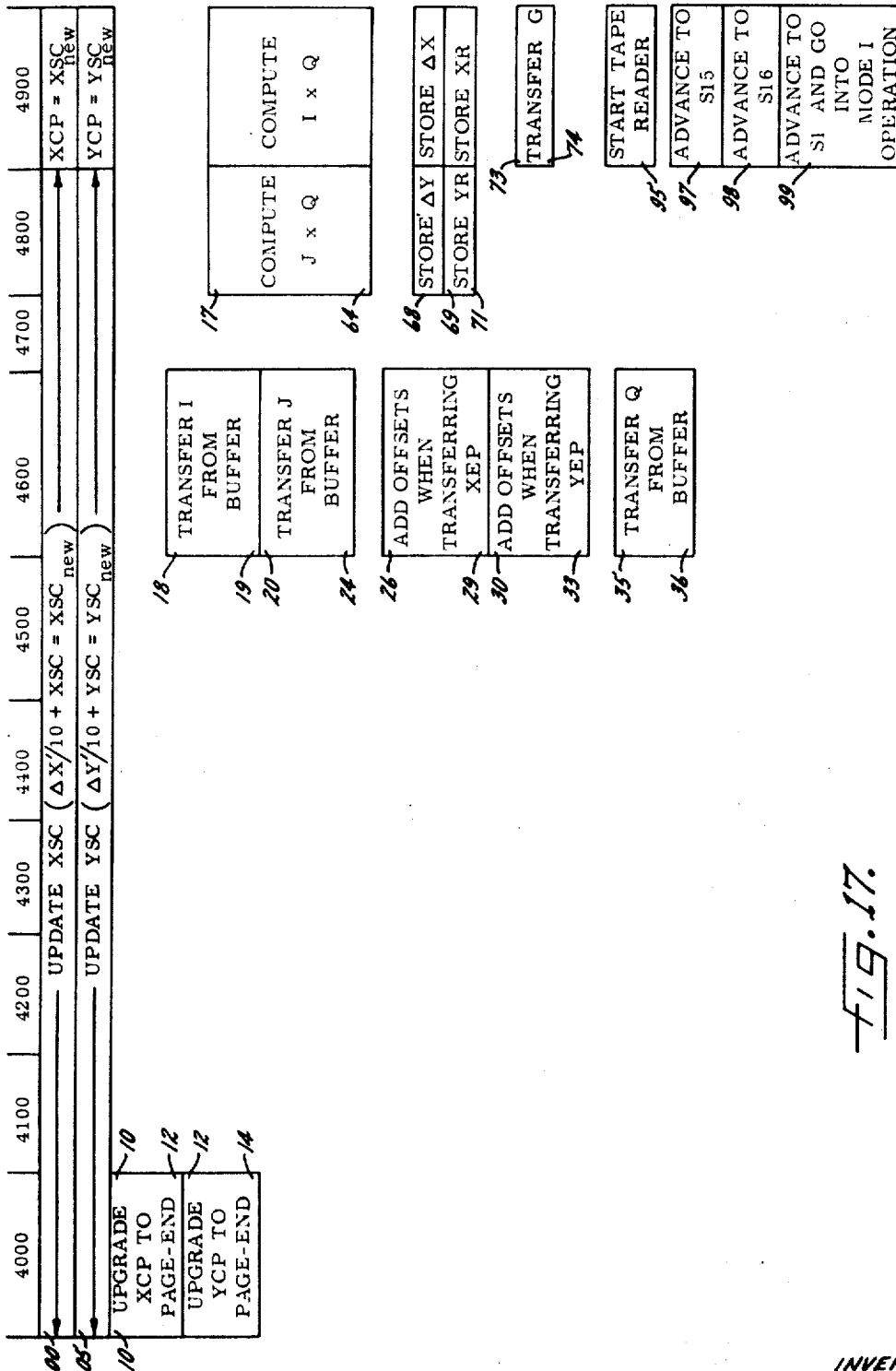


FIG. 17.

LAST (OR 10TH) ADJUSTED MACROMOVE OF A LINEAR PATH SEGMENT AND SET-UP CALCULATIONS FOR ENTERING ANOTHER LINEAR PATH SEGMENT

INVENTOR.
JOHN K. MCGEE
BY *Wolfe, Hubbard, Vontz, Osann*
ATTORNEYS.

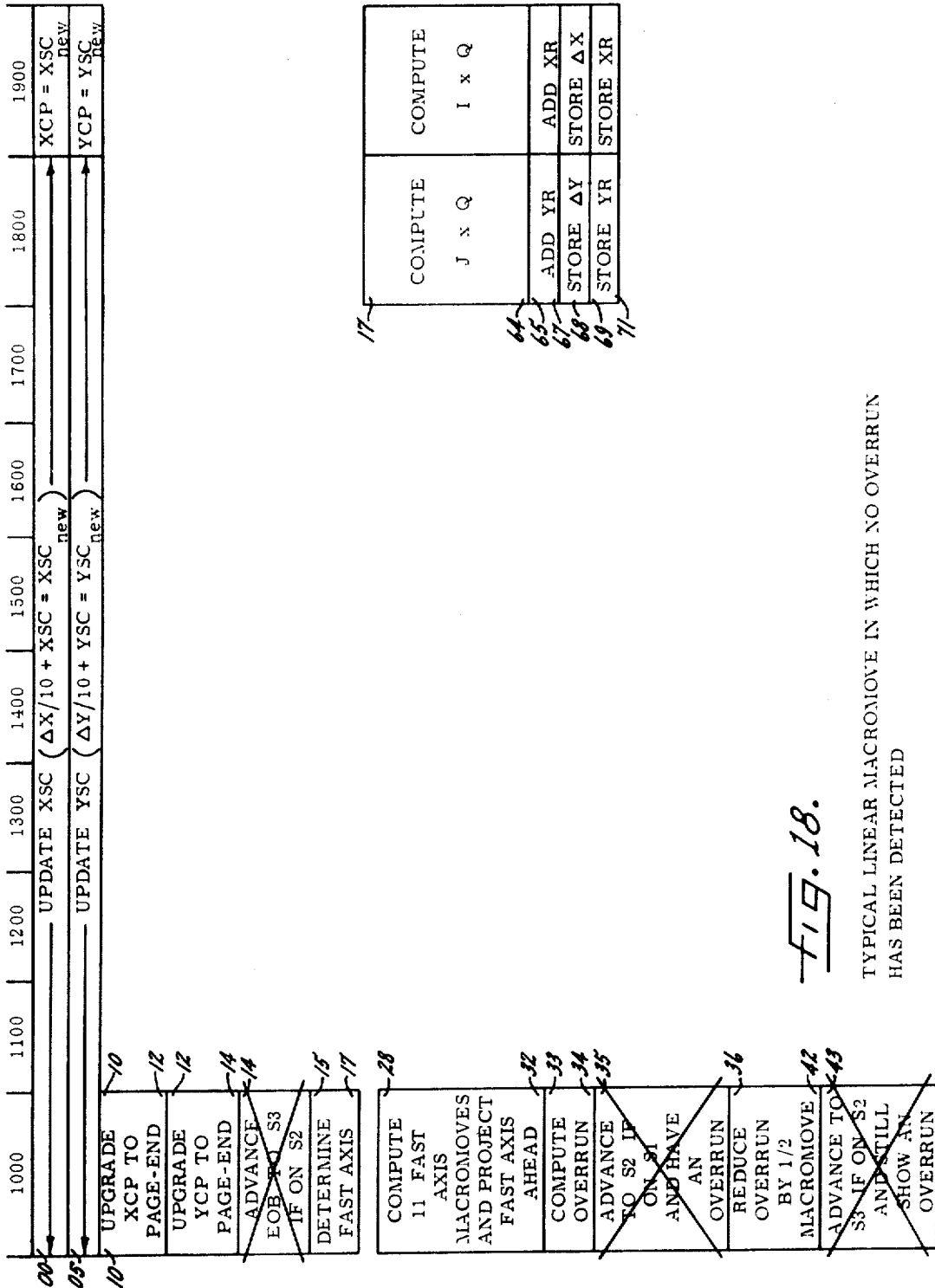
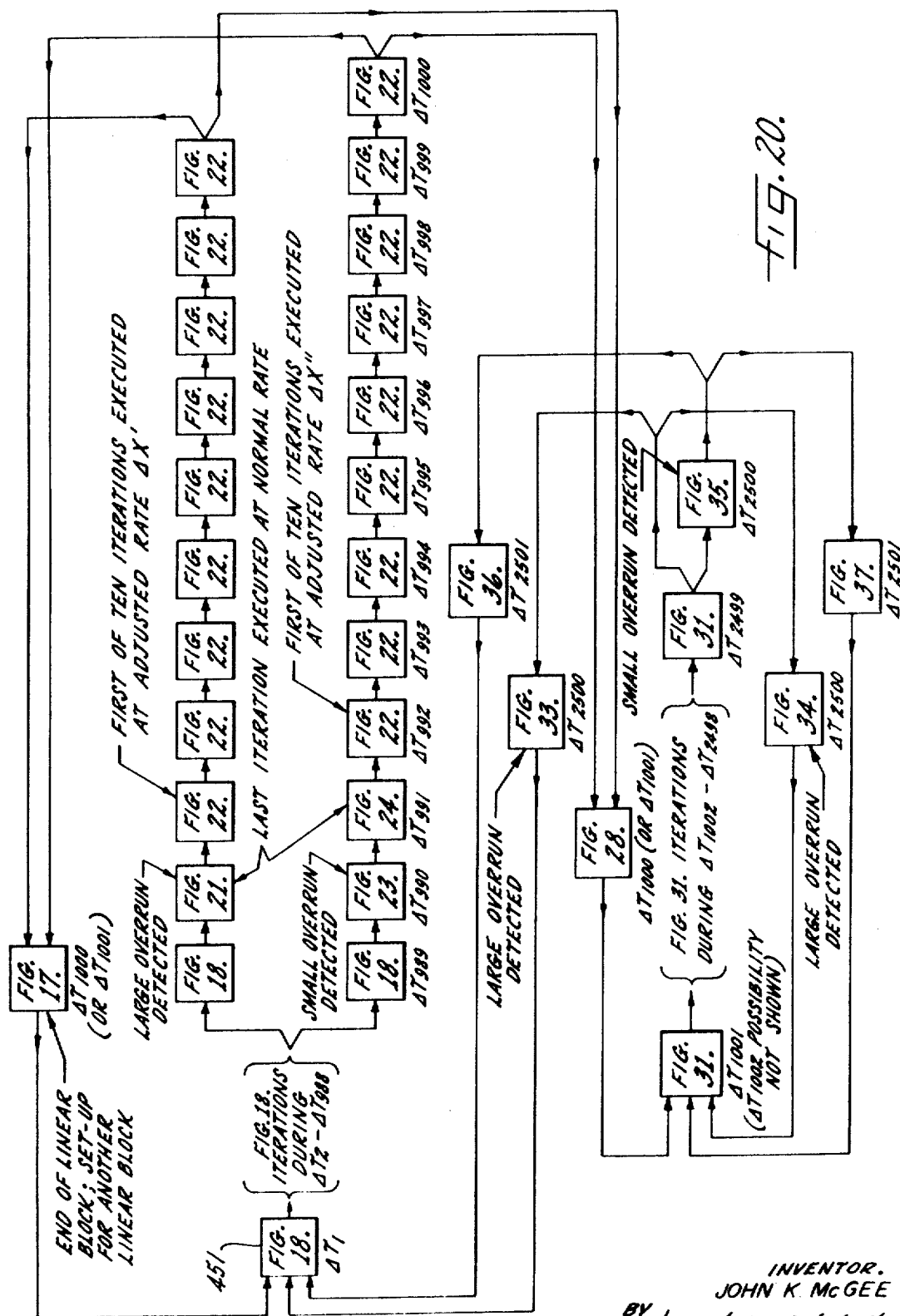


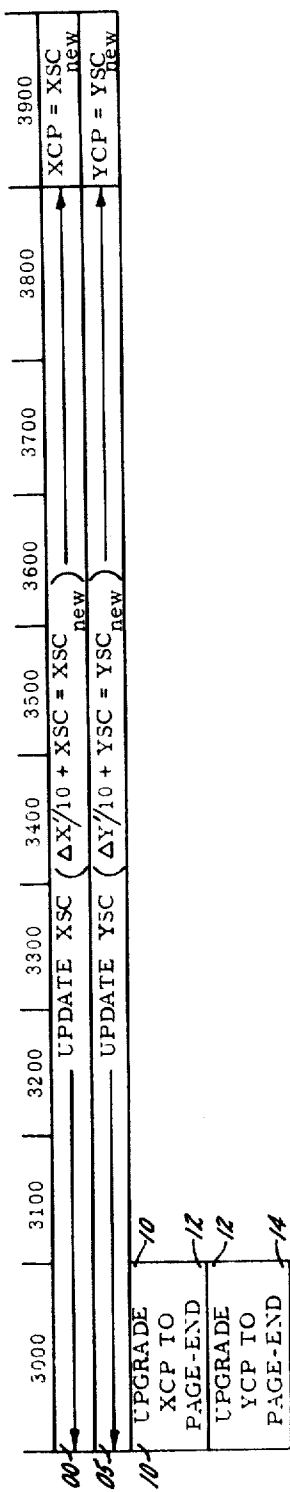
FIG. 18.

TYPICAL LINEAR MACROMOVE IN WHICH NO OVERRUN HAS BEEN DETECTED

INVENTOR.
JOHN K. MCGEE
BY *Wolfe, Hubbard, Voigt, O'Connell*
ATTORNEYS.



INVENTOR.
JOHN K. MCGEE
BY *Wolfe, Hubbard, Voigt & Osann*
ATTORNEYS.



ADVANCE
EOB BY
ONE
COUNT

FIG. 22.

1ST, 2ND, 3RD, 4TH, 5TH, 6TH, 7TH, 8TH and 9TH ADJUSTED MACROMOVES
OF 10-MACROMOVE LINEAR END-OF-BLOCK SEQUENCE

INVENTOR.
JOHN K. MCGEE
BY *Wolfe, Hubbard, Vont & Osann*
ATTORNEYS.

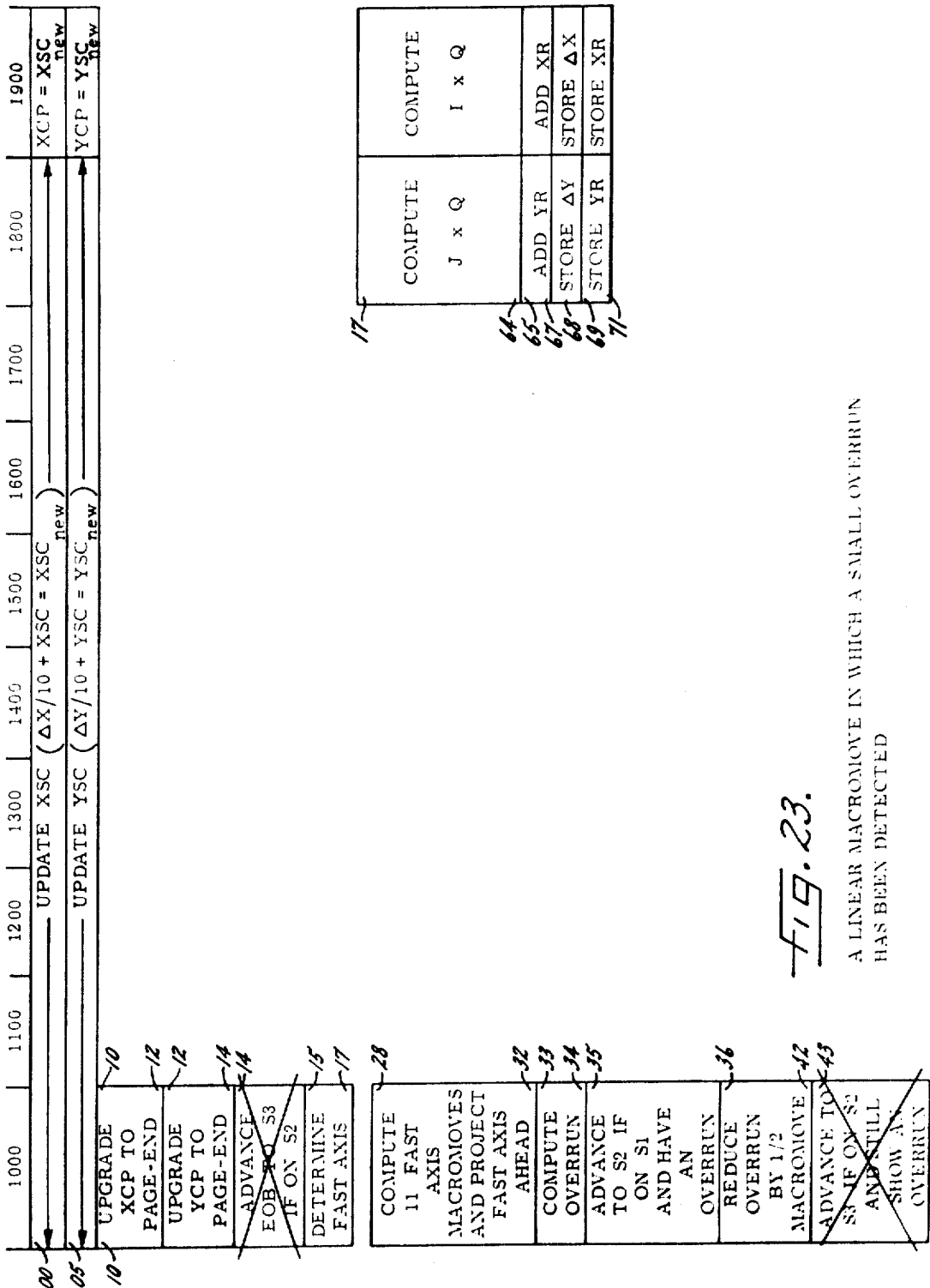
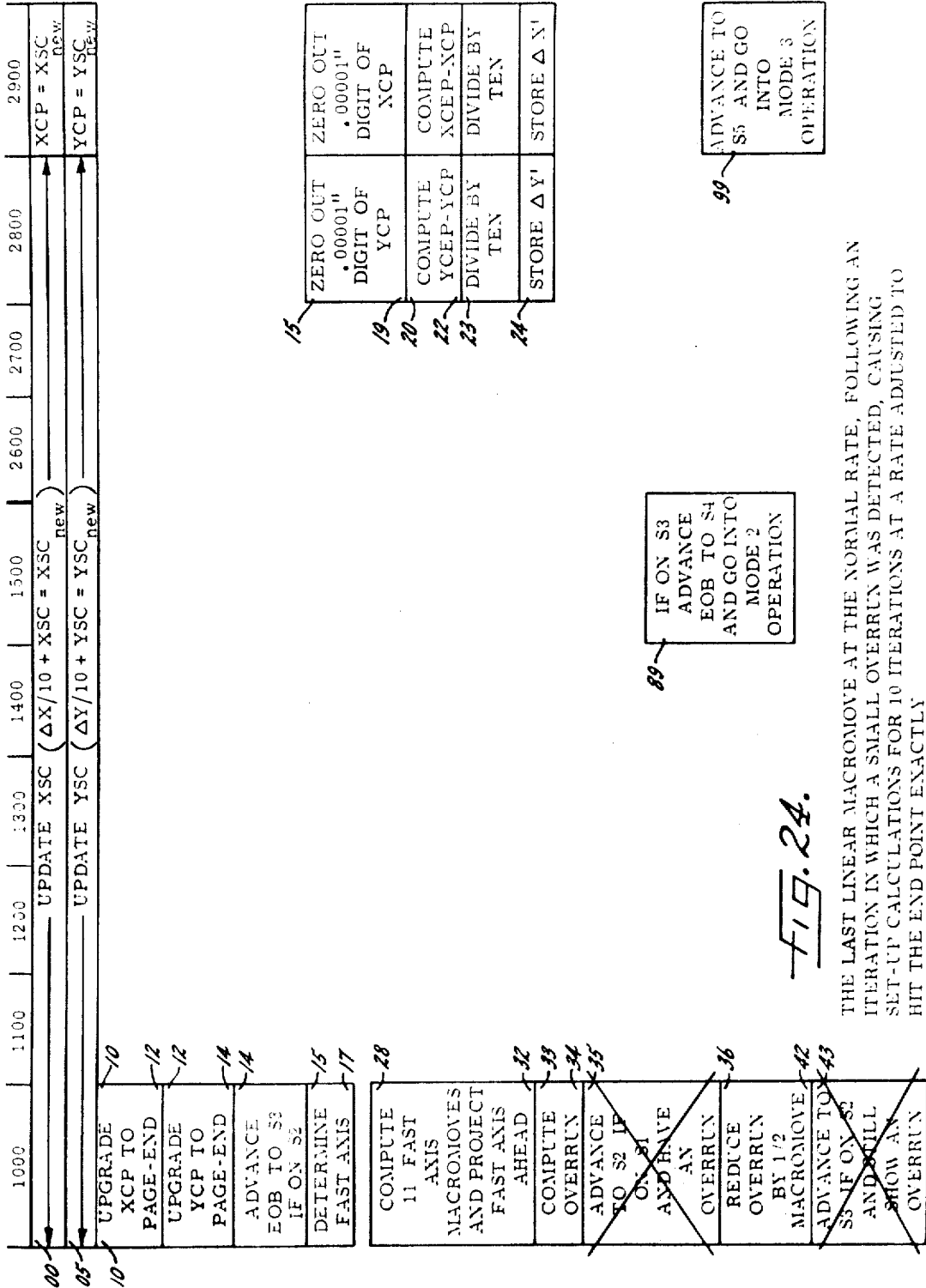


FIG. 23.

A LINEAR MACROMOVE IN WHICH A SMALL OVERRUN HAS BEEN DETECTED

INVENTOR.
JOHN K. MCGEE
BY *Wolfe, Hubbard, Voigt, Osann*
ATTORNEYS.



INVENTOR.
JOHN K MCGEE
BY *Wolfe, Hubbard, Voit & Orank*
ATTORNEYS.

Fig. 26a.

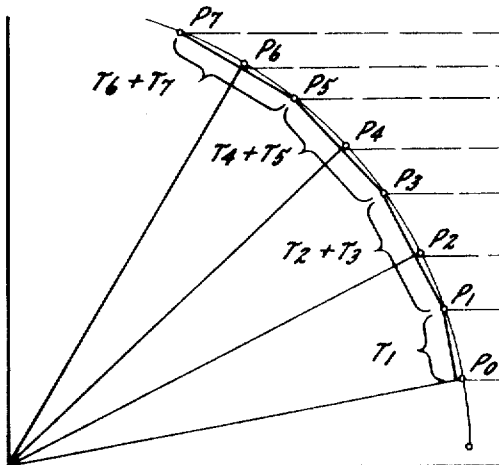
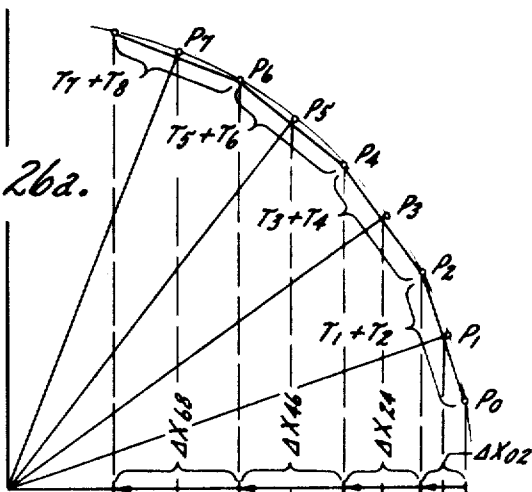


Fig. 26b.

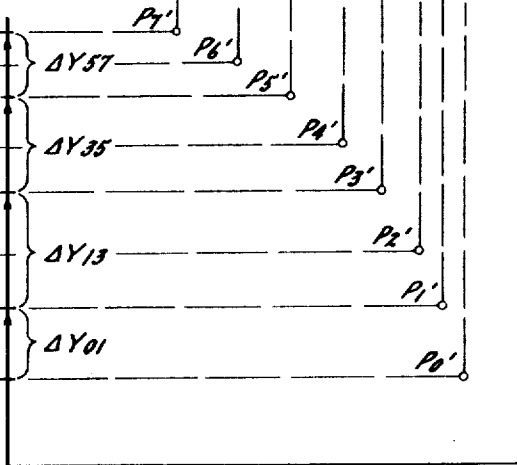


Fig. 26c.

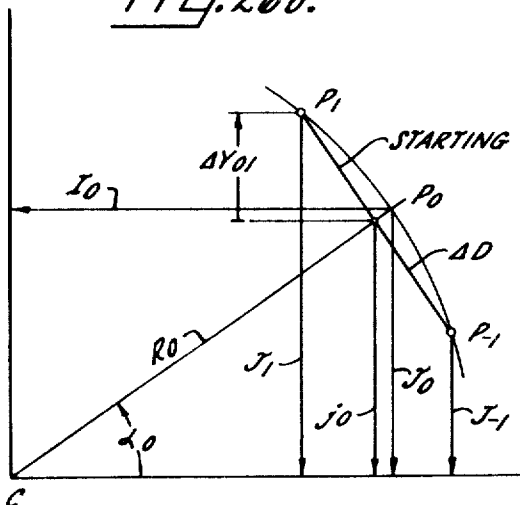


Fig. 27.

$$J_1 \approx J_0 - \Delta Y_{01}$$

$$\text{ERROR} = J_0 - j_0$$

INVENTOR.
JOHN K. MCGEE
BY *Wolfe, Hubbard, Voit, & Orann*
ATTORNEYS.

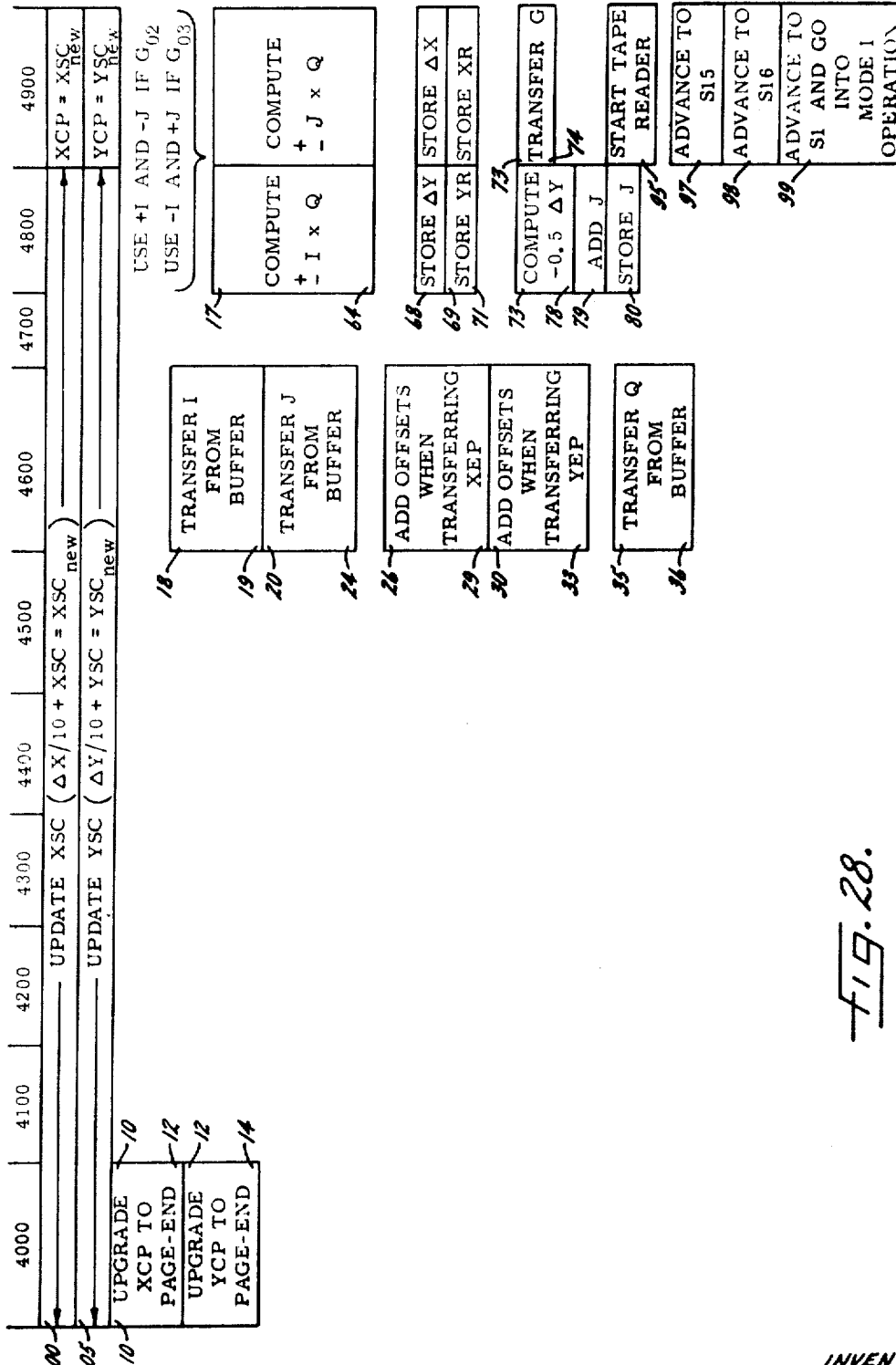
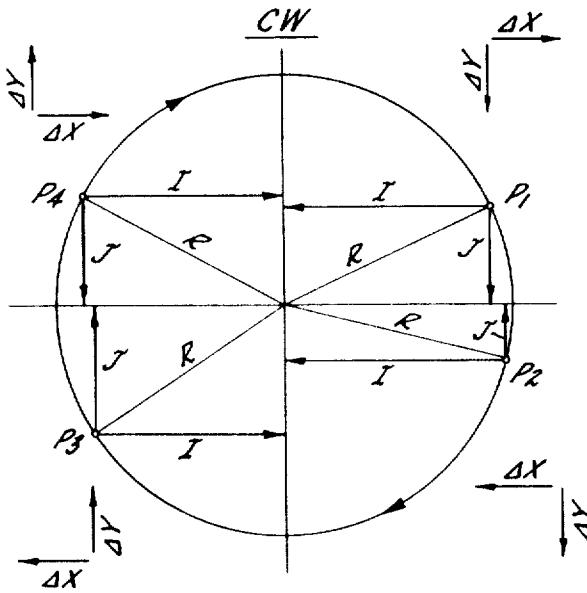


FIG. 28.

LAST (OR 10TH) ADJUSTED MACROMOVE OF A LINEAR PATH SEGMENT AND SET-UP CALCULATIONS FOR ENTERING A CIRCULAR PATH SEGMENT

INVENTOR.
JOHN K. MCGEE
BY *Wolfe, Hubbard, Voss, Clark*
ATTORNEYS.

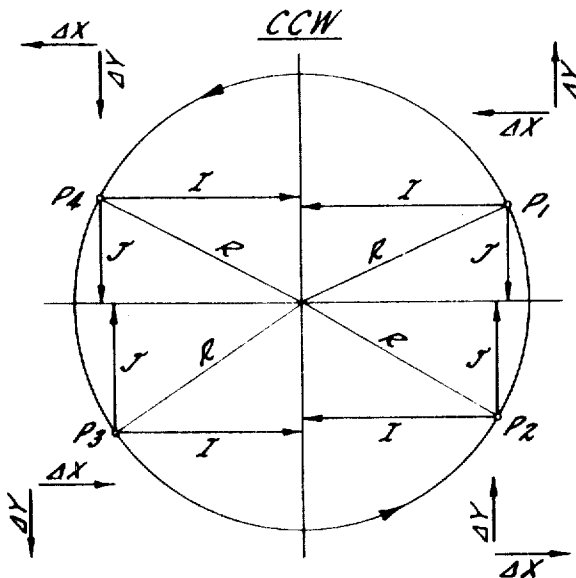


$$\Delta X = J \cdot Q \quad (18)$$

$$\Delta Y = I \cdot Q \quad (19)$$

$\therefore$ USE $-J$ AND $+I$

Fig. 29a.



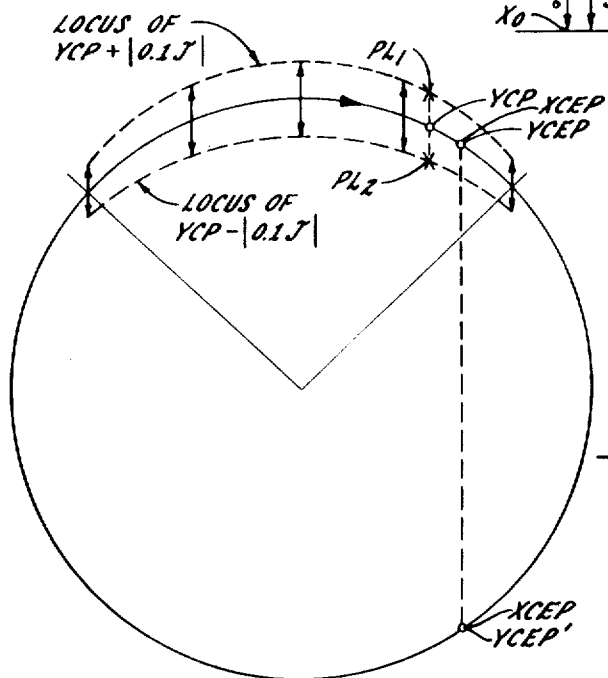
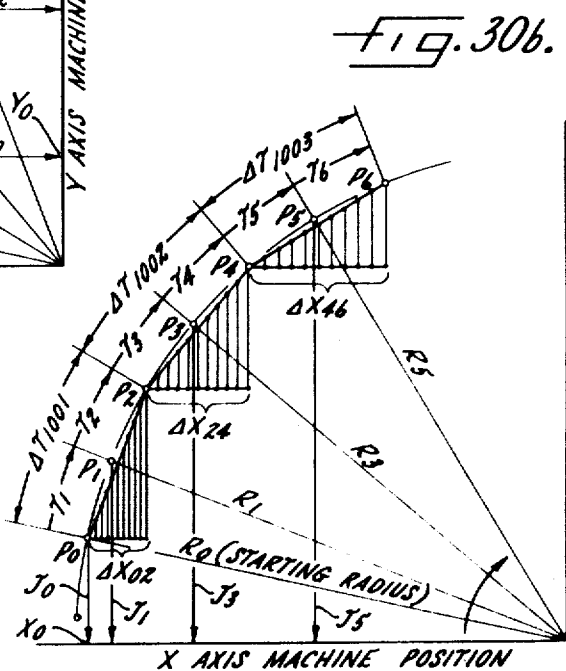
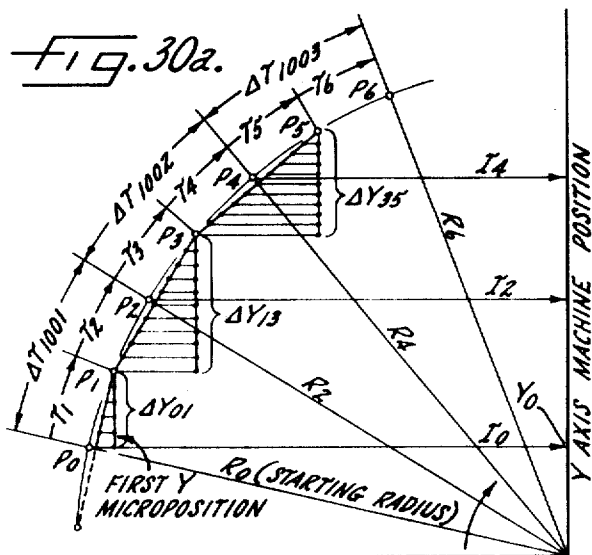
$$\Delta X = J \cdot Q \quad (18)$$

$$\Delta Y = I \cdot Q \quad (19)$$

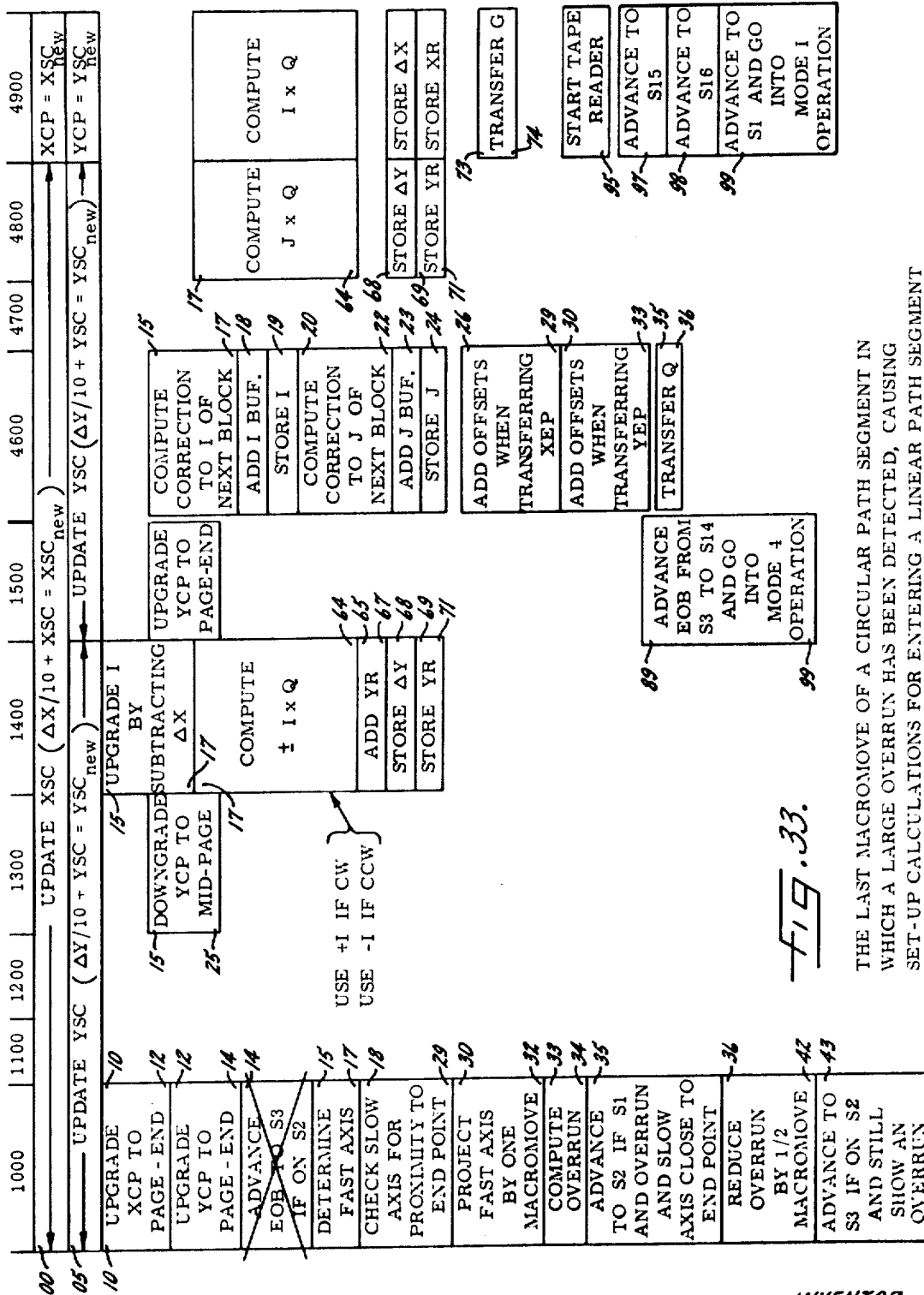
$\therefore$ USE $+J$ AND $-I$

Fig. 29b.

INVENTOR.
JOHN K. MCGEE
BY *Wolfe, Hubbard, Voigt, Osann*
ATTORNEYS.



INVENTOR.
JOHN K. MCGEE
BY *Wolfe, Hubbard, Vint & Crane*
ATTORNEYS.



INVENTOR.
JOHN K. McGEE
BY *Wolfe, Hubbard, Vent & Orann*
ATTORNEYS.

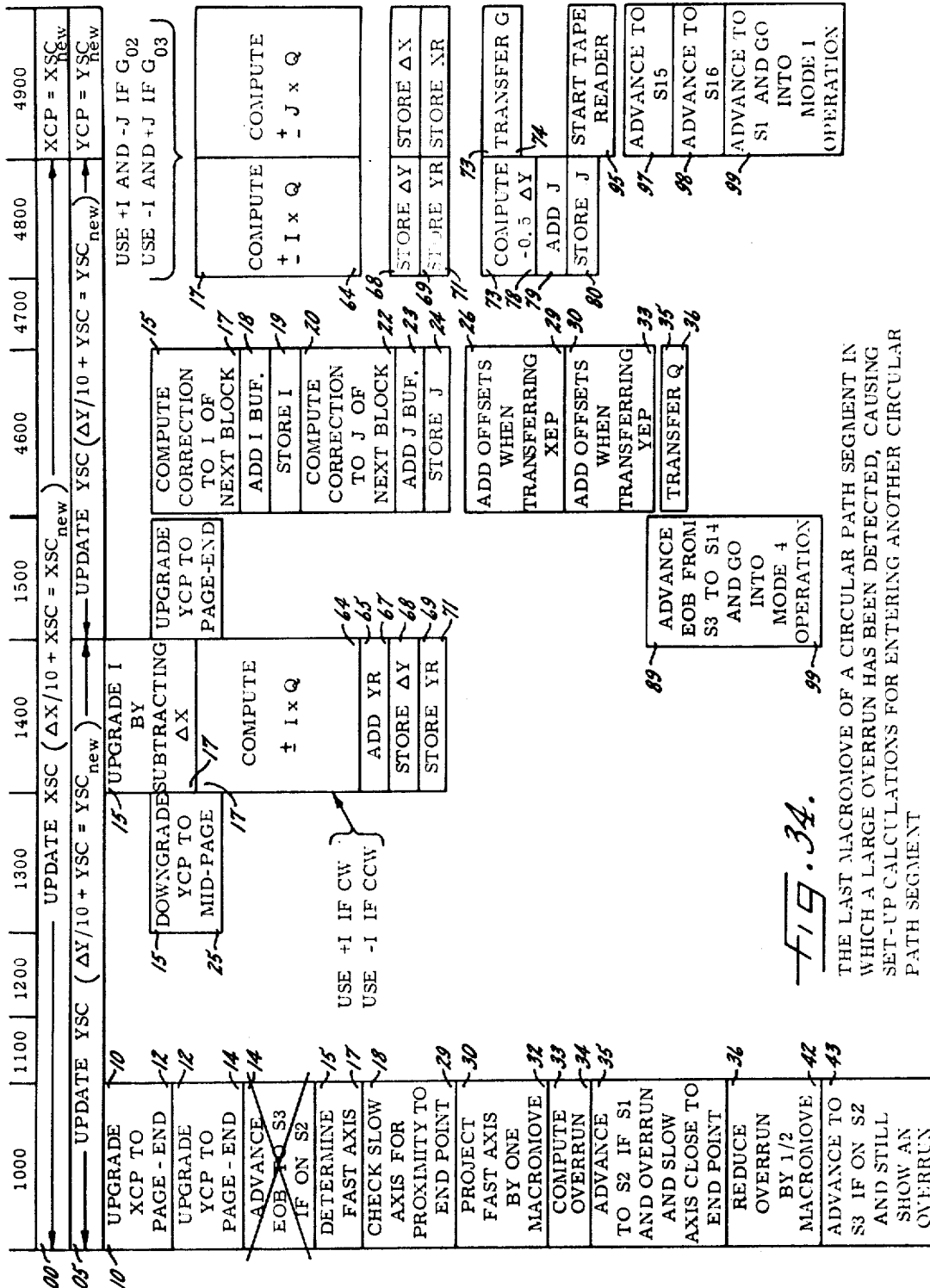


Fig. 34.

THE LAST MACROMOVE OF A CIRCULAR PATH SEGMENT IN WHICH A LARGE OVERRUN HAS BEEN DETECTED, CAUSING SET-UP CALCULATIONS FOR ENTERING ANOTHER CIRCULAR PATH SEGMENT

INVENTOR.
JOHN K. MCGEE
BY *Wolfe, Hubbard, Vortz, Orann*
ATTORNEYS.

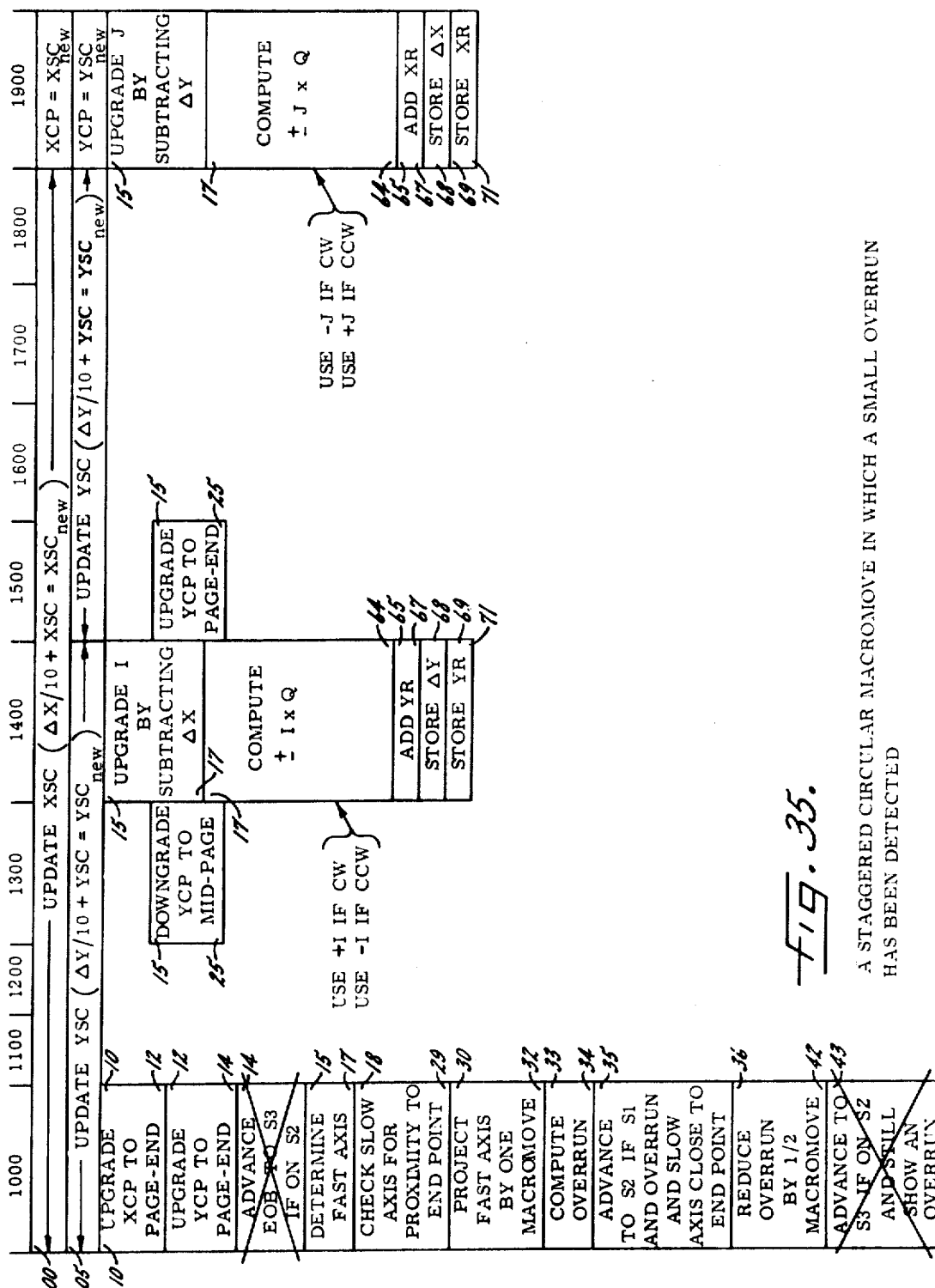


Fig. 35.

A STAGGERED CIRCULAR MACROMOVE IN WHICH A SMALL OVERRUN HAS BEEN DETECTED

INVENTOR.
JOHN K. MCGEE
BY *Wolfe, Hubbard, Vort, Osann*
ATTORNEYS.

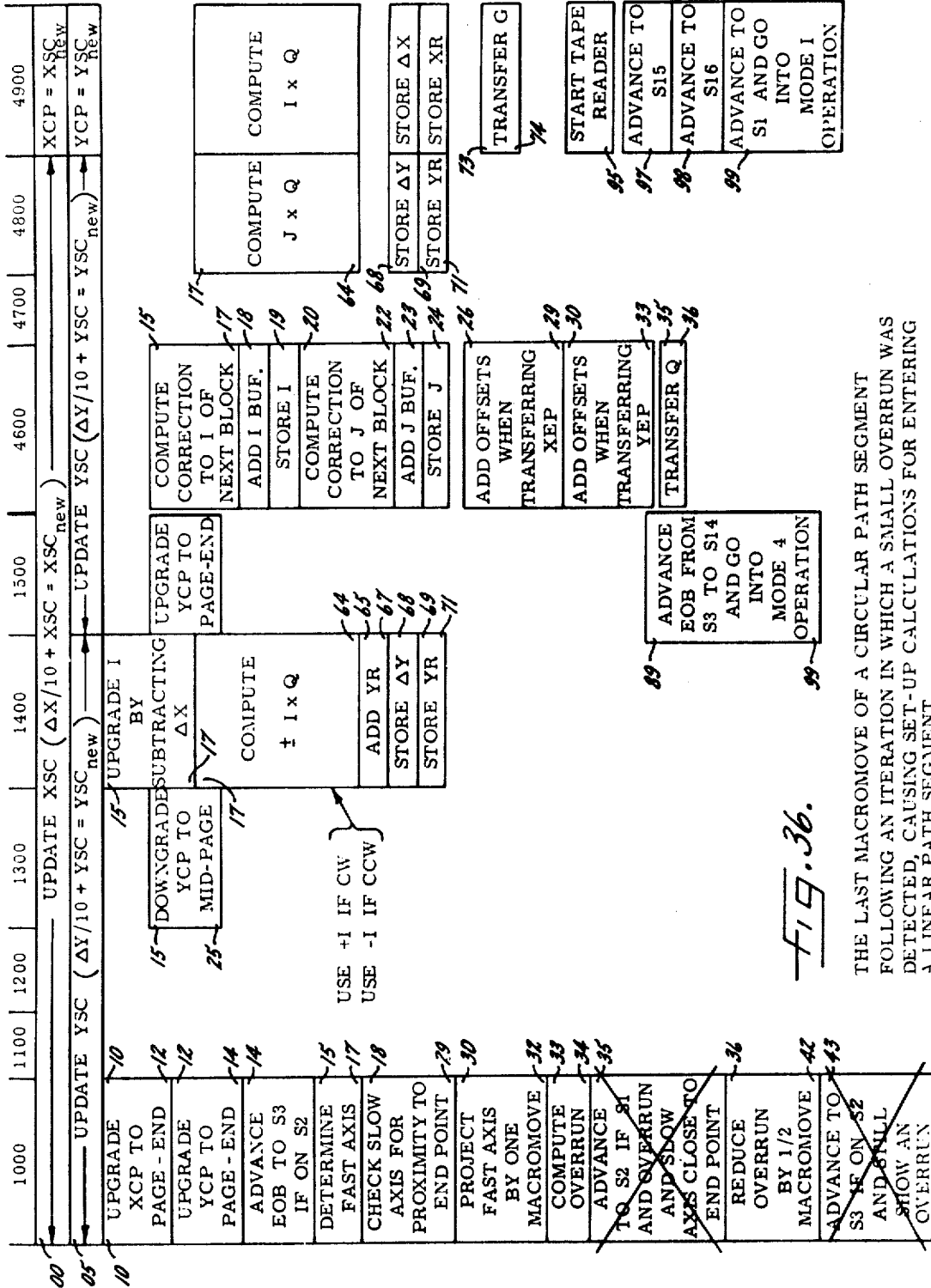
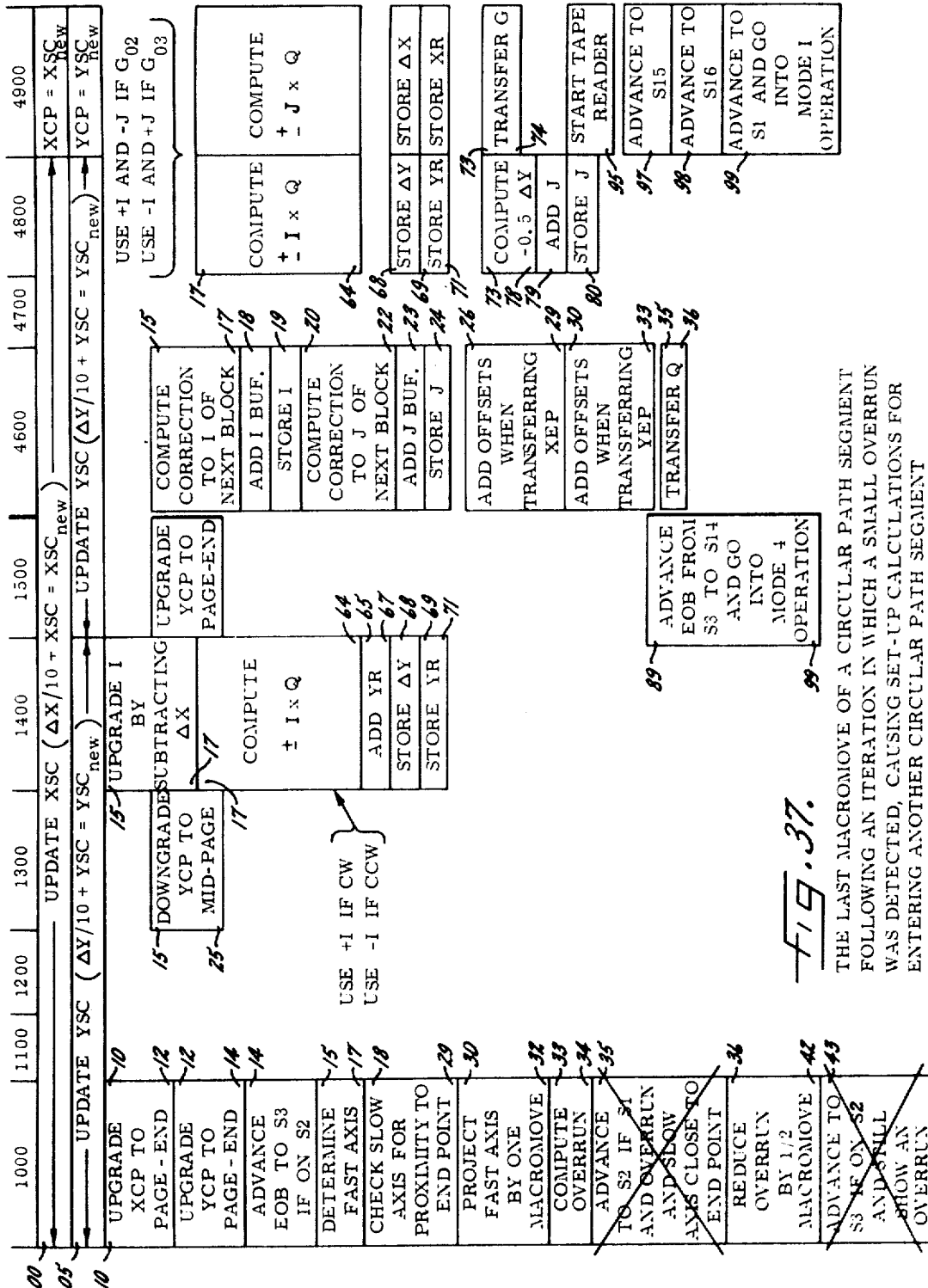


FIG. 36.

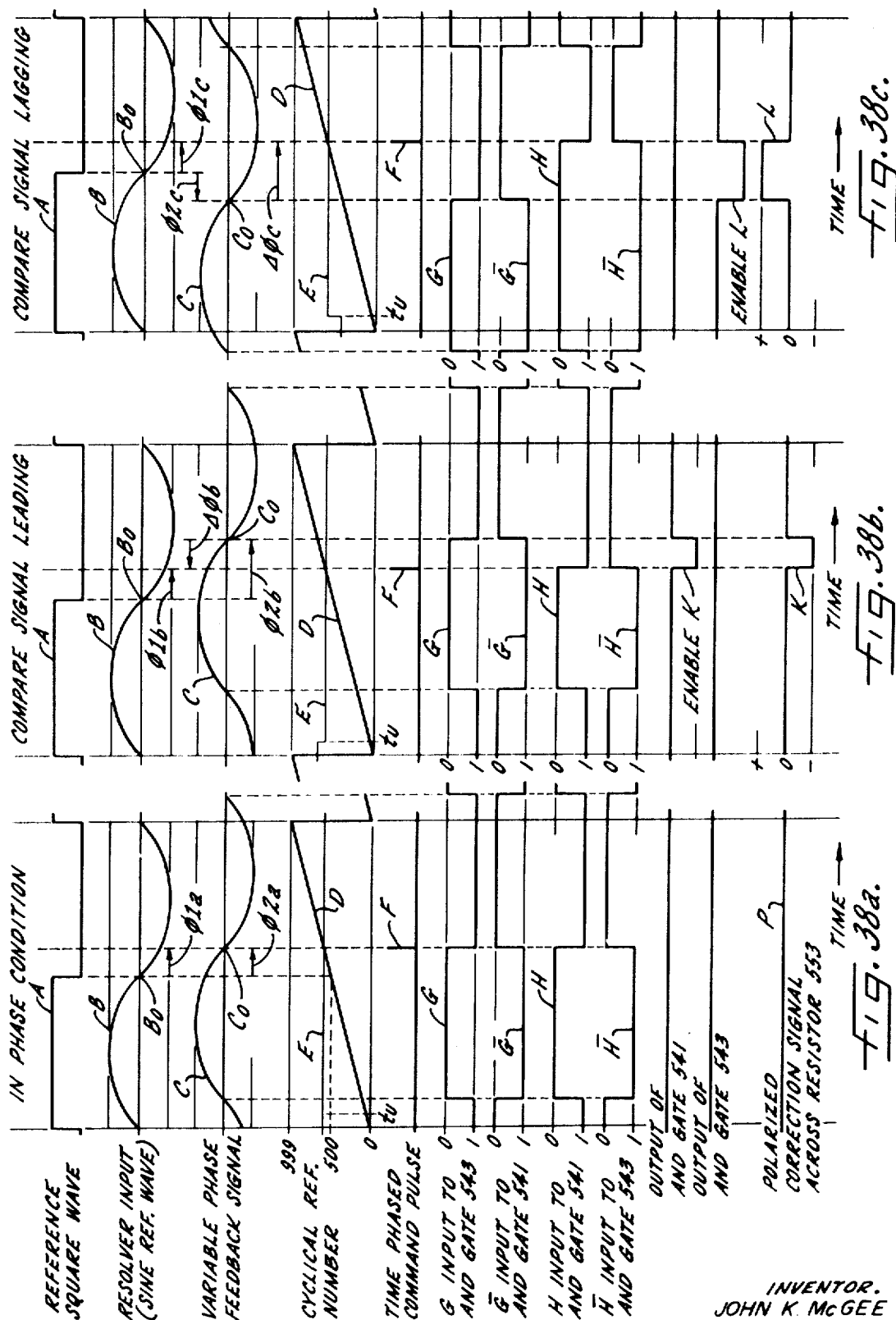
THE LAST MACROMOVE OF A CIRCULAR PATH SEGMENT FOLLOWING AN ITERATION IN WHICH A SMALL OVERRUN WAS DETECTED, CAUSING SET-UP CALCULATIONS FOR ENTERING A LINEAR PATH SEGMENT

INVENTOR.
JOHN K. MCGEE
BY
Wolfe, Hubbard, Voigt & O'Hara
ATTORNEYS.



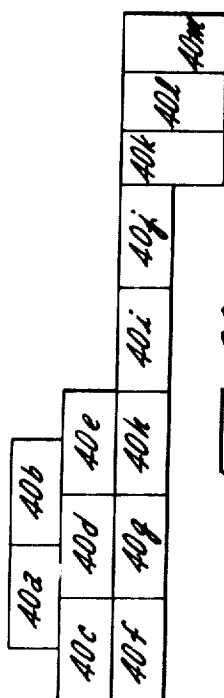
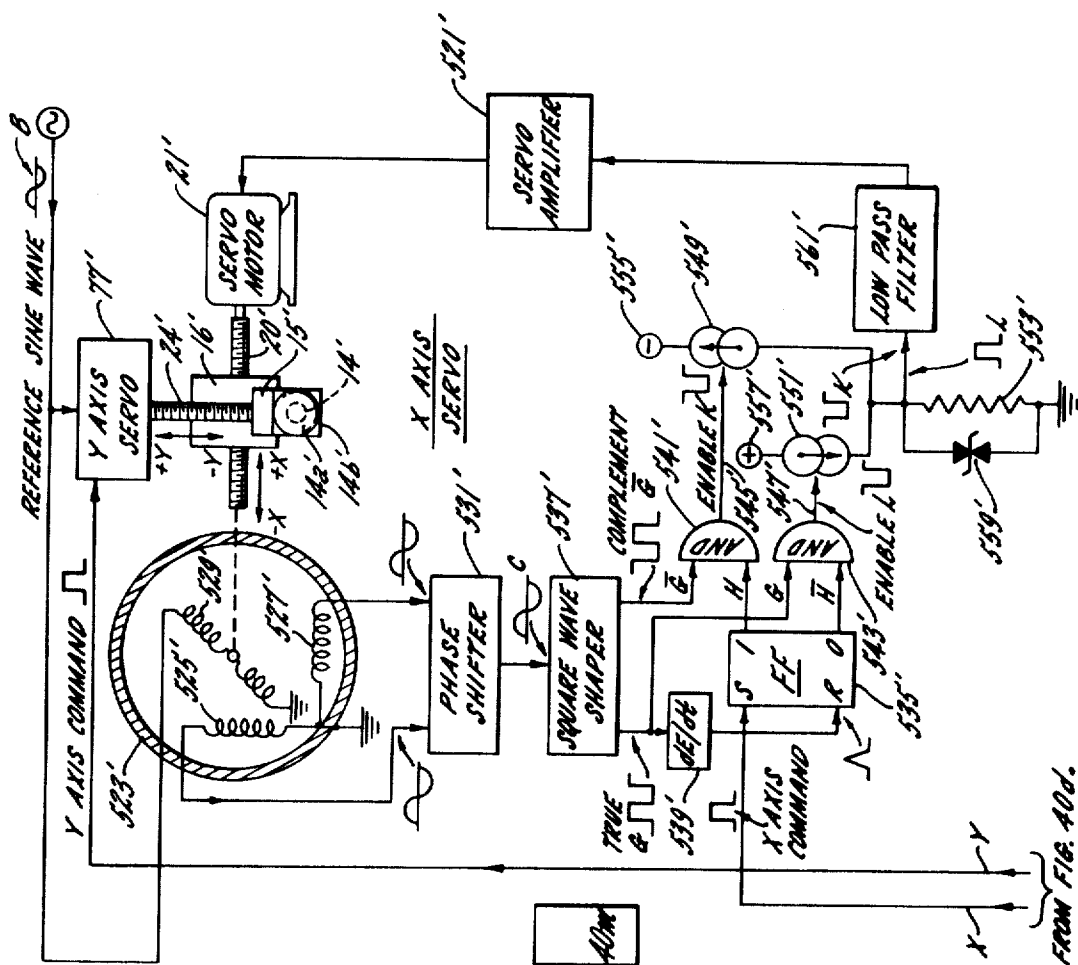
THE LAST MACROMOVE OF A CIRCULAR PATH SEGMENT FOLLOWING AN ITERATION IN WHICH A SMALL OVERRUN WAS DETECTED, CAUSING SET-UP CALCULATIONS FOR ENTERING ANOTHER CIRCULAR PATH SEGMENT

INVENTOR.
JOHN K. MCGEE
BY *Wolfe, Hubbard, Voigt & Osann*
ATTORNEYS.



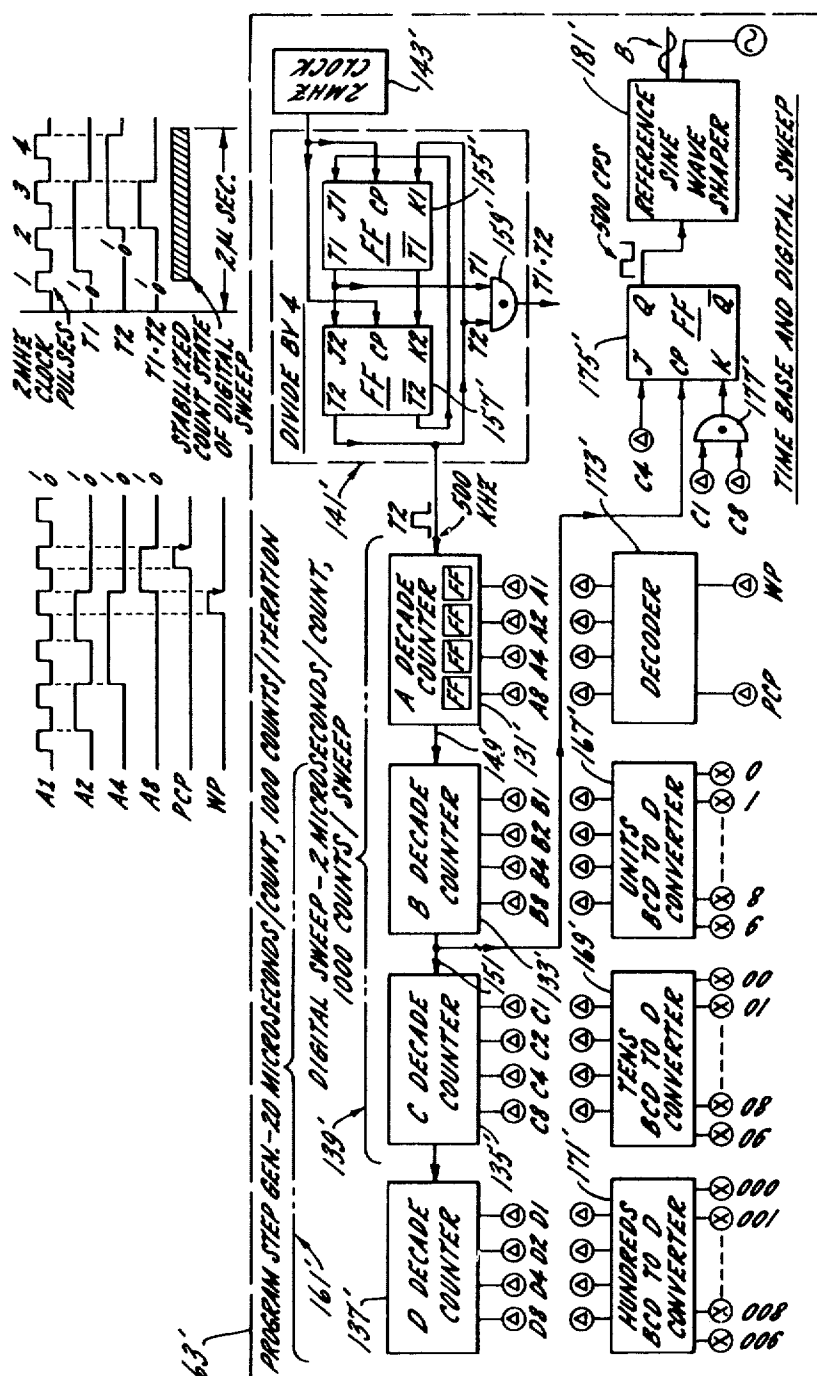
INVENTOR.
JOHN K. MCGEE

BY *Wolfe, Hubbard, Votz & Osann*
ATTORNEYS.



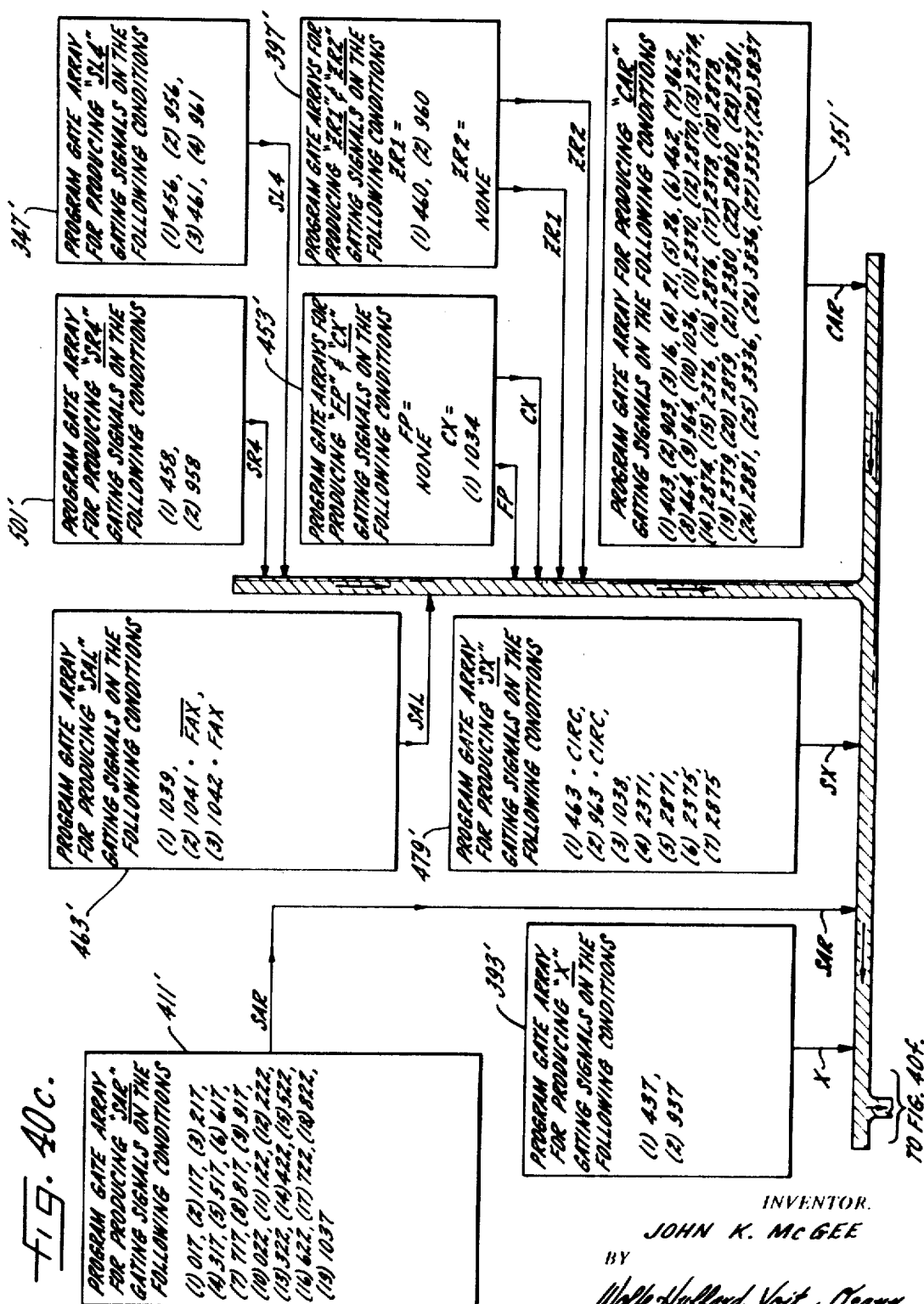
INVENTOR.
JOHN K. MCGEE

BY
Wolfe, Hubbard, Voigt & Osann
ATTORNEYS.



519. 406.

INVENTOR.
JOHN K. MCGEE
BY
Wolfe, Hallard, Voit & Osann
ATTORNEYS.

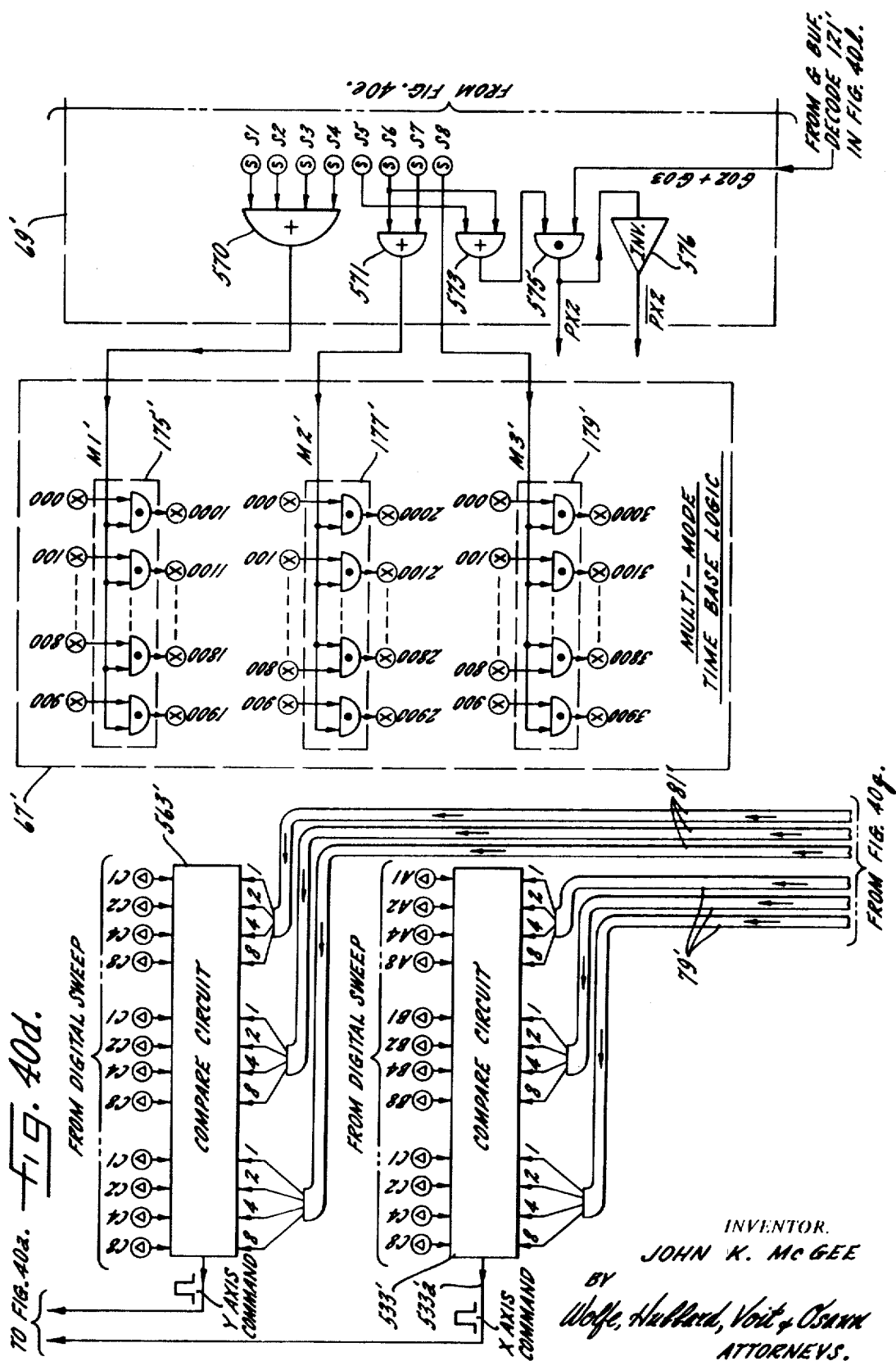


INVENTOR.

JOHN K. MCGEE

BY

Wolfe, Harland, Voit & Osann
ATTORNEYS.



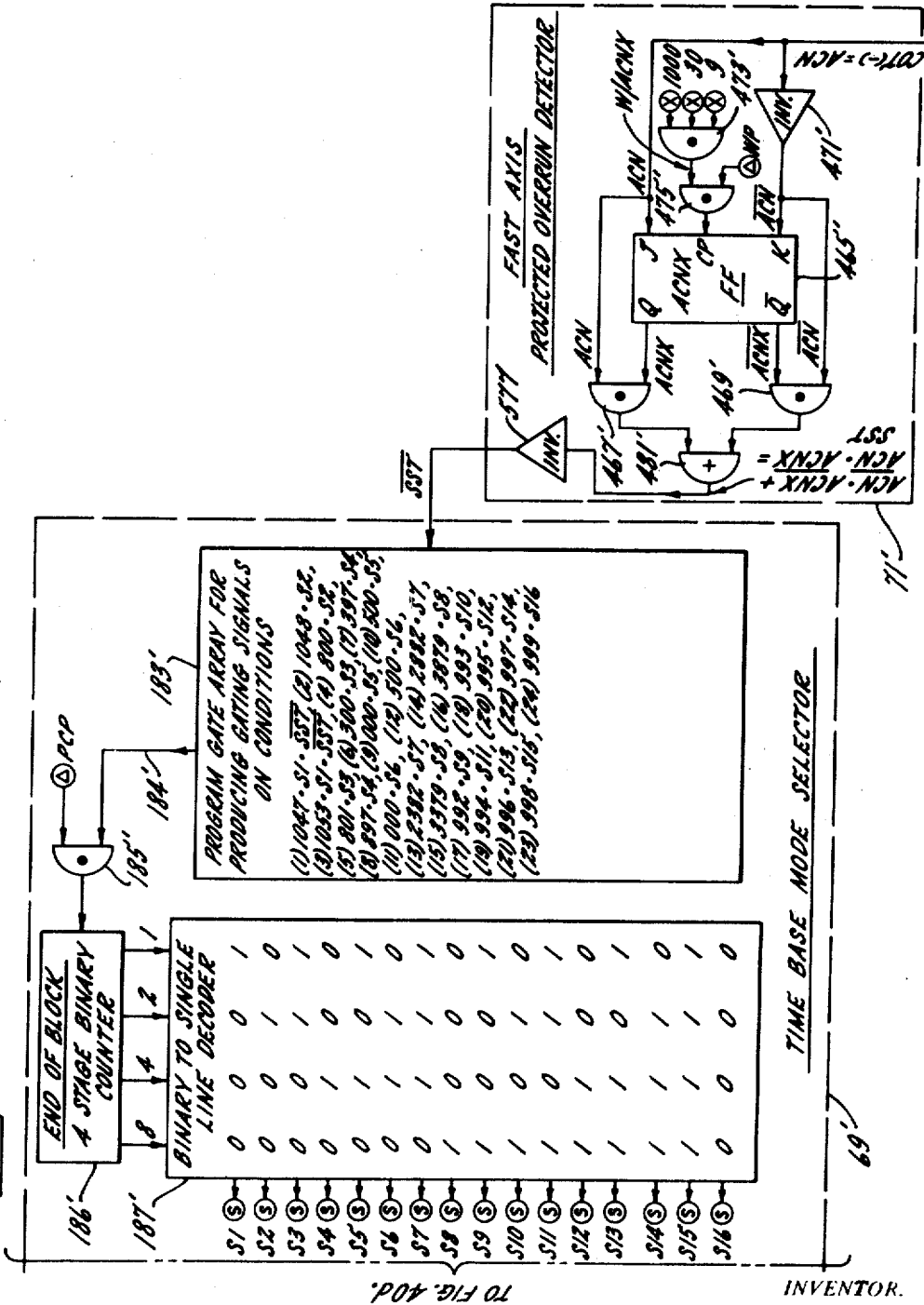
INVENTOR

JOHN K. MCGEE

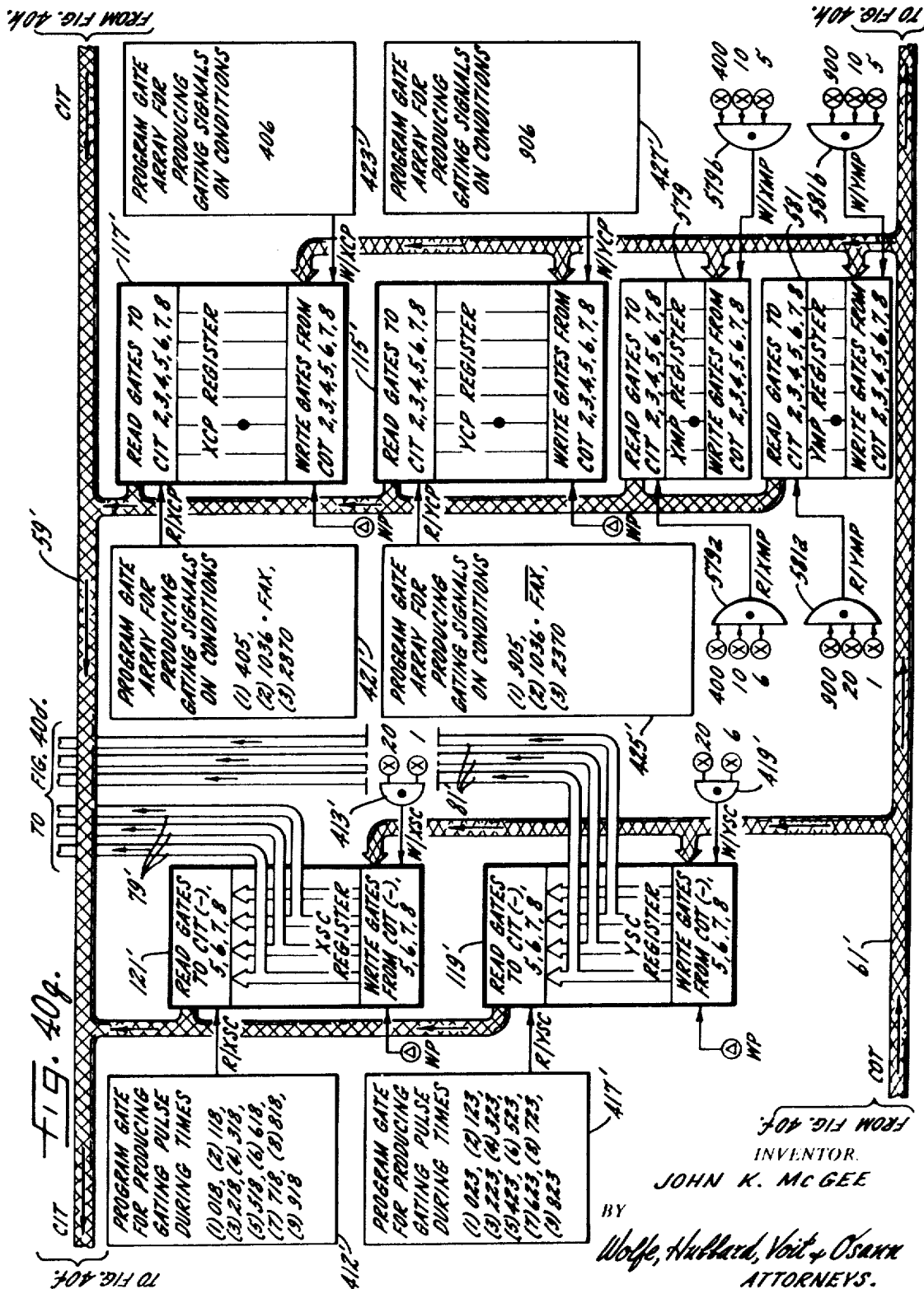
BY

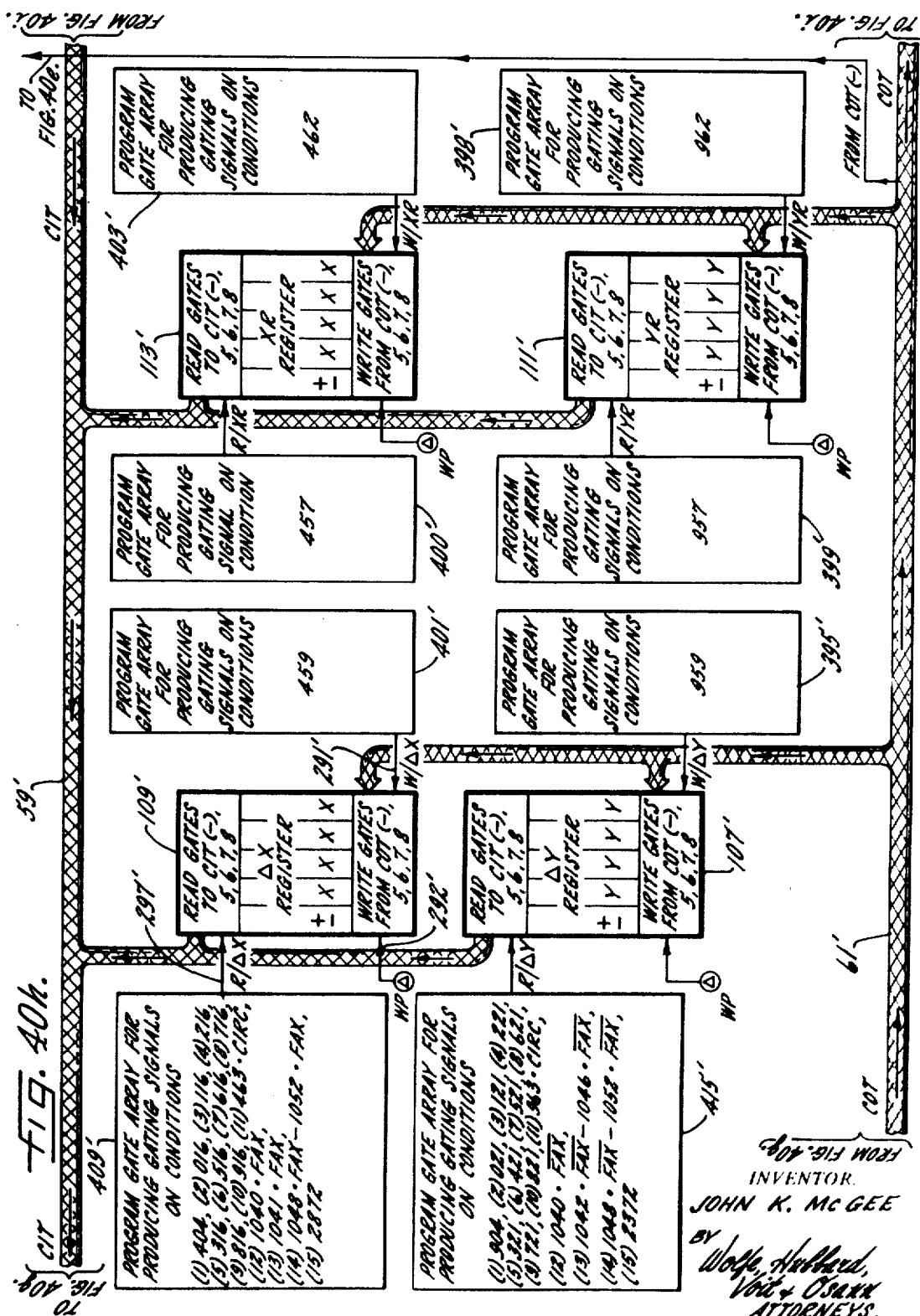
Wolfe, Hubbard, Voit & Osann
ATTORNEYS.

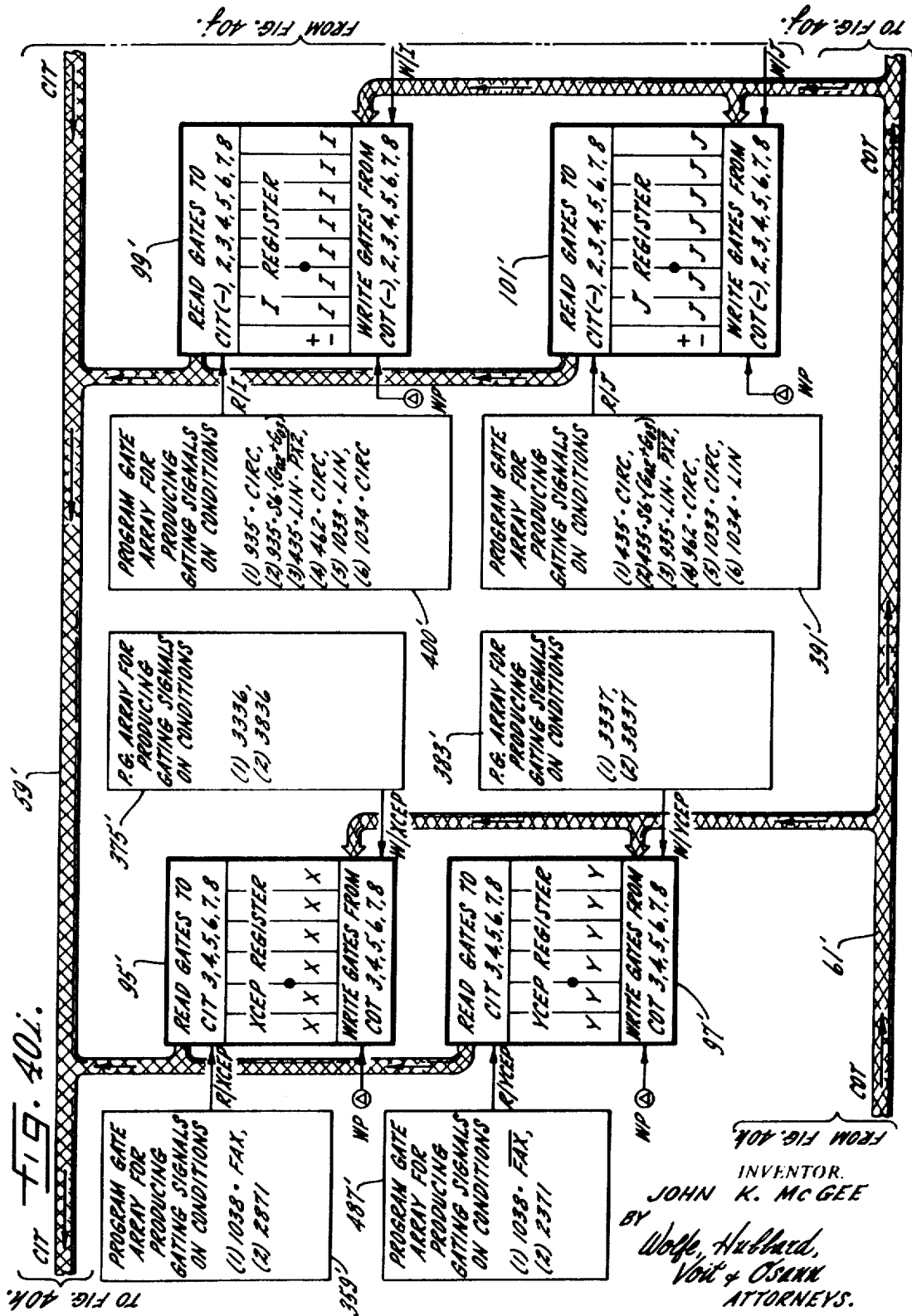
FIG. 40e.

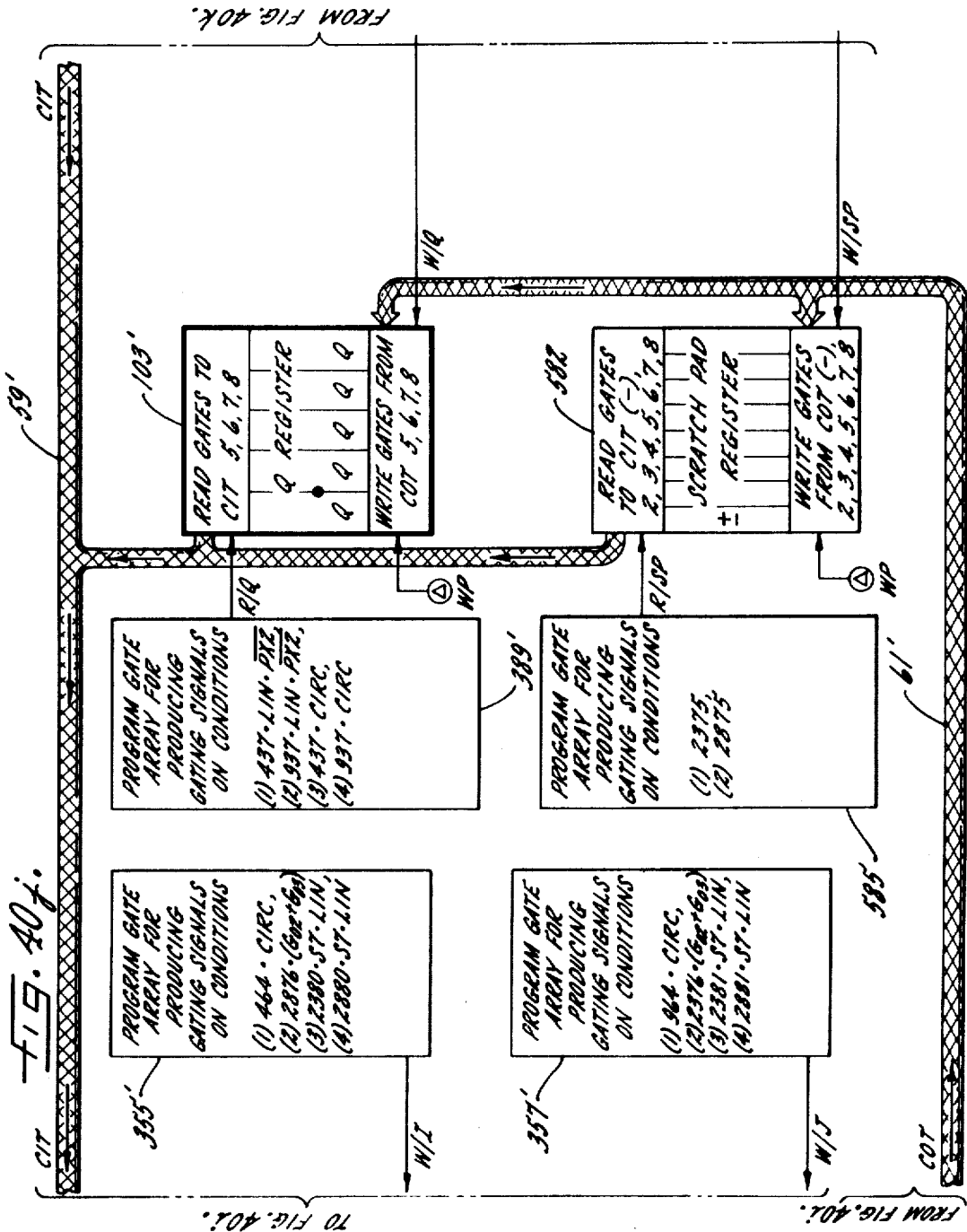


FROM FIG. 40k.







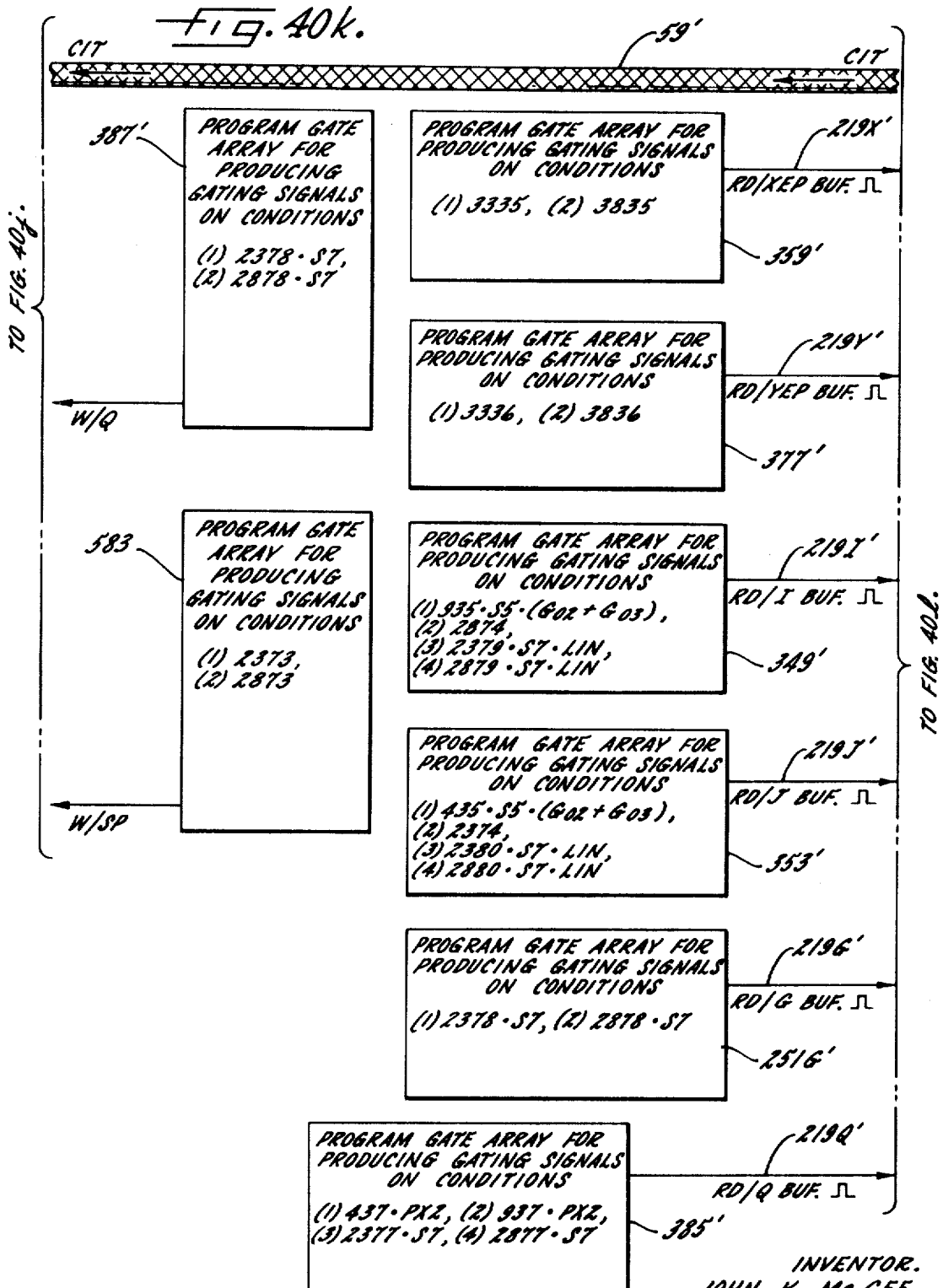


INVENTOR.

JOHN K. McGEE

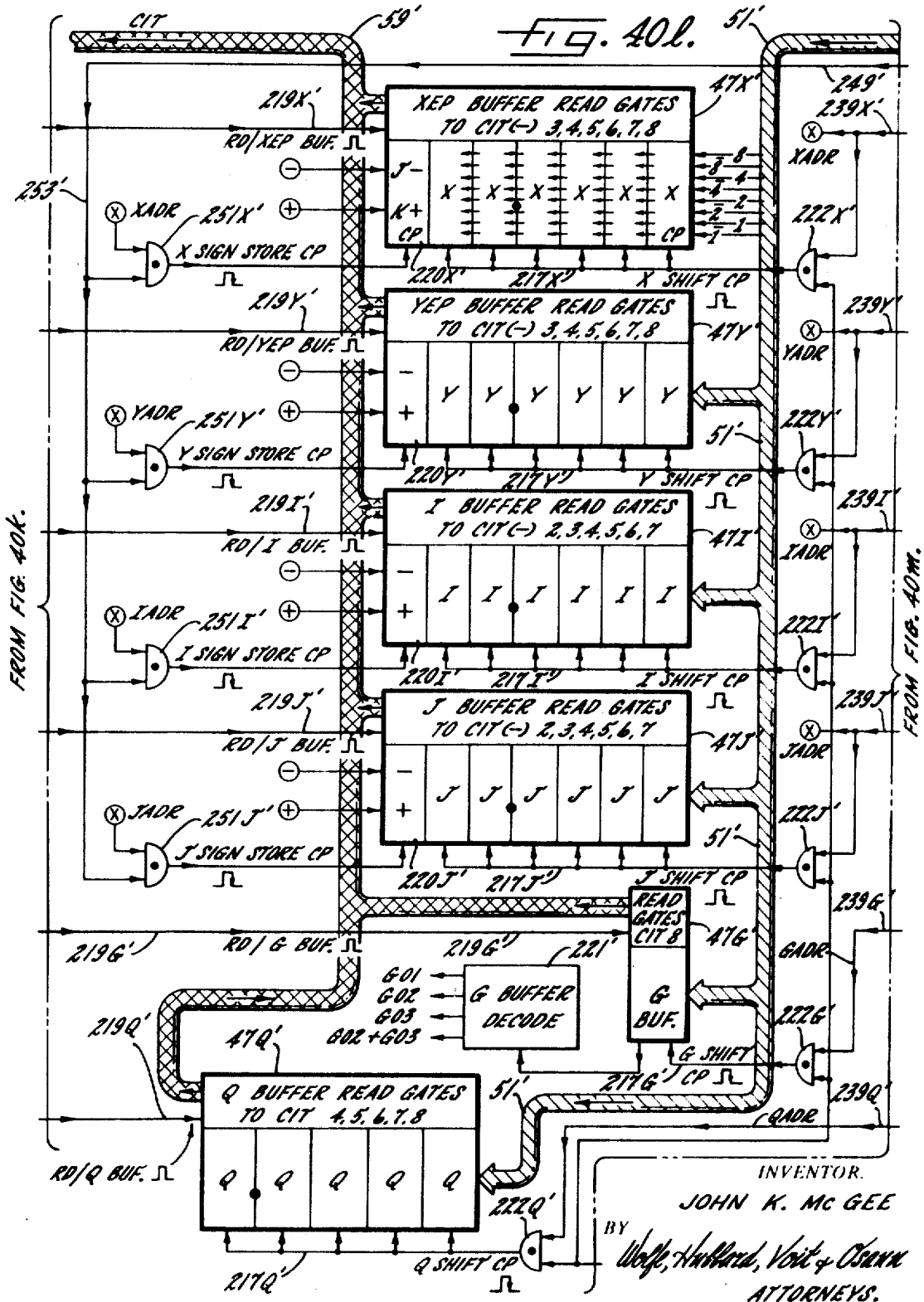
BY

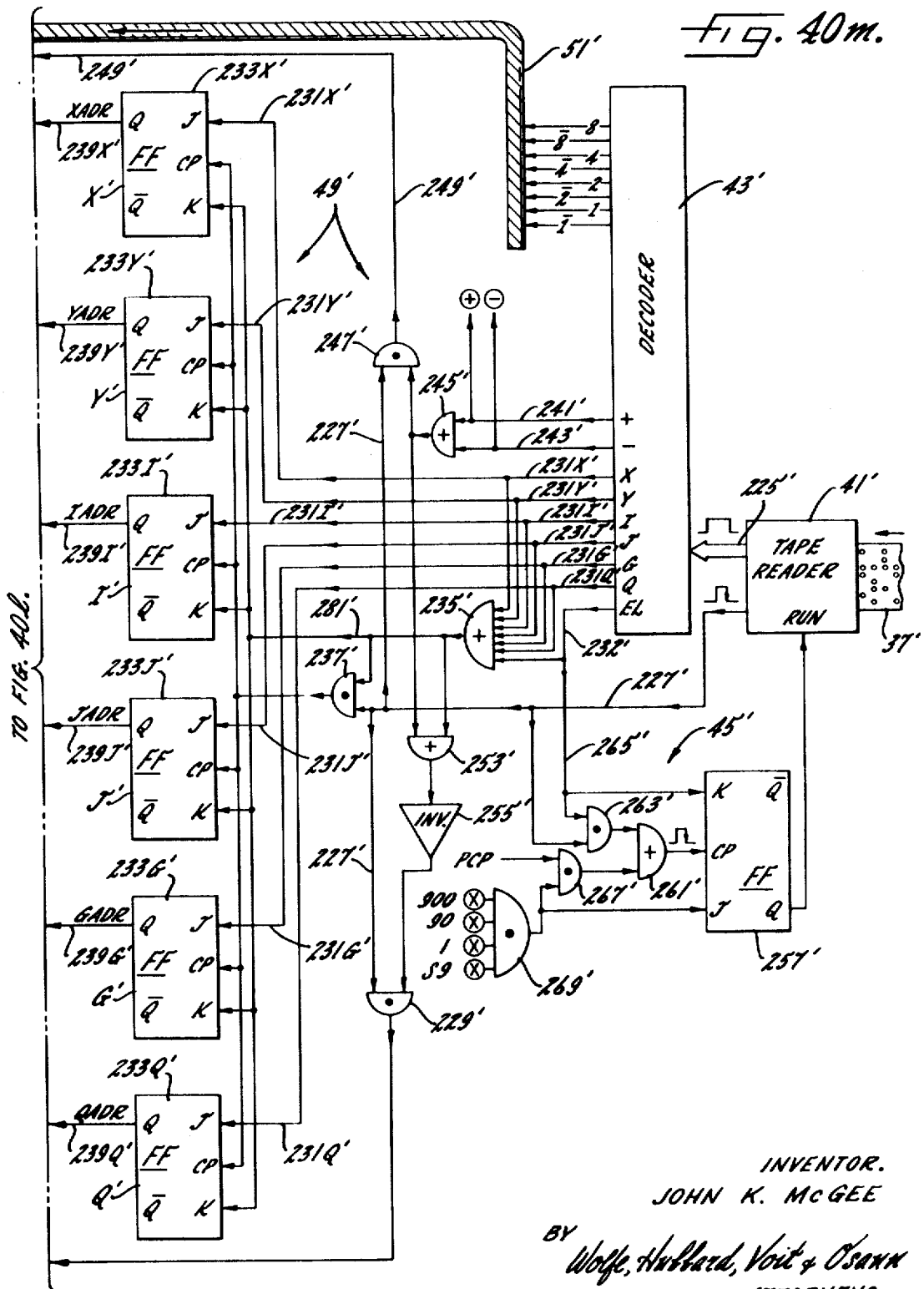
Wolfe, Hubbard, Voit & Osann
ATTORNEYS.



INVENTOR.
JOHN K. Mc GEE

BY
Wolfe, Hubbard, Voit & Osann
ATTORNEYS.





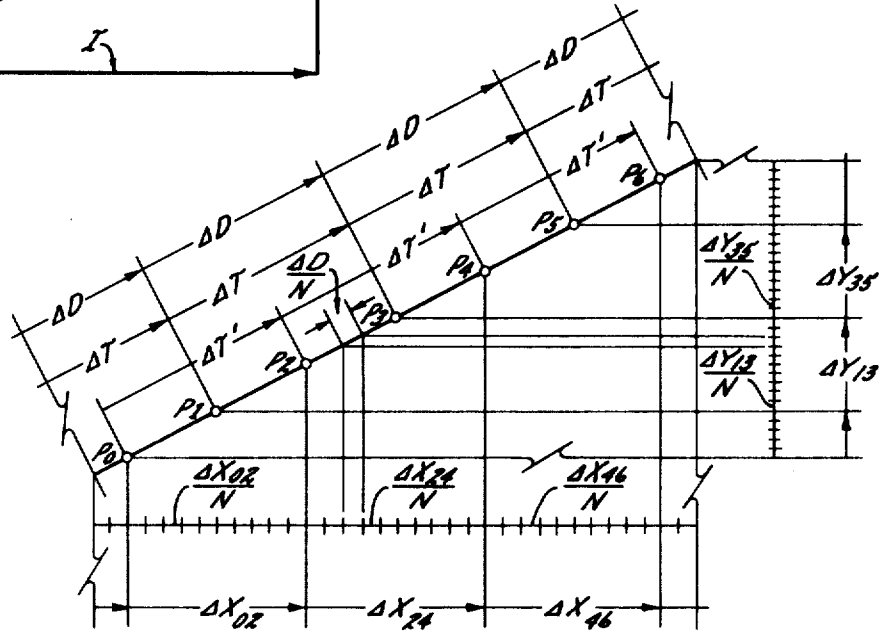
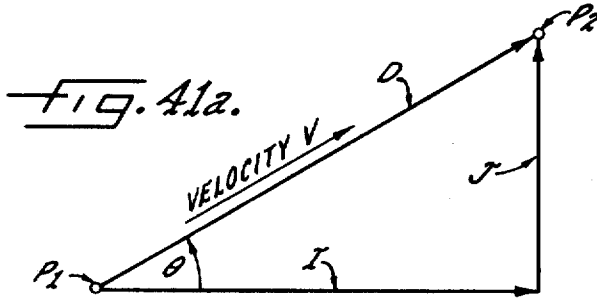


FIG. 41b.

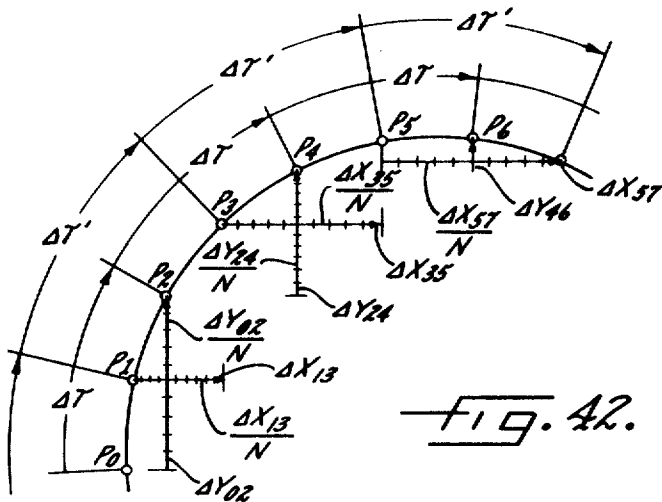


FIG. 42.

INVENTOR.
JOHN K. MCGEE
BY
Wolfe, Hubbard, Voit & Osann
ATTORNEYS.

Fig. 43.

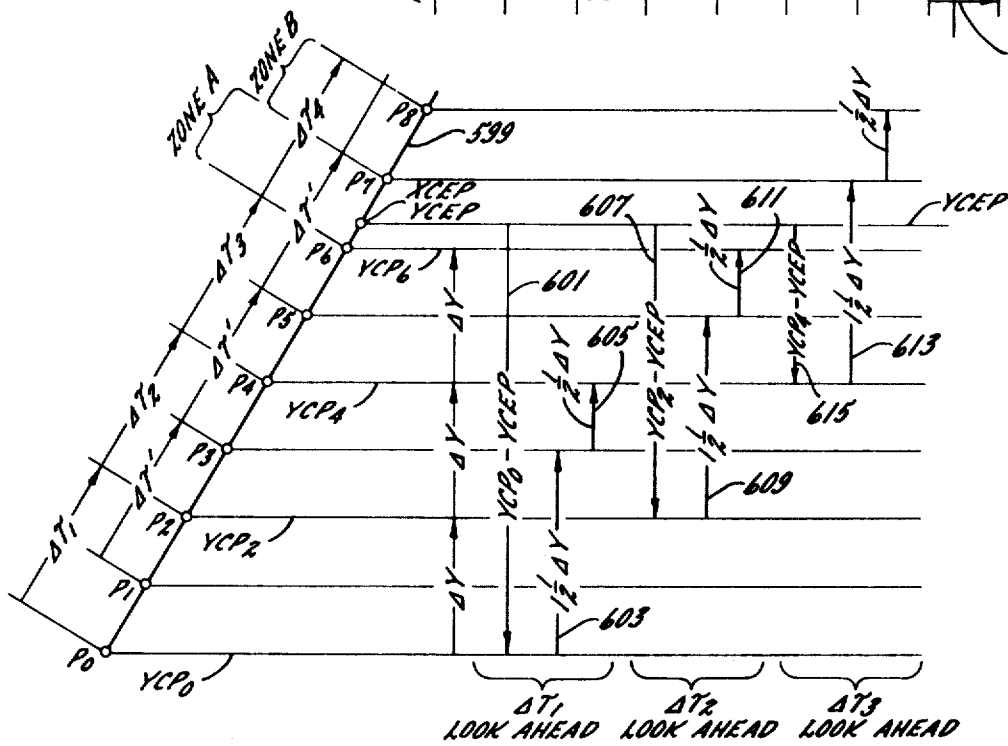
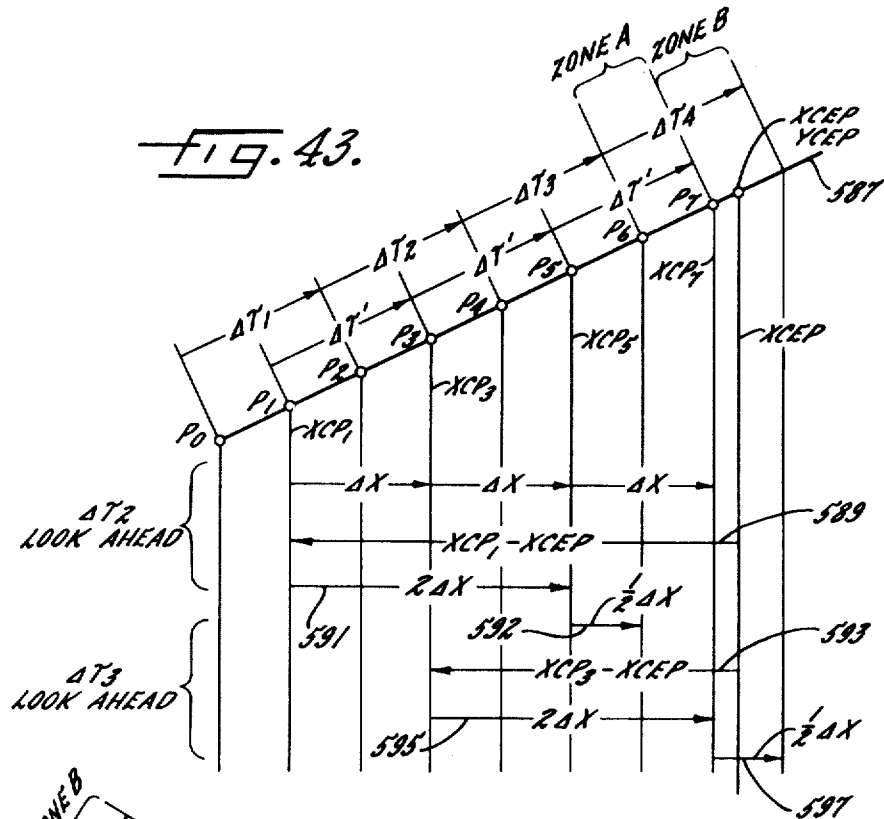
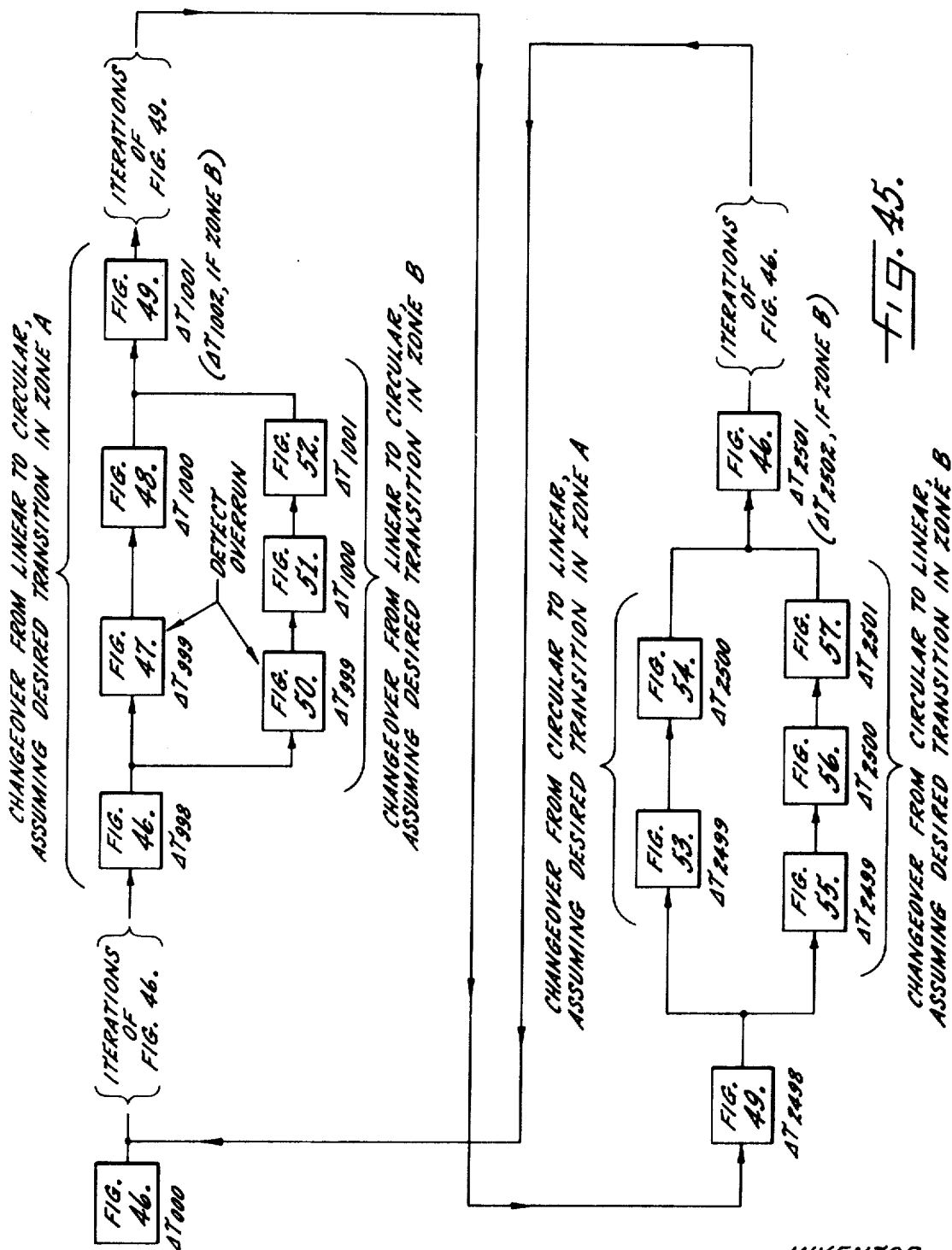


Fig. 44.

INVENTOR.
JOHN K. MCGEE
BY
Wolfe, Hubbard, Voit & Osann
ATTORNEYS.



INVENTOR.
JOHN K. MCGEE
BY
Wolfe, Hubbard, Voigt & Osann
ATTORNEYS.

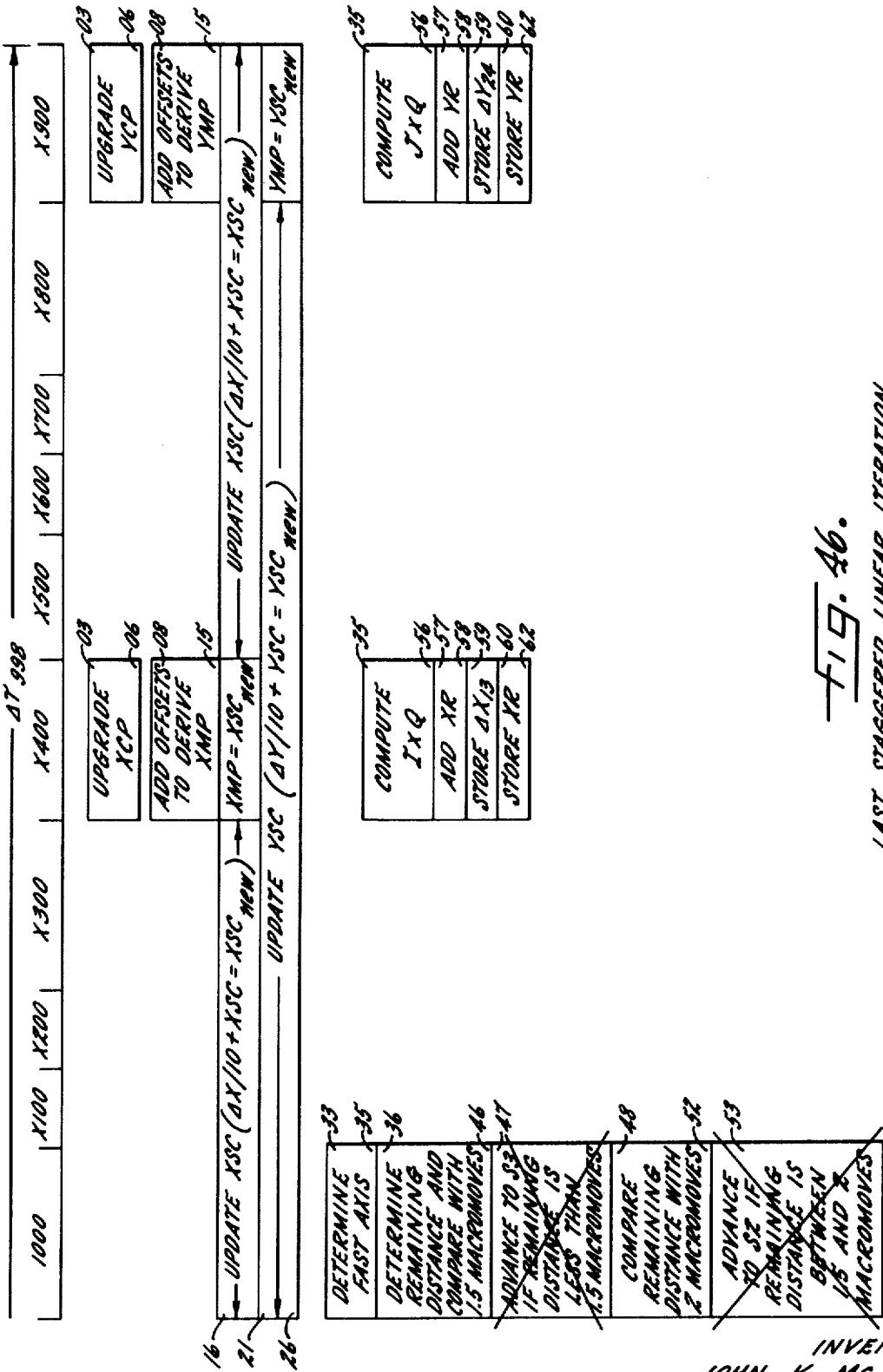


FIG. 46.

LAST STAGGERED LINEAR ITERATION
IN WHICH NO OVERRUN HAS BEEN DETECTED

INVENTOR.
JOHN K. MCGEE
BY
Wolfe, Hubbard, Voit & Osann
ATTORNEYS.

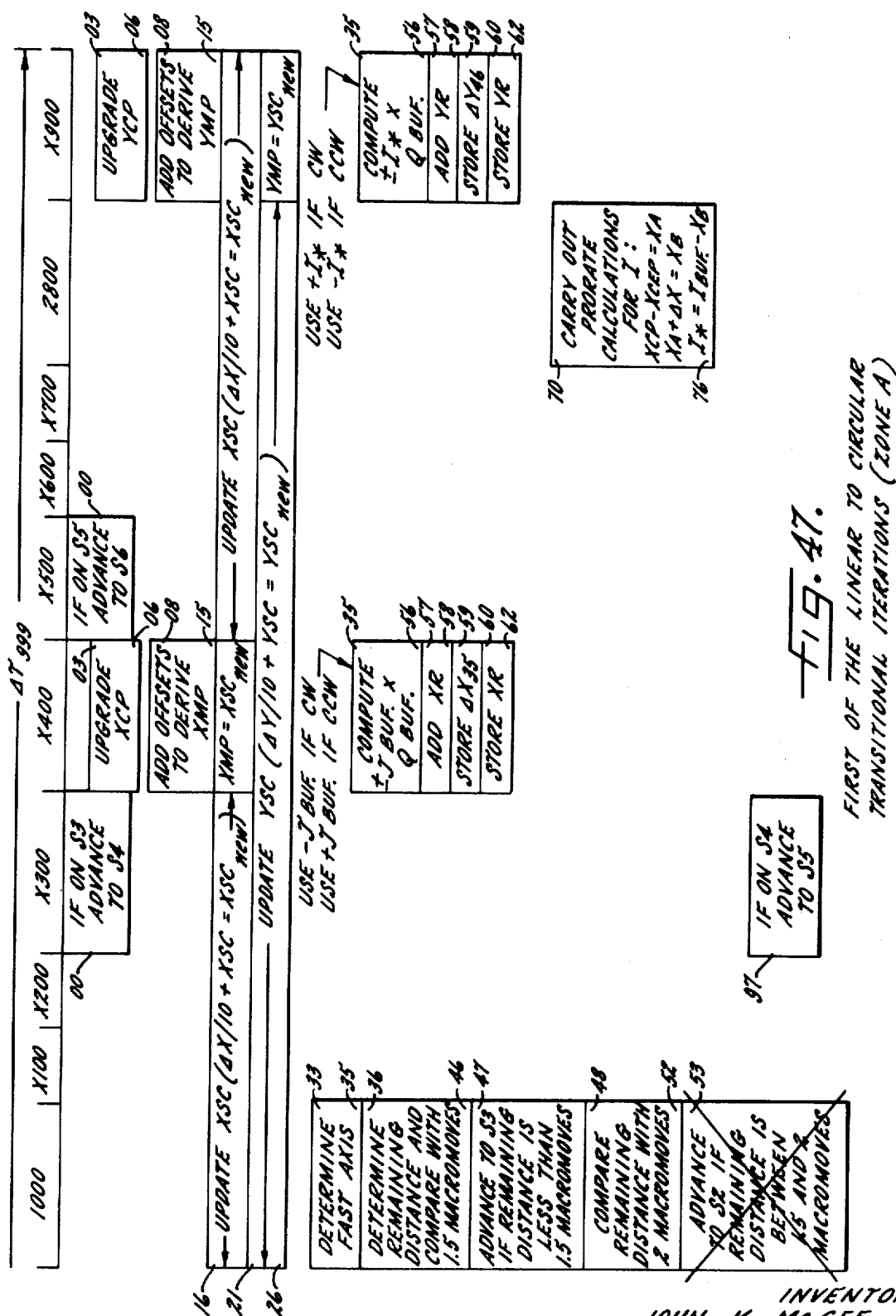


Fig. 47.

FIRST OF THE LINEAR TO CIRCULAR
TRANSITIONAL ITERATIONS (ZONE A)

INVENTOR.
JOHN K. MCGEE
BY
Wolfe, Hubbard, Voigt & Osann
ATTORNEYS.

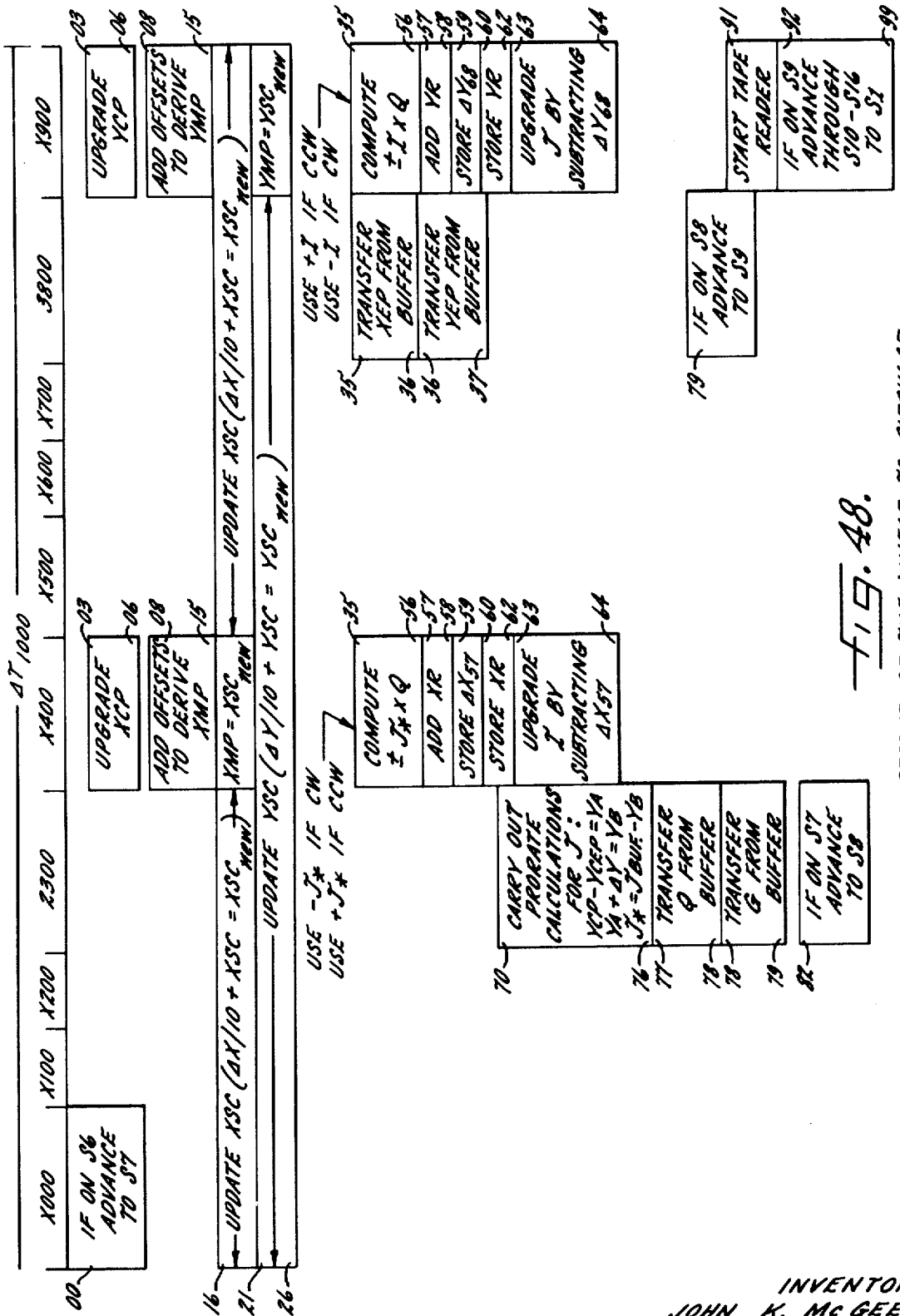


FIG. 48.

SECOND OF THE LINEAR TO CIRCULAR
TRANSITIONAL ITERATIONS (ZONE A)

INVENTOR.
JOHN K. MCGEE
BY
Wolfe, Hubbard, Voigt & Osann
ATTORNEYS.

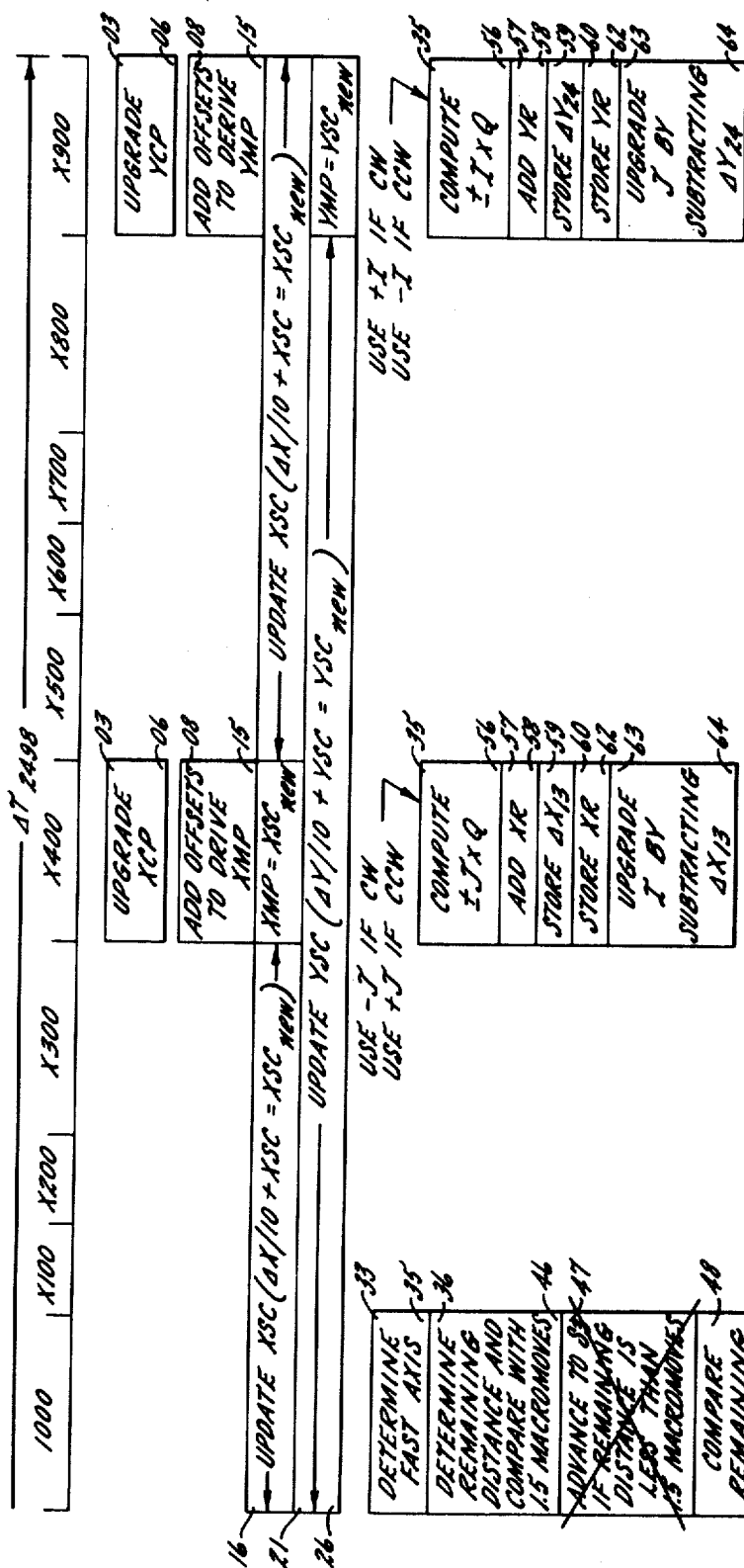


FIG. 49.

LAST STAGGERED CIRCULAR ITERATION
IN WHICH NO OVERRUN HAS BEEN DETECTED

INVENTOR.
JOHN K. MCGEE
BY
Wolfe, Hubbard, Voit & Osann
ATTORNEYS.

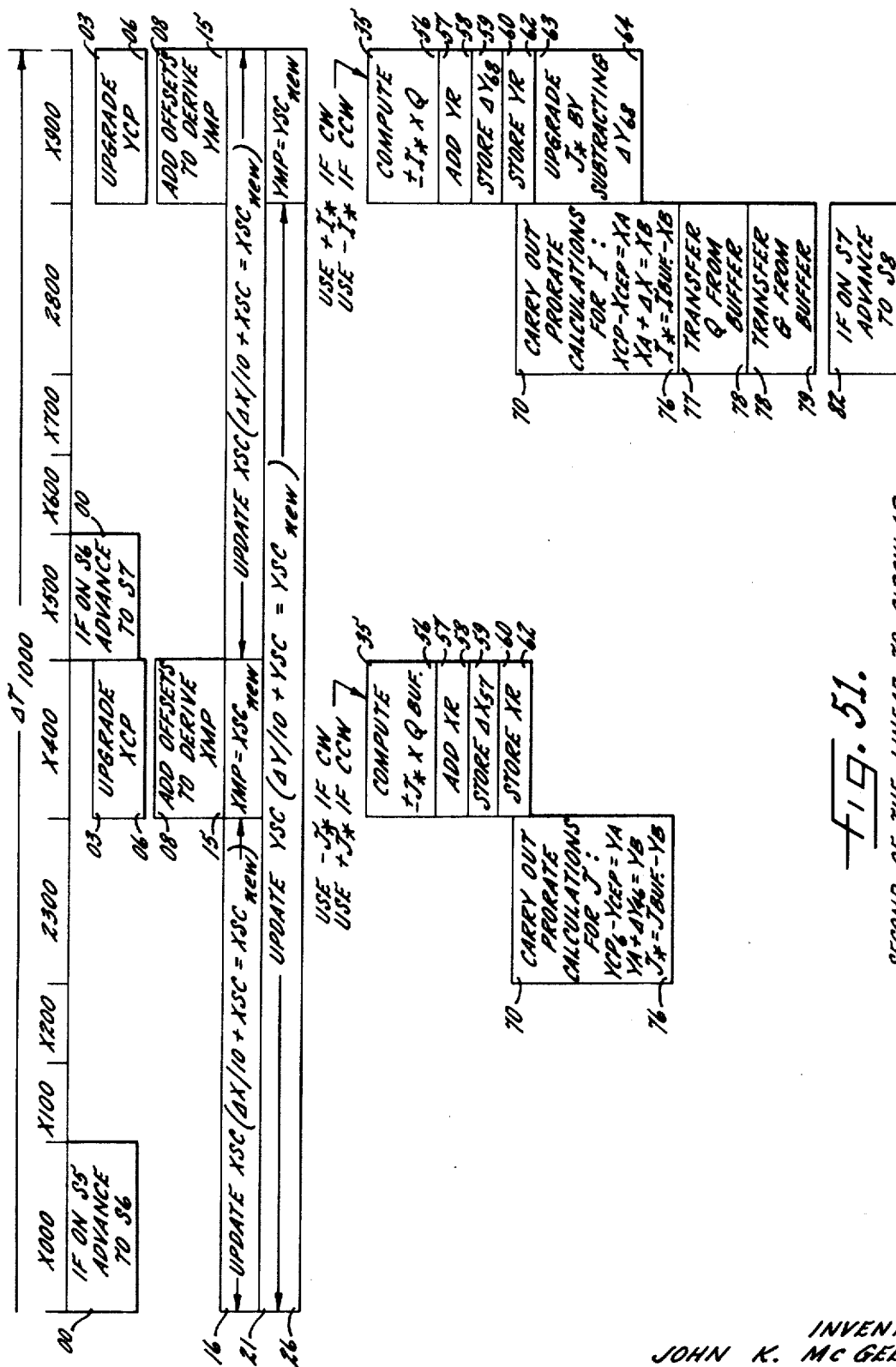


FIG. 51.

SECOND OF THE LINEAR TO CIRCULAR
TRANSITIONAL ITERATIONS (ZONE B)

INVENTOR.
JOHN K. MCGEE
BY
Wolfe, Hubbard, Vort & Osann
ATTORNEYS.

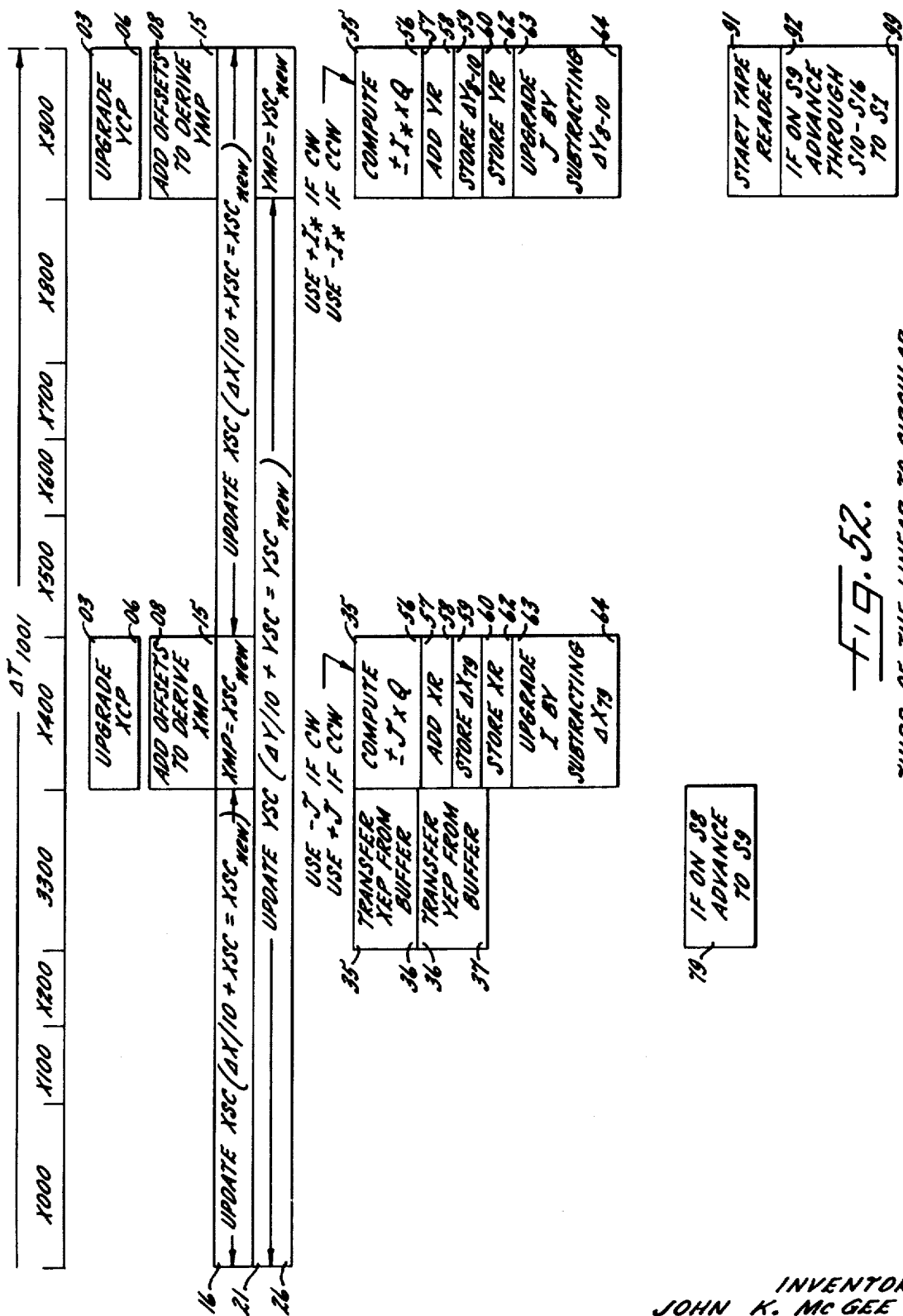


FIG. 52.

THIRD OF THE LINEAR TO CIRCULAR
TRANSITIONAL ITERATIONS (ZONE B)

INVENTOR.
JOHN K. MCGEE
BY
Wolfe, Hallard, Voigt & Osann
ATTORNEYS.

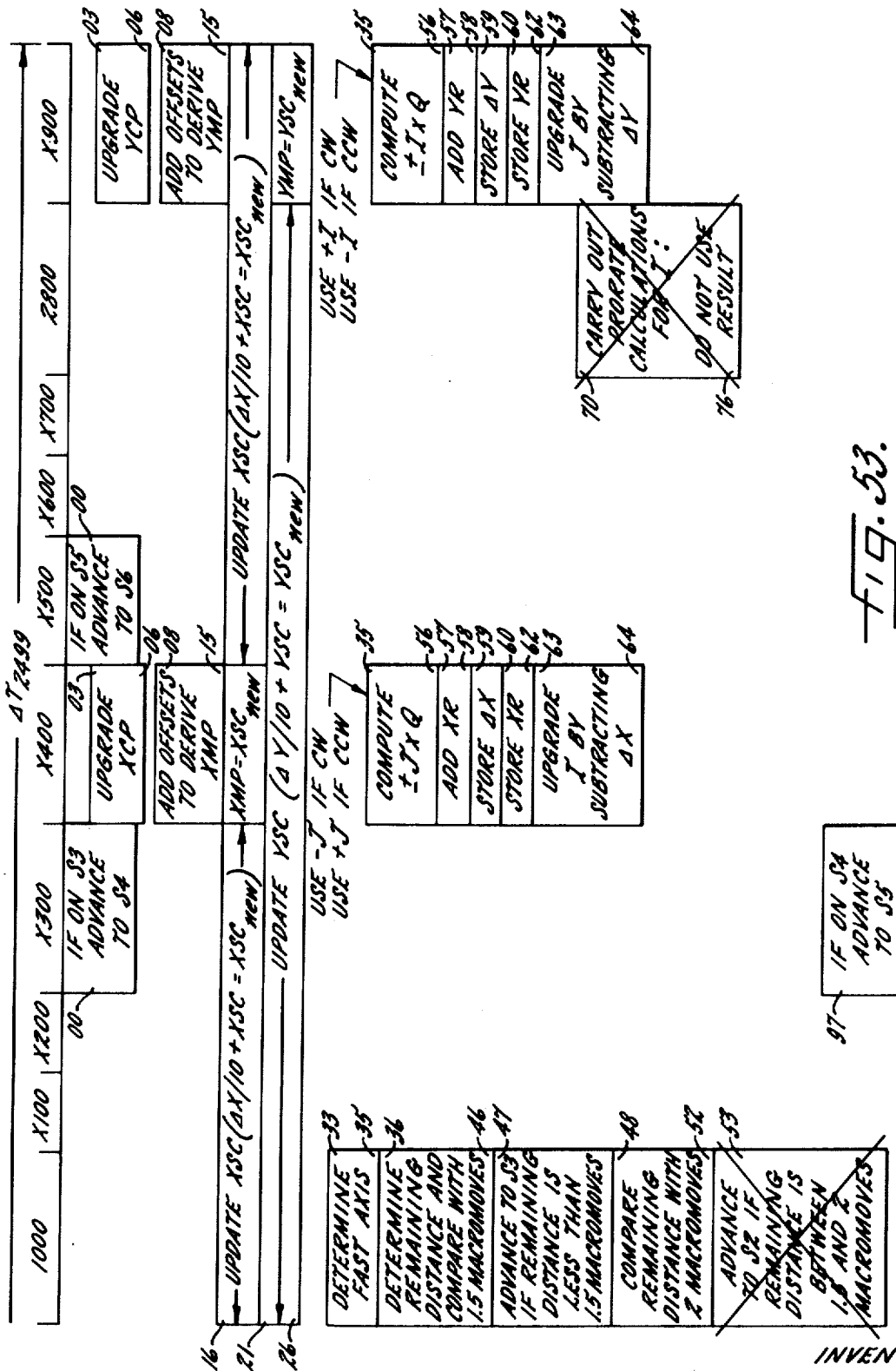
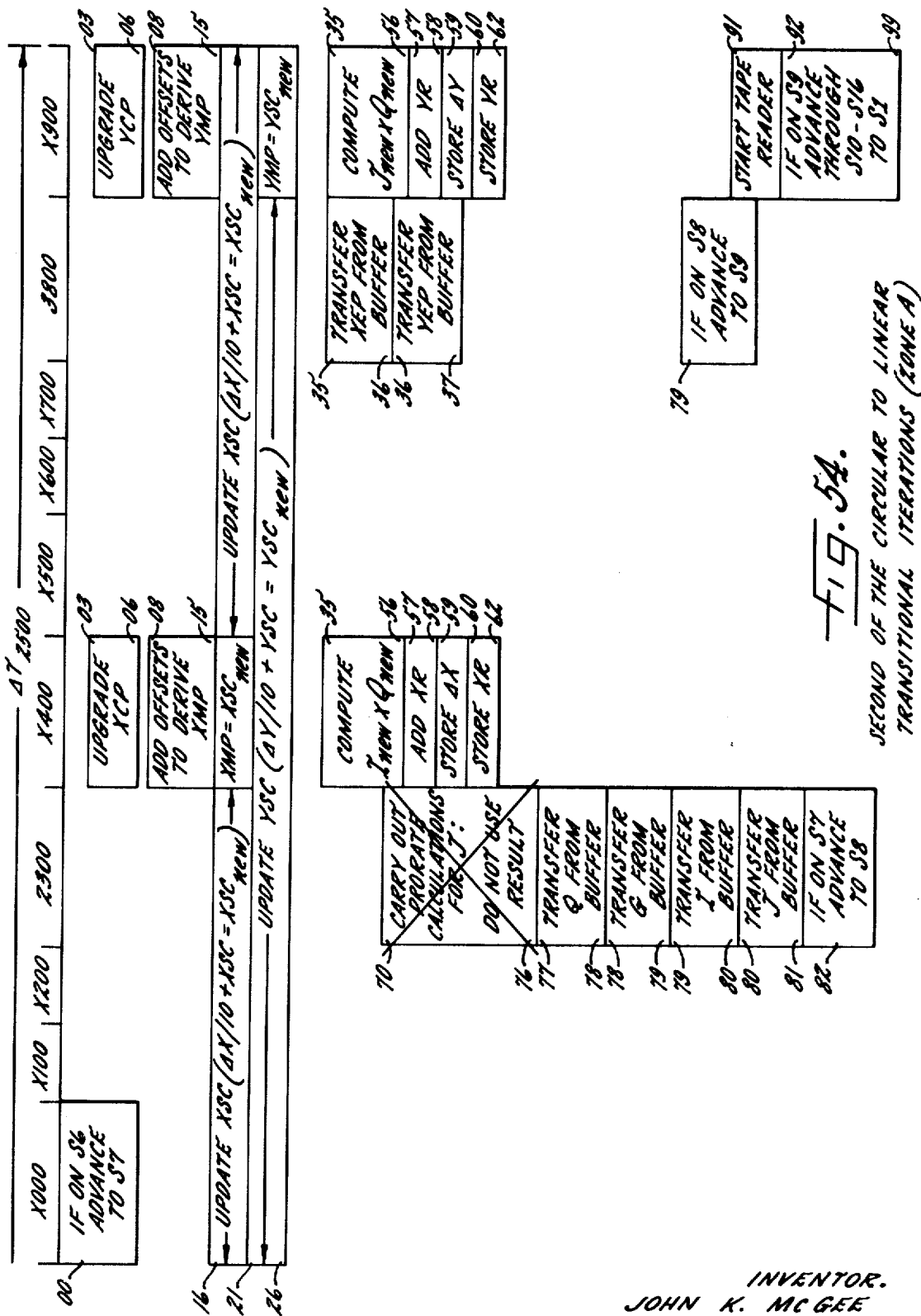


Fig. 53.

FIRST OF THE CIRCULAR TO LINEAR
TRANSITIONAL ITERATIONS (ZONE A)

INVENTOR.
JOHN K. MCGEE
BY *Wolfe, Hubbard, Vort & Osann*
ATTORNEYS.



INVENTOR.
JOHN K. MCGEE
BY
Wolfe, Hubbard, Voigt & Osann
ATTORNEYS.

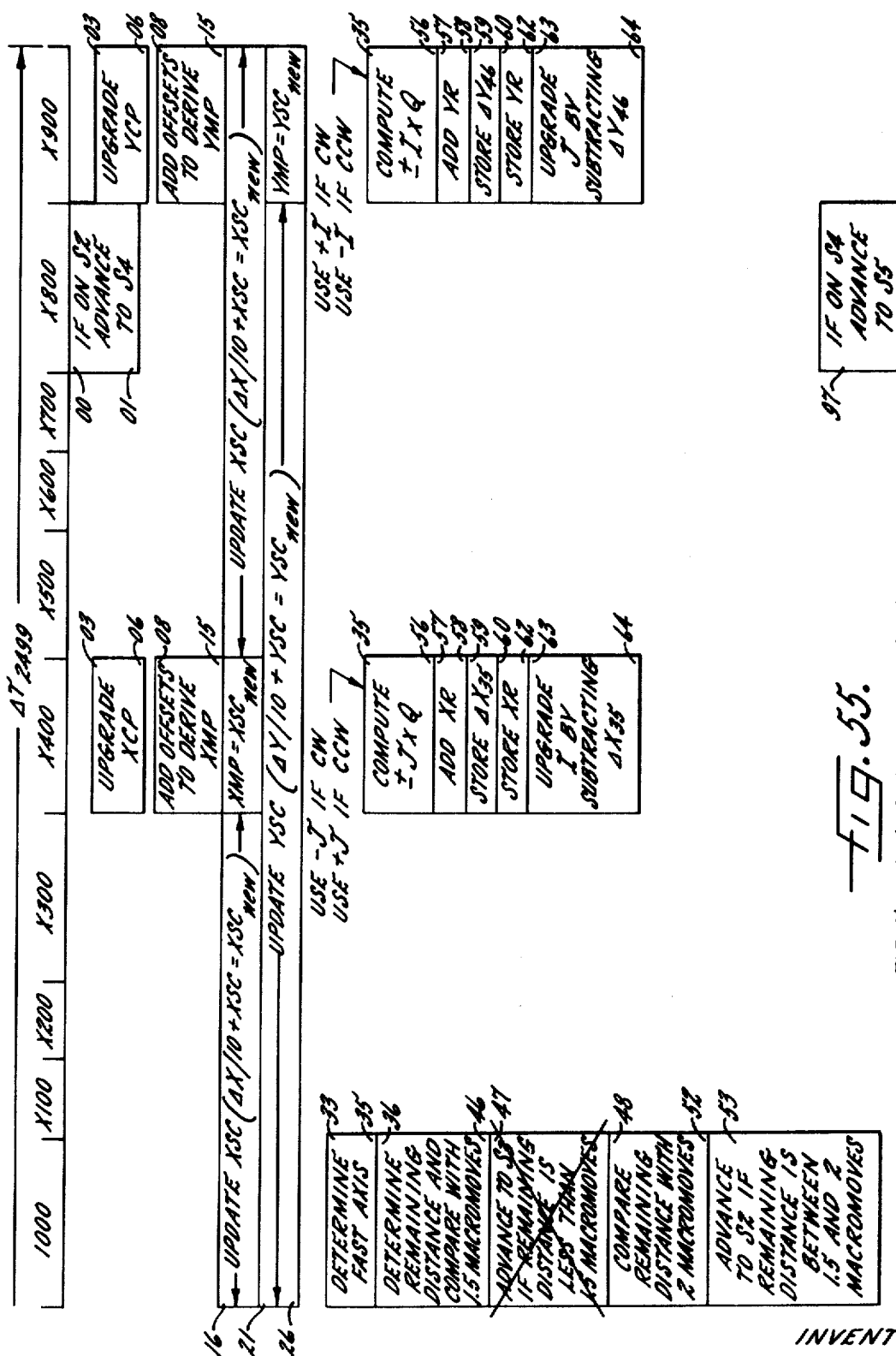


Fig. 55.

FIRST OF THE CIRCULAR TO LINEAR
TRANSITIONAL ITERATIONS (ZONE B)

INVENTOR.
JOHN K. MCGEE
BY *Wolfe, Hubbard, Vortz & Osann*
ATTORNEYS.

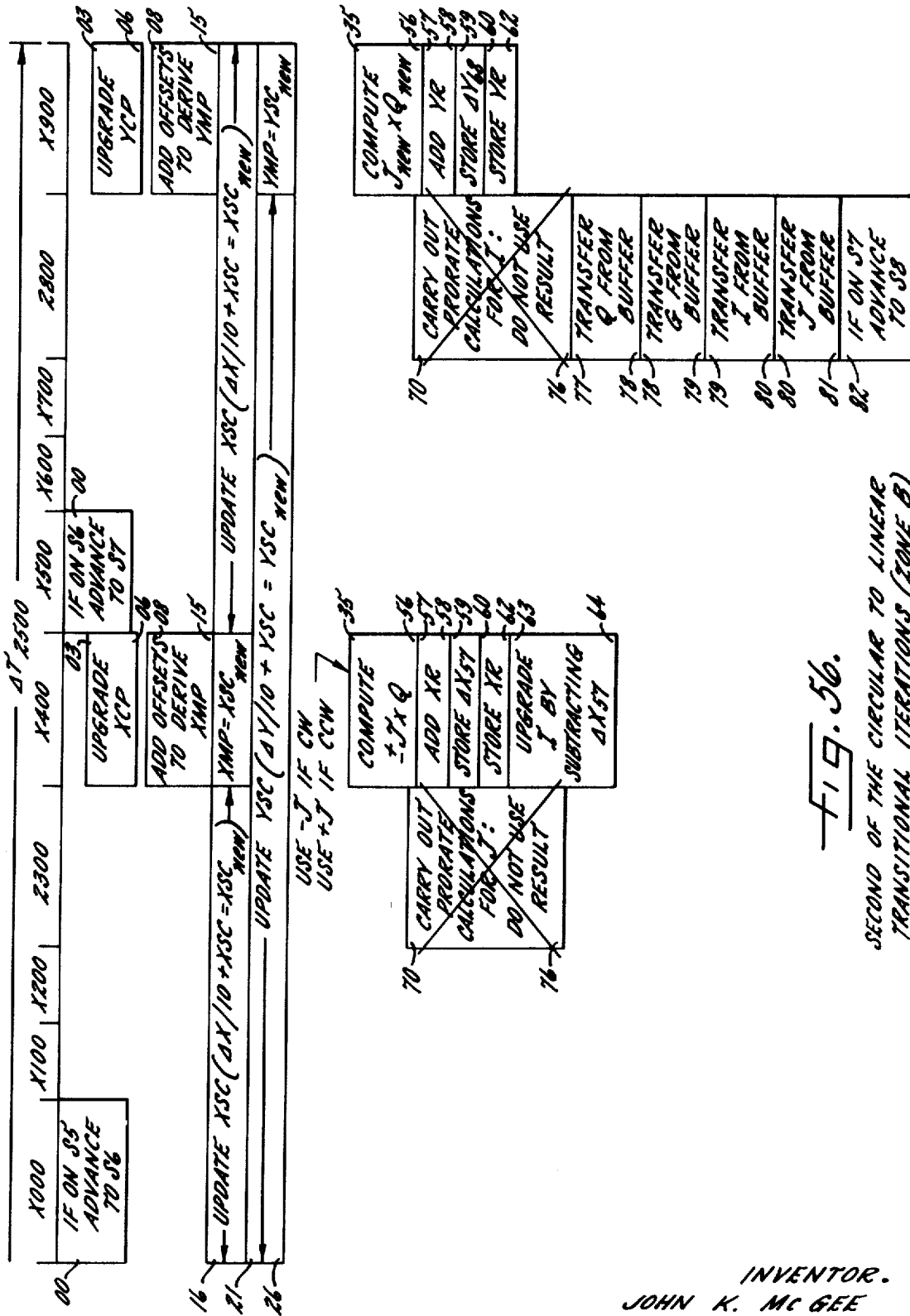
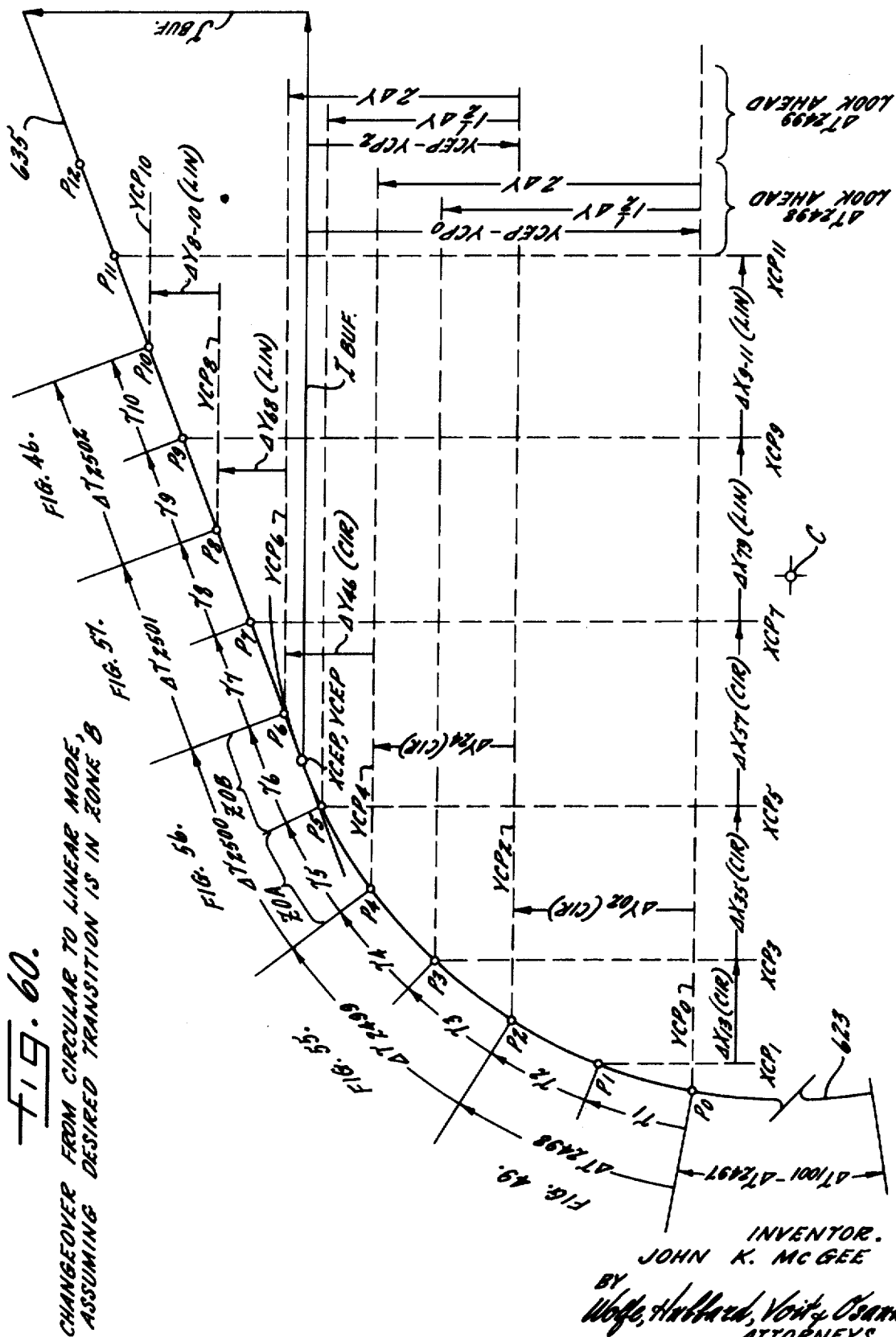


FIG. 56.

SECOND OF THE CIRCULAR TO LINEAR TRANSITIONAL ITERATIONS (ZONE B)

INVENTOR.
JOHN K. MC GEE
BY Wolfe, Hubbard, Vait & Osann
ATTORNEYS.



INVENTOR.
JOHN K. MCGEE

BY
Wolfe, Hubbard, Voth & Osawa
ATTORNEYS.

LINEAR AND CIRCULAR INTERPOLATION CONTOURING CONTROL USING REPEATED COMPUTATION

BACKGROUND OF THE INVENTION

The present invention relates in general to systems wherein information read from a punched tape or the like is supplied to data processing apparatus which in turn controls the operation of a utilization device. Such systems are exemplified by the well-known "numerical control" systems for machine tools or the like in which a numerically defined program of successive motions to be executed along a continuous path is represented on successive blocks of, and read from, a punched tape into a "director" which governs the servo drives of the movable elements of the machine tool.

In past numerical path controls for machine tools it has been customary to produce a separate stream of relatively evenly time spaced pulses for each separate axis of the machine tool and to assign a physical significance to each pulse so produced. For example, a common convention has been to regard each pulse as corresponding to an increment of movement such as 0.0001".

Because of the extremely wide speed range required for each axis, the pulse repetition rate may vary from 0 to a maximum through a very large number of different pulse repetition frequencies. For a machine having a top speed of 200" per minute, the maximum pulse repetition rate at 0.0001 inches per pulse would be 33-1/3 kHz.

A numerical path control built in accordance with past practice would, therefore, typically require the production of a separate stream of pulses for each of the separate axes, each of which can range from 0 to 33-1/3 kHz. Equipment limitations have in many cases dictated the reduction of the maximum pulse repetition frequency in path controls to a lower value, with the result that, to maintain the top speed of the machine, each pulse had to be assigned a larger value, such as 0.0002 or 0.0004", reducing the quality of obtainable surface finish.

Some systems have operated so as to change the pulse value depending on the speed range of the machine at the time. In this manner the quality of the surface finish produced by the cutting tool on the workpiece surface, and also the accuracy, can be retained by using a pulse value of 0.0001" per pulse for the relatively low speeds used during cutting, and to achieve high traverse rates between cutting operations by using a larger pulse value, such as 0.0002 or 0.0004" for the traversing operation. Nevertheless, due to the extremely large range of pulse frequencies that may be required of the system, to my knowledge a fixed time base for the production of such pulses has not been established.

Without a fixed time base for the production of the pulses for any of the separate axes of the machine it has been impossible to utilize a single time-shared set of equipment to serve all of the axes and the other control needs of the machine as well.

OBJECTS OF THE INVENTION

In view of the shortcomings of past numerical path controls as explained above, it is a principal object of this invention to depart from past practice wherein pulses representing uniform increments of position change are produced at relatively even intervals on an unknown and changing time base for each of the individual machine axes, and to employ instead a method wherein non-uniform increments of position change are produced at exactly recurring intervals on a known and unchanging time base for each of the individual machine axes.

A closely related object of the present invention is to discard the variable frequency multiple pulse train approach and to develop a new approach in which a group of digitally signaled command numbers are generated, where each command number of the group is the primary position command number for one axis, and where each primary position command number is stored in a primary register which is regularly updated at fixed time intervals.

It is a further object of the invention to generate the primary position command numbers at a relatively slow cyclic rate such that a single time-shared set of components can easily keep many machine axes supplied with updated primary command numbers.

As a corollary to the above, it is a subsidiary object of the invention to generate at a higher cyclic rate a secondary set of position command numbers, which are stored in secondary registers, which are updated at the higher cyclic rate, and which effectively achieve a multistep linear interpolation between successive primary position command numbers.

Yet another object which is related to that of producing regularly recurring command numbers is to develop the primary position command numbers at a relatively infrequent but regular rate such that the primary position command computations can be synchronized with the regularly recurring time base, hereinafter called the program iteration period, and furthermore, to develop an easily performed method for generating the secondary position command numbers from the primary position command numbers.

It is a still further object of the invention to develop the secondary position command numbers at a relatively more frequent but regular rate such that the secondary position command computations can be synchronized with the regularly recurring time base, and to update the secondary position command numbers at such a fast rate that the machine servos tend to move smoothly from one primary command position to the next, as urged by the smaller but more frequent changes of the secondary command numbers.

Within the framework of the foregoing objects, it is a more specific object of the invention to develop a method and apparatus for performing linear path generation, or interpolation, for straight line cuts, that can be extended beyond two axes.

Another object of this invention is to provide a method and apparatus for causing the programmed rate of motion during linear interpolation to be slightly altered during a selected part of the movement so that all command position numbers, both primary and secondary will converge to the programmed end point at the end of a program iteration period, thereby allowing the next program iteration to drive the machine servos into the next motion block.

Another more specific object of this invention is to develop an accurate method and apparatus for circular path generation, or interpolation, for circular cuts that can be applied to any two axes by appropriate switching devices.

In relation to both linear and circular interpolation it is another object of this invention to provide a method and means for sensing that the command numbers are approaching a selected end point. Closely related to the foregoing object and particularly related to the generation of a linear path is the object of providing a method and means for determining when and how to alter the final approach speed toward the end point so that the speed change will be held within an arbitrary small amount, and to provide a simple method for altering the final approach speed so as to hit the end point with a minimum speed change.

A further and subsidiary object of this invention is to reduce the amount of equipment required for linear and circular interpolation without affecting the degree of precision obtained by utilizing only a limited number of the most significant digits of a computed primary position command number, but carrying the remaining digits of the number to form a part of the computation of the next primary position command number.

Another principal object of the invention is to enable a single time-shared set of components to accept information from the punched tape at the appropriate time and to process that information as may be required prior to putting the information to use. In this connection it is a more specific object of the invention to allow the time-shared set of components to perform certain functions at certain time periods of the regularly recurring time base, or iteration, but to perform different functions during these same time periods under different conditions.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a perspective view of a milling machine shown to provide one example of a typical application of features of the invention.

FIG. 2 illustrates the path followed by a member of the exemplary milling machine under the control of command signals produced in accordance with a feature of the invention.

FIG. 3 illustrates the manner in which data representing successive segments of a desired tool path appears on punched tape.

FIG. 4 shows the arrangement of one block of data on the tape.

FIGS. 5a and 5b illustrate the vector quantities represented by numerical commands produced in the process of generating a linear path.

FIGS. 6a through 6b illustrate generally the vector quantities represented by numerical commands generated in the process of producing a circular path in accordance with features of the invention.

FIG. 7 formed of FIGS. 7a-7d when joined as indicated thereon is a general block diagram of a system embodying features of the present invention.

FIG. 8 is a guide for assembling FIG. 9 constituted of FIGS. 9a-9l.

FIG. 9 when assembled as shown in FIG. 8 is a detailed block diagram of a system organized in the manner shown in FIG. 7.

FIGS. 10a-10c illustrates the relationship between programming charts used to cause the system of FIG. 9 to operate in different modes during corresponding series of time periods.

FIG. 11 is a set of wave forms showing the time relationships between various timing pulses generated in the system of FIG. 9.

FIGS. 12, 12a, and 12b illustrate the last two stages of a shift register suitable for use as one of the buffer registers in FIG. 9l.

FIG. 13 illustrates one stage of a storage register suitable for use as one of the active registers 57 in FIG. 9.

FIG. 14 is a timing diagram for the computer of FIG. 9f.

FIG. 15 is a representation of all the storage registers shown in FIG. 9 and of the location of the decimal points of the numbers stored in those registers.

FIG. 16 illustrates an exemplary gating arrangement for deriving a gating pulse at a desired time period from program step signals produced by the time base generating components of FIG. 9.

FIG. 17 is a flow chart showing operations performed during the last modified macromove of a linear path segment, including calculations for entering another linear path segment.

FIG. 18 is a flow chart of operations performed during a macromove along a linear path segment.

FIG. 19 is a vector diagram illustrating the manner in which the commanded X and Y axis macromoves are adjusted in order to terminate a commanded linear segment exactly at the end of a macromove.

FIG. 20 is a flow chart showing the sequence in which different operating modes of the system illustrated in FIG. 9 follow one another in process of generating successive linear and circular path segments.

FIG. 21 is a flow chart of steps performed during the last macromove to be executed at a normal rate along the linear path, with a large overrun indication causing macromove calculations for ten subsequent time periods at a rate adjusted to hit the end point exactly at the end of the tenth drive period.

FIG. 22 is a flow chart of the operations performed during the first through the ninth of the ten iterations for which the macromove has been modified to hit the end point exactly.

FIG. 23 is a flow chart of operations performed during a macromove along a linear path segment in which a small projected overrun is detected.

FIG. 24 is a flow chart showing the steps performed during the last macromove to be executed at a normal rate following a time period (FIG. 23) in which a small projected overrun was detected, causing macromove calculations for ten time periods modified to hit the end point exactly.

FIGS. 25a-f are a series of vector diagrams showing the manner in which successive ones of the vector quantities necessary to generate a circular path are derived from initially given vector quantities.

FIGS. 26a-c are composite vector diagrams showing a series of X and Y macromove vectors produced in the manner shown in FIGS. 25a-f and also showing a series of points along a circle defined by the X and Y macromove vectors so produced.

FIG. 27 shows the manner in which the Y coordinate distance J1 between a point P1 on a circle and the center of the circle is derived from the Y coordinate distance J0 between a preceding point P0 and the center of the circle.

FIG. 28 is a flow chart showing operations performed during the last modified macromove of a linear path segment and set-up calculations for entering a circular path segment.

FIGS. 29a and 29b is a vector diagram showing the signs of I, J, ΔX and ΔY during clockwise and counterclockwise circular interpolation.

FIGS. 30a and 30b show the distribution of a series of micropositions along the X and Y axes produced as part of a process of generating a series of numerical commands for a circular path.

FIG. 31 is a flow chart showing operations during a typical iteration for describing a circular path segment.

FIG. 32 illustrates a method of eliminating ambiguity from a test which is performed to detect when the commanded position along a circular path is within a predetermined distance from the desired end point of that path.

FIG. 33 is a flow chart showing the data processing operations performed during an iteration in which a large projected overrun is detected, and which is to be followed by a linear path segment.

FIG. 34 is a similar to FIG. 33 but shows instead the data processing operations during an iteration in which a large projected overrun has been detected, and which is to be followed by a circular path segment.

FIG. 35 is a flow chart showing operations during an iteration in which a small projected overrun has been detected.

FIG. 36 is a flow chart showing the data processing operations performed during an iteration for a circular path segment following an iteration (FIG. 35) in which a small projected overrun was detected, and which is to be followed by a linear path segment.

FIG. 37 is a flow chart similar to FIG. 36 except that the iteration is to be followed by a circular path segment.

FIGS. 38a, b, and c illustrate the time relationships of signals in the servo system shown in FIG. 9a.

FIG. 39 is a guide for assembling FIG. 40 constituted of FIG. 40a-40m.

FIG. 40 is a second embodiment of a system for generating and executing both linear and circular path segments in the staggered mode.

FIGS. 41a and 41b illustrate the vector quantities represented by numerical commands produced by the second exemplary system in the process of generating a linear path in accordance with certain features of the invention.

FIG. 42 illustrates generally the vector quantities represented by numerical commands generated by the second exemplary system in the process of producing a circular path in accordance with features of the invention.

FIG. 43 is a vector diagram illustrating the manner in which proximity to the desired end point of a path segment is detected in the second exemplary system when movement along the X axis is faster than the movement along the Y axis.

FIG. 44 is a vector diagram illustrating the manner in which proximity to the end point is detected in the second exemplary system when movement along the Y axis is the faster.

FIG. 45 is a flow chart showing the sequence in which different operating modes of the system illustrated in FIG. 40 follow one another in the process of generating successive linear and circular path segments.

FIG. 46 is a flow chart of steps performed during the last staggered linear iteration in which no overrun has been detected.

FIG. 47 is a flow chart of steps performed by the second exemplary system during the first of two linear to circular transitional iterations which are performed when the linear path segment is expected to terminate during the first half of an iteration.

FIG. 48 is a flow chart of steps performed by the second exemplary system during the transitional iteration following that illustrated in FIG. 47.

FIG. 49 is a flow chart of steps performed by the second exemplary system during the last staggered circular iteration in which no overrun has been detected.

FIG. 50 is a flow chart of steps performed by the second exemplary system during the first of three linear to circular transitional iterations performed when the linear path segment is to terminate in the second half of an iteration.

FIG. 51 is a flow chart of steps performed by the second exemplary system during the transitional iteration following that shown in FIG. 50.

FIG. 52 is a flow chart of steps performed by the second exemplary system during the transitional iteration following that shown in FIG. 51.

FIG. 53 is a flow chart of steps performed by the second exemplary system during the first of two circular to linear transitional iterations performed when the circular path segment is to terminate in the first half of an iteration.

FIG. 54 is a flow chart of steps performed by the second exemplary system during the transitional iteration following that shown in FIG. 53.

FIG. 55 is a flow chart of steps performed by the second exemplary system during the first of three transitional iterations performed when the circular path segment is to terminate during the second half of an iteration.

FIG. 56 is a flow chart of steps performed by the second exemplary system during the transitional iteration following that shown in FIG. 55.

FIG. 57 is a flow chart of steps performed by the second exemplary system during the transitional iteration following that shown in FIG. 56.

FIG. 58 is a vector diagram illustrating the operations performed by the second exemplary system during the iterations shown in FIGS. 46, 47, 48 and 49 in executing a changeover from a linear to a circular path segment where the linear path segment is to end in the first half of an iteration.

FIG. 59 is a vector diagram illustrating the operations performed by the second exemplary system during the iterations shown in FIGS. 46, 50, 51, 52 and 49 in executing a changeover from a linear to a circular path segment where the linear path segment is to end in the second half of an iteration.

FIG. 60 is a vector diagram illustrating the operations performed by the second exemplary system during the iterations shown in FIGS. 49, 55, 56 and 57 in executing a changeover from a circular to a linear path segment where the linear path segment is to end in the second half of an iteration.

While the invention has been shown and will be described in some detail with reference to two exemplary embodiments thereof, there is no intention that it must be limited to such detail. To the contrary, it is intended to cover here all modifications, alternatives, and equivalents falling within the spirit and scope of the invention as defined by the appended claims.

E. AN EXEMPLARY MACHINE TOOL

In order to illustrate a particular environment in which the present invention will find especially advantageous use, it will be described here with reference to controlling paths described by an element of a numerically controlled milling

machine along two mutually orthogonal axes. A milling machine 10 (shown in FIG. 1) is typical of the many different types of machine tools which may be numerically controlled and whose numerical control requires digital command signals for operation. The exemplary milling machine includes a work table 11 movable horizontally along a bed 12 and adapted to carry a workpiece 13 which is to be machined. A milling tool 14 is mounted on, and vertically movable with, a ram 15 which is slidable within vertical ways (not shown) formed on a saddle 16, the latter in turn being movable horizontally along the ways of a rail 17 which is supported at its opposite ends by spaced columns 18 and 19. The milling tool 14 may thus be moved along horizontal and vertical X and Y axes relative to the workpiece 13 which in turn may be moved along a Z horizontal axis at right angles to the X horizontal axis.

By proper proportioning of the X, Y and Z axis components of commanded movement and velocity, the milling tool 14 moves through the workpiece 13 along a path and at a depth to cut a desired contour thereon. To produce such controlled movement of the workpiece 13 and the milling tool 14, the saddle 16 includes a nut (not shown) engaged with a lead screw 20 driven through suitable gears (not shown) by a reversible servomotor 21. As the latter motor is caused to rotate in one direction or the other at different speeds, the saddle 16 will be moved horizontally in +X or -X directions and at velocities determined by the speed of the motor. Correspondingly, the ram 15 carries a nut (not shown) engaged with a vertically disposed lead screw 24 driven by a reversible servomotor 25 so that energization of that motor in one direction or the other moves the milling tool 14 in +Y or -Y directions. Finally, the table 11 also carries a nut (not shown) which is engaged with a second horizontally disposed lead screw 22 driven by a reversible servomotor 23 whereby the table 11 and the workpiece 13 thereon may be moved in +Z or -Z direction. To simplify the description which follows, only that part of the numerical control associated with the X and Y axis drive motors 21 and 25 will be illustrated. It will be understood by those skilled in the art, however, that such a control may be modified without departing from the scope of the invention to accommodate the third or Z axis servomotor 23, and indeed to control simultaneous motions along as many axes as might be desired.

F. GENERAL DISCUSSION OF CONCEPTS AND METHODS

1. Positioning Controls v. Path Controls

The objective of a position control system is to locate a tool at a desired point in the shortest possible time. The nature of the path traveled by the tool between two successive points is of no importance. Thus a positioning control system will usually read a block of information from a record medium such as punched tape, the block including, as a minimum, the X and Y coordinates of the next desired point. In response to the block of data, the positioning control will generate electronic command signals which define the desired position of the next point for each of the X and Y coordinates. The command signals for the X axis are sent to the X axis servo drive and the corresponding signals for the Y axis are sent to the Y axis servo drive. Each servo drive includes its own feedback position transducer for signaling the actual position of the tool as well as a discriminating circuit for sensing the difference between the commanded position and the actual position of the tool along the respective axis. The discriminating circuit associated with each servo drive produces an error signal telling the servo drive how fast it should move and in which direction to reduce the position error for its axis to zero.

For an exemplary programmed path as illustrated in FIG. 2, if the X and Y servo drives had been directed to locate the tool at point P1 initially and were then directed to move it to point P2, they would both reach a very high speed initially due to the initial large error. Indeed, the initial speed of travel along the X and Y axes would be approximately equal, so that the

composite direction of travel imposed upon the driven tool by the servo drives would be about 45° from the horizontal. The Y axis servo drive would approach its target position before the X axis servo drive because, in the illustrated example, it has a shorter distance to travel. It would rapidly decelerate and come to a stop at point P2. Meanwhile, the X axis servo drive would continue on, without slowing down, until point P2 is approached, at which time it too would decelerate and come to a stop. Thus, it is seen that in a position control system of the type described, there is no control over the path followed by the controlled tool.

In contrast, in a contouring system the velocities of motion along the X and Y axes are controlled such that the resultant velocity vector, and the path of the tool, lie along a line 27b from points P1 to P2. The signals applied to the servo drives thus control not only axis distance but also velocities and they are reduced to zero when the tool reaches the point P2. Each axis velocity and the distance traveled along each axis in response to any given block of command data, are controlled, but in order to determine the net or resultant velocity, that is the feed rate which is of importance especially in metal cutting operations, it is necessary to change the axis velocities from block to block while keeping their ratio such that motion is at the desired angles. By making the successive straight line increments at different angles relatively short, and causing them to be executed in immediate succession, any arbitrary curve or contour may be produced with the desired degree of accuracy. For purely circular arcs, however, circular interpolation may be employed in which velocities along each axis are almost continuously readjusted to make the tool move from a first to a second point along the arc.

2. Contouring in Accordance with the Present Invention

Turning again to FIG. 2, illustrated therein is an exemplary path 27 through which a controlled member such as the tool 14 of FIG. 1 may be moved. Two sets of coordinate X and Y axes are shown, the first set of axes 29 and 31 representing a four quadrant coordinate system in which a person programming the path 27 may work. The second set of axes 33 and 35 represents the single quadrant coordinate system of the controlled machine into which the information developed by the programmer is eventually translated. For the present preliminary discussion concentrating on the manner in which the data representing the path 27 is initially presented, attention need be paid only to the first set of coordinate axes 29 and 31.

The path 27 consists of a series of segments, both straight line and circular, extending between a series of points P0, P1, P2, P3 and P4. The data representing the desired motion along the path 27 is usually presented in the form of a series of blocks of information, each block representing the desired motion along a different one of the path segments 27a - 27d. A record medium frequently used to carry these blocks of information is punched tape of the type illustrated in FIGS. 3 and 4. It will be recognized, of course, that punched tape is merely representative of the variety of record media which can be utilized to feed data to the apparatus which is to be described, but familiarity with the punched tape format and coding here illustrated will facilitate an understanding of the operation of the apparatus shown generally in FIG. 7 and in detail in FIGS. 9 and 40.

The elongated punched tape 37 fragmentarily shown in FIGS. 3 and 4 is made of paper, plastic or the like and contains eight longitudinal channels one through eight and a lengthwise column of sprocket holes to facilitate transport of the tape through a tape reader. Each transverse row on the tape may thus receive holes punched in different combinations of the eight channels to represent by any selective code notation different numbers, letters or symbols. In the present instance, the familiar E.I.A. standard coding is employed and the combination of holes in each row of the tape shown in FIG. 4 represents the letter number or symbol labeled at the left side. It may be observed that the ten decimal digits 0 through 9 utilize weighted combinations of holes applied in channels 1, 2, 3

and 4, these being assigned the respective weights 1, 2, 4 and 8 so that each decimal digit is represented in 1, 2, 4, 8, binary coded decimal notation.

The command data representing respective ones of the successive desired motions 27a, 27b, 27c, 27d appears on the punched tape 37 in successive "blocks," with each block being constituted by a plurality of successive transverse rows. One of these blocks of data, representing the desired motion along the path segment 27b, is shown in FIG. 4. It is seen to consist of a first group of eight rows representing the sign and magnitude of the X coordinate of the point P2. Of the eight rows of holes, the first identifies the quantity represented by the following seven rows of holes and is later converted into an "address" signal to help guide the information to its designated storage. The second row represents the sign of the number, and the last six rows represent the six digits of which the number is comprised. This quantity is identified by the symbol X and in the illustrated example is equal to -03.0000''

The next series of eight rows on the tape 37 represent the Y coordinate of P2 which is shown in FIG. 2 as +20.0000''. Also included in the block are the X and Y coordinate distances I and J from the point P1 to the point P2, shown as +06.0000 and +02.5000 respectively. The following three rows of holes identify the shape of the path segment that is to be generated, i.e., whether it is to be linear or circular. The next seven rows of holes signal a precomputed quantity Q which indicates the velocity at which movement is to occur along the path segment represented by the previous five groups of rows. The block ends with a standard EL symbol which indicates the end of the block.

In accordance with the present invention, the data stored on the punched tape 37 and representing a given path segment, such as the segment 27b of the path 27, is read from the tape and is converted into digital electrical signals. These signals are then continually processed to digitally represent at successive instants in time the X and Y coordinates of a point which progresses along the path 27b at the desired velocity. In the particular embodiment of the invention to be described herein, the point thus signalled is changed 500 times every second. Stated differently, a new target point or coordinate is generated for each of the X and Y servo drives every 2 milliseconds. The length of the path segment 27b in the example illustrated in FIG. 2 is slightly over 6 inches. Let it be assumed for simplicity of illustration that the path segment 27b is exactly 6 inches long. Let it be assumed further that the desired speed along the path segment 27b is 1 inch per minute, so that the total time required for the controlled machine tool to travel from point P1 to P2 along the path segment is 6 minutes. Significantly, during this time a total of 180,000 different digitally signalled command positions will be generated for each of the two servo drives. All of the numerous command positions generated during this 6 minute period are derived automatically from the relatively few numbers contained in the block of data stored on the punched tape 37.

3. Macromove and Micromove Calculations

Because of the large number of points signalled and because of the very short interval between the generation of successive points, it would be undesirable to compute each point through the same basic mathematical process. Instead, in accordance with an important feature of the invention, a two-step process is used. The first step involves the generation of signals for each of the X and Y axes representing successive positions or points along those axes. Successive ones of these points are defined herein as macropositions and the moves whereby the controlled tool is traversed from one of these positions to the next are defined as composite macromoves. In the exemplary embodiment, which will be described herein, such macroposition signals are generated 50 times each second or every 20 milliseconds. The computations whereby the successive macropositions are derived from the basic data appearing in a block of information on the punched tape are relatively lengthy.

As part of the second, and considerably shorter part of the process, signals representing a series of progressive micropositions which fall between each pair of successive macropositions are derived from the previously computed macroposition signals at 2 millisecond intervals. Summarizing, for each axis, a first series of target points, called macropositions, are generated through a detailed computational sequence, and a second and much larger series of intermediate points, called micropositions, which are equally approximately spaced and which lie between the macropositions, are derived from the signals representing the macropositions through a much shorter and simpler process.

A. LINEAR PATH GENERATION

FIGS. 5a and 5b show in greater detail the quantities and relationships involved in generating the path segment 27b from point P1 to point P2 of FIG. 2. FIG. 5a shows the path segment 27b as the diagonal vector D , its X component as the vector I , and its Y component as the vector J . In accordance with a basic feature of the invention, a time period ΔT is chosen which in comparison to the total time required to advance the desired position of the controlled machine element from point P1 to point P2 is very short. Consequently, a large number of such time periods ΔT will elapse while the desired position advances from point P1 to point P2. The distance by which the desired position is to be advanced during a single time period ΔT is shown in FIG. 5b as the hypotenuse ΔD of a triangle. This is the distance which was previously referred to as a composite macromove. Its X component forms the base of the triangle and is identified as the vector ΔX , and its Y component forms the vertical side of the triangle and is identified as the vector ΔY .

The size of the composite macromoves which are to be executed during the respective time periods ΔT are a function of the length of the time period ΔT and the desired velocity V as expressed by the equation

$$\Delta D = V \cdot \Delta T \quad (1)$$

From an inspection of the triangles illustrated in FIGS. 5a and 5b it is seen that they are geometrically similar, since the hypotenuse of the smaller triangle of FIG. 5b is merely a short portion of the hypotenuse D of the larger triangle of FIG. 5a. Consequently, it follows from the law of similar triangles that

$$\Delta X/I = \Delta D/D \quad (2)$$

Similarly, it follows from the law of similar triangles that

$$\Delta Y/J = \Delta D/D \quad (3)$$

$$\text{But } \Delta D = V \Delta T \quad (4)$$

$$\text{and } D = \sqrt{I^2 + J^2} \quad (5)$$

Substituting (4) and (5) into (2) and multiplying both sides of the equation by I gives

$$\Delta X = I \cdot \frac{V \Delta T}{\sqrt{I^2 + J^2}} \quad (6)$$

Substituting (4) and (5) into (3) and multiplying both sides of the equation by J gives

$$\Delta Y = J \cdot \frac{V \Delta T}{\sqrt{I^2 + J^2}} \quad (7)$$

For the exemplary embodiment to be disclosed it is assumed that, for any given path segment represented by a block of data on the tape, the quantities V , I , and J each have a single value. Furthermore, the time period ΔT is a chosen constant which remains the same for all of the segments of the path. This permits the use of a composite constant Q , hereinafter referred to as the "macromove quotient", (so named because it happens to be equal to $\Delta D/D$) in place of the term

$$\frac{V \cdot \Delta T}{\sqrt{I^2 + J^2}}$$

That is,

$$Q = \frac{V \cdot \Delta T}{\sqrt{I^2 + J^2}} \quad (8)$$

Substituting (8) in equations (6) and (7) yields the much simpler expressions

$$\Delta X = I \cdot Q \quad (9)$$

$$\Delta Y = J \cdot Q \quad (10)$$

From the foregoing series of equations it may be seen that the quantities ΔX and ΔY , respectively representing the desired movement of the commanded position during a time period ΔT along the X and Y axes, may be given in a partially precomputed form by using the term " Q " in place of the fraction

$$\frac{V \cdot \Delta T}{\sqrt{I^2 + J^2}}$$

Thus, given the quantities I , J , and Q , the desired X and Y macromove components ΔX and ΔY may be calculated relatively simply by the process of multiplication. However, even though multiplication is a relatively simple process, nevertheless it is a time-consuming one, especially where both the multiplier and the multiplicand are multi-digit numbers. It is for this reason that a series of intermediate points or micropositions lying intermediate a pair of adjacent macropositions is developed by an even more direct process. As shown in FIG. 5b a composite macromove ΔD is divided into N equal portions $\Delta D/N$. These smaller portions are referred to herein as composite micromoves and their X and Y components are shown in the smaller triangle of FIG. 5b as the quantities $\Delta X/N$ and $\Delta Y/N$. In order to take fullest advantage of the two-step process described here, " N " is made equal to ten and the quantities $\Delta X/N$ and $\Delta Y/N$ are derived by dividing respective ones of the quantities ΔX and ΔY by 10, which is very simply performed in a calculating machine or computed by shifting the signals representing those quantities by one decimal place.

After the quantities ΔX , ΔY , $\Delta X/N$ and $\Delta Y/N$ have been calculated, a series of macropositions and micropositions are developed by a process of successive additions. In particular, for the X axis the quantity ΔX is cumulatively added at intervals of ΔT to the X coordinate of the starting point P1 so that, during each time period ΔT , a new macroposition is generated for the X axis. By a similar process of successive additions a series of Y axis macropositions are developed at intervals of ΔT extending with the passage of time from near the Y coordinate of the point P1 to the Y coordinate of the point P2.

The micropositions are derived in a similar manner from the micromove components $\Delta X/N$ and $\Delta Y/N$. Thus, the first microposition for the X axis is produced by adding the quantity $\Delta X/N$ to the X coordinate of the initial point P1 and successive micropositions for the X axis are produced at intervals of $\Delta T/N$ by further additions of the quantity $\Delta X/N$ to the previously derived sum. The same procedure is followed for deriving Y micropositions from the Y coordinates of point P1 and micromove component $\Delta Y/N$.

The foregoing analysis has been principally in terms of geometrical relations. However, it should be remembered that in carrying out the invention, the successive macropositions and micropositions do not simply exist in space. Rather, they are numerically represented by digital electrical signals as are the quantities, I , J , V , ΔT , (or I , J , and Q) from which they are derived. These digital electrical signals representing successive target positions are then applied continuously to the X and Y servo drives in order to advance the position of the controlled machine tool shown herein as the milling tool 14. In this manner, the tool may be traversed along several consecutive straight line path segments, each of them being represented by a different block of data containing individual values for the quantities I , J , and Q (or V and ΔT if the quotient Q is not used). In carrying out the process, the digital signals representing the X and Y components I and J of the composite distance D are read by appropriate means from the tape on which they are initially stored, are converted into digital electrical signals and are processed by data processing apparatus to derive therefrom first and second sets of digital signals representing the ΔX and ΔY numbers, in accordance with the equations (6) and (7) given above. Having derived signals representing the ΔX and ΔY numbers, successive periods of ΔT are measured off in actual time and during each such time periods, the first and second sets of signals representing the quantities ΔX and ΔY are utilized to drive a controlled member through the distance ΔX along the X axis

and through the distance ΔY along the Y axis. In accordance with an additional, but optional, feature of the invention, signals are also produced to represent fractions $\Delta X/N$ and $\Delta Y/N$, and N times during each of the successive time periods ΔT these signals are utilized to drive the controlled member simultaneously along X and Y axes through intermediate distances $\Delta X/N$ and $\Delta Y/N$.

b. Circular Path Generation

The method of generating electrical signals representing successive target positions along a circular path is shown generally in FIG. 6a. It will be noted that the circular arc 39 there shown includes a quarter of a circle and thus does not correspond to the circular path shown in FIG. 2. Similarly, the points P0 through P5 appearing in FIG. 6a do not correspond to similarly labeled points appearing in FIG. 2. It should also be noted that the points P0 and P5 appearing in FIG. 6a are much further apart than they would normally be in order to clarify the geometrical explanation of the process.

Distributed along the arc 39 in FIG. 6a are a first series of points P0, P2 and P4, and a second series of points P1, P3 and P5 symmetrically interspersed with the points P0, P2 and P4. The first series of points P0, P2 and P4 are connected by chords of equal length ΔD as are the second series of points P1, P3 and P5. Each of the points P1 through P5 is connected to the center C of the circle, of which the arc 39 forms a part, by a radius R. The X components of the string of chords connecting the points P0, P2 and P4 are labeled ΔX , and the Y components of the second series of chords which connect the points P1, P3 and P5 are labeled ΔY . In accordance with the invention as it pertains to generating a circular path, signals representing the X coordinates of the points P0, P2 and P4 are produced at intervals of time ΔT , and signals representing the Y coordinates of the points P1, P3 and P5 are produced at time intervals $\Delta T'$ which are equal to but symmetrically staggered with respect to the time intervals ΔT associated with the points P2 and P4. Furthermore, in carrying out another feature of the invention, during each of the time periods ΔT which elapse between the generation of the points P0, P2 and P4, at intervals of $\Delta T/N$ signals are produced representing the X coordinates of equally spaced intermediate points lying along the chords of the circle connecting the points P0, P2 and P4. In a similar manner signals are also produced representing the Y coordinates of points intermediate the points P1, P3 and P5. This is best seen in FIG. 6b showing the ΔX vector components extending between P0 and P2, P2 and P4, and P4 and P6 and the vector components ΔY extending from P1 to P3 and from P3 to P5. By comparing the paths illustrated in FIGS. 5b and 6b, it will be seen that, (a) the successive vector quantities ΔX and ΔY represent successive macromoves along the X and Y axes, each of the macromoves along the respective axes being carried out over a period of time ΔT and $\Delta T'$ respectively, and (b) each of the macromoves ΔX is divided into N micromoves $\Delta X/N$ and similarly each of the macromoves ΔY is divided into N micromoves $\Delta Y/N$.

In generating the circular path a two step approach is used which is similar to the process explained with reference to linear path generation. As part of the first step, the quantities ΔX and ΔY are calculated and the X coordinates of the points P2 and P4 as well as the Y coordinates of the points P3 and P5 are derived by a processed of successive cumulative additions of the quantities ΔX to the X coordinate of a starting point and of the quantities ΔY to the Y coordinate of a starting point. As part of the second step, each of the quantities ΔX and ΔY is divided into N equal parts, preferably 10, and for each of the X and Y axes digital signals representing the X and Y coordinates of the intermediate target points, or micropositions, are generated. The signals representing the X and Y coordinates of the intermediate points are then applied to respective ones of the X and Y servo drives so that the X axis servo drive moves in close proximity with the X coordinates of a string of chords connecting points P0, P2, and P4, while the Y axis servo moves later in time by $\frac{1}{2} \Delta T$ in close proximity with the Y coordinates of string of chords connecting the points P1, P3

and P5. The resultant of these two motions is a path very closely approximating the arc of the circle along which the two strings of chords extend.

FIGS. 6c and 6d illustrate some of the principles underlying the process of circular path generation carried out in accordance with the present invention. Shown in FIG. 6c is the same arc appearing in FIG. 6a with the points P0 and P2 connected by a chord of length ΔD . With the chord ΔD as the hypotenuse, a triangle is formed by the X and Y components of the chord. Lying halfway between the points P0 and P2 on the circle is a point P1 and connecting the center C of the circle to the point P1 is a radius vector R1 having an inverse X component I_1 and an inverse Y component J_1 . Since the radius R_1 extends through the midpoint of the chord ΔD from the center of the circle, the chord and the radius vector are at right angles, and from this it follows that the triangles formed by the chord ΔD and its X and Y components on the one hand, and by the radius vector R_1 and its inverse X and Y components on the other hand are similar triangles.

Turning to FIG. 6d there is shown therein one of the strings of chords connecting the points P1, P3 and P5 in FIG. 6a. The particular chord shown in FIG. 6d connects the points P1 and P3 and forms the hypotenuse of a triangle whose other two sides are made up of the X and Y components of the chord. Lying midway between the points P1 and P3 on the arc in FIG. 6d is the point P2 connected to the center C of the circle by a radius vector R2. The radius vector R2 forms the hypotenuse of a second right triangle, in FIG. 6d, the other two sides of this triangle being made up of the inverse X and Y components I_2 and J_2 of the radius vector R2. Following the same reasoning advanced in connection with FIG. 6c, the chord of FIG. 6d and the radius vector R2 are at right angles and therefore the two right triangles of FIG. 6d are also similar triangles.

As part of the process of circular interpolation according to the invention, alternate ones of the radius vectors connecting the points P0 through P5 to the point C in FIG. 6a are represented by the X and Y coordinate distances from respective ones of them to the center of the circle C. The manner in which these X and Y coordinate distances are derived will not be described at this point. Let it be assumed instead that the radius vector R1 shown in FIG. 6c is represented by the Y coordinate distance J_1 from the point P1 to the point C and that the radius vector R2 is represented by the X coordinate distance from the point P2 to the point C. Given this information as well as the desired velocity along the chords of the circle and the duration of the chosen time period ΔT associated with each chord, the X component ΔX_{02} of the chord shown in FIG. 6c and the Y component ΔY_{13} of the chord shown in FIG. 6d may be derived.

Turning first to the generation of the chords shown in FIG. 6c, from the law of similar triangles it follows that

$$\frac{\Delta X_{02}}{J_1} = \frac{\Delta D}{R_1} \quad (11)$$

Similarly, with reference to the chord shown in FIG. 6d, it follows from the law of similar triangles that

$$\frac{\Delta Y_{13}}{I_2} = \frac{\Delta D}{R_2} \quad (12)$$

But

$$\Delta D = V \cdot \Delta T \quad (1)$$

$$R_1 = \sqrt{I_2^2 + J_1^2} \quad (13)$$

and

$$R_2 = \sqrt{I_2^2 + J_2^2} \quad (14)$$

Therefore, substituting (1) and (13) into (11) and multiplying both sides of the equation by $-J_1$ yields

$$\Delta X_{02} = -J_1 \frac{V \cdot \Delta T}{\sqrt{I_1^2 + J_2^2}} \quad (15)$$

Similarly, substituting (1) and (14) into (12) and multiplying both sides of the equation by I_2 yields

$$\Delta Y_{13} = I_2 \frac{V \cdot \Delta T}{\sqrt{I_2^2 + J_2^2}} \quad (16)$$

It is apparent from inspection that

$$\frac{V \cdot \Delta T}{\sqrt{I_2^2 + J_2^2}} = \frac{V \cdot \Delta T}{\sqrt{I_1^2 + J_1^2}} = Q \quad (17)$$

Therefore, substituting (17) into (15) and (16)

$$\Delta X_{02} = -J_1 \cdot Q \quad (18)$$

and

$$\Delta Y_{13} = I_2 \cdot Q \quad (19)$$

It is helpful to compare the equations (18) and (19), which relate to generating a circular path, with the equations (9) and (10) relating to the generation of a linear path. It is seen that they are the same, except that the positions of I and J are interchanged, and $-J$ is used. As written, equations (18) and (19) apply only for a clockwise (CW) progressing circular path. For CCW progression, the sign of the right side of each equation is simply reversed.

The information appearing in FIGS. 6c and 6d permits not only the calculation of the vector quantities ΔX_{02} and ΔY_{13} representing X and Y axis macromoves, but they also permit calculation of the Y coordinate distance from the point P3 to the center of the circle C and therefore the representation of the next radius vector R3 which, in turn, is used to calculate the X component of the next chord extending from the point P2 to the point P4 as seen in FIG. 6a. This is achieved by algebraically adding the quantities $+J_1$ and $-\Delta Y_{13}$, which respectively represent the Y coordinate distance from the point P1 to the center C of the circle, and from the point P3 to the point P1. This step of addition follows in the process the step of multiplication indicated by equation 16 (or alternatively equation 19) and once it is performed and the Y coordinate distance from point P3 to the center of the circle C is derived, then the X component of the next chord, extending between the points P2 and P4 and representing the next X axis macromove is calculated by another process of multiplication similar to that expressed by the equation 15 (or alternatively by the equation 18). In this manner, successive points are generated along a circular arc, alternate ones of the points being represented by their X and Y coordinates. These points are the X and Y macropositions mentioned with reference to FIGS. 6a and 6b. Also, as part of the process, X and Y axis macromoves ΔX and ΔY are alternately calculated and may then be divided into equal segments as shown in FIG. 6b by a process of division, permitting the generation of a series of X and Y axis micropositions along the X and Y axes from the respective series of chords shown in FIG. 6a.

4. THE DISCLOSED SYSTEM

FIG. 7 is a general block diagram of a system designed to carry out the process of linear and circular interpolation as described briefly in the foregoing sections. Two versions of the system will be described in greater detail with reference to detailed block diagrams respectively shown in FIGS. 9 and 40. Systems for carrying out the inventive processes are disclosed herein in detail in order to aid in understanding some of the specific aspects of the processes. Alternative ways for carrying out these specific aspects will occur to those skilled in the art of numerical controls and neither the details of the disclosed apparatus nor the specific intermediate steps performed thereby should be thought of as necessarily essential to the invention.

The system of FIG. 7 comprises four major groups of components. The first group serves to read a block of data from the record medium upon which it is initially stored, and to store it temporarily while the rest of the system is processing the previously read block of data. This group of components includes a tape reader 41, a decoder 43, and a tape reader start control 45. Also part of the group are a plurality of buffer registers 47. There are six such buffer registers, one for each of the six numbers which comprise a block of data on the punched tape 37 shown in FIG. 3. Connected between the decoder 43 and the buffer registers 47 is a buffer register shift control 49. When it is desired to transfer a block of data from

the punched tape 37 to the buffer registers 47, a transport mechanism within the tape reader 41 advances the tape past a reading mechanism which converts the indicia on the tape into electrical signals row by row. By means of the decoder 43, address signals are derived from the data rows representing the address symbols I, J, X, Y, G and Q and these signals in turn cause the buffer register shift control 49 successively to feed the signals following the respective I, J, X, Y, G , and Q address indicia into respective ones of the buffer registers 47 over the trunk line 51.

The second principal group of components is made up of a time shared digital computer 53, a plurality of computer function control arrays 55, and a series of "active" registers 57, so called because the data stored in them is continuously being processed by the computer 53. The output of each of the active registers 57 is connected through a common computer input trunk (CIT) line 59 to the inputs of the digital computer 53, and each of the active registers 57 receives at its inputs the signals produced by the digital computer 53 through a common computer output trunk (COT) line 61. In addition to serving as a path for transferring signals stored in the active registers 57 to the computer 53, the computer input trunk line 59 also carries the signals which are stored temporarily in the buffer registers 47 to be transferred to the active registers 57 by first entering them in the computer 53 over the computer input trunk line 59, and then transferring the signals from the computer 53 to the active registers 57 through the computer output trunk line 61.

It is the principal function of the computer 53 to process the digital data stored in the active registers 57 and to deposit the resulting data in appropriate ones of those registers. This operation is performed in a regularly recurring cyclical manner so that the information may be thought of as circulating counterclockwise, out of the active registers 57 through the input trunk line 59 into the computer 53, and out of the computer 53 through the output trunk line 61 back into the active registers 57. The selection and timing of the operations of the computer 53 necessary to perform this processing cycle is determined by the computer function control arrays 55. Specifically, each of the control arrays 55 applies to a respective control input of the computer 53 a control signal at the instant when the computer is to perform the operation associated with the particular control array involved. Just as the functioning of the digital computer must be timed, so must the flow of data from the active registers 57 onto the computer input trunk line 59. It is by selecting a particular one of the active registers 57 to feed data onto the input trunk line 59 that the data to be processed at a particular instant is selected. For this reason, each of the active registers 57 is provided with a set of READ gates which, when enabled, apply the signals then stored in the register to the computer input trunk line 59. Each of the READ gates is controlled by an array of program gates, the latter being labeled PGA in FIG. 7. When it is desired to transfer the data stored in a particular one of the active registers 57 into the digital computer 53 for processing, the program gate array associated with that register is caused to apply an enabling signal to the READ gates which are provided for that register. It is also desirable to have control over the particular one of the active registers 57 into which a particular group of signals produced by the computer 53 is to be deposited. Toward this end, each of the registers is provided with a set of WRITE gates which are operative, when enabled, to enter the signals, then on the computer output trunk line 61 into their associated register. Associated with each set of WRITE gates is an array of program gates (PGA) similar to those associated with the READ gates so that when it is desired to deposit the numbers produced by the computer 53 in a particular one of the registers 57, the program gate array associated with that register is caused to apply an enabling signal to the WRITE gates provided for the register.

To provide a time base for the operation of the entire system is the function of the third major group of components shown in FIG. 7. This group includes a Time Base and Digital Sweep 63, which, in the exemplary embodiment, is capable of producing at a plurality of outputs, a thousand different signal

combinations, each representing a different one of a thousand cyclically repeating time periods. This basic set of timing signals is applied over a trunk line 65 to a Multimode Time Base Logic 67, which in effect, converts the single set of timing signals capable of producing a thousand different time periods into any selected one of four different groups of timing signals, each group representing the same series of a thousand time periods, but under different conditions. These conditions are determined in the third component 69 of the timing system, labeled the End of Block Responsive Time Base Mode Selector. The Time Base Mode Selector 69 responds to signals produced by the fourth component 71 of the timing system labeled the End of Block Anticipating System which in turn, receives the information upon which it acts from the digital computer 53 through the computer output trunk line 61. The End of Block Anticipating System 71 in combination with the Time Base Mode Selector 69 is operative to change the timing signals being produced at the outputs of the Multimode Time Base Logic 67 when the *X* and *Y* axis positions represented by the command signals which are being generated are near to the end of the path represented by the block of data then stored in the active registers 57. The purpose of this feature is to cause the computer 53 to operate in different modes as the end of the data block which is being processed is approached, and different operations become necessary.

The timing signals which appear at the outputs of the Multimode Time Base Logic 67 are distributed throughout the rest of the system by a trunk line network 73. Thus, these timing signals are applied to the computer function control arrays 55, to the program gate arrays associated with the active registers 57, and to the program gate arrays associated with the buffer registers 47. The timing signals are also distributed by the trunk line network 73 to the Tape Reader Start Control 45, so that the tape reader 41 may be started in proper time relationship with the termination of the processing of data in the active registers 57, and with the transfer of data into those registers from the buffer registers 47 through the computer 53.

Comprising the last major group of components forming the system of FIG. 7, are the *X* and *Y* axis servos 75 and 77. Over a pair of trunk lines 79 and 81, the servos 75 and 77 continuously receive regularly the microposition signals which are recomputed at frequent and regularly spaced intervals of time and which are periodically stored in two of the active registers 57 as the end result of the data processing operation carried out by the digital computer 53.

In the present application, a single method is disclosed for performing circular interpolation. It is the "staggered" method, so called because the points whose *X* coordinates are calculated by the basic computational process and described generally with reference to FIG. 6 are not the same ones as the points for which the *Y* coordinates are so produced.

In contrast to the single method for performing circular interpolation, two different methods are disclosed for performing linear interpolation. They are the "simultaneous interpolation" method and the "staggered interpolation" method. In the simultaneous linear interpolation method, the processes or steps performed to derive *X* and *Y* coordinate points are performed concurrently for each of a succession of points along the path to be described. In other words, for each point for which the basic computational process is carried out to generate an *X* coordinate, there is also performed a similar basic computational process for producing the *Y* axis coordinate. On the other hand, in staggered linear interpolation, the process is staggered in the same way as it is in staggered circular interpolation. In other words, a given one of a succession of points along the path is represented by its *X* coordinate derived by the basic computational process, and the next point on the path is represented by its *Y* coordinate similarly derived. In the detailed description which will follow, two systems are disclosed. One of them, shown in FIG. 9, is operative to produce both linear and circular paths through the processes simultaneous linear interpolation and staggered circular interpolation. A second system, which is similar to the first, is shown in FIG. 40 and is operative to generate linear

and circular path segments through the process of staggered linear interpolation and staggered circular interpolation. The point that is to be observed here is that the system which is generally shown in FIG. 7 is representative of the organization of both of the systems disclosed in detail in FIGS. 9 and 40. The structural features which distinguish the two systems illustrated in FIGS. 9 and 40 reside principally in the details of their associated program gate arrays, and in their computer function control arrays. Otherwise, the general organization of the two systems is the same.

5. OPERATION OF THE SYSTEM DISCLOSED GENERALLY IN FIG. 7

The first step in processing a block of data taken from the punched tape 37 is to transfer the data for temporary storage into the buffer registers 47. It should be noted that all of the data blocks, except the very first to be processed, will be entered in the buffer registers 47 just after the transfer of the previous data block from the buffer registers to the active registers 57. The data block thus entered in the buffer registers 47 remains there for substantially the entire time during which the previous data block, just transferred to the active registers 57, is being processed through the computer 53 to generate the path represented by that previous block of data. Thus, at substantially all times two blocks of data are being stored in the system, one block in the active registers 57 and the other in the buffer registers 47.

Turning now to the particular ones of the buffer registers 47, two of the registers 47*X* and 47*Y*, are provided for receiving the *X* and *Y* coordinates of the end point that is to be reached in accordance with the block of data. They are labeled "*X* Axis Block Target Buffer Register (XEP BUF)" and "*Y* Axis Block Target Buffer Register (YEP BUF)." In terms of the data described with reference to the exemplary data block in FIG. 4, the information which will be stored in register 47*X* comprises the rows of indicia making up the *X* coordinate of the point P2, and the data to be stored in the register 47*Y* comprises the rows of indicia making up the *Y* coordinate of the point P2 in FIG. 2.

The next pair of buffer registers 47*I* and 47*J* receive the data representing (in the case of a linear block) the sign and distance along the *X* and *Y* axes of the movement to be carried out in accordance with the block of data. The *X* and *Y* sign and distance data are referred to (in the case of a linear block) as the quantities *I* and *J* respectively, and the registers storing them are labeled "*X* Axis Component For Next Block (*I* BUF)," and "*Y* Axis Component For Next Block (*J* BUF)" respectively.

A fifth buffer register 47*Q* labeled "Macromove Quotient Buffer Register (Q BUF)" is provided to receive the signals derived from the tape indicia representing the macromove quotient *Q*. Finally, a sixth buffer register 47*G* labeled "Path Mode Buffer Register (G BUF)" receives the signals corresponding to the tape indicia which represent the number *G*.

As noted previously, the entire block of data now residing in the buffer registers 47 was transferred there shortly after the previous block of data had been transferred from the buffer registers 47 into the active registers 57. The present block of data residing in the buffer registers 47 remains there until the previous block of data has served its purpose in the active registers 57 and is no longer needed. This occurs very shortly before the move along the path segment represented by the previous block of data and residing in the active registers 57 has been carried out. It may be assumed that this previous path segment is the segment 27*a* between points P0 and P1 in FIG. 2, and that the block of data now to be transferred from the buffer registers 47 into the active registers 57 represent the following path segment 27*b* extending from point P1 to point P2 in FIG. 2.

Concentrating now on the active registers 57 which are about to receive the block of data residing in the buffer registers 47, they may be divided functionally into two groups, 57*A* and 57*B*. The first group 57*A* is comprised of six registers which correspond to the six buffer registers 47, and which are

provided to receive the data when it is transferred from the buffer registers 47. Specifically, the data stored in the buffer registers 47X and 47Y is received and stored in a pair of active registers 95 and 97 respectively labeled "X Axis Block Target Register (XCEP)" and "Y Axis Block Target Register (YCEP)," the symbols XCEP and YCEP designating "X command end point" and "Y command end point."

A second pair of active registers is provided to receive the information transferred from the buffer registers 47I and 47J. They are the "X Axis Component For Present Block (I)" register 99 and the "Y Axis Component For Present Block (J)" register 101.

The last pair of the first group of active registers 57 serve to store the data read from the QBUF and GBUF registers 47Q and 47G, and are identified in FIG. 7 as the "Macromove Quotient Register (Q)" 103 and the "path mode register (G)" 105.

The data stored in the foregoing group 57A of six registers serves as the basic information from which additional signals representing intermediate macropositions and micropositions are derived by the digital computer 53 through the process of interpolation in accordance with the present invention.

In general, the basic data stored in the six active registers 95 - 105 is utilized in the process of interpolation by the computer 53 by transferring the data over the computer input trunk line 59 to the computer, storing the data in appropriate storage devices (not shown) provided within the computer and operating on the data to derive desired further data therefrom.

To receive the signals produced by the computer 53 as a result of the computations which form part of the interpolation process is the function of the second group 57B of the active registers 57. Let us assume that the block of data representing the desired motion along the path segment 27b from point P1 to point P2 in FIG. 2 has been transferred from the buffer registers 47 to the six active registers 95 - 105. The computational process for performing interpolation is then ready to begin. The first step is to produce the X and Y axis macromove numbers ΔX and ΔY corresponding to the macromove distances ΔX and ΔY illustrated previously in FIG. 5. It will be noted that all numbers produced by the system under discussion are represented by digital signals.

It should be understood that when mention is made herein of reading, writing, adding, storing or otherwise processing a "number" the reference being made is to the signals which represent that number, rather than to the number per se.

It will be recalled from the discussion of FIG. 5b that, for a linear path segment,

$$\Delta X = I \cdot Q \quad (9)$$

and

$$\Delta Y = J \cdot Q \quad (10)$$

In carrying out the invention therefore, the numbers stored in the I register 99, the J register 101, and the Q register 103 are transferred into corresponding storage devices in the digital computer 53. The computer then produces a pair of axis macromove numbers $I \cdot Q$ and $J \cdot Q$. One of these numbers, the number $J \cdot Q$, is entered over the computer output trunk line 61 into a pair of storage registers 107 and 111, respectively identified in FIG. 7 as the "Y Axis Macromove Magnitude Register (ΔY)" and as the "Y Axis Macromove Remainder Register (YR)." The reason for using two registers to store a single number need not be understood at this point. Suffice it to say that in the exemplary embodiment to be described herein, the four most significant digits of the macromove number $J \cdot Q$ are stored in the ΔY register 107, and that the remaining digits of the number are stored in the YR register 111. Subsequently, the computer produces an axis macromove number $I \cdot Q$ and it is stored in another pair of active registers 109 and 113, respectively labeled "X Axis Macromove Magnitude Register (ΔX)" and "X Axis Macromove Remainder Register (XR)."

The four most significant digits of the product $J \cdot Q$ stored in the ΔY register 107, are used to produce a number represent-

ing the Y coordinate of the point which is to be reached at the end of the axis macromove ΔY . The Y coordinate number, resulting from this operation is stored in a register 115, designated as "Y Axis Macroposition Register (YCP)." The operation is performed after each calculation of the number ΔY by adding in the computer the numbers then stored in the registers 115 and 107, and storing the sum in the YCP register 115.

Similarly for the X axis, a register labeled the "X Axis Macroposition Register (XCP)" 117 is provided for storing the X command position number XCP. Thus, the number representing the last XCP macroposition is transferred from the register 117 to the computer 53 followed by the ΔX number from register 109. Their sum is then produced by the computer and replaces the previously stored XCP number in the register 117.

In addition to producing XCP and YCP macropositions, the system is also capable of producing micropositions designating the points along the X and Y axes to be reached at the ends of successive micromoves such as those shown in FIG. 5b. The numbers representing Y axis micropositions are received in a register 119 labeled the "Y Axis Microposition Register (YSC)," and the numbers representing the successive X axis micropositions are received in a similar register 121 labeled "X Axis Microposition Register (XSC)." The numbers stored in the registers 119 and 121 may be thought of as the end product of the computations carried out by the computer 53, since it is these numbers which are applied in the form of digital electrical signals over the trunk lines 79 and 81 to the X and Y axis servos 75 and 77. It is these numbers therefore, which serve as the Y Servo Command and X Servo Command numbers. Hence the abbreviation YSC and XSC to designate the numbers stored in the registers 119 and 121.

The Y axis microposition number YSC is derived in a manner similar to that for deriving the Y axis macroposition number YCP. Thus, the number stored in the register 119, and representing the last microposition for the Y axis is taken from the register and is transferred to the computer 53. To this number is then added a number representing a micromove, or $\Delta Y/N$. The latter quantity is derived by taking the number from the ΔY register 107, transferring it to the computer 53, and dividing it by N. The result, $\Delta Y/N$, is added to the number previously read from the YSC register 119, and the sum is transferred from the computer 53 back into the YSC register 119. The process of adding the number $\Delta Y/N$ to the number YSC taken from the YSC register 119, is repeated in the exemplary embodiment every 2 milliseconds, thus serving to make available to the Y axis servo 77, signals representing a new microposition number 500 times every second. Successive microposition numbers are also derived for representing successive micropositions along the X axis. These numbers, as stated previously, are stored in the register 121, and they are derived from the ΔX number stored in the register 109 in a manner which is identical to that for deriving the YSC number in the register 119 from the number ΔY in the register 107.

The computational process whereby data is periodically taken from the active registers 107-121, applied to the computer 53, processed by the computer to generate successive position numbers, and returned to the registers continues until the system detects that the process is about to be completed. This occurs when the axis macroposition number stored in one of the registers 115 and 117 approaches within a predetermined distance of the desired end point for the corresponding axis, as represented by the number stored in one of the block target registers 95 and 97. In a preferred embodiment of the invention, a test is carried out for one of the axes after each macroposition has been computed for that axis. Furthermore, in the preferred embodiment of the invention, in order to enhance the sensitivity of the test, the axis for which this test is performed is that axis along which the commanded movement is the faster. Thus, for example, in computing the macropositions for the path 27b in FIG. 2, the test would be performed for the X axis, since the desired movement and the required speed for executing such movement along the X axis are

greater than the corresponding quantities for the Y axis. In performing the test to determine when the computational process for the data associated with the path 27b has been completed, each time a new macroposition number XCP is stored in the XCP register 117 that number is compared by the digital computer 53 with the number stored in the XCEP register 95. When it is detected that the macroposition number XCP is within a predetermined distance of the target point XCEP, the computer 53 produces a signal indicating this condition and this signal is applied over the computer output trunk line 61 to the End of Block Anticipating System 71. In response to an "end of block" signal from the system 71, the Time Base Mode Selector 69 changes the timing signals which are produced by the Time Base Logic 67, and this change in timing signals in turn changes the sequence of operations performed by the computer 53. When the system is performing simultaneous linear interpolation, the new sequence of operations results in a modification of the position signals applied to the servo devices 75 and 77 so as to bring the last macroposition numbers stored in the registers 115 and 117 as close as possible to the corresponding target points XCEP and YCEP for the block. Another effect of the change in the sequence of operations is to initiate the transfer into the active registers 57 of the data which up to that point was being held in the buffer registers 47 and which is associated with the data block representing the next path segment 27c. The change in the timing signals produced by the Multimode Time Base Logic 67 is also effective to cause the Tape Reader Start Control 45 to start the transport mechanism of the tape reader 41 so as to cause the tape reader to read the following block of data corresponding to the path segment 27d from the punched tape 37 and to transfer that data into the buffer registers 47.

The purpose of the foregoing description has been to provide a general understanding of the manner in which a system capable of performing linear and circular interpolation might be organized. Sufficient details have not been given up to this point to understand exactly the operation or organization of the system. To provide such an exact understanding will be the purpose of the detailed description of two exemplary systems which follow. As stated earlier, both of the systems which will be described herein conform substantially to the organization of the system shown generally in FIG. 7 and an occasional reference back to the foregoing general description and to FIG. 7, as the detailed description unfolds may be found helpful.

G. DETAILED DESCRIPTION OF SIMULTANEOUS LINEAR PATH GENERATION

1. The System in Greater Detail

The process by which macroposition numbers XCP and YCP are generated for each of a series of equally spaced points will be described in detail in connection with a system, shown in FIG. 9, which is capable of performing both simultaneous linear interpolation and staggered circular interpolation. Before considering the operation of the system whereby linear path generation is carried out, selected parts of the system will first be described. It will be apparent to those skilled in the art that the methods here to be described are carried out through the utilization of digital processing apparatus, but the specific apparatus so employed may take any of a wide variety of forms.

A. THE CENTRAL TIMING SYSTEM

Provision is made in the numerical control system of FIG. 9 for producing a digital sweep formed of signals which digitally represent a reference number repeatedly and cyclically changing through a progressively increasing series of values during successive time intervals. Provision is also made for providing a sinusoidal reference signal for the X and Y axis servo drives 75 and 77, and for deriving from the digital sweep, signals which control the time periods during which various operations of the numerical control are performed. A particularly effective combination for performing all of these functions are the Time Base and Digital Sweep 63 and the Time Base Mode Selector 69 shown respectively in FIGS. 9a and 9e.

1. PRODUCING THE DIGITAL SWEEP

Forming part of the Time Base and Digital Sweep 63 is a series of four cascaded binary coded decimal pulse counters 131, 133, 135, and 137. The digital sweep is derived from the outputs of the first three stages, 131, 133, and 135, respectively labeled the A, B, and C decade counters and collectively identified as the Digital Sweep Generator 139. The decade counters 131-137 are of similar construction and are well known to those skilled in the art. A typical unit of this type includes four cascaded flip-flops interconnected so that the decade repeatedly counts from 0 to 9 in response to successive input pulses, being reset to 0 by every 10th input pulse and signaling its contents on four output lines, or terminals, in 8, 4, 2, 1 binary code. Additionally, each time the unit is reset to 0, it produces a signal on a fifth or "carry" line.

The A decade counter 131 is stepped by output pulses produced at a repetition frequency of 500 kilohertz by a Divide by 4 circuit 141 which in turn is driven by pulses from a 2 megahertz clock 143. The output pulses produced by the clock 143 and the Divide by 4 circuit 141 are shown in FIG. 11 as the waveforms 145 and 147 respectively. The units digit of the digital sweep is signaled on output terminals of the A decade counter 131, labeled A1, A2, A4 and A8 to signify the binary weight of the signals appearing on the respective terminals. Since the pulse repetition frequency of the pulses applied to the A decade counter 131 is 500 kilohertz, or a pulse every 2 microseconds, the A decade counter cycles through its 10 count states every 20 microseconds. Through line 149 the "carry" output of the A decade counter 131 is applied to the input of the B decade counter 133 and consequently that counter cycles through its 10 count states every 200 microseconds and on its four output terminals, labeled B1, B2, B4 and B8 is signaled the tens digit of the digital sweep. The "carry" output of the B decade counter 133 is applied to the input of the C decade counter 135 through the line 151, causing that counter to cycle through its 10 count states every 2 milliseconds. It is on the output terminals C1, C2, C4, and C8 of this counter that the binary coded decimal signals representing the hundreds digit of the digital sweep are derived.

Collectively, the cycling decade counters 131, 133, and 135 produce a digital sweep formed of signals representing a reference number which repeatedly cycles during successive time intervals from a first predetermined value (here 000) to a second predetermined value (here 999) by uniform increments (here 1) and at uniformly time-spaced instants (here 2 microseconds). The signals produced by the Digital Sweep Generator 139 are applied to other components of the system and in order to maintain clarity in the drawings, their connections are not indicated by lines. Instead the output terminals of the Digital Sweep Generator 139 contain the symbol Δ and terminals of devices elsewhere in the system, such as the compare circuits in FIG. 9d, which are connected to terminals of the Digital Sweep Generator, contain similar symbols, with an additional indication of the particular Digital Sweep Generator terminal to which they are connected.

2. PRODUCING GATING PULSES TO ASSURE UTILIZATION OF THE DIGITAL SWEEP WHILE ITS COUNT STATES ARE STABLE

The progression of the reference number collectively signaled by the cycling decade counters of the Digital Sweep Generator 139 is shown in FIG. 11 as the staircase shaped waveform. It will be seen that each new count state begins with the negative going edge of the waveform 147 and lasts until the next such negative going edge. The wavy line during the initial portion of each count state represents an instability period during which the counters are tumbling from their previous count states. It is the principal function of the Divide by 4 circuit 141 to produce a gating pulse which occurs during the central portion of the stable period of each count state so that units in the system which utilize the output of the Digital Sweep Generator may be gated open to receive the outputs of the Digital Sweep Generator during its stable count states.

The Divide by 4 circuit 141 includes two gated flip-flops 155 and 157. The flip-flop 155 has a pair of inputs J1 and K1 for receiving gating signals and a third input labeled CP for receiving clock pulses. The flip-flop also has a pair of outputs labeled T1 and $\bar{T}1$ characterized by the fact that when a logic "1" level signal appears at one of the outputs, a logic "0" level signal is produced at the other. The flip-flop 157 has inputs and outputs corresponding to those of flip-flop 155 and are labeled J2, K2, CP, T2 and $\bar{T}2$. Flip-flops of the type illustrated for use in the Divide by 4 circuit are commonly referred to as J-K flip-flops and are well known to those skilled in the art. It will therefore suffice here to describe the manner in which they operate under four possible conditions.

Condition 1: if neither the J input nor the K input receives a logic 1 level signal, i.e., if neither J nor the K input is "qualified," while a clock pulse is applied to the flip-flop, then the clock pulse does not change the state of the flip-flop. Condition 2: if the K input is qualified during application of a clock pulse, the negative-going edge of the clock pulse following qualification of the K input of the flip-flop will reset it to its "0" state, in which state its $\bar{T}$ output produces a logic 0 level and its T output produces a logic 1 level. Condition 3: if the J input is qualified, application of a clock pulse will set the flip-flop to its "1" state, that is, to that state in which its T output produces a logic 1 signal and its $\bar{T}$ output produces a logic 0 signal. With respect to the last two conditions, it will be understood that, if the flip-flop is already in the 0 state when its K input is qualified, it will simply stay in that state after a clock pulse is applied to it. Similarly, if the flip-flop is already in the 1 state when its J input is qualified, it will simply stay in that state after receiving a clock pulse. Condition 4: if both the J and K inputs of the flip-flop are qualified at the time when a clock pulse is applied to the flip-flop, application of a clock pulse to the flip-flop will cause it to reverse its state from whichever state it had been in prior to application of the clock pulse.

With this basic understanding of the J-K flip-flops which form the Divide-by-4 circuit 141, operation of the circuit can be simply described. It is the well-known switch tail or Johnson counter and is basically a two-stage serial shift register connected in a loop with the outputs cross-connected to the inputs. The T1 and $\bar{T}1$ outputs of Flip-flop 155 are connected to the J2 and K2 inputs of flip-flop 157 respectively and the T2 and $\bar{T}2$ outputs of flip-flop 157 are connected to the K1 and J1 inputs respectively of the flip-flop 155. The clock pulse inputs of both flip-flops 155 and 157 receive pulses from the 2 megahertz clock 143. Assuming that initially both flip-flops are in the reset condition, logic 1 signals are applied by the $\bar{T}1$ and $\bar{T}2$ outputs of the flip-flops to the K2 and J1 inputs of the two flip-flops. This is indicated by the waveform diagrams shown above the Divide by 4 circuit 141. Consequently, upon receiving the negative going or switching edge of the first clock pulse, the first flip-flop 155 becomes set but the second flip-flop 157 remains in the reset condition.

With the flip-flop 155 set, when the negative-going edge of the second clock pulse occurs, the T1 output of the flip-flop 155 will be qualifying the J2 input of flip-flop 157 while the $\bar{T}2$ output of the flip-flop 157 will continue to qualify the J1 input of flip-flop 155 so that the negative-going edge of the second clock pulse will leave the flip-flop 155 in the set state but will also cause the second flip-flop 157 to switch from the reset to the set state. Thus, when the third clock pulse arrives, the T1 and T2 outputs of flip-flops 155 and 157 will qualify the J2 and K1 inputs of flip-flops 155 and 157. Since the K1 input of flip-flop 155 and the J2 input of flip-flop 157 are qualified when the trailing edge of the third clock pulse occurs, that trailing edge will cause flip-flop 155 to be reset while leaving flip-flop 157 in the set condition. This last change causes the $\bar{T}1$ output of flip-flop 155 to qualify the K2 input of flip-flop 157 and the $\bar{T}2$ output of flip-flop 157 to qualify the K1 input of flip-flop 155. With this set of conditions existing, arrival of the trailing edge of the fourth clock pulse resets the flip-flop 157 while leaving the flip-flop 155 reset. This was the initial

condition of the two flip-flops. From this point, the cycle repeats.

The T2 pulse train, produced by the flip-flop 157, is the one used to drive the Digital Sweep Generator 139, which is advanced by one count with each negative-going edge of the T2 waveform. Thus, the time which elapsed during the four clock pulses represents one count state of the Digital Sweep Generator 139. Specifically, the A decade counter 131 changes its count state on the trailing edge of the T2 pulse, delayed by the propagation delay of its flip-flops. Decade counter 133 changes its count state, as signaled at its output terminals, on the trailing edge of the carry output of decade counter 131, delayed by the delays of the flip-flops in counter 133. Thus the delays in successive counters 131-137 cumulate. In the diagram above the Divide by 4 circuit 141, the shaded area spanning the second, third, and fourth clock pulses begins at the instant when the C decade counter 135 has stabilized and ends when the A decade counter 131 begins to change. Thus the shaded area represents the approximate time period during which all of the counters 131-135 of the Digital Sweep Generator 139 are in a stable state. The unshaded area, coinciding approximately with the first clock pulse, represents the brief time period during which some of the individual counters of the Digital Sweep Generator are tumbling to their new count states.

To produce a gating pulse which falls approximately during the mid-portion of the shaded area representing the stable state of the Digital Sweep Generator, the T1 output of flip-flop 155 and the T2 output of flip-flop 157 are applied to an AND gate 159. From the waveforms related to the Divide by 4 circuit 141, it is seen that the T1 and T2 outputs of flip-flops 155 and 157 are concurrently at a logic 1 level during the mid-portion of the shaded area so that during this time period a logic 1 or enabling voltage level appears at the output of AND gate 159. This signal is represented by the logic symbol $T1 \cdot T2$.

3. GENERATING A BASIC SERIES OF PROGRAM STEPS

The Time Base and Digital Sweep 63 also includes means for producing timing signals during different predetermined states of the Digital Sweep Generator so as to time the operation of various elements in relation to particular count states of the Digital Sweep Generator 139. The particular arrangement used here comprises a Program Step Generator 161 formed of the B and C decade counters 133 and 135 of the Digital Sweep Generator 139, and the D decade counter 137, the latter being driven by the carry output of the C decade counter 135. Just as the ABC decade counters 131, 133, and 135, when taken together, comprise a Divide by 1,000 counter which cycles through a thousand count states 500 times per second, so that the BCD decade counters 133, 135, and 137, when taken together, comprise a Divide by 1000 counter which cycles through a thousand different count states 50 times per second. Stated differently, the count signaled by the ABC decade counters 131, 133, and 135, which comprise the Digital Sweep Generator 139, changes 10 times for every change in the count signaled by the BCD decade counters 133, 135, and 137 which comprise the Program Step Generator 161.

The relationship just described is graphically illustrated in FIG. 11 wherein the waveform 163 illustrates the steps through which the Program Step Generator 161, formed of the BCD decade counters 133, 135, and 137, progresses. Each of these steps is a program step, and it is seen from the figure that, during the same period in which the program steps progress from 0 to 100, the Digital Sweep Generator 139 goes through a complete cycle from 000 to 999.

The relationship between digital sweep numbers and program steps is also illustrated in FIG. 10a. It shows a rectangular chart vertically divided into 10 columns, each column divided into 10 equal zones and each zone being further divided into ten equal steps. Thus, the chart is divided into one thousand equal steps, and each step represents one count state

of the Program Step Generator 161. The top step in the first column is labeled 000 and represents the initial count state of the Program Step Generator 161. The bottom or last step of the first column is labeled 099 and represents the 100th count state of the Program Step Generator. Steps in the remaining nine columns have a similar significance; thus, the top step of the second column represents count state number 101, the bottom step of that column the 200th count state, and so on through the entire chart until the last step of the last column, representing the 1,000th count state of the Program Step Generator.

As pointed out earlier, for each step or count state of the Program Step Generator 161 the Digital Sweep Generator 139 goes through ten steps or count states of its own. Thus, during each 100 steps represented by successive columns of the programming chart, the Digital Sweep Generator cycles through a thousand different count states from 000 to 999 and this is indicated in FIG. 10b by the leftward sloping line 165 which is an approximate representation of the count states of the Digital Sweep Generator 139.

Continuing with the consideration of the Program Step Generator 161, it is seen that it generates a series of 1,000 potential program steps from which any step may be selected by means which are responsive to the particular set of signals existing on the outputs of the Program Step Generator during that step. This selection might be performed by devoting a separate set of AND gates to each program step that is to be selected. Obviously this would require quite a large number of such AND gates since each program step is signaled on 12 output terminals, B1-B8, C1-C8, and D1-D8. Consequently, to reduce the complexity of the equipment required to select various program steps for performance of differently timed functions, a series of three binary coded decimal to decimal code converters 167, 169 and 171 are provided for reducing the number of signals by which different program steps are represented. Each of the three code converters has four input terminals for receiving a binary coded decimal digit from one of the three decade counters of the Program Step Generator 161 and ten output terminals. The units converter 167 signals the units digit of the program step on one of 10 terminals labeled zero through nine; the tens converter 169 signals the 10s digit of the program step on one of 10 terminals labeled 00 through 90; and the hundreds converter 171 signals the hundreds digit of the program step on one of its 10 outputs labeled 000 through 900.

Through the use of the code converters, each of the 1,000 program steps shown in the programming chart in FIG. 10a is represented by signals on a different combination of three output terminals among the thirty output terminals of the code converters 167, 169, and 171. As an example, assume that the Program Step Generator is in its count state number 746, or with reference to the programming chart of FIG. 10a, in the 47th step of the eighth column. This count state and program step will be represented at the output terminals of the Program Step Generator 161, by logic 1 signals appearing on terminals B2 and B4, C4, and D1, D2, and D4. In contrast, the same count state and program step is represented at the outputs of the binary coded decimal to decimal converters 167, 169, and 171 by logic 1 signals on three output terminals: the 700 terminal of the hundreds converter 171, the 40 terminal of the tens converter 169, and the 6 terminal of the units converter 167. Thus, if it were desired to initiate operation of some part of the system during count state 746, it could be accomplished by use of means such as an AND gate responsive to concurrent signals on the three code converter output terminals just listed. This is indeed the way in which many of the timing signals are produced in the system, and the AND gates used for this purpose will be referred to as "program" gates.

The Time Base and Digital Sweep 63 also includes a decoder 173 for producing a program clock pulse (PCP) and a WRITE pulse (WP). Each of the PCP and WP pulses occur once during each cycle of the A decade counter 131 and their relative times of occurrence during the ten counts of each

cycle of the A decade counter 131 are shown in the waveform diagrams directly above the counter in FIG. 9b. The PCP pulse initiates each program step and the WP pulse signals the instant when data is to be stored in a storage register during a particular program step. The details of the decoder 173 will be obvious to one skilled in the art and will not be described. It uses the outputs A1, A2, A4 and A8 of the decade counter 131 and may also use T1, T2 of the Divide by 4 circuit 141. The decoder 173 may include suitable inverters, AND gates, and flip-flops arranged in accordance with the logic required to produce the PCP and the WP pulses so as to avoid marginal timing problems. In particular, the PCP pulse should end before the count state of the Program Step Generator changes.

4. GENERATING ALTERNATIVE SERIES OF PROGRAM STEPS

The Program Step Generator 161 as thus far described can produce 1,000 timing signals, each defining a different 20 microsecond time period during each cycle of the Program Step Generator. It has been stated that, to initiate a certain operation during a given count state of the Program Step Generator 161 such as count state 746, three inputs of a program gate would be connected to the 700, 40, and 6 output of the converters 171, 169, and 167. Let it be assumed, however, that a certain operation is to be initiated during count state 746 only if a given condition exists, a different operation only if another condition exists, a third operation only if a third condition is met and yet another operation only if a fourth condition has been reached. Let it be assumed also, that a fifth operation is to be carried out during count state 746 regardless of whether or not any of the first, second, third and fourth conditions have been met. This is the function of the Multimode Time Base Logic 67 shown in detail in FIG. 9d. It includes four groups of AND gates 175, 177, 179, and 181. Each group includes 10 AND gates of which only the first and last two are shown. Within each group, successive ones of the individual AND gates have one of their inputs connected to successive ones of the individual outputs 000 to 900 of the converter 171 in FIG. 9d.

The output terminals of the 10 AND gates 175 are identified as 1000, 1100....1900, and may be caused to receive the signals present on the outputs 000, 100, ..., 900 of the converter 171 by an enabling signal on the input line and M1. The remaining AND gate groups 177, 179, and 181 have outputs identified as 2100-2900, 3100-3900, and 4100-4900 respectively. Signals corresponding to those on the outputs 000-900 may be placed on the outputs 2100-2900, 3100-3900, and 4100-4900 by enabling signals on respective ones of the input lines M2, M3, and M4. If the first of the four conditions is signaled by an enabling signal on the line M1, and the second, third and fourth conditions are signaled on the lines M2, M3 and M4 respectively, then to produce a timing signal during the count state 746 of the Program Step Generator 161 only when the first condition exists, two of the three inputs of the program gate will remain connected to the 40, and the 6 outputs of the converters 169 and 167 but the third input will have to be connected to the 1700 output of the AND gates 175 instead. In this way, a timing signal will be produced at the output of the program gate during count state 746 only if the first condition is in fact signaled on the M1 line. By similar reasoning, to produce a timing signal during count state 746 only if the second, third, and fourth conditions have been fulfilled, the third input to the program gate will have to be connected to the 2700, 3700, and 4700 output terminals of the AND gates 177, 179, and 181. If on the other hand, a timing signal is to be produced during count state 746 of the Program Step Generator 161 regardless of the existence of the four different conditions, then the inputs of the program gate should be connected directly to the 700, 40, and 6 outputs of the converters 171, 169, and 167.

Thus the generation of a timing signal during any one of the 1,000 different count states of the Program Step Generator

161 may be conditioned on the existence of one of four conditions, and for each one of the four conditions there exist 1,000 potential timing signals which may be derived with the aid of respective ones of the AND gates 175, 177, and 179, and 181. The 1,000 potential time periods or program states that may be derived directly from the outputs of the converters 171, 169, and 167 were shown in FIG. 10a in the form of a programming chart. Similar programming charts exist for the four different groups of 1,000 timing signals which can be derived in response to signals on the lines M1, M2, M3, and M4. One such programming chart is shown in FIG. 10c. It is identical to the programming chart of FIG. 10a except that its 10 columns are numbered 1,000-1900 in keeping with the numbers and the AND gates 175 which must be enabled to produce the timing signals represented by the programming chart.

Summarizing, the programming chart shown in FIG. 10a represents 1,000 different time periods corresponding to the 1,000 successive count states of the Program Step Generator 161 during which a timing signal may be derived by means of a program gate connected to the outputs of the converters 171, 169, and 167. Each of the 1,000 different timing signals represented by the programming chart on FIG. 10a may be thought of as requiring a single program gate connected to an appropriate three of the 30 output terminals of the converters 171, 169, and 167. Thus, if it were desirable to produce a timing signal during each of the 1,000 program steps of the Program Step Generator 161, 1,000 program gates, each connected to a different combination of three output terminals of the converters might be used. In such a case, 1,000 different timing signals would be produced, one during each of the count states of the Program Step Generator 161, regardless of the presence of a signal on the lines M1-M4. If, in a similar manner 1,000 program gates were connected to the AND gates 175, and to the outputs of the converters 169 and 167 so as to generate the 1,000 possible timing signals represented by the programming chart in FIG. 10c, these timing signals would be produced only if an enabling signal were received by the AND gates 175 over the input line M1. Further, if the timing signals thus produced were used to initiate 1,000 different operations in the system in succession, these operations too would occur only when the M1 input line was energized. In actual practice, four groups of programming gates (or equivalent devices) are provided, each for producing a series of timing signals under a different one of the four conditions referred to. A different series of operations is initiated by each of the four series of timing signals, each series being repeated with each cycle of the Program Step Generator 161. In this manner, the system of FIG. 9 may be caused to carry out any one of four series of operations, or to operate in any one of four different "modes," by energizing an appropriate one of the four input lines M1-M4.

The four different modes in which the system of FIG. 9 is capable of operating are required to perform four different types of operation as a path consisting of linear and circular segments is generated. In its first mode of operation the system generates either a linear or a circular path segment. This mode of operation is typically repeated a large number of times in the course of generating a path, each repetition being called an iteration, extending over a period of time which is equal to the cycling time of the program step generator 161, and resulting in the generation of a new pair of macropositions for the X and Y axes.

Modes 2 and 3 relate exclusively to the generation of a linear path and relate in particular to a corrective operation performed during the last few iterations associated with the generation of the linear path which is designed to bring the last macroposition for each of the X and Y axes in coincidence with the desired end point for the linear path segment.

The last of the four modes of operation is put into effect during the last sequence of operations associated with generating either a linear or a circular path. In this mode of operation the system transfers data from the buffer registers 47 to the active registers 57 and processes the data so as to

generate the macromove numbers for the following path segments to be executed.

It can be stated generally that in a typical operational sequence the system operates a large number of times and through a large number of the same, mode 1, iterations in generating a linear or circular path segment. It is toward the end of the path segment that a relatively rapid change occurs in the operating mode of the system. This rate of change may be so rapid as to shift the system through different modes of operation within one iteration, or within less than 20 milliseconds. This indeed is the case where the system is completing a circular path. It will be seen that, when generating a circular path, the system shifts abruptly from its first to its fourth mode of operation toward the end of the path. In contrast, in performing simultaneous linear interpolation, and in carrying out a feature of the invention which is yet to be discussed as the end of the linear path is approached the system shifts from operating mode 1 to mode 2, then into mode 3 and operates in that mode for nine iterations and only during the last iteration is it switched into mode 4 operation in which the macromove numbers for the following path segment are derived. To permit the system to step through its successive operating modes at different rates, depending on whether a circular or a linear path segment is being generated, is the function of the Time Base Mode Selector 69 (FIG. 9e).

5. THE TIME BASE MODE SELECTOR

Generally, the Time Base Mode Selector 69 responds to certain pre-selected conditions occurring in the system to take a step toward changing the operating mode of the system in a prearranged manner. One condition to which the Time Base Mode Selector 69 responds is the nearing of the end of the path segment being generated by the system. This condition is sensed by the Fast Axis Projected Overrun Detector 71, and is indicated on its output line 82 by a signal which is one of the many signals that are applied to the Time Base Mode Selector 69. In all, in the exemplary system of FIG. 9 there are over 20 different conditions to which the Time Base Mode Selector 69 is designed to respond. Each of these conditions is detected by a different logic gate or gating array. The various conditions thus responded to are listed in a block 183 which thus symbolically represents the array of program gates which are used for this purpose. The various conditions represented in the block 183 will be explained subsequently in connection with the operation of the system. It need only be noted at this point that, in response to each condition an output signal is produced by the program gate array 183 and is applied over a line 184 to one input of an AND gate 185 whose other input is energized by the PCP output of the Time Base and Digital Sweep 63. Therefore, every time an output signal is produced by the program gate array 183, a program clock pulse is gated through the AND gate 185. These gated program clock pulses are applied to a four stage binary counter 186 whose outputs in turn are decoded by a binary to single line decoder 187. Binary counters and decoders of the type suitable for use in the Time Base Mode Selector 69 are well known to those skilled in the computer arts and in fact the combination illustrated in the FIG. 9e is shown and explained in *Digital Computer Principles* by the Burroughs Corporation, McGraw-Hill Book Company, Inc., 1962, at page 327. The decoder 187 is provided with a series of 16 outputs numbered S1 through S16. In combination the binary counter 186 and the decoder 187 operate in a manner analogous to a stepping switch in that, in response to each pulse received from the AND gate 185 the binary counter is stepped to its next higher count state, signaled at its outputs in binary coded decimal form. These output signals in turn appear at successive ones of the 16 outputs S1-S16 of the decoder 187. The binary counter 186 has a count capacity of 16 and is cyclic in operation. Thus, the first program clock pulse gated through the AND gate 185 causes an output signal to appear on the S1 output terminal of the decoder 187, and the 16th gated pulse results in a signal on the S16 decoder out-

put terminal. The following pulse applied to the counter 186 again sets it in its first count state and again causes a signal to appear on the S1 output of the decoder 187.

The output terminals S1-S16 of the decoder 187 are operatively connected in four groups to the four control lines M1-M4 of the Multimode Time Base Logic 67. The first three output terminals S1-S3 are connected through an OR gate 189 to the input line M1. The fourth output terminal S4 is connected directly to the second input control line M2. The fifth through the 13th output terminals S5-S13 are connected through a second OR gate 190 to the third input line M3, and the last three outputs S14-S16 are connected through a third OR gate 191 to the fourth input line M4.

When the system is performing linear path generation, the binary counter 186 is stepped generally once, or in some cases, twice during each of the last few iterations performed during the generation of the linear path. As a result, the system is caused successively to change its mode of operation through the first, second, third and fourth modes. On the other hand, when the system is generating a circular path, as the end of the path is approached, in one or two iterations the counter 186 is caused by the program gate array 183 rapidly to step through its second through 14th or S2 through S14 stages, thus causing the system to shift from its first mode of operation to its fourth in one or two iterations.

6. PRODUCING A REFERENCE SIGNAL FOR THE SERVO DRIVES

A third function performed by the Time Base and Digital Sweep 63, is to provide a sinusoidal reference signal for the servo drives 75 and 77. By means of a flip-flop 175, a square wave which is in phase with the cycling periods of the digital sweep is generated. The flip-flop 175 may be of the same type as those described in connection with the Divide by 4 circuit 141 and it is so shown. To drive the flip-flop 192 in phase with the digital sweep, its J input is connected to the C4 output of the C decade counter 135 and the "ANDed" C1 and C8 outputs of the same decade counter are applied by means of an AND gate 177 to the K input of the flip-flop. Finally the clock pulse (CP) input of the flip-flop is connected to carry output line 151 of the B decade counter 133. Consequently, in accordance with the operation of the J-K flip-flop as explained previously, the flip-flop 175 is set every time the Digital Sweep Generator 139 changes from its 499th count state to its 500th count state and is reset every time the Digital Sweep Generator changes from its 999th count state to its 000 state. The desired square wave voltage appears at the Q output of the flip-flop 175 and is shown in FIG. 10b as the waveform A to the immediate right of the programming chart. The 500 c.p.s. output of the flip-flop 175 is applied to a reference sine wave shaper 181 and is converted thereby into a 500 c.p.s. sinusoidal reference voltage B which, as seen in FIG. 10b, is in phase with the square wave A. Suitable sine wave shapers are well known in the art and need not be described here.

B. TAPE READER AND BUFFER REGISTERS

Having considered in detail the manner in which various timing signals are produced for the system of FIG. 9 attention will next be directed to the peripheral, input portion of the system whereby information is read from the punched tape 37 and stored temporarily in the buffer registers 47. The XEP BUF register 47 is exemplary of all the buffer registers with exceptions which will be noted in connection with particular ones of registers. It consists of two parts, a six stage shift register for storing six binary coded decimal digits of a number, and a seventh stage for storing the sign of that number. Each digit stage of the shift register is made up of four J-K flip-flops which are identical to the flip-flops 155 and 157 described with reference to the Divide By 4 circuit of FIG. 9b, with the sole difference that the letters Q and $\bar{Q}$ are used in place of the letters T and $\bar{T}$ to denote the outputs. Input signals are applied to the register 47X on a set eight input lines labeled 8, $\bar{8}$, 4, $\bar{4}$,

2, $\bar{2}$, 1 and $\bar{1}$. Successive pairs of eight input lines are applied to the J and K inputs of successive ones of the four flip-flops comprising the lowest order digit position of the register. The two lowest order positions are shown in detail in FIG. 12.

Flip-flops 201, 203, 205 and 207 comprise the lowest order digit stage, and the second stage includes another four flip-flops 209, 211, 213 and 215, each of which receives at its J and K inputs the Q and $\bar{Q}$ outputs of the corresponding flip-flop of the preceding stage. The first two digit stages are exemplary of the other four stages which are not shown. They too contain four flip-flops per stage and the flip-flops comprising each stage are connected to the flip-flops comprising the preceding stage in the same manner in which the second stage flip-flops 209-215 are connected to the first stage flip-flops 201-207.

To enter data into the shift register, groups of binary coded signals are periodically applied to its eight input lines and synchronously with these signals clock pulses are applied at regular intervals over an "X shift" input line 217 X to the CP inputs of all the flip-flops of the shift register. With each shift clock pulse the binary signals on the eight input lines are stored in the lowest order stage of the shift register. With each succeeding clock pulse, the signals are shifted to the next higher order stage and a new group of signals is entered into the lowest order stage of the shift registers until after the sixth clock pulse and the sixth group of signals, the first group of signals has been transferred all the way to the last or highest order stage of the register and the last group of signals has been stored in the first, or lowest order, stage of the register.

In the operation of the system of FIG. 9 it is desirable to transfer simultaneously all of the signals which have been entered serially into the digit stages of the buffer register 47X. For this purpose, the register also includes a group of READ gates. In particular, four READ gates 218 are provided for each digit stage of the register and each of them receives one of the eight outputs of the stage. Thus, a total of 24 READ gates 218 comprise the group of gates for simultaneously transferring or "reading" the contents of the six digit stages of the register 47X onto the computer input trunk line 59.

Each READ gate is comprised of a standard two-input positive logic AND gate and an isolating diode as illustrated in FIG. 12a. As shown in FIG. 12a, a typical two-input READ gate includes a pair of diodes 208 and 210 whose cathodes receive the two logic input signals applied to the gate and whose anodes are connected together and through a resistor 212 to the positive terminal of a voltage source. The output of the AND gate is applied to one of the conductors 59-1 of the trunk 59 through an isolating diode 214. The other end of the trunk line conductor 59-1 is typically connected to the negative terminal of the voltage source through a path within the computer 53, represented in FIG. 12a by the resistor 216.

The READ gate, comprised of the components 208-214 just described and connected to the trunk line conductor 59-1 may be assumed to be one of those shown as part of the buffer register 47X. Several other buffer registers have corresponding READ gates whose outputs are also connected to the trunk conductor 59-1. One of these READ gates is shown in FIG. 12a as being comprised of the resistor 212a, a pair of diodes 208a and 210a, the foregoing three components comprising the AND gate portion of the READ gate, and an isolating diode 214a. The isolating diodes 214 and 214a serve to prevent the interaction of the several READ gates which are part of different registers but whose outputs are connected to the same trunk conductor.

To clarify the correspondence between the symbol used for a READ gate and the actual circuit represented by that symbol, the two READ gates which are drawn in in FIG. 12a in detail are shown in FIG. 12b, with the READ gate symbols being substituted for the individual components of the gates.

To time the simultaneous transfer of data from the buffer register 47X, a gating signal is applied over a line 219X labeled "RD/XEP BUF" to a second input of each of the READ gates 218.

In addition to the six digit stages of the register 47X, it also includes a seventh stage (shown in FIG. 9k to the left of the highest order digit stage) for storing the sign of the number whose digits are stored in the other six register stages. This sign is not shown in the detailed representation of the register in FIG. 12. It comprises a single J-K flip-flop 220X with the J and K inputs of the flip-flops receiving signals respectively representing a negative and a positive sign. To time the entry of the sign signal into the sign flip-flop 220X, a sign store clock pulse is applied to its CP input. And to permit transfer of the sign information stored in the flip-flop 220X along with the digit data stored in the six digit stages of the register 47X, its group of READ gates 218 includes an AND gate connected between an output of the sign store flip-flop and the trunk line 59 and gated concurrently with the other gates in the group from the line 219X.

The foregoing description of the XEP BUF buffer register 47X applies equally to the buffer registers 47Y, 47I, and 47J, each of which also includes a single stage for storing a sign signal and six digit stages for storing a six digit number. Due to the much smaller number to be stored therein, the G BUF register 47G has fewer stages than the other four registers 47X, Y, I and J. From the discussion related to FIG. 4 it will be recalled that the G number comprises three digits, the first designating the letter G and the second and third designating a two digit number. Only the second digit of the number is actually used. Consequently the G BUF register 47G comprises a single digit stage of four flip-flops connected exactly as the flip-flops 201-207 in FIG. 12. The first digit to be entered in the register is simply destroyed when the second number is subsequently entered and this second number is then converted into a control signal by means of a G BUF decoder 221.

The decoder 221 has four control outputs respectively labeled G01, G02, G03, and G02+G03, and is operative to produce a logic 1 signal at those of its outputs which correspond to the number stored in the G BUF register 47G. If the number stored in the register is 1, a logic 1 signal is produced at the G01 output of the decoder and this represents to the rest of the system that the numbers stored in the registers 47X, Y, I and J represent a linear path segment. If the number stored in the register 47G is 2, a logic 1 signal appears on the G02 output of the decoder 221, indicating that the path segment to be generated is circular, progressing clockwise. A counterclockwise progressing circular path segment is indicated by a logic 1 signal on the G03 output of the decoder and this occurs when the number stored in the register 47G is 3. And, if a 2 or a 3 is stored in the register 47G, the G02 + G03 line of the decoder 221 is energized with a logic 1 signal, indicating that the path to be generated is circular but that it may be either clockwise or counterclockwise.

The Q BUF register 47Q differs from the XEP BUF register 47X, first in that it has only 5 digit positions and secondly in that it does not have a sign storing flip-flop. The reason for these differences arises out of the fact that the Q number which is stored in the register 47Q is represented on the tape in "floating point" terminology. According to this terminology a multi digit number is represented by a sign followed by a digit called the "characteristic," which is in turn followed by a series of digits which are collectively called the "mantissa." The characteristic digit serves to indicate where the decimal point should be placed. If the characteristic digit is zero, the decimal point should be placed immediately before the first digit of the mantissa. If the decimal point should be placed after the first digit of the mantissa, the value of the characteristic will be +1. If there should be a zero between the decimal point and the first digit of the mantissa, the characteristic digit will be a -1. Furthermore, if the decimal point should appear after the second, third, or fourth digit of the mantissa, this will be indicated by a characteristic digit of +2, +3, and +4 respectively. Similarly, if there should be 2, 3 or 4 zeros between the decimal point and the first digit of the mantissa, respective ones of these conditions will be indicated by the characteristic which is -2, -3 and -4. Thus the particular

number represented by the indicia on the block illustrated in FIG. 4 is 0.001, meaning that the ratio of the macromove ΔD to the total movement D to be executed in response to the block of data is 0.001 and that, therefore, it will require 1,000 macromoves to complete the path segment involved. For practical purposes the value of the Q number will always be less than 0.1 and, therefore, the sign of the characteristic of the Q number will always be negative, and the Q BUF register 47Q need not have a sign store flip-flop.

To provide individual control over the stepping of data in different ones of the buffer registers 47, each of the registers is provided with an individual shift control line for supplying clock pulses to associated shift registers. In addition to the X shift clock pulse line 217X associated with the register 47X, five other clock pulse lines 217Y, 217I, 217J, 217G, and 217Q are associated with respective ones of the registers 47Y, I, J, G and Q. Each of the six clock pulse lines 217 receives clock pulses through an individual one of six clock pulse control AND gates 222X, Y, I, J, G, Q from a common shift clock pulse line 223 connected to a first input of each of the AND gates. Means are provided individually to enable each of the AND gates 222 thereby passing shift clock pulses only to a selected one of the registers 47. The clock pulses are derived directly from the sprocket holes on the punched tape 37, thus insuring that they will be in synchronism with the numbers being read from the tape. Specifically, the tape reader 41 includes a nine channel photoelectric hole detector, eight channels for the eight information channels on the tape 37 and a ninth channel for the sprocket holes of the tape. As each row of holes is transported past the detector, it produces data channel pulses on eight output lines, shown collectively in FIG. 91 as a trunk line 225, and a timing pulse on a ninth output line 227. Since the sprocket hole is smaller than the data holes, the timing pulse produced in response thereto is shorter in duration than the data channel pulses as indicated by the waveforms to the left of the tape reader 41. This assures that any shifting operation initiated by the clock pulses will occur between the leading and lagging edges of the data channel pulses.

The sprocket hole pulses are applied to the shift clock pulse line 223 through an AND gate 229 which is part of a logic gating arrangement for permitting only those pulses that are derived from sprocket holes in digit data rows to reach the line 223.

The data channel pulses of the tape reader 41 are applied over the tape reader output trunk line 225 to the decoder 43 which is shown in block form only, since decoders are well known to persons skilled in the art of data processing. It need only be noted that the decoder 43 is provided with eight inputs connected to receive the signals on the trunk line 225 and with a first group of eight outputs labeled 8, $\bar{8}$, 4, $\bar{4}$, 2, $\bar{2}$, and 1, $\bar{1}$. As row after row of the indicia on the tape 37 are read by the tape reader 41, the decoder 43 is operative to produce at its first group of eight outputs electrical signals representing each row of indicia in binary coded decimal form. The aforesaid eight outputs comprise the decoder output trunk line 51 noted previously with reference to FIG. 7 and are connected to the similarly labeled eight inputs of each of the buffer registers 47.

A principal function of the decoder 43 is to generate over individual ones of a second group of output lines 231 signals in response to those rows of indicia on the tape 37 which represent the characters X, Y, I, J, G and Q. Thus by means of appropriate logic gating devices signals are produced on successive ones of six output lines 231X, 231Y, 231I, 231J, 231G, and 231Q of the decoder 43 as respective ones of the data rows representing the characters I, J, X, Y, Q, and G are sensed by the tape reader 41. At a seventh output 232, marked EL, the decoder also produces a signal when the last, or EL row of block has been read. The signals which are produced at the output lines 231 are called the address signals and are used to control the flow of clock pulses to that one of the buffer registers which is to receive the particular number being read from the tape. Toward this end six J-K flip-flops 233 are provided, one for each of the six buffer registers 47. Each of the

six address output lines 231 of the decoder 43 is connected through a common OR gate 235 to the K inputs of all of the flip-flops 233 so that all of the flip-flops are reset each time an address character is read from the tape. Furthermore, respective ones of the six address output lines 231 are individually connected directly to the J input of a different one of the six flip-flops 233 so that, as successive ones of the address characters X, Y, I, J, G, and Q are decoded, successive ones of the flip-flops 233X, Y, I, J, G and Q receive a signal at their J inputs. A second input of the AND gate 237 is connected to the output of the OR gate 235 so that each time an address character is read from the tape and an address signal is produced on one of the six lines 231, a sprocket hole pulse is gated through the AND gate 237 to the CP inputs of all of the flip-flops 233.

As the tape 37 is transported through the tape reader 41, each time an address row of indicia is read and decoded, all six of the flip-flops 233 receive a pulse at their K inputs and also at their CP inputs. Furthermore, one of the flip-flops will also receive a pulse on its J input directly from one of the six address outputs 231 of the decoder 43. Only those five flip-flops which receive signals at their CP and K inputs will be reset, and the sixth flip-flop, receiving signals at all three of its inputs, will be caused to reverse its state from a reset to a set condition. Consequently, as each address row on the tape is read and decoded, a different one of the flip-flops 233 will be set. Thus in response to the X address row, containing holes in channels 1, 2, 3, 5 and 6, a signal will be produced by the decoder 43 on its output line 231X and this signal will be fed to the J input of the flip-flop 233X causing that flip-flop to be the only one whose state is reversed and leaving the flip-flop in its set state. As a result, a logic 1 signal appears at the Q output of the flip-flop. This output, labeled the "X ADR" (X address) is connected by a line 239X to the second input of the shift clock pulse control AND gate 222X. The logic 1 signal thereon permits shift clock pulses to pass from the line 223 onto the X shift clock pulse line 217X.

The Q outputs of the other five flip-flops 233Y, I, J, G, and Q are similarly connected by lines 239Y, I, J, G, Q to supply enabling signals to the AND gates 222Y, I, J, G, and Q through which sprocket hole pulses are fed to the other five buffer registers 47Y, I, J, G, and Q.

The next row of indicia after the X address row represents a minus sign and is presented by the decoder 43 on the second of a pair of special "+" and "-" output lines 241 and 243. The + and lines 241 and 243 are connected to the J and K inputs of the sign flip-flops of the buffer registers 47X, Y, I, J. The lines 241 and 243 are also connected to the inputs of an OR gate 245 whose output is connected to one input of an AND gate 247. The other input of the AND gate 247 receives sprocket hole pulses through the line 227 from the tape reader 41 and the output of the AND gate is connected over a line 249 to one input of each of four AND gates 251X, 251Y, 251I, and 251J. Respective ones of the AND gates 251X, Y, I, J, have their outputs connected to the CP inputs of the sign store flip-flops 220X, Y, I, J and their second inputs connected to the Q outputs of respective ones of the flip-flops 233X, Y, I, J. By means of the gates 245 and 247, sprocket hole pulses are passed to the inputs of the AND gate 251 only when a sign signal is present on one of the decoder output lines 241 and 243, and by means of the AND gate 251, the sprocket hole pulses are applied to the sign store flip-flop 220 of that register which is about to receive the digits associated with the sign signal. Thus the sign signal on the lines 241 or 243 will be entered in the proper sign store flip-flop 220.

To prevent shift clock pulses from reaching the register 47X before the first X digit has been read from the tape, the AND gate 229 which controls the supply of sprocket hole pulses to the clock pulse line 223 is disabled while the X sign is read from the tape and is enabled only when a digit is being read from the tape. For this purpose, the output of the OR gate 235 is fed to an OR gate 253 whose output, in turn, is applied through an inverter 255 to a second input of the AND gate

229. As a result, whenever the OR gate 235 receives a logic 1 signal from one of the address lines 231 of the decoder 243, that signal is applied through the OR gate 253 to, and is inverted by, the inverter 255 to be applied as a logic 0 signal to the AND gate 229. A sprocket hole pulse is thus prevented from being passed through the AND gate 229 whenever the data being read from the tape is an address row.

Further, to prevent a sprocket hole pulse from passing through the AND gate 229 when a sign signal or an EL row is being read, the output of the AND gate 245 is connected to a second input of the OR gate 253, and the EL output 232 of the decoder 43 is connected to an input of the OR gate 235.

Following the reading, decoding and storing of the sign of the X coordinate number, the next row of indicia, representing the first digit of the X coordinate, a 0, is read and decoded. Since the row represents a digit, none of the input lines to the OR gates 235 or 245 carries a logic 1 signal so that a logic 0 voltage level appears at the output of the OR gate 253. This signal level is inverted by the inverter 255 and is applied as a logic 1 signal level to the AND gate 229, enabling it. As a result, a sprocket hole pulse is passed through the AND gate 229 onto the digit store clock pulse line 223. This sprocket hole pulse is gated through the AND gate 222X by the address signal from the address flip-flop 233X, causing the data appearing on the digit output lines of the decoder 43 to be shifted into the lowest order digit stage of the register 47X. As successive digit rows are read, five more sprocket hole pulses are applied through the AND gate 222X and over the line 217X as shift clock pulses to the register 47X, causing the other five digits of the X coordinate to be shifted into the register. After the last shifting operation, the first read digit will be stored in the highest order stage of the register and the last read digit will be stored in the lowest order stage. In the same manner, the sign and digit rows of the Y, I, and J numbers are stored in the registers 47Y, 47I, and 47J.

The storage of the G and Q numbers in the registers 47G and 47Q differs from the foregoing process only in that neither of the G and Q numbers needs a sign. Consequently, the step associated with the reading and storing of the sign of the number is omitted.

When the last digit of the G number has been stored in the G BUF register 47G, the tape reader 41 is stopped until the next block of data is to be transferred from the tape 37 to the buffer registers 47. The stopping and starting of the tape reader is synchronized with the remainder of the system of FIG. 9 by means of the tape reader start control 45, principally comprising a J-K flip-flop 257. The tape reader 41 is provided with a RUN input 259 connected to the Q output of the flip-flop 257 and is operative to begin transporting the tape 37 and reading the data thereon when a logic 1 voltage level is applied to the RUN input by the flip-flop. The clock pulse input (CP) of the flip-flop 257 is connected to the output of an OR gate 261 and one of the inputs of the OR gate is, in turn, connected to the output of an AND gate 263. One input of the AND gate 263 and the K input of the flip-flop 257 are both connected to the EL output of the decoder 43 through a line 265. The other input of the AND gate 263 is connected to the sprocket hole pulse line 227. Consequently, assuming that the flip-flop 257 was set to start the tape reader 41 at the start of the block, when the last row of data in the block, representing the end of line symbol EL is read by the tape reader 41, a logic 1 signal is applied to the K input of the flip-flop and shortly thereafter a sprocket hole pulse is gated through the AND gate 263 and the OR gate 261 to the clock pulse input of the flip-flop. As soon as the EL symbol has been read, the flip-flop 257 is reset and applies a logic 0 voltage level to the RUN input 259 of the tape reader 41, causing the tape reader to stop just short of reading the first row of data in the following block.

The OR gate 261 receives at a second input the output of an AND gate 267. One input of the AND gate 267 is connected to the PCP output of the decoder 173 in FIG. 9b and the other input of the AND gate is connected to the output of a program gate 269 which produces a program pulse on program step

4,995. Program step 4,995 marks the time when the data in the active registers 57 has served its purpose in generating the path segment derived therefrom and occurs after the data from the buffer registers 47 has been transferred into the active registers 57 for generating the next path segment. The output of the program gate 269 is also connected directly to the J input of the start control flip-flop 257. Therefore, during program step 4995, a logic 1 voltage level is applied to the J input of the flip-flop 257 and during this same program step a clock pulse is applied to the CP input of the flip-flop so that it is set and applies a logic 1 voltage level to the RUN input 259 of the tape reader 41, causing the tape reader to start transporting the tape 37.

C. THE ACTIVE REGISTERS

In contrast to the serial transfer of data from the decoder 43 to the buffer registers 47, the flow of data between the buffer registers 47 or the active registers 57 and the digital computer 53 is characterized by parallel, or simultaneous transfer. Generally, as one of the first operations in the process performed by the system of FIG. 9, the data stored in respective ones of the six buffer registers 47 is read in parallel into the computer 53 on six different program steps by enabling the READ gates of the respective buffer registers. The data so placed on the computer input trunk line (CIT) 59 is temporarily stored in the computer 53 without removing it from the register from which it was read and is then put by the computer on the computer output trunk line (COT) 61 during the following program step to be stored in a designated one of the active registers 57. The trunks 59 and 61 each contain four conductors for each digit of the number to be sent over them and an additional line for the sign of the number. The exemplary system has a capacity for processing numbers with up to seven digits and therefore each of the trunks 59 and 61 includes a total of 29 conductors. FIG. 13 shows the trunks 59 and 61 in greater detail, illustrating the four conductors associated with the lowest order digit to be carried on the output trunk 61 as the conductors 61-1, 61-2, 61-3, and 61-4. Similarly, the four conductors associated with the lowest order digit to be carried from the active registers 57 to the computer 53 are shown as the conductors 59-1, 59-2, 59-3, and 59-4. Also shown in FIG. 13 is the lowest order digit stage of the ΔX register 109. The register stage illustrated in FIG. 13 is representative of all of the other six digit stages of the register 109. The sign stage of the register is similar to one quarter of the digit stage. Thus, the ΔX register has a total of 17 flip-flops, and is representative of all of the other active storage registers, except that some of them have more digit stages, others have fewer digit stages, and others yet do not have a sign stage.

Referring to FIG. 13, the lowest order digit stage of the ΔX register includes four R-S flip-flops 281, 283, 285, and 287. There is no provision for shifting the data stored in the flip-flops 281-287 to a higher stage. Rather, data is transmitted over the four trunk line conductors 61-1, -2, -3, and -4 to the flip-flops 281-287 and is subsequently sent from the flip-flops over the four trunk line conductors 59-1, -2, -3, and -4. To permit simultaneous transmission of signals over the trunk line 61 a group of 17 inverters and 17 pairs of AND gates, which collectively comprise the WRITE gates associated with the register, are provided. Those of the inverters and AND gates that are associated with the lowest order digit stage of the register are shown in FIG. 13 as the inverters 288 and the AND gates 289 connected between the trunk line conductors 61-1, -2, -3, and -4 and the R-S inputs of the flip-flops 281-287. One input of four of the AND gates 289 is connected to a respective one of the conductors 61-1, -2, -3, and -4 and the output of each of these four AND gates is connected to the S input of a different one of the four flip-flops 281-287. The second group of four AND gates are connected to the outputs of the four INVERTERS 288, which in turn are connected to respective ones of the lines 61-1, 61-2, 61-3, 61-4.

The second group of four AND gates have their outputs connected to respective ones of the R inputs of the four flip-flops 281-287. With the common input to the eight AND gates 289 held at logic "0," the four R-S flip-flops 281-287 are held in their existing state. When a number is to be written into the flip-flops of the ΔX register, the line 291 is brought to the logic "1" level by appropriate program gating. At the appropriate time within the program step thus selected, the formerly blocked WP pulse is allowed to pass through the AND gate 290 to the common input of the 8 AND gates 289. When this occurs, the common inputs are enabled, and the R-S flip-flops 281 through 287 acquire the data from lines 61-1, 61-2, 61-3, and 61-4. A logic "1" presented at the S input sets an R-S flip-flop; a logic "1" presented at the R input resets the R-S flip-flop. When a logic "0" is presented on both inputs, the flip-flop does not change. The combination of logic "1" on both inputs is prevented from occurring by INVERTERS 288.

To transfer a stored number from the register 109 to the computer 53, a group of seventeen READ gates, one for each of the seventeen flip-flops of the register 109, is provided. Each READ gate is a coincidence responsive logic device, four of which are shown as the READ gates 295, each having one input connected to the Q output of a respective one of the four flip-flops 281-287 and all of them having a second input connected to a common READ control line 297. The outputs of the four READ gates 295 are individually connected to respective ones of the trunk line conductors 59-1, -2, -3, and -4 so that, when a READ control signal (R/ ΔX) is applied over the conductor 297 to the READ gates 295 the information stored in the four flip-flops 281-287 is simultaneously applied via the READ gates to the aforesaid conductors.

D. AN EXEMPLARY TIME SHARED DIGITAL COMPUTER

The destination as well as the source of almost all of the signals stored in the active registers 57 is the time shared digital computer 53. Such computers are well known and the discussion of their details would overburden this description. As a general example, reference may be made to Chapter 21 in the textbook "Digital Computer Principles" by Burroughs Corporation published in 1962 by McGraw-Hill Book Company, Inc. Because only ordinary skill in the art is required to organize the arithmetic computer, and the latter may take a variety of specific forms, there will only be here defined, in detail, the functions that are performed by the computer 53 and the times during which those functions are performed.

To satisfy the varied demands of the system of FIG. 9 in its different modes of operation, the computer 53 is capable of performing several arithmetic operations such as algebraic addition and subtraction, and multiplication. The basic operating mode of the computer 53 is that of addition and all other operating modes are brought about by special control signals. The computer receives its inputs from the computer input trunk line 59 over eight branches of four conductors each, respective branches being labeled in FIG. 9f as CIT-1 through CIT-8 and representing successively lower order digits of a number. In addition to the 32 conductors in the eight branches, a 33rd input conductor CIT (-) is provided for carrying the sign of the number signaled on the other conductors. The output of the computer 53 is stored in an Answer Register 53a which includes stages one through eight corresponding to the eight numerical digits received by the computer over the trunk line 59 and a sign stage corresponding to the sign signal received by the computer.

Each of the eight digit stages includes means for storing four binary signals representing a decimal digit and these may be four J-K flip-flops such as those used in the buffer storage registers in the system. The sign stage includes means for storing a single binary signal representing the sign of the number stored in the other stages of the register and may comprise a single J-K flip-flop. The numbers stored in the Answer Register 53a appear on the conductors of the computer output

trunk 61 which are connected to the Q outputs of the respective flipflops comprising the register through eight output branches COT-1 through COT-8 and a ninth output line COT (-). Each of the eight branches COT-1 through COT-8 has four conductors, and respective ones of the groups are connected to successive ones of the stages 1 through 8 of the Answer Register 53a. The single COT (-) line is connected to the single flip-flop comprising the sign stage of the Answer Register.

A digit signaled on a particular one of the input line branches CIT-1 through CIT-8 is processed in the corresponding stage of the Computer Register 53a. Thus, a digit signaled on branch CIT-5 will be processed in stage 5 of the Computer 53a. In performing addition, the computer 53 adds to each digit stored in the Answer Register 53a that digit of the following number which is signaled on the corresponding branch of the trunk line 59. Each decade of the computer, except the last, also includes means for carrying any excess over 9 of the sum developed therein to the next higher order stage. Thus, for instance, if stored augend of +346 is stored in stages 6, 7, and 8 of the Answer Register 53a and subsequently an addend of +778 is applied over branches CIT-6, -7, and -8 to the computer, the result of +1,124 will appear in stages 5, 6, 7, and 8 of the Answer Register.

An important attribute of the computer 53 is that it is time-shared so as to perform many different operations upon data received by it from diverse sources in the system. Toward this end the computer is synchronized with the rest of the system. The relationship between the timing of the computer 53 and the rest of the system of FIG. 9 will be best understood by recalling the concept of program steps explained with reference to FIGS. 10 and 11. As explained there, certain operations of the system may be timed to occur during selected ones of a plurality of time periods, called program steps, each of which is 20 microseconds long. For example, a number stored in a particular active storage register may be caused to be transferred to the computer input trunk line 59 in response to a gating signal applied to the READ gates of that register during a particular one of the program steps. The point to observe is that, in the exemplary system the time during which such a gating signal is being applied to the READ gates of the program storage register coincides with the program step involved.

Each operation of the computer 53 is also performed in response to a control signal produced during a particular program step, but the operating cycle of the particular computer which is illustrated herein does not coincide with that program step. Instead, the operation begins during the middle of the program step in which that operation was initiated and ends at the middle of the following program step. The particular operation of the computer to which reference is being made is that of addition. Thus, the computer is so timed that, in response to a control signal during a given program step, it commences its operation for performing addition, called the Adder Cycle, halfway through that program step and terminates that adder cycle halfway through the following program step. Since the time periods during which numbers from various ones of the buffer and active storage registers 47 and 57 are gated onto the trunk line 59 coincide with the various program steps, the adder cycles are staggered relative to the time periods during which data is supplied to the computer 53. This staggered time relationship is illustrated in FIG. 14. Shown against a time scale extending from 0 to 60 microseconds are three program steps N, N+1, and N+2 represented by a continuous row of three blocks 311, 313, and 315. The adder cycles N and N+1 corresponding to the first two program steps are shown staggered therewith, as the second row of blocks 317 and 319.

An example will serve to illustrate how the exemplary computer 53 would add two numbers stored in different registers of the system. Let it be assumed that a number stored in the ΔX register 109 is added to the number stored in XCP register 117. During program step N the XCP number is read from the

register 117 over the trunk line 59 into the digital computer 53 over its input branches CIT-2 through CIT-8. During the same program step the computer is instructed by means of a control signal over its CAR control input to clear its Answer Register 53a and its Overhanging Register 53b of any number that might have been stored in it. These events are indicated in FIG. 14 by the blocks 321 and 323. Addition begins during the adder cycle N and is represented in FIG. 14 by the block 325.

During this period the Answer Register 53a is effectively "cleared" by supplying zeroes at the augend input terminals of the computer 53 in place of the numbers in the Answer Register which are normally used as the augend, while supplying the XCP number at the addend input terminals of the computer from the trunk line 59. Since the Answer Register 53a has thus been made to appear empty, the result of the first adder cycle N is that only the XCP number will be stored therein. The above operation for causing the Answer Register to appear empty will be referred to throughout this specification simply as "clearing the Answer Register." Furthermore, since the result of "clearing" the Answer Register is to prevent the number which had been stored in it from being, in effect, re-entered in the same register as the augend of the next addition, clearing the Answer Register, in effect, "removes" the number which had been stored in it, and for this reason throughout the specification the step of clearing the Answer Register 53a, or the larger Accumulator Register 53ab, will be referred to as "removing" a number from those registers.

During the next program step N+1 the sign and four digits of the ΔX number are read onto branches CIT (-), and CIT-5, -6, -7, and -8 of the trunk line 59 and through them to the computer 53. This operation is shown to occur during program step N+1 and is represented by the block 327.

To complete the addition, the computer 53 is signaled halfway through the program step N+1 to commence its next adder cycle N+1. In its normal, "add mode" operation within the system, the computer receives such a signal during each program step. In response to the signal indicating commencement of the adder cycle N+1, and with the computer operating in the ADD mode, the number signaled on the trunk line 59 is algebraically added by the computer to the previously stored XCP number and the resulting sum is substituted in place of the previously stored XCP number in the Answer Register 53a, as indicated by the block 329. In the actual operation of the system of FIG. 9, after the addition just described has been performed, the sum is substituted for the number previously stored in the XCP register 117. This occurs during the second half of the program step N+2, as indicated by the block 331.

In the foregoing example the seven digits of the XCP number were entered into the computer 53 over its input line branches CIT-2 through CIT-8 and the four digits of the ΔX number were then entered over the input line branches CIT-5 through CIT-8. Entry of the ΔX number into the computer caused it to be added immediately so as to produce the correct sum of the two numbers in the Answer Register 53a. The reason why the addition was performed so directly is that the digits stored in corresponding stages of the registers 117 and 109 from which the numbers XCP and ΔX were taken have the same significance. In the case of the XCP register 117, the significance of the digits residing in its various stages can be readily seen in FIG. 9g by observing that a decimal point has been placed between the second and third highest order digit positions. Thus, the digit stored in the highest order stage of the register 117 in the exemplary system represents tens of inches. The next lower order stage represents units inches and successively lower order stages to the right of the decimal point represent tenths, hundredths, thousandths, 10 thousandths, and 100 thousandths inches. Since the ΔX register 109 has only four digit stages in the exemplary system, its highest order stage represents hundredths of inches, so that the largest possible ΔX number that can be stored therein is 0.09999 or approximately 0.1 inches. This simply means that all macromove numbers are assumed to be less than 0.1 inches

when the time period ΔT during which such a macromove is carried out is 20 milliseconds. This is a reasonable assumption since an excursion of 0.1 inches during 20 milliseconds would be the equivalent of 5 inches per second or 300 inches per minute.

Unlike the register 117 the ΔX register 109 does not carry an indication of the decimal point, because its highest order digit stage is one place removed therefrom. Instead, this information, as well as the placement of the decimal point and the digital significance of numbers in all of the other registers, is shown in FIG. 15. The registers of the system are shown in FIG. 15 as stacked upon one another, with each register being represented as a row of squares and identified by appropriate symbols to its left. FIG. 15 readily shows that the decimal points of the numbers XCP and ΔX coincide and that they may be added to one another in the manner explained hereinabove. This is not always the case, however, since FIG. 15 shows that not all of the numbers have their decimal points in the same place. When two numbers with differently placed decimal points are to be added, the first number is usually shifted in the Accumulator Register 53ab to bring its decimal point into alignment with that of the second number and only then is the addition performed.

So long as the computer continues to operate in the ADD mode, every time a number is read from one of the registers 57 onto the computer input trunk line 59 during a program step, that number will be automatically added during the following adder cycle to the sum previously stored in the Answer Register 53a. The addition operation is usually terminated by enabling the WRITE gates of a selected one of the active registers 57 during the particular program step in which the desired sum is available in the Answer Register 53a.

If it is desired that the computer 53 subtract a number signaled on the input trunk line 59 from a number stored in its Answer Register, a control signal is applied to its SX input concurrently with the reading onto the trunk line 59 of the number that is to be subtracted.

The computer is also capable of performing multiplication, this operation being initiated by a control signal on its X control input. This operation may be assumed to involve a series of successive additions of the multiplicand by a number which is equal to the least significant digit of the mantissa of the multiplier (which is Q, represented in characteristic mantissa form, see section G1(b)), followed by a shift of the partial sum stored in the Answer Register 53a one position to the right, then another series of successive additions of the multiplicand a number of times equal to the second least significant digit of the mantissa of the multiplier, followed by another shift to the right of the partial sum in the Answer Register 53a, and so on until the multiplicand has been repeatedly added a number of times equal to the most significant digit of the mantissa of the multiplier. At this time the sum in the Answer Register 53a is shifted to the left a number of positions which is equal to the characteristic of the multiplier. It can be shown that the product will then be located correctly in the Answer Register 53a.

To store the digits which are shifted toward the right from stage 8 of the Answer Register 53a an "overhanging" register 53b is provided. This register is shown as including four digit stages and forms a continuation of the Answer Register 53a. Indeed, the Answer Register 53a and the overhanging register 53b will be considered as a single 12 stage accumulator register 53ab. However, only the first eight stages of this 12 stage register receive sum data from the computer 53 and only those eight stages feed the sum data to the output trunk line 61. It is evident that the Accumulator Register 53ab has means for shifting its contents both toward the right and toward the left. The shifting operations performed during multiplication are internally initiated by the computer and these are not individually described herein. During certain other operations carried out by the computer, however, it is instructed to shift a number stored in its Accumulator Register 53ab to the right and also to the left and on occasion by four positions during a

single adder cycle. These shifting operations are externally initiated by application of control signals during designated program steps to the SAR (shift accumulator right), SAL (shift accumulator left), SR4 (shift right 4), and SL4 (shift left four) control inputs of the computer. The timing of such control signals and shifting operations initiated thereby is shown by block 333 and waveforms 335 and 337 (FIG. 14).

Provision is also made in the computer 53 for selectively clearing either the Answer Register 53a or the Overhanging Register 53b. Such selective clearing operations are performed by the computer in response to control signals on the ZR1 (zero register portion 1) and the ZR2 (zero register portion 2) control inputs.

Another operation performed by the computer 53 is that of comparing two numbers and signaling on a pair of outputs 339 and 341 the relative magnitudes of the compared numbers. In carrying out this operation, the first of the two numbers to be compared is read onto the trunk line 59 during program step N and the second of the two numbers is read onto the trunk line during program step N+1, concurrently with a "classify" control signal (block 343) which is applied to the CX control input of the computer 53. The result of the comparison operation appears on one of the two outputs 339 and 341 in time to be used during the following program step (block 345). If the first of the two numbers is less than the second, a logic 1 signal appears on the output line 341. If the first number is equal to or greater than the second, this condition is indicated by a logic 1 signal on the output line 339.

The last mode of operation performed by the computer 53 is referred to as "force positive." In this mode of operation the computer treats the number which is entered into it as a positive number, regardless of what its sign is. To cause the computer to operate in this mode, a control signal is applied to the FP control input of the computer concurrently with the reading of the second number onto the trunk line 59.

E. PATH TYPE INDICATOR FOR THE CURRENT PATH SEGMENT

In describing the buffer registers 47 in FIG. 9k reference was made to the G register 47G and to the G decoder 221. It was stated that the decoder 221 is operative to produce at its outputs a signal indicative of the type of path represented by the signals stored in the other buffer registers 47. This information is useful in the present system because it can be used to control the type of operations to be performed on the data stored in the other buffer registers, in accordance with the shape of the path segment represented by them. Such a path indicative signal is also useful during the continuous processing of data in the active registers 57 which occurs while the path is being executed, that is, while the controlled machine element is being moved along that path. It is for this reason that the G number residing in the buffer register 47G is transferred along with the numbers in the other buffer registers 47 to the active registers 57 and in particular to the path mode, or G, register 105.

The G register 105 appears immediately to the right of the computer 53 in FIG. 9f, and is comprised of two R-S flip-flops, each for storing a binary signal. When the G number is transmitted from the G BUF register 47G to the computer 53, it is carried to the computer on the CIT-8 branch of the trunk line 59 and, in particular, on the one and two bit lines of that branch. When the number is subsequently sent from the computer to the G register 105, it is carried on the one and two bit lines of the COT-8 branch of the output trunk line 61 to the WRITE gates associated with the flip-flops of the register 105. In this manner, the digitally signaled G number, whether it be 1, 2, or 3, is transferred from the G BUF register 47G to the G active register 105. To convert the binary coded representation of the G number in the active register 105 into a single line representation, a decoder 106, which is similar to the G BUF decoder 221 is provided. The decoder 106 is shown in detail in FIG. 9f and includes an array of three AND gates

106a, 106b, and 106c, connected to one another and through a trunk line 108 to the outputs of the G register 105 so that a logic 1 voltage level appears at the output of respective ones of the AND gates 106a, 106c and 106b, in response to a logic 1, 2, and 3 stored in the register. The decoder also includes an OR gate 106d connected to the outputs of the AND gates 106b and 106c so that a logic 1 signal appears at its output when either of the AND gates 106b and 106c is enabled. Since G numbers of 1, 2, and 3 signify paths which are linear, clockwise circular, and counterclockwise circular respectively, logic 1 signals on the outputs of the AND gates 106a, 106c, and 106b are respectively designated as LIN, CW, and CCW. Finally, a logic 1 signal at the output of the OR gate 106d is designated CIRC, meaning that a logic 1 signal at the output of the OR gate 106d is used to initiate operations which are to be performed whenever a circular path is to be generated, be it clockwise or counterclockwise.

F. THE PROGRAM GATE ARRAY (PGA)

The type of operation to be performed by the computer 53 and the particular program step during which this is to occur depends on several factors. Among these is the type of path to be generated and the stage to which the generation of that path has progressed. Both of these factors are detected by means of a large number of program gates, each of which is connected to respond to the particular condition that is to initiate a particular computer operation. In most cases, a given type of computer operation will be required to be performed in response to a unique combination of several different conditions, each combination being detected by an individual program gate. All of the program gates associated with a given computer operation, such as shifting four places, are collected in a group, and are referred to as a Program Gate Array, or PGA for short. All of the PGA's associated with the computer 53 are shown in FIG. 9c and one of these, the PGA 347 for initiating the shift left four (SL4) operation, is also shown in detail in FIG. 16. For reasons which will become apparent during the description to follow, the computer will be instructed to shift the contents of its Accumulator Register 53ab four positions to the left during program steps 1465 and 1470 when generating a circular path, program steps 1865 and 1870 when generating a linear path, and during program steps 1965, 1970, 4870, and 4970 for both a linear path and a circular path. In response to each of the foregoing eight conditions, therefore, a control signal is produced at the output of the Program Gate Array 347. Each of the eight control signals produced under the eight different conditions is derived by means of a different one of eight program gates 347a, b, c, d, e, f, g, and h. Taking the first program gate 347a as an example, it is an AND gate having four inputs, the first three being connected to the 1400 output of the Multimode Time Base Logic 67 and to the 70 and 0 outputs of the Time Base and Digital Sweep 63. The fourth input of the program gate 347a is connected to the CIRC output of the G decoder 106 so that, when the path being generated is circular, regardless of whether it is clockwise or counterclockwise, a pulse is produced during program step 1470 at the output of the program gate.

The second program gate 347b is similarly connected to produce a program pulse at its output during program step 1870 when a LIN signal is produced by the G decoder 106. The remaining six program gates 347c-h are connected in a similar manner to appropriate outputs of the Multimode Time Base Logic 67, the Time Base and Digital Sweep 63 and the Path Type indicator 106 to produce at their respective outputs program pulses during program steps 1970, 4870, 4970, 1465-CIRC, 1865-LIN, and 1965. By means of an OR gate 347i the outputs of all of the five program gates 347a-h are collected and made to appear at the output of the OR gate. This single output constitutes the output of the entire Program Gate Array 347.

Program gate arrays are also used to time the reading of numbers into the computer trunk line 59 from, and the writing

of numbers from the computer output trunk line 61 into the active registers 57. Each register is seen to have next to it two such PGA's, one connected to the WRITE gates and the other to the READ gates of the register. The different conditions under which a signal is produced by a particular PGA are listed therein in the same manner as done for the PGA's shown in FIG. 9c.

2. OPERATION OF THE SYSTEM IN GENERATING A LINEAR PATH

a. Steps Performed Prior to the Execution of the First Macromove Along a Linear Path to Generate ΔX and ΔY Numbers

In describing the process of linear path generation, and subsequently the process of circular path generation, reference will be made to the hypothetical path 27 shown in FIG. 2. Thus, the description will proceed on the assumption that the path 27 is being generated, and will begin with a point on the path just short of the end of the path segment 27a. The description will then follow the generation of the linear path segment 27b and subsequently the generation of the circular path segment 27c. Interlaced with the description of the linear and circular path generating process will be a description of the means provided for carrying out the processes, that is, the system of FIG. 9. By the nature of the processes to be described, each of them involves a very large number of steps, some of which are related to optional features which are not essential to the basic processes themselves. The steps performed by the apparatus in carrying out linear and circular path generation will be described, with a few exceptions, in chronological order, that is, in the order in which they occur in time. This will permit an orderly presentation of the operation of the exemplary system. However, in many cases, steps which are directed to carrying out a particular feature do not follow one another, and the relationship between these steps will not be apparent. Therefore, periodically as the description proceeds, the various features and sub-processes which have been carried out will be summarized.

Beginning with steps associated with generating the linear path segment 27b in FIG. 2, let it be assumed that the system is in the process of completing the previous path segment 27a, and that it is presently executing the last macromove associated with that path segment. Some of the events taking place during the time period are shown in FIG. 17 which is a flow chart of operations performed by the system of FIG. 9 during a complete iteration or system cycle when the system is operating in mode 4. The flow chart is, in effect, a programming chart such as those shown in FIGS. 10a and 10c. It has 10 columns numbered 4000 through 4900 and each column is understood to have a hundred 20 microsecond time periods, during each of which a different operation may be performed. However, rather than attempt to show each single step performed during the iteration represented by the flow chart in FIG. 17, these steps are identified only by the operation toward which they are directed.

Reviewing the programming chart concept briefly, it will be recalled that each iteration or machine cycle represented by a programming chart such as that shown in generalized form in FIG. 10a, takes 20 milliseconds, this being the cycling time of the Program Step Generator 161. It will be understood of course, that this iteration time is entirely optional and has been chosen only for sake of a specific example.

The steps and operations carried out during the first six columns of the FIG. 17 chart relate entirely to the execution of the previous path segment 27a. They will become of interest when the termination of a linear path is considered and will be discussed at that time. The operations related to the generation of the next straight line path segment 27b begins during the seventh column, numbered 4600, of the chart.

1. TRANSFERRING OF DATA REPRESENTING NEXT PATH SEGMENT FROM BUFFER REGISTERS TO ACTIVE REGISTERS

The initial operations carried out during the 4600 column are directed to transferring the numbers associated with the next block segment 27b from the buffer registers 47, where they had been stored up to this time, to the active registers 57. First, the I and then the J number is transferred and then the XEP and the YEP numbers are transferred from the buffer registers 47X and 47Y to their designated active registers 57. Next, the Q number is transferred from the buffer register 47Q to the Q active register 103 and, finally, the G number is transferred from the buffer register 47G to its designated active register.

The foregoing series of operations, whereby numbers are transferred "from buffer to active," are shown in detail, step by step, in Table I.

TABLE I
Steps performed in Mode 4 (Part 1)

Finishing linear block		Finishing circular block	
4,600 . LIN	Comments	4,600 . CIRC	Comments
1 to 14 g See Table III			
15. CAR		CAR	Compute X and Y overrun correction and modify I and accordingly when transferring J from buffer to active
16. R/IBUF	Transfer I and J from buffer to active registers	R/XCEP	
17. W/I		SAL	
19. CAR		R/XCP, SX	
20g. CAR		R/IBUF	
21. R/IBUF		W/I	
22g. W/J		CAR	
23. CAR	Transfer programmed end point from buffer to active, translating it into machine coordinates	R/YCEP	
24. R/IBUF		SAL	
25. W/J		R/YCP, SX	
		R/IBUF	
26. CAR		W/J	Transfer programmed end point from buffer to active, translating it into machine coordinates
27. R/XEPBUF	Transfer programmed end point from buffer to active, translating it into machine coordinates	CAR	
28. R/50.0000"		R/XEPBUF	
29. R/XOFF		R/50.0000"	
29. W/XCEP		R/XOFF	
CAR		W/XCEP	
30. R/YEPBUF		CAR	
31. R/50.0000"		R/YEPBUF	
32. R/YOFF		R/50.0000"	
33. W/YCEP		R/YOFF	
34. CAR		W/YCEP	
35. R/QBUF		CAR	
36. W/Q		R/QBUF	
37 to 99.	No further computations		

This table includes two columns, the first labeled 4600 LIN and related to finishing a linear path segment or block of data, and the other labeled 4600 CIRC, related to finishing a circular data block. They are shown side by side to permit comparison of the steps performed during corresponding time periods in carrying out linear and circular path generation. The first 14 steps during each column of the FIG. 17 programming chart are devoted to utilizing the results of data generated by the system and are separately shown in another table. The transferring of data from buffer to active begins on step 18 (time period 18) of column 4600. By means of a program gate 349 a READ gating signal is applied to the I BUF register 47I during program step 4618, causing the I number stored therein to be placed upon branches CIT-2 through CIT-7 of the computer input trunk line 59, from which it is entered in the computer 53 during the adder cycle associated with the program step 4618.

During the following program step 4619 a WRITE gating signal is applied to the I register 99 by its WRITE PGA 355 (FIG. 9j) causing the I number appearing during this program step in the Answer Register 53a of the computer to be entered into the I register over trunk line branches COT-2 through COT-8. Since the I BUF number stored in the Answer Register 53a had only six digits the least significant of which was

stored in digit stage 7 of the register, a zero is stored in the least significant digit stage of the I register 99. The reason for this additional digit will become apparent later in this description.

Having transferred the I number from buffer to active through the computer 53, its Answer Register 53a is cleared in the next step 20 by application of a control signal to its CAR input. This clearing operation is performed many times during the operation of the system and to produce the control signals required to initiate the clearing operation a program gate array 351 (FIG. 9c), comprising many individual program gates is provided. A series of conditions under which the program gate array 351 is to produce a CAR signal is listed in the block representing the array. The particular program step 4620 is 14th on the list.

To transfer the J number from buffer to active, a READ pulse is applied to the J BUF register 47J by a program gate 353 during program step 4623. In response to this READ signal the J number is read into the trunk line 59 during program step 4623 and appears in the Answer Register 53a during the following program step. The J number is stored during the program step 4624 by a WRITE signal applied to the J register 101 by its associated WRITE PGA 357.

Having transmitted the J number from the Answer Register 53a to the J register 101, the Answer Register is cleared on program step 4626 by means of a CAR signal from the program gate array 351 (condition 15). Also during program step 4626 the XEP number is read from the XEP BUF register 47X by application of a gating pulse derived by means of a program gate 359. As a result, the programmed X axis end point is transmitted to the Answer Register 53a over trunk line branches CIT-3 through CIT-8, causing the number to be stored in stages 3 through 8 of the register. Before this number is stored in the active registers 57, however, it is modified to translate it from the programmer's coordinate system into the machine's coordinate system. This is an entirely optional step and may be omitted if so desired. Indeed, in certain applications it is desirable not to translate the data from the programmer's coordinates (also known as "part coordinates") in order to permit display of the data in the form which is familiar to the programmer and to the operator of the machine with which the system is used. However, in the exemplary system the data is transformed and the nature of the changes made in making the transformation will be understood by referring to FIG. 2.

As seen in FIG. 2, the zero point of the machine's single quadrant coordinate system is offset by 50 inches along both the X and the Y axes with reference to the center of the programmer's four quadrant coordinate system. The direction of the fixed X and Y axis offsets is such that the position of any point defined with reference to the programmer's coordinate system may be translated into the machine's coordinates of that point.

Another optional feature for which allowance is made in the detailed program shown in Table I is the machinist's "dialed in offset" correction. By means of this optional feature the machinist operating the machine tool can enter an additional offset on either or both of the X and Y axes to compensate for dimensional inaccuracies of the cutting tool used. To implement the 50 inch fixed offset correction along both the X and Y axes, a pair of READ gates 361 and 363, shown in FIG. 9f, are provided, their respective outputs connected to the one and four bit lines of trunk line branch CIT-3. Each of the READ gates 361 and 363 has two inputs, the first of both being connected to a source of logic 1 voltage. The other inputs of the READ gates are parallel connected to a 2 program gate array so as to be simultaneously enabled thereby. Consequently, when the READ gates 361 and 363 are enabled they pass a logic 1 voltage level over the 1 and 4 bit lines of the trunk line branch CIT-3 to the one and four bit flip-flops of stage 3 of the Computer 53, thereby entering digitally the number 5 therein. By defining the decimal point between

stages 4 and 5 of the Answer Register for this offset number, it is made to represent 50.0000 inches, as seen in FIG. 15.

The machinist's X and Y offset numbers are produced by a pair of registers 365 and 367, respectively labeled in FIG. 9f as X OFF register and the Y OFF register. Each of the registers 365 and 367 has six digit stages and a sign stage. Each digit stage may be set by appropriate means digitally to signal any number from 0 to 9 and similarly the sign stage may also be caused to signal either + or -. Because the X OFF and Y OFF registers 365 and 367 are independently settable and do not receive their data from the buffer registers 47, they do not have WRITE gates as do the other active registers. They are provided, however, with READ gates which may be identical to those shown in FIG. 13 with reference to the XΔ register 109. The READ gates of the registers 365 and 367 are connected to apply the signals stored in the digit stages of the registers to trunk line branches CIT-3 through CIT-8 and through them to cause the numbers to be entered in stages 3 through 8 of the Computer 53.

With the X axis end point number which had been read from the buffer register 47X still in the Answer Register 53a, a READ gating signal is applied to the READ gates 361 and 363 during program step 4627 in order to cause them to apply a digitally signalled number 5 to the computer as previously described. This pulse is produced by a program gate 369 whose output is applied through an OR gate 371 to the inputs of the READ gates 361 and 363. As a result, a 50.000 inch offset is added during the adder cycle associated with program step 4627 to the X coordinate in the Answer Register 53a. The machinist's offset is added to or subtracted from, the sum in the Answer Register 53a during the following program step 4628 in response to a READ signal applied to the X OFF register 365 from a program gate 373. The final result, representing the X coordinate fully translated into machine coordinates, appears in the Answer Register 53a during program step 4629 and during this program step is transmitted over the output trunk line 61 to the XCEP register 95 in response to a WRITE command signal from a Program Gate Array 375 associated with the XCEP register 95. This completes the transfer and translation into machine coordinates of the X axis end point number from the buffer register 47X to the active register 95.

The manner in which the Y axis block end point is transferred during the next four program steps from the buffer register 47Y to the active register 97 is exactly the same as the foregoing, except that the steps occur at different times and are initiated by different program gates, and that different registers are involved. Thus, first the Answer Register 53a is "cleared" during program step 4630 by a CAR pulse from the Program Gate Array 351 (condition 16). During the same program step the YEP BUF number is read from buffer register 47Y in response to a READ gating signal from a program gate 377. During the following program step 4631 a 50.000 inch offset is read into the computer through the READ gates 361 and 363 in response to a READ gating signal produced by a program gate 379, and fed through the OR gate 371 to the READ gates. During the next program step 4632 the machinist's Y offset is read from the Y OFF register 367 in response to a READ gating signal from a program gate 381 shown immediately to the left of the register. The result, representing the YEP BUF number translated into machine coordinates, is read from the Answer Register 53a into the YCEP register 97 by a WRITE gating signal applied thereto from a Program Gate Array 383 associated with that register.

Transfer of the Q number from the buffer register 47Q to the corresponding active register 103 is carried out during program steps 4635 and 4636. The Q number is read from the register 47Q during program step 4635 by a READ gating signal applied to the register from a program gate 385 and during the same program step the Answer Register 53a of the computer 53 is cleared by a CAR signal from the PGA 351 (condition 17). Stored in stages 4 through 8 of the Answer Re-

gister 53a during the following program step 4636, the Q number is written into the Q active register 103 from the Answer Register 53a by a WRITE gating signal from a program gate 387.

This completes the first sequence preparatory to generating the linear path segment 27b during which the I and J numbers and the X and Y coordinate numbers were transferred from buffer to active. No further operations are performed during the 4600 column. It will be noted that the G number has been left in the G BUF register 47G. The reason for this is that the G number associated with the path segment 27a which is currently being generated is still stored in register 105, and cannot as yet be discarded.

2. CONVERTING THE DATA TRANSFERRED TO THE ACTIVE REGISTERS INTO MACROMOVE AND MICROMOVE NUMBERS

In carrying out the invention, the digital signals representing the X component I of the composite distance to be covered in executing the path segment 27b are processed to derive a set of signals digitally representing a ΔX macromove number which is equal to $I \cdot \Delta T \cdot V/D$ (Equation 20). Additionally, the digital signals representing the Y component J of the composite distance D are processed to derive therefrom a Y axis macromove number ΔY , which in turn is equal to $J \cdot \Delta T \cdot V/D$ (Equation 21). These equations will be recognized as being equivalent to those shown as equations (6) and (7) in FIG. 5b. In accordance with a more specific feature of the invention, several of the numbers comprising the equations (6) and (7) are precomputed and collectively represented as the number Q. The numbers which are so computed and their relation to the number Q is shown in equation (8) in FIG. 5b. Thus, the two computations referred to above as equations (20) and (21) are shortened to the form represented by the equation (9) and (10) of FIG. 5b. Reproduced here for convenient reference they are: $\Delta X = I \cdot Q$ (9); $\Delta Y = J \cdot Q$ (10).

In the exemplary embodiment of FIG. 9 the number Q is precomputed externally of the system and is supplied thereto as part of the data block representing a path segment. It will be understood, however, that this precomputation could also be performed internally. Thus, instead of the number Q the block of data representing a given path segment might include the number V. ΔT is a known constant. In such a case, the system would use a digital computer capable of squaring, square rooting, and dividing as well as adding and multiplying and the operations called for by the equations (6) and (7) would be carried out internally by the computer.

It should also be noted that the X and Y components I and J of the path segment D need not be given on the tape if the X and Y coordinates of the desired end point are supplied. Instead, the I and J numbers could be derived by the system simply by subtracting from the X and Y end point coordinate numbers the corresponding numbers of the previous data block. The disclosed system merely represents the simplest of several possible alternatives.

A. MULTIPLYING J×Q AND SPLITTING OFF THE REMAINDER

Referring now to FIG. 17, the basic computations represented by equations (9) and (10) are carried out during corresponding time periods of the adjacent programming sheet columns 4900 and 4800. The first quantity to be computed is $J \times Q$ and the quantity $I \times Q$ is computed subsequently. The steps involved in carrying out these computations are shown in greater detail in Table II.

TABLE II
Steps performed in Mode 4 (Part 2)

	Entering linear block		Entering circular block	
	4800 · G 01	4900 · G 01	4900(G 02+G 03)	4900(G 02+G 03)
1 to 15	See Table III			
17...	R/Q	R/Q	R/Q	R/Q
18...	R/J and multiply by Q	R/I and multiply by Q	R/±I and multiply by Q	R/±J and multiply by Q
19 to 64...	Carry out J × Q	Carry out I × Q	Carry out ±I × Q	Carry out ±J × Q
	multiplication	multiplication	multiplication	multiplication
65 to 87...	No computations			
88...	W/ΔY	W/ΔX	W/ΔY	W/ΔX
89...	ZR1	ZR1	ZR1	ZR1
90...	SL4	SL4	SL4	SL4
91...	W/YR	W/XR	W/YR	W/XR
92...		CAR R/GBUF	CAR R/ΔY	CAR R/GBUF
93...		W/G	8X	W/G
94...		Advance J by ½ Y axis macromove	R/ΔY	8X
95...			R/ΔY	8X
96...			R/ΔY	8X
97...			R/ΔY	8X
98...			SAR	
99...			R/J	
100...			W/J	
101 to 104...	No computations			
105...		Start tape reader		Start tape reader
106...		Advance EOB to S15		Advance EOB to S15
107...		Advance EOB to S16		Advance EOB to S16
108...		Advance EOB to S1		Advance EOB to S1

The first pair of columns in the table pertain to computations performed in preparation to entering a linear block and are of immediate interest. The third and fourth columns of the table pertain to computations in preparation for performing circular interpolation and will be discussed in a subsequent section. As in the case of Table I, the corresponding steps pertaining to linear and circular path generation are shown side by side to facilitate comparison.

The operations performed during the first 15 time periods represented by the Table II are not shown for the same reason mentioned in connection with Table I. These time periods are assigned to certain repetitive operations which relate to the execution of the previous linear path segment 27a and will be discussed with reference to a separate table directed to those operations.

The steps listed in the first column of Table II correspond to those indicated generally in the 4800 column of the FIG. 17 iteration. Therefore, each of the steps or operations performed during that first column in Table II is initiated by a signal on the 4800 output of the Multimode Time Base Logic 67 and also by a signal on the G01 output of the G BUF decoder 221, if the step is to be performed only when the next path segment is linear.

To begin the computation of J×Q the multiplier Q is read during step 4817-G01 from the Q active register 103 in response to a READ gating pulse from the Program Gate Array 389 (FIG. 9j) associated with the register. This step is also performed during program steps 4817-G02 and 4817-G03, and hence the PGA is conditioned to produce a gating signal simply in response to the signals representing program step 4817 (condition 3). The Q number thus read is stored in the auxiliary storage devices (not shown) provided for that purpose in the computer 53. For sake of example, it may be assumed that such a storage device is identical in structure to the Q active register 103.

Next, during program step 4818-G01, the number J is read from the J active register 101 in response to a program gating pulse from its associated PGA 391 (condition 9). The J number thus read is also stored in a storing device (not shown)

in the computer 53 which may be identical to the J active register 101.

Concurrently with the reading of the J number from register 101 a multiply instruct signal is applied to the X (multiply) control input of the computer 53 by PGA 393 (FIG. 9c) in response to the fourth of its seven listed responsive conditions. The multiplication of J×Q is carried out in the exemplary computer during steps 19 through 64. This allows for 10 program steps for each of the four mantissa digits of the Q number read from the active register 103 and additional steps for shifting the product in accordance with the characteristic of the Q number. The product appears in the Answer Register 53a during step 4864-G01, correctly positioned relative to the decimal point of the multiplicand I, which is between digit positions 3 and 4 in the register (See I in FIG. 15).

For practical values of path segment length and desired velocity along that path segment it may be assumed that the product stored in the Accumulator Register 53ab will have a zero in position 4 and that the first non-zero digit of the product will be in stage 5 or in an even lower digit stage of the register. In all, the non-zero digits of the product will be stored in some or all of stages 5 through 12 of the register, the last four of these being part of the Overhanging Register 53b and the digits stored in them having a digital significance of millionths, 0.1 millionths, 0.01 millionths, and 0.001 millionths (or billionths) inches. Because of their extremely small size these digits are not sufficiently significant to be utilized by the servo drives 75 and 77. Thus, it would be desirable to discard them and to use only the four digits of the product J-Q that are stored in stages 5, 6, 7, and 8 of the Answer Register 53a. This would simplify the handling of the product J-Q. However, if the four least significant digits which are stored in stages 9 through 12 were simply discarded, an undesirable error would eventually accumulate. To achieve simplification without affecting accuracy, in accordance with another feature of the present invention there is provided a method, and apparatus for performing that method, for storing the first group of digits of J-Q, those in the Answer Register 53a, in a first register and for storing a second group of digits, those in the Overhanging

Register 53b, in a second register. In carrying out this aspect of the invention, only those digits stored in the first register are processed to develop macro-positions along the X axis, thereby reducing the storage capacity required for the computations involved in producing them.

The computation of J·Q is performed at regularly recurring instants, and as a part of each such computation the last four digits of the previous product J·Q, which had been formed in the second register as part of the previous computation of J·Q, are added to the result. The last four digits of the sum are then split off again and separately stored. In this manner each time the computation J·Q is performed, it is performed to an accuracy of 1-billionths of an inch. Even though the last four digits of the product are not used at this time, they are carried along as an unused part of the product, to be included in the next computation so as to insure that no significant accumulation of error occurs in the overall results.

Turning again to Table II, the first step in the operation of "splitting off the remainder" occurs in the step 4868 · G₀₁ when the WRITE gates of the ΔY register 107 are opened by a gating signal from the Program Gate Array 395 (condition 4) provided for that register. As a result, the sign and digits of the product J·Q stored in stages ±, 5, 6, 7, and 8 of Answer Register 53a are transferred into the ΔY register 107 and are stored therein.

During the following program step 4869 · G₀₁ a control signal is applied to the ZR1 control input of the computer 53 from a Program Gate Array 397 (FIG. 9c) (condition 4). In response to this control signal, section R1 of the Accumulator Register 53ab is cleared so that stages 1 through 8 of the register all contain zeros. It now remains to transfer the contents of stages 9 through 12 of the Accumulator Register 53ab representing the "remainder" digits of J·Q, into the YR (for "Y Remainder") register 111. Toward this end, during the following program step 4870 · G₀₁ the numbers stored in the Overhanging Register 53b are shifted four positions to the left by applying a control signal to the SL4 input of the computer 53. This signal is produced by the Program Gate Array 347 (condition 7) and in the manner explained with reference to FIG. 14 causes the contents of the register 53ab to be shifted 4 positions to the left during the same program step. By so doing the numbers previously stored in the Overhanging Register 53ab which does not have means for reading out its contents to the computer output trunk line 61 have been transferred to stages 5 through 8 of the Answer Register 53a which does have the appropriate connections, the trunk line branches COT-5 through COT-8. Through these branches, the sign and the four least significant digits of the product J·Q, now in stages ± and 5 through 8, are transferred to the YR register 111 by a gating pulse applied to the WRITE gates of that register from a PGA 399 (condition 3) provided to control the entry of data into that register.

The computation of I·Q follows the same steps during Column 4900 · G₀₁ as those performed in calculating the product J·Q except that, instead of the number J it is the number I which is fed to the computer and instead of the registers ΔY and YR 107 and 111, it is the ΔX and XR registers 109 and 113 in which the product is stored. Briefly, during step 4918 · G₀₁ the number I is read from the I register 99 (PGA 400 condition 8) and is transferred over the trunk line 59 to a storage device within the computer 53 (not shown). The Q number, serving as the multiplier, is again read into its storage device within the computer 53 during program step 4917 · G₀₁.

Also during step 4918 · G₀₁ a multiply command signal is applied to the computer from the PGA 393 (condition 3) in response to which the computer carries out the I·Q multiplication during program steps 4919 · G₀₁ - 4964 · G₀₁. The sign and the first four digits of the product, stored in stages ±, 5, 6, 7, and 8 of the Answer Register 53, are transmitted during program step 4968 · G₀₁ to the ΔX register 109 and are entered therein by a WRITE gating signal from a PGA 401 (condition 2) associated with that register. The first four digits are then

eliminated from the Answer Register 53a during program step 4969 · G₀₁ by a control signal from the "ZR1 + ZR2" PGA 397 (condition 5). The last four digits of the product are then transferred into stages 5, 6, 7 and 8 of the Answer Register 53a from stages 9 through 12 of the Overhanging Register 53b by a "Shift Left 4" operation carried out during step 4970 · G₀₁ in response to a control signal produced by the PGA 347 (condition 8). During the next program step 4971 · G₀₁ the sign and the last four digits of the product are transferred from stages 5 through 8 of the Answer Register 53a to the XR register 113 and are read into that register in response to a gating signal from its associated PGA 403 (condition 2).

As a result of the foregoing two series of computations, the products J·Q and I·Q have now been stored in their designated active storage registers. It is now appropriate to transfer the G number from the G BUF buffer register 476 to the G active register 105.

This may be done now because the execution of the path segment 27a is complete, so that the G number presently in the G active register is no longer needed.

To transfer the G number from buffer to active, the Answer Register 53a is cleared during step 4973 · G₀₁ by a control signal from CAR PGA 351 (condition 18) and during the same step the number stored in the G BUF register 476 is read therefrom by a gating signal from a program gate 405. The G number is transferred over trunk line 59 to the computer 53 and appears in its Answer Register 53a during the following program step 4974 · G₀₁ at which time it is entered into the G active register 105 in response to a gating signal applied thereto by a program gate 407. This completes the transfer from buffer to active of all of the basic data required to generate the following linear path segment 27b. Accordingly, during program step 4995 · G₀₁ a control signal is applied by the program gate 269 to the tape reader control flip-flop 257 (FIG. 91). The data representing the next path segment 27c is then read from the tape 37 and is stored in the buffer registers 47.

b. Going into Mode 1 Operation

Having transferred all the necessary information into the active registers 57 and having started the transfer of data from the tape to the buffer registers for the following path segment, it is now time to change the mode of operation of the system from that associated with the transfer and basic computation of data to that associated with the interpolation of the transferred and basically computed data for the generation of a linear path. Accordingly, means are provided in the form of one of the array of program gates 183 to advance the end of block counter from its 14th count state through its 15th and 16th count states into its first count state wherein its decoded output appearing on output S1 of the binary to single line decoder 187 enables the M1 input of the Time Base Mode Selector 69. This, it will be recalled, places the system in its mode 1 operation.

In order to step the End of Block Counter from its 14th to its first count state the Program Gate Array 183 includes three program gates for successively applying a stepping pulse to the End of Block Counter 186 in response to conditions 4997, 4998, and 4999 (conditions 19, 20, and 21). As a result, at the end of program step 4999 · G₀₁ the system is placed in its mode 1 operation. In this mode, a thousand successive program steps, from 1000 to 1999 and corresponding to a programming chart such as that shown in FIG. 10c, are recurrently signaled at the outputs of the AND gates 175 of the Multimode Time Base Logic 67 in conjunction with the tens and units converters 169 and 167 of the Time Base and Digital Sweep 63 in FIG. 9b.

b. Upgrading and Updating With Macromove and Micromove Numbers to Produce Macropositions and Micropositions

It will be helpful to summarize the operations which have been performed up to this point and also the operations which are to be performed next in the generation of the linear path segment 27b. In preparation for generating the path segment 27b and while the system was executing the previous path seg-

ment 27a the macromove numbers ΔX and ΔY of a given computation were computed on the basis of information transferred from the buffer registers 47 plus the XR and YR unused remainders from the previous computation. Only the first four digits of the ΔX and ΔY macromove numbers will be used, those stored in the ΔX and ΔY registers 109 and 107, and the new remainders XR and YR carried over. These macromove numbers represent the movements which are to be executed along the X and Y axes during the next time period or iteration ΔT which, for the illustrated system, is 20 milliseconds long. (The phrase "time period ΔT , or iteration" denotes a designated segment in time, which in the exemplary embodiment is 20 milliseconds long, as determined by the cycling period of the Program Step generator 161, and which repeats). What remains to be done and what is performed by the system in Mode 1 operation is to measure off successive periods ΔT in actual time and to utilize the ΔX and ΔY numbers stored in the registers 109 and 107 to generate for each successive time period ΔT corresponding X and Y axis macropositions which are to be reached at the end of the time period.

The means for measuring off successive periods ΔT in actual time has already been described. It is the program step generator 161 cycling at a repetition rate of once every 20 milliseconds. The macroposition numbers for the X and Y axes are also referred to as the X command position (XCP) and Y command position (YCP). Means are provided in the form of

are also provided for initially storing in the registers 121 and 119 XSC (X servo command) and YSC (Y servo command) signals representing the coordinates of the starting point P1 of the linear path segment 27b. Moreover, means are provided for causing the digital computer 53 during each successive period $\Delta T/N$ to process the signals stored in the ΔX register 109 to derive therefrom the quantity $\Delta X/N$ and thereafter to operate upon the signals stored in the XSC register 121 so as to change the XSC coordinate value represented thereby by the quantity $\Delta X/N$. Similarly, means are provided for causing the computer 53 during each period $\Delta T/N$ to process the signals stored in the ΔY register 107 so as to derive from them the quantity $\Delta Y/N$ and thereafter to operate upon the signals stored in the YSC register 119 so as to change the YSC coordinate value which they represent by the quantity of the $\Delta Y/N$.

FIG. 18 is a general programming chart of the operations performed by the system during each iteration in mode 1. Among these are the generation of macropositions and micropositions. A new pair of micropositions is generated during the initial ten steps of each 100 step column of the programming chart. A new pair of macropositions is produced only once during an iteration, and this occurs during steps 10-14 of the first column of the programming chart. These steps are shown in detail in Table III and it will be noted that the steps there shown are common to all modes of operation of the system.

TABLE III
Steps common to all modes

	X000	X100	X200	X300	X400	X500	X600	X700	X800	X900
00.....	CAR, R/ ΔX									R/XCP
01.....	BAR									
02.....	R/XSC									
03.....										
04.....	W/XCS, CAR									
05.....	R/ ΔY									R/YCP
06.....	BAR									
07.....	R/YSC									
08.....										
09.....	W/YSC									
10.....	CAR, R/XCP									
11.....	R/ ΔX									
12.....	W/XCP									
13.....	CAR, R/YCP									
14.....	R/ ΔY									
15.....	W/YCP									
16.....	CAR									

the XCP register 117 and the YCP register 115 to store signals initially representing the X and Y coordinates of the starting point P1 of the linear path segment 27b and means for supplying these particular coordinates to the registers 115 and 117 are also provided. In generating macropositions, the signals stored in the XCP and YCP registers 117 and 115 are "upgraded" during each time period ΔT so as to cause them to represent a new macroposition for each of the X and Y axes. To do this, means are provided to cause the computer 53 to utilize the ΔX and ΔY numbers stored in the registers 109 and 107 to perform the upgrading of the numbers stored in the XCP and YCP registers 117 and 115. In accordance with a more specific aspect of the invention, there is also provided a method, and means therefor, for generating signals which represent intermediate micropositions for each of the X and Y axes representing points which are to be reached at instants distributed in time along a given time period ΔT . In carrying out this aspect of the invention, successive periods $\Delta T/N$ in actual time are measured off, N such time periods spanning each of the time periods ΔT . Furthermore, there are derived from the signals stored in the ΔX and ΔY registers 109 and 107 microposition signals which are changed at the end of each time period $\Delta T/N$ by incremental amounts of $\Delta X/N$ and $\Delta Y/N$.

For the above purpose there are provided a pair of storage means, the XSC and YSC registers 121 and 119, and means

The 10 columns of the programming chart shown in Table III are identified as X000, X100, X900, indicating that the first digit of the program step number, denoting the operating mode of the system, is immaterial, and that this chart applies to all operating modes. Consequently, all of the program gates provided to initiate the operations shown in Table III are connected to receive their highest order input from the hundreds converter 171 of the Time Base and Digital Sweep 63 in FIG. 9b.

Turning now to a detailed consideration of producing micropositions, the first step is to clear the Answer Register 53a. This is timed to occur during program step X000 (Table III) by a CAR signal from PGA 351 (condition 1). During the same step a READ gating signal is applied to the ΔX register 109 by an associated program gate array 409 (condition 1), causing the ΔX number to be stored in the Answer Register 53a. The next step in the process, performed during step 001, is to derive from the macromove number ΔX a micromove number $\Delta X/N$. In the present example, $N=10$ in conformance with the number of smaller time periods into which the time period ΔT had been divided. Had ΔT been divided by 20, i.e., if the programming chart had 20 columns, then a micromove for such a time period would be 1/20th of a macromove number.

To divide the macromove number ΔX by 10, it is simply shifted one position to the right during program step 001 in

response to a Shift Right control signal applied to the SAR control input of the computer 53 by a Program Gate Array 411 (condition 1) provided for that purpose and shown in FIG. 9c. As a result of the shifting to the right of the ΔX number, the Answer Register 53a will contain during the following program step 002, the X axis micromove number $\Delta X/10$ representing the movement to be executed along the X axis substantially during the first 1/10 of the time period ΔT .

To produce the first X microposition number for the time period ΔT , the X micromove number stored in the Answer Register 53a is added to the X microposition number presently stored in the XSC register 121. This presently stored microposition number is the X coordinate of the starting point of the linear path segment 27b and was stored therein at the end of the generation of the previous path segment 27a. To add the micromove $\Delta X/10$ to the last X microposition number XSC, the latter is transferred to the computer 53 from the XSC register 121 during program step 002, by application of a READ gating signal to the XSC register 121 from a Program Gate Array 412 associated with that register (condition 1). As explained previously in section G1 (d), the reading into the computer of a number without a special control signal to one of the 10 control inputs to the computer causes the number to be added algebraically to the number stored in the Answer Register 53a. Consequently, at the end of the adder cycle associated with program step 002, there is stored in the Answer Register 53a the sum of the previous X microposition number read from the XSC register 121 and the current micromove number $\Delta X/10$. The sum, representing the first X microposition for the time period ΔT under discussion is transferred from the Answer Register 53a to the XSC register 121 during program step 005 by the application of a WRITE gating signal to the register from a program gate 413 shown to the right of the register in FIG. 9g.

The foregoing four steps, performed during program steps 000, 001, 002, and 005, represent the updating of the microposition number stored in the XSC register 121. A similar updating sequence is next performed for the Y axis microposition stored in the YSC register 119. The process begins during program step 005 with the clearing of the Answer Register 53a to prevent the X axis microposition number XSC from reaching the addend input of the computer. For this purpose the PGA 351 is operative to apply a CAR signal to the corresponding control input of the computer 53 during program step 005. This is indicated as condition 2 in the program gate array 351, shown as 05. The reason why the condition is not shown as 005 is that, for reasons which will be shown hereinafter, a CAR signal is also produced during program steps 105, 205...905 and therefore these 10 conditions are adequately defined by the program step number 05. Other such multiple steps are produced by 00, etc.

Also during program step 005, the Y axis macromove number ΔY is transferred from the ΔY register 107 to the Answer Register 53a by applying a READ gating signal to the register from a Program Gate Array 415 (condition 1). During the next program step 006 the macromove number stored in the register 53a is shifted to the right by one position, thereby dividing it by 10 and producing therein the Y axis micromove number $\Delta Y/10$. The shifting to the right by one position is initiated by a control signal applied to the SAR input of the computer 53 by the Program Gate Array 411 (condition 2).

The Y micromove number $\Delta Y/10$ stored in the Answer Register 53a is added to the previous Y microposition number YSC by transferring that number from the YSC register 119 to the computer 53, the latter operating in the ADD mode. This is done during program step 007, and is initiated by a READ gating signal applied to the YSC register 119 by a program gate array 417 (condition 1). The sum, representing the next Y microposition number YSC, is transferred from the Answer Register 53a over the trunk line 61 to the YSC register 119 during program step 010 by means of a WRITE gating signal applied to the register 119 by a program gate 419.

Summarizing the foregoing sequence of operations carried out during steps 000-010, the XSC and YSC registers 121 and 119 have both been updated so as to store in them the first of the 10 microposition numbers for the time period ΔT .

The first step in upgrading the XCP and YCP registers for producing new macroposition numbers is that of clearing the Answer Register 53a and is carried out during program step 010. During this program step a CAR signal is applied to the appropriate control input of the computer 53 by the CAR PGA 351 (condition 3). Next, during the same program step, the X macroposition number presently stored in the XCP register 117 is read therefrom by applying to the register a READ gating signal from its associated Program Gate Array 421 (condition 1) and is stored in the Answer Register 53a. During program step 001, the X macromove number ΔX is read into the computer 53 from the ΔX register 109 by a READ gating signal supplied to the register from its associated Program Gate Array 409 (condition 10) and is automatically added to the stored macroposition number XCP. The sum, representing the upgraded X macroposition number XCP appears in the Answer Register 53a during the following program step 012 and is written into the XCP register 117 during the same program step by a WRITE gating signal applied to that register from its Program Gate Array 423 (condition 1).

A similar process is next performed to upgrade the YCP register 115 and commences with the clearing of the Answer Register 53a during program step 012 to remove the upgraded XCP number therefrom (CAR PGA 351, condition 4). During the same program step the current Y axis macroposition number is added into the cleared Answer Register 53a from the YCP register 115 by the application of a READ gating signal to the register from its Program Gate Array 425 (condition 1). During the next program step 013 the Y axis macromove number ΔY is added to the current YCP number stored in the Answer Register 53a by reading the ΔY number from the register 107 onto the computer input trunk 59. For this purpose, a READ gating signal is applied to the register by its Program Gate Array 415 (condition 10). The sum, appearing in the register 53a during program step 014, is entered in the YCP register 115 by a WRITE gating signal applied thereto by its Program Gate Array 427 (condition 1). This completes the upgrading of the XCP and YCP registers 117 and 115. They now contain macroposition numbers representing the X and Y axis coordinates to be reached at the end of the time period ΔT represented by the programming chart of FIG. 18. In order to prepare the computer for other operations that are to follow, its Answer Register 53a is cleared during the next program step 015 (CAR PGA 351, condition 5).

The updating operations performed during the first 1/10 of the iteration represented by column X000 in Table III are repeated during the corresponding time periods in each of the following eight-tenths of the iteration represented by columns X100 through X800 of Table III. This is indicated in the table by horizontal arrows originating from particular operations performed during the column X000 and running through the corresponding steps of the columns to its right. Thus, for example, the horizontal arrow originating from R/ ΔX (read ΔX) in step 00 of column X000 in Table III signifies that the same operation of reading the number from ΔX register 109 is also performed during step 00 of each of the columns X100 through X800, i.e., during program steps X100, X200, ...X800. The gating signals for initiating these reading operations are indicated in the PGA 409 in FIG. 9h as the conditions 2 through 9 inclusive. It may be confirmed by reference to other portions of FIG. 9 that other gating signals for initiating all of the other operations indicated for columns X200 through X800 of Table III are similarly shown in their respective Program Gate Arrays. By these means a new updated microposition is written into the XSC register 121 during the sixth program step of each of nine successive but time-separated series of 10 program steps. Similarly, a new microposition for the Y axis is written during the eleventh pro-

gram step of each series into the YSC register 107. Each updated XSC value differs by $\Delta X/10$ from the previous XSC value and similarly each YSC value varies by $\Delta Y/10$ from the previous YSC value.

To compensate for any potential errors, the last updating operation during Column X900 in Table III is carried out by using the macroposition numbers XCP and YCP. These numbers, stored in the registers 117 and 115, are free of the small errors inherent in the microposition numbers XSC and YSC since they were produced by adding the entire ΔX and ΔY numbers to the previous XCP and YCP numbers. Therefore, they represent the correct values of the X and Y axis micropositions XSC and YSC for the 10th Column X900 of the Table III. Accordingly, during the first step 00 of the Column X900, the Answer Register 53a is cleared by a CAR signal from the CAR PGA 351 (condition 1) and during the same program step the X macroposition number is read from the XCP register 117 and added into the cleared Accumulator Register 53a in response to a gating signal applied to the XCP register from its associated PGA 421 (condition 2). The XCP macroposition number is then written into the XSC register 121 during program step X905 by a gating signal applied to the register from the program gate 413. During the same program step, as in all of the previous columns of Table III, the Answer Register 53a is cleared by a pulse from CAR PGA 351 (condition 2) and the Y axis macroposition number is added into it from the YCP register 115 by a gating signal from its PGA 425 (condition 2). The number so stored is then transmitted from the Answer Register 53a to the YSC register 119 as the new YSC microposition number during step X910 by a WRITE signal applied to the register by the program gate 419.

Referring again to FIG. 18, the updating operations of the XSC and YSC registers 121 and 119 are represented by the two horizontal strips extending along the top of the programming chart of FIG. 18 and the upgrading operations of the XCP and YCP registers 117 and 115 comprise the first two operations indicated in block form following the updating operations performed during the first column of the programming chart. These operations are common to all of the programming charts used in performing linear interpolation and are performed during each iteration, or once every 20 milliseconds.

The generation of macropositions and micropositions shown on the programming chart of FIG 18 is continuously repeated, causing the generated positions to progress and to be executed along the desired path segment until the macroposition associated with one of the X and Y axes reaches within a predetermined distance of the programmed end point P2 of the path segment 27b for that axis. In accordance with this aspect of the invention, provision is made for predicting M time periods ΔT in advance, the specific time period ΔT during which an overshoot of the end point P2 will occur, i.e., during which the macroposition numbers XCP and YCP will be changed to represent X and Y coordinates which are past the desired end point P2 of the path segment 27b as represented by the end point coordinates XCEP2 and YCEP2. Toward this end the computer 53 is caused during each of the time periods ΔT to process the signals representing a chosen one of the end point coordinates XCEP and YCEP, the corresponding one of the macropositions XCP and YCP, and the corresponding one of the ΔX and ΔY numbers to produce a signal representing the comparative magnitudes of:

1. the chosen end point coordinate, and
2. the sum of the corresponding macroposition number and M times the corresponding macromove number.

In response to the signal produced by the computer, representing the comparative magnitudes of (1) and (2), a warning signal is generated during the first time period ΔT in which the initially lesser of the compared quantities (1) and (2) becomes the greater. It will be seen in the following more detailed explanation that this reversal of relative magnitudes is indicative of a predicted overshoot of the desired end point

coordinate. This feature of predicting a future overshoot condition will be referred to as "look-ahead."

c. Geometrical Analysis of Look-Ahead with Macromoves Adjusted to Hit the End Point

1. General Discussion

The purpose of the look-ahead feature is to enable the system to reach the programmed end point for a linear path segment such as 27b exactly at the end of an iteration. In order to understand the usefulness of this feature, it will be helpful to understand why this is unlikely to occur without some adjustment in the magnitudes of at least some of the calculated macromoves ΔX and ΔY . The basic reason for computing and executing ΔX and ΔY macromoves at regularly recurring time periods ΔT is to cause movement along a designated path segment at a predetermined velocity. Thus, the primary significance of the macromove numbers ΔX and ΔY is velocity and not distance. Consequently, the calculations for deriving the macromove numbers ΔX and ΔY are directed to producing movement along a predetermined path segment toward a designated target point at a predetermined velocity. It is not the objective of these computations to reach the end point at the end of any particular macromove. To do so would require unduly high computational accuracy. However, this objective can be attained without increasing the computational accuracy otherwise required, by adjusting for certain of the time periods ΔT the magnitudes of the ΔX and ΔY macromove numbers which are to be executed during those time periods. Stated very simply, the movements to be carried out during these time periods ΔT are either "stretched" or "compressed" by amounts which will result in the target point being reached exactly at the end of a time period ΔT .

In previous sections there have been described, with reference to a path segment 27b extending toward an end point P2, certain means causing the computer 53 to operate over a first period of time at intervals of ΔT on the signals stored in the XCP register 117 and the YCP register 115 so as to "upgrade", or change, the coordinate values represented thereby by the quantities ΔX and ΔY , respectively. In accordance with the feature to be described next, means are also provided for causing the computer 53 to operate, over another given period of time and at similar intervals ΔT , on the signals in the XCP and YCP registers so as to upgrade them by the quantities $\Delta X'$ and $\Delta Y'$ whose values are such that after one of the similar time intervals ΔT , the upgraded XCP and YCP values agree closely with the X and Y coordinates of the end point P2.

Referring to FIG. 19, in accordance with the invention a preferred method for attaining the above objective is illustrated with reference to the path segment 27b previously shown in FIG. 2. Shown in FIG. 19 are the ΔX and ΔY movements executed along the X and Y axes during a series of time periods ΔT_1 - ΔT_{1000} and the composite movement representing the path 27b is also shown. The X axis excursion I for the path segment 27b was previously given as 6 inches and the macromove quotient Q was stated to be 0.001. Accordingly, the X axis macromove ΔX is 0.006 inches. In view of this assumption the end point P2 would be reached after exactly 1,000 iterations of 0.006 inches each, exactly at the end of iteration ΔT_{1000} . Of these only the first two and the last 14 are fully shown in FIG. 19.

In addition to the end point P2 two alternative end points P2' and P2'' are also shown. Let it be assumed first that, instead of the end point P2, the end point P2' had been programmed on the tape 37. It can be shown that the magnitudes computed for the ΔX and ΔY macromoves would not have been affected by this very small change and that the point P2, which is short of the point P2' would still have been reached at the end of the 1,000th ΔT iteration as before. If movement were stopped at this point, the desired end point P2' would not be reached. On the other hand, if one more iteration were carried out, the movement executed during that iteration would carry past the desired end point P2'.

2. The Two Tests for Detecting the Presence and Sign of a

Projected Overrun

In accordance with a specific aspect of the invention related to reaching the end point exactly at the end of an iteration, a distinction is made, based upon the amount by which this last macromove would carry past the actual end point. If the amount of "overrun" would exceed half of a macromove, that is, if the end point would be passed during the first half of the time period ΔT , it is classified as a large overrun. If not, it is classified as a "small" overrun. Thus it is seen that, since the point $P2'$ lies less than one-half macromove away from the point $P2$, the execution of another full macromove past the point $P2$ would result in a large overrun.

A small overrun is illustrated by the second alternative end point $P2''$ lying more than one-half macromove past the point $P2$.

In carrying out the invention, two tests are performed during each time period ΔT while movement along the path segment 27b progresses toward the end point, taken initially to be the point $P2'$ to illustrate a large overrun. The first test detects proximity to the desired end point $P2'$ far enough in advance to permit subsequent macromoves to be so adjusted as to cause the end point to be reached at the end of one of them. The second test is used to determine whether the detected overrun condition is large or small. If large, one type of corrective operation is carried out and if small, the corrective operation is of a second type. Specifically, the number of iterations for which the macromove numbers ΔX and ΔY are modified are made sufficiently large to prevent the resulting change in speed from being excessive. A convenient number for computation is 10. In the illustrated embodiment of the invention, it is the last 10 iterations for which a modified pair of macromove numbers ΔX and ΔY are computed.

3. The First Test

To allow sufficient time to modify the macromove numbers ΔX and ΔY for the last 10 iterations for the path segment 27b, the overrun prediction is made 11 time periods ΔT in advance. It is made by producing during each time period ΔT , as part of the iteration performed during that time period, a point projected 11 macromoves forward along a selected one of the coordinates X and Y measured from the point that would be reached at the end of the current iteration. Several of these projections, produced during iterations $\Delta T987$ – $\Delta T990$, are shown in FIG. 19 as the horizontally extending vectors 431a–d. In the exemplary system the look-ahead test projection is always done for the axis along which desired movement is faster in order to enhance the sensitivity of the test. In the illustrated example, this is the X axis.

Each of the projected points along the selected coordinate is compared with the corresponding coordinate of the end point to detect the first iteration ΔT during which an overrun would occur if movement continued along the path at the existing rate, i.e., with the existing macromoves. The first three of the look-ahead projections 431a–c fall short of the X coordinate $XCEP2'$ of the end point $P2'$ and as a result, the following iteration is carried out with the "standard" macromove numbers $X\Delta$ and $Y\Delta$. However, the fourth projection 431d, produced during iteration $\Delta T990$, extends past the coordinate $XCEP2'$ and this condition is detected and is used to initiate appropriate corrective action. The corrective action undertaken in response to detecting this projected overrun condition depends, however, on the result of the second test, which is related to the size of the overrun. If the second test indicates a large overrun, greater than half a macromove, the system is permitted to complete the iteration during which the overrun was detected but is forced to perform the following 10 iterations with increased macromoves $\Delta X'$ and $\Delta Y'$. On the other hand, if a small overrun is detected, the system is permitted to complete the iteration during which the detection occurred and also the following iteration at the normal rates. The subsequent 10 iterations are then carried out with slightly shortened macromoves $\Delta X''$ and $\Delta Y''$.

4. The Second Test

The second test is represented in FIG. 19 as a short vector extending leftward from the end of one of the longer vectors

431a–d. Each of the short vectors 433a–d is half a macromove, i.e., $1/2 \Delta X$, long and represents an operation whereby the amount of overrun indicated by the first test is compared with $1/2 \Delta X$ in order to determine whether it would be "small" or "large". In the case of the vector 431d the second test represented by its corresponding short vector 433d indicates that the projected overrun revealed by the first test represented by the vector 431d would be a large one because the short vector 433d extending backward from the end of the first vector 431d does not quite extend to the X coordinate $XCEP2'$ of the assumed end point $P2'$. In other words, even if the projection were shortened by half a macromove it would still extend past the X coordinate $XCEP2'$.

5. Calculating Adjusted Macromoves $\Delta X'$, $\Delta Y'$ for a Large Projected Overrun

Continuing with the assumption that $P2'$ is the end point and that a large projected overrun had been predicted to occur during iteration $\Delta T1001$, the iteration $\Delta T990$ during which the projected overrun was detected is carried out at the normal ΔX and ΔY macromoves. However, during the same iteration, while movement along the path 27b is advancing from command point (or macroposition) $CP989$ to command point $CP990$, calculations are being made to produce adjusted X and Y axis macromoves to be executed during the following ten iterations so that at the end of the last one of them the end point $P2'$ will be reached along both the X and the Y axes. These adjusted macromoves for reaching the point $P2'$ are identified in FIG. 19 by the symbols $\Delta X'$ and $\Delta Y'$.

To compute the adjusted X axis macromoves $\Delta X'$ the difference between the X coordinates of the end point $P2'$ and of the macroposition $CP991$ is determined. This difference is the quantity $10 \Delta X'$. It is divided by 10, resulting in the adjusted X axis macromove $\Delta X'$ which is then repeatedly executed once during each of the following 10 iterations, thereby insuring that at the end of the last of them, movement along the X axis will have reached the X coordinate $XCEP2'$ of the end point $P2'$ exactly. The results of these computations are identified in FIG. 19 as the vector 435, representing the quantity $10 \Delta X'$ and by a series of 10 shorter vectors 437, together equal in length to the vector 435 and individually representing the adjusted X axis macromoves $\Delta X'$. Since the vector $10 \Delta X'$ is slightly longer than $10 \Delta X$, each of the adjusted X axis macromoves $\Delta X'$ is longer than a normal macromove ΔX and the end point $XCEP2'$ along the X axis is reached exactly at the end of iteration $\Delta T1000$.

The adjusted macromoves $\Delta Y'$ for the Y axis are calculated in a similar manner. During iteration $\Delta T990$ the distance remaining between the Y axis coordinate of the macroposition $CP990$ to be reached at the end of the iteration and the Y coordinate $YCEP2'$ of the point $P2'$ is determined. This distance is identified in FIG. 19 as the vector 439 and it represents the quantity $10 \Delta Y'$. During this same iteration the quantity $10 \Delta Y'$ is divided by 10, producing the adjusted Y axis macromoves $\Delta Y'$ shown in FIG. 19 as the string of vectors 441. During the following 10 iterations $\Delta T991$ – $\Delta T1000$ the 10 adjusted macromoves $\Delta Y'$ are executed concurrently with the adjusted X axis macromoves $\Delta X'$.

6. Calculating Adjusted Macromoves $\Delta X''$, $\Delta Y''$ for a Small Projected Overrun

Let it be assumed next that the actual programmed end point is $P2''$. This assumed end point has X and Y axis coordinates $XCEP2''$ and $YCEP2''$ and is less than half a macromove short of the point scheduled to be reached at the end of iteration $\Delta T1001$. It, too, is detected during iteration $\Delta T990$ by use of the forward projected point represented by the vector 431d. Thus, it becomes known during the iteration $\Delta T990$ that an overrun would occur during iteration $\Delta T1001$. By means of the second test carried out during iteration $\Delta T990$, represented by the shorter vector 433d, it is further determined that the projected overrun along the X axis would be less than $1/2 \Delta X$. As a result, in accordance with the invention, instead of increasing the X and Y axis macromoves in order to reach the end point $P2''$ at the end of iteration $\Delta T1000$, as was done in the case of a large overrun, a number

of the remaining iterations are modified by shortening the X and Y axis macromoves so as to reach the end point P2'' only at the end of iteration $\Delta T1001$. To simplify calculations, such modified macromoves are calculated for the last 10 iterations $\Delta T992 - \Delta T1001$ by determining the difference between the X coordinate of the macroposition CP991 that will be reached at the end of iteration $\Delta T991$ and the X coordinate XCEP2'' of the end point P2''. The result is divided by 10, yielding the adjusted macromove quantity $\Delta X''$ for the X axis, ten of which are then successively executed during the time periods $\Delta T992 - \Delta T1001$. The quantities $10 \Delta X''$ and $\Delta X''$ calculated in this manner are shown in FIG. 19 as the vectors 443 and 445, respectively. Also, during iteration $\Delta T991$ the quantities $10 \Delta Y''$ and $\Delta Y''$ are calculated, these being represented in FIG. 19 by the vectors 447 and 449, respectively.

7. Summary and Restatement in Data Processing Terms

Summarizing the foregoing, if the overrun predicted for a given iteration, such as iteration $\Delta T1001$ in FIG. 19 is more than half a macromove, then the macromoves ΔX and ΔY are increased for the 10 iterations immediately preceding the given iteration and the programmed end point for the path segment is reached at the end of the iteration $\Delta T1000$. On the other hand, if the overrun during a given iteration such as iteration $\Delta T1001$ is predicted to be less than $1/2$ macromove, then the macromoves ΔX and ΔY are reduced for the nine iterations immediately preceding the given iteration and additionally for the given iteration itself and the programmed end point for the path segment is caused to be reached at the end of the given iteration $\Delta T1001$.

The time periods ΔT during which the first series of adjusted macromoves $\Delta X'$ and the second series of adjusted macromoves $\Delta X''$ are executed are shown by diagonal lines extending from the ends of the respective adjusted macromoves to the vertical lines designating the time periods $\Delta T991 - \Delta T1001$. The corresponding adjusted macromoves $\Delta Y'$ and $\Delta Y''$ are executed concurrently with the adjusted X axis macromoves $\Delta X'$ and $\Delta X''$.

A principal advantage of providing two possible corrections of the macromoves ΔX and ΔY , depending upon the magnitude of the projected overrun, is that, in the instance of adjusting the last 10 macromoves, the greatest amount of change in the magnitude of the macromoves and therefore in the rate at which motion is executed along the path 27b is 5 percent. Assume, for example, that an attempt had been made to reach the programmed end point at the end of iteration $\Delta T1000$ in the case of a small overrun such as illustrated by the assumed end point P2''. Specifically, assume that the programmed end point P2'' is only 0.1 macromove short of the point which would be reached at the normal rate at the end of iteration $\Delta T1001$. If it were desired to reach this point at the end of iteration $\Delta T1000$ instead, each macromove along both the X and Y axes would have to be increased by 0.09 macromoves or by 9 percent. But the end point P2'' can instead be reached exactly at the end of iteration $\Delta T1001$ with only a 1 percent change in speed by reducing each of the macromoves by 0.01 macromove.

It will be understood that the numbers used in the above example have been chosen because they lend themselves most readily to rapid calculation. However, it would be equally within the scope of the invention to use a longer look-ahead projection and to approach the end point so as to hit it exactly at the end of a given iteration with 20 instead of 10 adjusted macromoves.

To better illustrate the full scope of this feature of the invention it will be restated using symbols and data processing terms in place of specific numbers, but with the corresponding numbers in the previous example being shown in parentheses. Thus, during each time period ΔT , first and second sets of signals are produced representing the amount, if any, by which motion would be directed past the programmed end point along a selected axis at the end of the following $M(11)$ and $M - \frac{1}{2}(10.5)$ time periods. If it is predicted that motion would be directed past the programmed end point along the selected

axis within $M - \frac{1}{2}(10.5)$ time periods, the macromove numbers along the X and Y axes are increased by equal amounts for each of the $M-1(10)$ time periods ΔT which precede the time period ΔT for which an overrun was predicted so as to cause the programmed end point to be reached substantially at the end of the last one of the $M-1(10)$ time periods ΔT . On the other hand, if it is determined that motion would be directed to extend past the programmed end point along the selected axis within $M(11)$ time periods ΔT but not within $M - \frac{1}{2}(10.5)$ time periods ΔT , the macromove numbers for the specific time period ΔT during which an overrun was predicted to occur ($\Delta T1001$) as well as the macromoves for the $M-2(9)$ time periods ΔT immediately prior thereto are reduced by equal amounts so that motion will end exactly at the programmed end point at the end of the specific time period ($\Delta T1001$).

d. Data Processing Steps Carried Out by the Exemplary System in Performing Look-Ahead and Macromove Adjustment

1. Sequence of Iterations During Generation of Linear Path Segment

To explain the sequence of iterations performed by the system of FIG. 9 corresponding to the time periods $\Delta T1 - \Delta T1001$ illustrated in FIG. 19, reference will first be made to FIG. 20. Illustrated therein are a series of blocks, each block representing a programming chart such as those shown in FIGS. 17 and 18. The first iteration, performed during time period $\Delta T1$ is represented by the block 451 and is seen to be the type of iteration illustrated in FIG. 18 and discussed previously. The iterations of FIG. 18 are repeated during time periods $\Delta T2 - \Delta T988$. During each of these, a standard macromove is executed along the X and Y axes, causing a resultant motion to progress along the linear path segment 27b toward the programmed end point of that path segment. Following this series of iterations two possible options, corresponding to the two possible corrective actions discussed with reference to FIG. 19, are shown. The first option, undertaken when a large overrun is detected, is shown as the top row of blocks extending from iteration $\Delta T988$. The second option, chosen when a small overrun is detected, is represented by the second row of blocks, also extending from iteration $\Delta T988$.

Taking the top row of blocks first, agreeably with the example in FIG. 19, one more FIG. 18 iteration is performed during time period $\Delta T989$ with the standard macromove along the X and Y axes. During the following time period $\Delta T990$, a large overrun is detected by means of the look-ahead feature. As a result, the operations performed during the following time period $\Delta T990$ include calculations whereby modified macromoves are computed. The operations performed during this type of an iteration are illustrated in FIG. 21 and will be discussed in greater detail hereinafter.

During each of the following 9 time periods $\Delta T991 - \Delta T1000$ yet another type of iteration is carried out, these being shown in FIG. 22. Iterations of this type are performed with the system in mode 3 operation and are characterized by the fact that during each of them adjusted macromoves are executed. After the last of the nine FIG. 22 iterations, one more iteration is performed with the adjusted macromoves during time period $\Delta T1000$ in accordance with the operations shown in FIG. 17. These were discussed in detail previously in Section G-2.a and will be recalled to have included the computation of the standard macromoves for performing or generating the initial portion of the next linear path segment.

Referring now to the second row of blocks shown in FIG. 20 for time periods $\Delta T989 - \Delta T1000$ in FIG. 20, the first of the iterations, during $\Delta T989$, is again of the FIG. 18 type and precedes the iteration in which a small overrun is detected. The latter occurs during time period $\Delta T990$ and causes the operations performed during this time period to assume a sequence which is shown in FIG. 23. Because the detected overrun is small, however, adjusted macromoves are not computed until the following time period $\Delta T991$ and these are shown in FIG. 24. The adjusted macromoves are executed during the next 10 iterations $\Delta T992 - \Delta T1001$, the first 9 of

which are of the type shown in FIG. 22 and the last of which is in accordance with FIG. 17.

2. Data Processing Steps Associated with Detecting a Projected Overrun

a. Determining the Fast Axis

Turning again to FIG. 18 for a closer analysis of the look-ahead feature, the first step is to determine for which of the X and Y axes are the computed macromoves ΔX and ΔY larger, i.e., which of the axes is the "fast" axis. The identification of the fast axis is carried out during program steps 1015-1017 of the programming chart shown in FIG. 18 and are shown in detail in Table IV.

TABLE IV
Steps performed in Mode 1 linear

	1000 · LIN	1500 · LIN	1800 · LIN	1900 · LIN
0 to 13...	See Table III			
14.....	Advance E.O.B.			
15.....	? S2			
16.....	R/I			
17.....	CX			
18.....	P/FAX ? ≥			
19 to 27.....	C/FAX ? <			
28.....	CAR R/ΔX?FAX			
29.....	R/ΔY?FAX			
30.....	SAL			
31.....	R/ΔY ? FAX			
32.....	R/ΔY ? FAX			
33.....	W/ACNX from			
34.....	ACN			
35.....	R/XCP ? FAX			
36.....	R/YCP ? FAX			
37.....	SAR			
38.....	SX, R/XCEP ? FAX			
39.....	R/YCEP ? FAX			
40.....	See Note 1			
41.....	SAL			
42.....	SAL			

NOTE 1: On step 1035 advance E.O.B. counter to S2 ? S1 · (ACN · ACNX + ACN · ACNX) · LIN

During program step 1015 the programmed coordinate excursion I is read from the I active register 99 by a gating pulse from its PGA 400 (condition 1). The I number so read is transferred to the computer 53 and is stored therein. During the following step 1016 the programmed Y coordinate excursion J is read from the J active register 101 in response to a gating signal from its PGA 391 (condition 5). This information, too, is stored in the computer 53. Concurrently with reading of the J number a "compare" control signal is applied to the CX input of the computer 53 by a Program Gate Array 453 shown in FIG. 9c and labeled "FP+CX" (condition 3). In response to the control signal, the computer 53 carries out a compare operation whose results appear at one of its outputs 339 and 341 during program step 1017. If I is equal to or greater than J, a logic 1 signal appears on the output 339. If I is less than J, a similar signal is produced on the output line 341. To preserve this indication, a J-K flip-flop labeled FAX FF 455 (fast axis X) is provided. Through a pair of AND gates 457 and 459 the flip-flop 455 is so connected to the compare outputs 339 and 341 that if I is greater than J during program step 1017 the flip-flop is set, and if I is less than J the flip-flop is reset. Accordingly, if the X axis is the fast axis, a logic 1 signal appears at the Q output of the flip-flop 455 during program step 1017 and this output is labeled FAX. Conversely, if the Y axis is the fast axis, a logic 1 signal appears at the $\bar{Q}$ output of the flip-flop and this output is labeled FAX.

(b) Producing the Look-Ahead Point and Performing the First Test

Following the identification of the fast axis, a look-ahead point is subsequently generated along that axis beginning with program step 1028. It will be understood, of course, that the particular program steps during which the operations which are being described here are performed, are of no significance except where indicated and thus if it were desired, the computations which will be described next could have been initiated during any of the program steps 1018-1027. The first opera-

tion, performed during program step 1028, is to clear the Answer Register 53ab and to read the number stored in the ΔX register 109 if the X axis was identified as the fast axis or the number in the ΔY register 107 if the Y axis was so identified. Toward this end the PGA 409 associated with the ΔX register 109 is conditioned to apply a READ gating signal to the ΔX register 109 during program step 1028 if an FAX signal is produced by the flip-flop 455 (condition 11). Conversely, the PGA 415 is conditioned to apply a READ gating signal to the ΔY register 107 during program step 1028 if an FAX signal is produced by the flip-flop 455 (condition 11). Since the present description relates to the path segment 27b

whose X-component I is greater than its Y component J, the ΔX number will be read into the computer during program step 1028.

The ΔX number read into the computer 53 during program step 1028 appears in its Answer Register of 53a during program step 1029 and is effectively multiplied by 10 by shifting the contents of the Answer Register one position to the left. The appropriate control signal is applied to the SAL control input of the computer 53 by Program Gate Array 463 in FIG. 9c (condition 1). This multiplication is the first step in "building up" an 11 macromove look-ahead number. As a result of the operation, a number which is equal to 10 ΔX is stored in the Answer Register 53a. To increase this number to 11 ΔX , the number ΔX is again read from the register 109 into the computer 53 during program step 1030 in response to a READ signal from PGA 409 (condition 12). The result appears in the Answer Register 53a during program step 1031, but its numerical digits are not immediately utilized. Instead, during program step 1031 the sign of the number 11 ΔX is stored in a sign store flip-flop 465, labeled "ACNX" and shown in FIG. 9e as part of the Fast Axis Projected Overrun Detector 71.

In addition to the flip-flop 465, the detector 71 also includes a pair of AND gates 467 and 469. In the computer 53, the sign stage was previously described as a single JK flip-flop. In the exemplary system of FIG. 9, a (-) sign is represented by a logic 1 signal on the Q output of the sign flip-flop, and this is the output supplied to the COT(-) line of the computer output trunk 61. The output line COT(-) is directly connected to the J input of the ACNX flip-flop 465 in the detector 71 and is referred to as ACN (Accumulator Negative).

The inverse of the ACN signal, $\bar{ACN}$ is applied to the K input of the flip-flop 465 by means of an inverter 471 whose input is connected to the COT(-) line. Consequently, if the sign of the number 11 ΔX is negative, the ACNX flip-flop 465 is set and a logic 1 signal, referred to as ACNX, appears at its

Q output. On the other hand, if the number $11 \Delta X$ is positive, the ACNX flip-flop 465 is reset and a logic 1 signal labeled ACNX appears at its Q output.

Continuing with the look-ahead process, the sign is read into the ACNX flip-flop 465 during program step 1031 by means of a program gate 473 which is connected to the CP input of the flip-flop through an additional AND gate 475 enabled by a WP pulse from the decoder 173 of the Time Base and Digital Sweep 63 (FIG. 9b). Having stored the sign of the look-ahead number $11 \Delta X$ and with that number still in the Answer Register 53a during the following program step 1032, the upgraded macroposition number XCP, representing the position to be reached at the end of the current time period ΔT , is additively read into the computer 53 from the XCP register 117 by a READ gating signal from the Program Gate Array 421 (condition 3). The resulting sum ($XCP + 11 \Delta X$) appears in the Answer Register 53a during the following program step 1033 and represents the desired look-ahead point projected forward $11 \Delta X$ macromoves from the macroposition which is programmed to be reached at the end of time period $\Delta T1$. (See FIGS. 19 and 20).

Having produced the look-ahead point, it now remains to compute the projected overrun, if any. This is achieved in the exemplary system by subtracting the programmed X coordinate end point XCEP from the look-ahead point and using the sign of the difference to detect an overrun condition. The minuend of this operation is in the Answer Register 53a; however, the subtrahend, which is the XCEP number, cannot be read subtractively into the computer because its decimal point is located one position to the right of the decimal point of the subtrahend. (Compare XCEP and XCP in FIG. 15). Accordingly, during program step 1033 the number $XCP + 11 \Delta X$ is shifted one position to the right in the Answer Register 53a in response to a control signal applied to the SAR control input by the Program Gate Array 411 (condition 23).

With the number representing the look-ahead point $11 \Delta X$ in proper decimal point position, the XCEP number is subtractively read into the computer 53a during program step 1034 from the XCEP register 95 in response to a READ gating signal from its Program Gate Array 359 (condition 1). The computer 53 is put in the subtractive mode by a control signal applied to its SX input from the Program Gate Array 479 (condition 1).

In accordance with a specific aspect of the invention related to detecting an overrun, a comparison is made during program step 1035 between the sign of the remainder produced during the program step 1034 and the sign of the quantity $11 \Delta X$, previously stored in the ACNX flip-flop 465 in FIG. 9e during step 1031. It may be readily shown that, so long as the two signs are opposite, an overrun condition has not occurred. Thus, for the test performed during iteration $\Delta T1$ the look-ahead quantity $11 \Delta X$ is positive, since motion is to be toward the right in FIG. 19. When the end point coordinate $XCEP_2$ is subtracted from the look-ahead point projected forward $11 \Delta X$ past the upgraded macroposition XCP_1 , the remainder will be negative, since $XCEP_2$ is greater than $XCP_1 + 11 \Delta X$. Thus, the sign of the quantity $11 \Delta X$ is opposite that of the quantity $XCP_1 + 11 \Delta X - XCEP_2$. However, during the first iteration that the look-ahead point $XCP_1 + 11 \Delta X$ extends past the end point coordinate $XCEP_2$, so that the quantity which had been the lesser becomes the greater, the sign of the remainder of $XCP_1 + 11 \Delta X - XCEP_2$ will become the same as the sign of the quantity $11 \Delta X$.

The foregoing condition is detected by applying the ACN signal from the Answer Register 53a directly to a second input of the AND gate 467 which also receives the stored ACNX signal from the Q output of the flip-flop 465. Similarly, in addition to receiving the output $\overline{ACNX}$ from the $\overline{Q}$ output of the flip-flop 465, the AND gate 469 is also connected to receive the signal $\overline{ACN}$ directly from the output of the inverter 471. As a result, during the program step 1035 a logic 1 output is passed through the AND gate 467 and appears at its output only if both the quantity $11 \Delta X$ and the remainder of XCP_1

$+11 \Delta X - XCEP_2$ are negative. On the other hand, the second AND gate 469 produces an output only if both of the aforesaid quantities are positive. The outputs of the two AND gates 467 and 469 are fed to an OR gate 481.

During each of the time periods $\Delta T1 - \Delta T989$ in which the path segment 27b is generated, the look-ahead point XCP_1 , ΔX will be less than the programmed end point $XCEP_2$, the signs applied to the detector 71 during program steps 1031 and 1035 will differ, and no signal will appear at the output of the OR gate 481 during step 1035. During the first iteration, however, in which the initially lesser look-ahead point $XCP_1 + 11 \Delta X$ ($\Delta T990$ in the example of FIGS. 19 and 20) becomes the greater, the sign of $XCP_1 + 11 \Delta X - XCEP_2$ is reversed and a logic 1 signal is applied to the OR gate 481 by the AND gate 469 during step 1035. The resulting signal which appears at the output of the OR gate 481 is in effect an end of block warning signal utilized by the system to change its mode of operation a predetermined number of iterations before the specific time period ΔT in which a projected overrun is predicted to occur.

As indicated in the discussion of FIG. 19 for the exemplary path segment 27b, a projected overrun condition will not be indicated or predicted, for many iterations. Means are provided, however, to initiate in response to a signal from OR gate 481 of the detector 71 a change in the mode of operation of the system such that during selected ones of the following iterations adjusted macromoves are computed for the X and Y axes. Toward this end the output of the OR gate 481 is connected to the Program Gate Array 183, to serve as the input of the Time Base Mode Selector 69 in FIG. 9e. The output from the detector 71 is identified by the symbols SST. ("Same sign twice"). One of the program gates of the array 183 represented by the first condition listed therein (1035-LIN/S1-SST) is connected to produce a signal in response to an SST signal from the overrun detector 71 during program step 1035 if the system is generating a linear path and if the End of Block Counter 186 is in its first count state as indicated by a logic 1 level on the S_1 output of the decoder 187. Therefore, in response to an SST overrun indication, whether it be caused by a small projected overrun or a large one, the PGA 183 allows a PCP pulse to be applied to the End of Block Counter 186, advancing it at the end of program step 1035 from its first to its second count state, indicated by a signal on the S_2 output of the decoder 187.

The operation of advancing the End of Block Counter 186 to state S_2 , which can only occur when a projected overrun has been detected, appears in Column 1000 of FIGS. 21 and 23.

c "Retracting" the Look-Ahead Point by $\frac{1}{2}$ Macromove and Performing the Second Test

A second test, directed to determining the magnitude of an overrun, if any, is carried out during steps 1036-1042 of the iteration shown in FIG. 18 and Table IV.

First, the projected overrun which is still stored in the Answer Register 53a during program step 1036 is shifted two positions to the left in the register during program steps 1036 and 1037 in response to Shift Left control pulses from the PGA 463 (conditions 2 and 3). The rest of the iteration is shown in Table IV-A.

During steps 1038-1042 the macromove number stored in the ΔX register 109 is subtractively read into the computer 53 five times. Due to the shifting of the minuend in the register 53a two positions to the left, each subtractive reading of the ΔX number in the computer has the effect of reducing the number remaining in the Answer Register by $0.1 \Delta X$ so that, after the last of the five program steps 1038-1042, $\frac{1}{2} \Delta X$ will have been subtracted from the projected overrun. The necessary gating and control signals for carrying out the foregoing operations are listed in PGA 409 of the ΔX register 109 as condition 14. It will be noted that condition 14 depends on the presence of an FAX signal at the output of the FAX flip-flop 455. Had this not been present; that is, had the Y axis been the fast axis, the number in the ΔY register 107 would have been

read instead in response to gating signals indicated as condition 13 in its associated PGA 415.

tion shown in FIG. 21 is initiated during program step 1589. During this program step the End of Block Counter 186 is ad-

TABLE IV A

	1000 · LIN	1500 · LIN	1800 · LIN	1900 · LIN
38.....	SX, R/ΔX ? FAX			
39.....	R/ΔY ? FAX			
40.....	SX, R/ΔX ? FAX			
41.....	R/ΔY ? FAX			
42.....	SX, R/ΔX ? FAX			
43.....	R/ΔY ? FAX			
44 to 64.....	See note 1			
65.....			SL4	SL4
66.....			R/YR	R/XR
67.....			SR4	SR4
68.....			W/ΔY	W/ΔY
69.....			ZR1	ZR1
70.....			SL4	SL4
71.....			W/YR	W/XR
72 to 87.....	No computations			
88.....	Advanced E.O.B.			
89.....	1 coun ? S3			
90 to 98.....	No computations			
99.....				

Note 1: On step 1043 advance E.O.B. to S3 ? S2 · (ACN · ACNX + ACN · ACNX)

During step 1043 a test is performed on the remainder in the Answer Register 53a to determine whether or not the projected overrun exceeded a ½ macromove. The test is identical to that performed previously during step 1035; that is, the sign of the remainder present in the register 53a during step 1043 is compared with the sign of the look-ahead quantity 11 ΔX. If they are the same, it is an indication that even with the projected overrun reduced by a ½ macromove an overrun would occur; or, in other words, that the projected overrun is equal to or greater than a ½ macromove.

Referring to FIG. 9e, if movement is in a positive direction, a logic 1 signal will be applied to the OR gate 481 by the AND gate 469 on step 1043 during the first iteration in which the sign of the remainder in the register during step 1043 is the same as the stored sign of the look-ahead quantity 11 ΔX. Similarly, a logic 1 signal is applied by the AND gate 467 to the OR gate 481 during the first time period ΔT during which a similarity of signals occurs if movement is in a negative direction. By means of a program gate represented in the PGA 183 by condition 3, if a projected overrun condition was detected during a given time period ΔT as a result of the first test, causing the S2 output of the decoder 187 to be energized, then a second SST signal at the output of the detector 71 caused by a positive result from the second test will cause the End of Block Counter 186 to be advanced to its third count state and the decoder 187 to produce a signal on its S3 output.

3 Changing System Operation into Mode 2 in Response to Detection of a Large Projected Overrun

Since the projected overrun detected during program step ΔT990 is assumed to be large, a changeover needs to be made from the FIG. 18 iteration to an iteration which includes computational steps for deriving the adjusted macromove numbers which will be required during the following 10 iterations. FIG. 21 illustrates this iteration.

Referring to FIG. 21 it will be observed that the periodic updating of the XSC and the YSC registers 121 and 119 are the only operations performed during the second, third and fourth columns 1100-1400 of the programming chart. Up to this point the only difference between the FIG. 21 type iteration and that shown in FIG. 18 is that, during the first column on the programming chart in FIG. 21 both of the tests performed have resulted in projected overrun indications. For this reason the two blocks in FIG. 21, column 1000, calling for advancement of the EOB Counter to S2 and then to S3 are not crossed out. The changeover to operations characteristic of the itera-

tion shown in FIG. 21 is initiated during program step 1589. During this program step the End of Block Counter 186 is advanced to its fourth count state, provided that it is already in its third count state. This further advancement of the binary counter 186 has the effect of producing a signal on the S4 output of the decoder 187, which, in turn, enables the AND gates 177 of the Time Base Logic 67 thereby placing the system in its mode 2 operation. The significance of this change is that, in its mode 2 operation the system carries out the computations required to produce the adjusted macromoves necessary to hit the end point exactly.

To advance the binary counter 186 from its third to its fourth count state, the Program Gate Array 183 includes a gate, represented as condition 5, which is connected to produce the required stepping signal during program step 1589, provided a signal is present on the S3 output of the decoder 187. This does not occur for the example illustrated in FIG. 19 until time period ΔT990 and then only if the assumed end point is P2'. During that time period, however, with P2' being the assumed end point, in response to the condition created by advancement of the End of Block Counter 186 from state S1 to state S4 during the same time period ΔT, after program step 1589, or slightly after the first half of time period ΔT990, the system is switched over into its mode 2 operation as indicated by the headings 2600-2900 on the last four columns of the programming chart of FIG. 21. During each of the previous 989 time periods ΔT1-ΔT989 the system operates in its mode 1 and the operations which it performs are those shown in FIG. 18.

4. Data Processing Steps Associated with Calculating Adjusted Macromoves

In addition to the frequently repeated updating operations performed during each column, two series of computations are performed during columns 1800 and 1900 of FIG. 18. The first of these is almost identical to the operations performed in column 4800 of FIG. 17 and, similarly, the second series of operations performed in column 1900 of FIG. 18 is almost exactly the same as those carried out in column 4900 of FIG. 17.

Turning first to the computations performed during column 1800 in FIG. 18 during steps 1817-1864 the product J·Q is derived from the numbers stored in the registers 101 and 103. Thus, during program step 1817 (Table IV) the number Q is read from the register 103 in response to a gating signal applied thereto by its PGA 389 (condition 1). During the following program step 1818 the number J is read from the Register 101 by a gating signal from its PGA 391 (condition 6). During the same program step, a multiply command signal is applied

to the computer 53 from its PGA 393 (condition 5). In response to the multiply command signal, the operation J·Q is performed by the computer during program steps 1819-1864. This is followed by adding the "remainder" digits which were split off the product formed during the previous iteration. Toward this end, the product J·Q is shifted four positions to the left in the Answer Register 53a during program step 1865 by a Shift Left 4 control signal from the PGA 347 (condition 2). The four remainder digits are then read from the YR register 111 during program step 1866·LIN by a gating signal from its Program Gate Array 399 (condition 1). With the product of J·Q correctly aligned in the Answer Register 53a with the incoming digits from the YR register 111 the latter is directly added to the product and the sum appears in the Answer Register 53a during program step 1867·LIN. It is then shifted to the right four positions by Shift Right 4 control signal from the PGA 487 in FIG. 9c (condition 2). The digits remaining in the Answer Register 53a after the shifting operation represent the macromove number ΔY and are transferred into the ΔY 107 Register by a WRITE signal from its Program Gate Array 395 (condition 1).

During the following three steps the remainder digits of the product J·Q just completed are transferred to the YR register 111. First, during program step 1869 the digits in the Answer Register 53a are zeroed by the ZR1 control signal from the PGA 397 (condition 2). Next, during program step 1870, the four digits stored in the overhanging register 53b are transferred to the left by four positions by a Shift Left 4 control

undisturbed in separate registers. This is so because these quantities remain the same for all of the iterations during which one linear path segment is generated at a constant velocity until an overrun condition is detected. The reason why these quantities have been shown as being newly computed during each iteration is that it may be desirable in some instances to provide for modification of velocities during a segment by manual override or adaptive controls, and also in the case of circular interpolation, described below.

Returning now to time period $\Delta T990$ and continuing with the assumption that a large overrun has been detected, the first series of operations carried out by the system in Mode 2 during Column 2800 of FIG. 21 is directed to generating an adjusted macromove number $\Delta Y'$. Generally, this is achieved by subtracting the macroposition YCP from the end point coordinate YCEP, dividing the result by 10 and storing the resulting quotient. Preceding this sequence, however, the last digit of the macroposition number YCP is eliminated. The reason for this is that the YCP macromove number has one more digit than the YCEP end point coordinate from which it is to be subtracted. Were this extra digit not eliminated, it would prevent the subsequent division from producing adjusted macromove numbers that would fit in the ΔY register 107. The Y servo command register YSC also carries a digit corresponding to the extra digit of YCP (see FIG. 15), but this digit is not transmitted to the Y servo 77 over the lines 81.

The detailed steps performed by the system in Mode 2 are shown in Table V.

TABLE V
Steps performed in Modes 2+3 linear

	2800 · LIN	2900 · LIN	3900 · LIN
1 to 13.....	See Table III		
14.....			
15.....	R/YCP	R/XCP	
16.....	SAR	SAR	
17.....	ZR2	ZR2	Zero out the .00001 digit of XCP and YCP
18.....	SAL	SAL	
19.....	W/YCP	W/XCP	
20.....	CAR, R/YCEP	CAR, R/XCEP	
21.....	SAL	SAL	Compute YCEP-YCP and XCEP-XCP
22.....	R/YCP, SX	R/XCP, SX	
23.....	SAR	SAR	
24.....	W/ ΔY	W/ ΔX	Divide the result by 10 Store $\Delta X'$ and $\Delta Y'$
25.....			
26.....			
27 to 98.....	No computations		
99.....	Advance E.O.B. to S5		Advance E.O.B. by one count

signal from the PGA 347 (condition 5). Finally, during the program step 1871 the four digits now residing in stages 5, 6, 7 and 8 of the Answer Register 53a are transferred to the YR Register 111 by a WRITE signal from its Program Gate Array 399 (condition 2).

During the corresponding steps of column 1900 the same operations are performed as those carried out during 1800, but with the number "I" being used instead of the number "J" and with the ΔX and XR registers being used instead of the ΔY and YR registers to receive the result. These operations will not be described in detail. They are shown in Tables IV and IV-A, and the appropriate program and control gates required to perform them are identified in FIG. 9.

It should be noted that the multiplication of J·Q and I·Q to produce ΔY and ΔX macromove numbers need not be performed during each FIG. 18 iteration because these products remain the same for those iterations during which one linear path segment is generated at constant velocity. Thus, having derived the ΔY and ΔX quantities during the first FIG. 18 iteration during time period $\Delta T1$, these stored and left

To eliminate the last digit of the YCP number, it is read into the computer 53 from the YCP Register 115 during program step 2815 by a gating signal from the PGA 425 (condition 7). Lodged in the Answer Register 53a during the following program step 2816, the YCP number is shifted one position to the right by a shift signal applied to the computer by the PGA 411 (condition 26). The last digit of the YCP number is now in stage 9 of the Overhanging Register 53b and is eliminated by a ZR2 signal applied to the corresponding control input of the computer by the PGA 397 (condition 6). With its lowest order digit eliminated, the YCP number is shifted back toward the left by one digit position during program step 2818 in response to an appropriate control signal from the SAL PGA 463 (condition 6). The YCP number is now in a proper position in the Answer Register 53a to be transmitted over the output trunk line 61 and it is stored during program step 2819 in the YCP register 115 by a WRITE gating signal from its PGA 427 (condition 2). This completes the elimination of the lowest order digit of the macroposition number YCP.

To subtract the modified YCP number from the YCEP

number residing in the YCEP Register 97 the YCEP number is transferred to the computer 53 during program step 2820 by a READ gating signal applied by its associated PGA 487 (condition 2). During the same program step, just prior to the reading of the YCEP number, the Answer Register 53a is cleared by a CAR signal from PGA 351 (condition 12). Next, the YCEP number, the minuend just stored in the Answer Register, is shifted one position to the left to bring its decimal point into alignment with that of the subtrahend YCP (see FIG. 15). Accordingly, a Shift Left control signal is applied to the computer 53 during program step 2821 from PGA 463 (condition 7).

With the minuend YCEP properly positioned in the register 53, the subtrahend YCP is subtractively read into computer 53 during program step 2822. Toward this end a subtract control signal is applied to computer 53 from the PGA 479 (condition 3) and a READ gating signal is applied to the YCP register 115 by its PGA 425 (condition 8). The remainder, stored in the Answer Register 53a during program step 2823, is $10 \Delta Y'$, representing the sum of 10 adjusted Y axis macromoves, at the end of which the end Point YCEP will be reached exactly.

To derive from the $10 \Delta Y'$ number the adjusted Y axis macromove number $\Delta Y'$, the $10 \Delta Y'$ number in the Answer Register 53a is shifted one position to the right during program step 2823 by means of a control signal applied to the SAR input of the computer 53 from its PGA 411 (condition 27). The result is transmitted from the Answer Register 53a to the Y axis macromove register 107 during program step 2824 by the application to the register of a WRITE gating signal from its PGA 395 (condition 3). Thus, during the ninth column of the programming chart of FIG. 21, the adjusted macromove $\Delta Y'$ has been calculated and stored in the ΔY register 107. A similar series of calculations are performed during the tenth portion of the iteration represented by the column 2900 of the programming chart to derive from the macroposition number XCP and the end point coordinate XCEP the adjusted macromove number $\Delta X'$. These steps and the necessary gating means are indicated in Table V, and in FIG. 9 and will not be described in detail. Suffice it to say that, at the end of the series of operations, during program step 2924-LIN, the adjusted macromove number $\Delta X'$ becomes stored in the ΔX register 109.

5. Changing System Operation into Mode 3 in Order to Execute Adjusted Macromoves During Subsequent Iterations

With the adjusted macromoves $\Delta X'$ and $\Delta Y'$ computed and stored, the only operation remaining during the FIG. 21 iteration is to cause the system to change to its mode 3 operation. In this mode the system executes during each iteration one of the macromoves $\Delta X'$ and $\Delta Y'$ along the X and Y axes, as indicated in FIG. 22. Changeover of the system from its mode 2 operation to its mode 3 operation is initiated during program step 2999 by advancing the End of Block Counter 186 to its fifth count state signaled by a logic 1 voltage level on the S5 output of the decoder 187. Since the End of Block Counter is already in its count state 4, a single signal is sufficient and this signal is applied to the counter 186 by the PGA 183 (FIG. 9e) during program step 2999 as indicated by condition 17 in the PGA. In response to the signal on the S5 output of the decoder 187, the third group of AND gates 179 of the Multi Mode Time Base Logic is enabled, causing timing signals 3000-3999 to be recurrently generated, as indicated by the heading of the 10 columns in FIG. 22.

6 Operation of the System in Mode 3

The first of the FIG. 22 iterations is performed during time period $\Delta T991$ and is repeated eight more times, once during each of the following time periods $\Delta T992$ - $\Delta T999$. Each of these iterations comprises principally only two types of operations. First, the XSC and YSC registers are updated during each of the ten time periods $\Delta T/10$ represented by the 10 columns 3000-3900 of the FIG. 22 iteration. Second, the XCP and YCP registers 117 and 115 are upgraded during the 3000 column of the iteration. A comparison of the Programming

Chart of FIG. 22 with the basic Programming Chart shown in FIG. 18 will reveal that the updating and upgrading operations are performed during the same time periods. The only difference between the operations is that different numerical values are used. Consequently, Table III and the description given in reference to it in Section G.2.b. is equally applicable to the steps performed during the FIG. 22 iterations carried out during the time periods $\Delta T991$ - $\Delta T999$.

7 Changing the System to Mode 4 Operation Following Execution of the 9th Pair of Adjusted Macromoves

An additional step performed once during each of the FIG. 22 iterations, but not during the FIG. 18 iteration, is directed to registering the number of times that the FIG. 22 iterations have been performed. Toward this end, means are provided for advancing the End of Block Counter 186 one count state for each FIG. 22 iteration. This is achieved by applying a stepping pulse to the End of Block Counter 186 during program step 3999 of each FIG. 22 iteration. The stepping pulses originate from the PGA 183 and the particular Program Gate for generating them is indicated as condition 18. The FIG. 22 iteration is permitted to repeat nine times. After the last such iteration, the system is directed to leave its mode 3 operation and to enter its fourth mode. In this mode, previously discussed with reference to FIG. 17, computations are performed to derive a macromove for the next path segment after the end point of the current path segment has been reached. To bring about the desired changeover from mode 3 to mode 4 operation, the S14 output of the decoder is connected through the OR gate 191 to the fourth set of AND gates 181 in the Time Base Logic 67. These AND gates, as explained previously, cause the system to operate in its fourth mode when they are enabled. The S14 output of the decoder 187 becomes enabled at the end of the ninth FIG. 22 iteration, shown in FIG. 20 as appearing at the end of time period $\Delta T999$. By the end of this time period, nine iterations have been performed with the adjusted macromoves, leaving only one iteration to be performed with those adjusted macromoves in order to reach the programmed end point P2 exactly. This is the iteration shown in FIG. 17. Thus, during time period $\Delta T1000$ the updating and upgrading operations associated therewith are carried out with the adjusted macromove quantities $\Delta X'$ and $\Delta Y'$.

The FIG. 17 iteration also includes computational steps associated with producing another linear path segment. A slightly different type of iteration is also available for commencing a circular path segment, and this will be disclosed in a subsequent portion of this specification. The computational steps performed during the columns 4600-4900 of FIG. 17 have been discussed previously, in section G.2.a, in connection with initiating the generation of the path segment 27b. Thus, the completion of the updating and upgrading steps of the FIG. 17 iteration during time period $\Delta T1000$ completes the operations associated with generating the linear path segment 27b.

8. Changing System Operation from Mode 1 to Mode 2 in Response to the Detection of a Small Projected Overrun

The foregoing detailed description with reference to FIG. 20 of the iterations shown in FIGS. 18, 21, 22 and 17 was based on the assumption that a large overrun was detected during time period $\Delta T990$. It was seen that, in keeping with the analysis offered with reference to FIG. 19, it was necessary for the system to switch over to its next mode of operation during the same time period in which a projected large overrun was detected ————this, in order to calculate the adjusted macromove before the following time period ΔT begins. The iteration during which such a prompt switchover occurs is that shown in FIG. 21.

Let it be assumed, however, that instead of a large overrun, a small overrun was detected during time period $\Delta T990$. In this event it is desirable to defer the switchover of the system for one time period so that it occurs during the time period $\Delta T991$ following the time period $\Delta T990$ (during which the small projected overrun was detected). In this way, the first

FIG. 22 iteration during which adjusted macromoves are executed is also deferred, and does not occur until time period $\Delta T992$ (see FIG. 20).

It is in order to prevent a switchover from mode 1 to mode 2 in the time period ΔT during which a small overrun is detected that advancement of the End of Block Counter 186 through its second and third count states is made necessary before the further change of the counter to its fourth count state, and with it, switchover of the system to mode 2 can occur. This series of events cannot occur, however, if the detected overrun is a small one because advancement of the End of Block Counter 186 into its third count state will not occur during the same time period ΔT in which such an overrun was detected. It will, however, occur during the following time period ΔT .

Specifically, means are provided, in the form of a Program Gate represented by condition 4 of the PGA 183, for advancing the End of Block Counter 186 from its second to its third count state during a given time period ΔT if during the previous time period ΔT the counter was advanced only to its second count state.

The steps carried out during the time period in which a small overrun is detected and during the following time period are shown as the iterations appearing in FIGS. 23 and 24. A comparison of the FIG. 23 iteration with the FIG. 21 iteration shows that the principal difference between the two is that the event represented in FIG. 21 by the last block in its 1000 column is crossed out in the corresponding block in FIG. 23. The legend in the block in question states that the End of Block Counter 186 shall be advanced from its second to its third count state if the second test associated with overrun indicates that the overrun was greater than $\frac{1}{2}$ macromove. If, as assumed, the overrun is small and this test yields a negative result, the counter 186 is not advanced to its third count state, but instead, the iteration is completed with the counter still in its second count state as shown by the last block in Column 1900 of FIG. 23.

As a result of the failure of the End of Block Counter 186 to advance from its second to its third state during the Column 1000 of the FIG. 23 iteration, the further advancement of the counter into its fourth count state, shown to occur during the FIG. 21 iteration toward the end of its fifth Column 1500, also fails to occur so that the iteration is completed with the system in its Mode 1.

As a second result of the End of Block Counter 186 not advancing to its third state S3 during program step 1043 of time period $\Delta T990$, the following iteration, shown in FIG. 24 begins with the system still in Mode 1. However, during the FIG. 24 iteration, that is, during time period $\Delta T991$, the End of Block Counter 186 is advanced to S3 and finally, about halfway through $\Delta T991$, the operation of the system is changed from Mode 1 to Mode 2 in the same manner as was previously seen to occur during the FIG. 21 iteration performed during time period $\Delta T990$. Thus, the End of Block Counter 186, having been previously advanced to its third count state S3 on step 1014 of the FIG. 24 iteration is advanced to its fourth count state S4 during program step 1589 of the FIG. 24 iteration. With the counter 186 thus advanced, all events thereafter are the same as if a large overrun had been detected during the time period $\Delta T991$ during which the iteration of FIG. 24 is performed. Consequently, during the next nine time periods $\Delta T992 - \Delta T1000$, the iterations of FIG. 22 are carried out, following which, provided the next path segment is to be linear, the system is caused to switch during time period $\Delta T1001$ into operating mode 4 and the iteration shown in FIG. 17. The End Point is reached at the end of time period $\Delta T1001$.

H. Detailed Description of Staggered Circular Path Generation

1. The Concept

a. Assuming Components of Two Radius Vectors are Given:

Circular path generation in accordance with the present invention was explained in a previous section with reference to FIGS. 6a-6d and will be explained in greater detail presently

with reference to FIGS. 25a-25f. The latter figures show successive steps in generating a circular path progressing counter-clockwise. Referring in particular to FIG. 25a, it is assumed that movement is to be accomplished along a circular arc generated by a radius vector of length R which passes from an initial point P_0 on the arc through a series of points P_1, P_2, P_3, P_4, P_5 on the arc at the end of respective ones of a consecutive series of equal time periods T_1, T_2, T_3, T_4 and T_5 . It is also assumed that the data initially furnished includes the X component I_0 of the inverse of the radius vector R_0 which extends from the center C of the circle to the initial point P_0 on the arc, and that the information also includes the Y component J_1 of the inverse of the radius vector R_1 extending from the center C of the circle to the point P_1 . The inverse of the radius vector R_0 is simply that radius vector turned head for tail, as a straight line extending from the point P_0 to the circle center C . The reason why the X and Y components of the inverse of the radius vectors are given is that this convention has been adopted by the numerical control industry to define circular paths. Had the radius vector and its X and Y components been so adopted, the following description would be little changed. The same fundamental principles would still apply. In conformance with the present convention, in the following paragraphs and in the claims the quantities I and J will be referred to by the short names "inverse X component" and "inverse Y component" of a radius vector.

In generating the circular path in accordance with the invention, prior to the respective time periods T_1, T_3 and T_5 , signals are produced respectively representing numerically the X coordinate distances ΔX_{02} and ΔX_{24} between the points P_0, P_2 and P_4 . These quantities are shown vectorially in FIGS. 25b and 25f. In a similar manner, prior to the respective time periods T_2 and T_4 , signals are also produced respectively representing numerically the Y coordinate distances ΔY_{13} and ΔY_{35} between the points P_1, P_3 and P_5 . The first of these quantities is shown vectorially in FIG. 25d.

The ΔX signals are utilized during respective pairs of the time periods such as $T_1 - T_2$, and $T_3 - T_4$, during which a controlled member is driven under the control of the ΔX signals through the distances ΔX_{02} and ΔX_{24} along the X axis. Correspondingly, the ΔY signals are utilized during respective pairs of time periods $T_2 - T_3$ and $T_4 - T_5$ by driving the controlled member along the Y axis through the distances ΔY_{13} and ΔY_{35} . In accordance with a more detailed aspect of the invention, motion along both axes is kept substantially uniform during each of the pairs of time periods.

The manner in which the ΔX and ΔY quantities are developed will be explained next. The ΔX_{02} number, representing motion to be executed along the X axis during time periods T_1 and T_2 is derived from the inverse Y axis component J_1 of the radius vector R_1 in accordance with equation (22) in FIG. 25b. The derivation of the equation was given previously with reference to equation (15) in FIG. 6. The differences between equation (15) and (22) are due to the fact that the direction of motion shown in FIG. 25b is counter-clockwise, whereas FIG. 6c, to which equation (15) applies, shows motion in a clockwise direction. Secondly, the term $V \cdot \Delta T$ in equation (15) is replaced in equation (22) by ΔD , representing the chord which extends from point P_0 to point P_2 . Finally, the term, $\sqrt{I_1^2 + J_1^2}$ in equation (15) is replaced by (R) , the radius vector.

Having produced the macromove number ΔX_{02} prior to the time period T_1 , the macromove is executed by moving the controlled element along the X axis by a distance ΔX_{02} during the time periods T_1 and T_2 at substantially uniform speed. The next quantity to be derived is the Y axis macromove number ΔY_{13} . As shown in FIG. 25d, in order to calculate ΔY_{13} , the X coordinate distance from the point P_2 to the center C of the circle, that is, the inverse X component I_2 of the radius vector R_2 , must be known. I_2 is determined by advancing the inverse X component of R_0 by means of the X component ΔX_{02} of the chord CH_{02} which was derived in the previous step. This is achieved during time period T_1 by processing the previously stored signals representing I_0 and ΔX_{02} so as to derive

therefrom further signals numerically representing the quantity $I_0 - \Delta X_{02}$ in accordance with equation (23) in FIG. 25c. Having thus derived signals representing the inverse X component, I_2 of the radius vector R_2 , the Y component ΔY_{13} of the next chord CH_{13} , extending from P_1 to P_3 in FIG. 25d, is produced by processing the signals representing I_2 in accordance with equation (24) in FIG. 25d to derive therefrom signals numerically representing the quantity

$$-I_2 \cdot \frac{\Delta D}{R}$$

and storing the resulting signals. With the ΔY_{13} signals stored, the macromove ΔY_{13} along the Y axis is executed at substantially uniform speed during time periods T_2 and T_3 , as indicated in FIG. 25d. The stored ΔY_{13} macromove number is also used to produce the Y axis inverse component J_3 of the radius vector R_3 required to derive the next X axis macromove number ΔX_{24} . As shown in FIG. 25e, the component J_3 is derived by advancing the inverse Y component J_1 of the radius vector R_1 by means of the Y component ΔY_{13} of the chord CH_{13} . In data processing terms, during time period T_2 the stored signals representing J_1 and ΔY_{13} are processed to derive from them signals which numerically represent the quantity $J_3 = J_1 - \Delta Y_{13}$ (equation 25). Shortly thereafter and still during the time period T_2 , the derived signals are further processed in accordance with equation (26) in FIG. 25f to derive therefrom signals numerically representing the quantity

$$\frac{J_3 \cdot \Delta D}{R}$$

This quantity is the X component of chord CH_{24} extending from point P_2 to point P_4 in FIG. 25f. The signals representing the macromove number ΔX_{24} are stored and are used to control movement along the X axis during time periods T_3 and T_4 by an amount corresponding to the number ΔX_{24} .

In equations (22) and (26) of FIGS. 25b and 25f, the right-hand side of the equation is positive, whereas in FIG. 25d a negative sign appears. It will be understood that the particular signs shown in FIGS. 25b, 25c, 25d, and 25f are characteristic of a circular path progressing counterclockwise. As will be explained subsequently with reference to FIG. 29a, the signs of equations 22, 24, and 26 will be different when the movement is clockwise.

Recapitulating the foregoing analysis of staggered circular interpolation given with reference to FIG. 25, movement is along two series of staggered chords. The first series of chords is represented by CH_{02} and CH_{24} in FIGS. 25b and 25f. The second series of chords is represented by CH_{13} in FIG. 25d. It is not exactly accurate to say that movement is along either one of the two series of chords. Rather, movement along the X axis is represented by the X coordinate of the first series of chords CH_{02} and CH_{24} and movement along the Y axis is represented by the Y components of the second series of chords. Furthermore, movements along the two series of chords are mutually time-staggered so that it is rather difficult to visualize the actual path covered by a controlled member whose movements along the X and Y axes are represented by the X and Y components of the first and second series of chords referred to.

To clarify the actual path in space defined by the X and Y components of the two series of chords, FIG. 26a shows a first series of macromove chords extending from point P_0 to P_7 whose X components ΔX_{02} , ΔX_{24} , ΔX_{46} and ΔX_{68} are the macromoves executed along the X axis during time periods $T_1 - T_2$, $T_3 - T_4$, $T_5 - T_6$ and $T_7 - T_8$, respectively. FIG. 26b, on the other hand, shows a second series of macromove chords, including a half chord extending from point P_0 to P_1 and three full chords extending from P_1 to P_3 , P_3 to P_5 and P_5 to P_7 . The Y components ΔY_{01} , ΔY_{13} , ΔY_{35} and ΔY_{57} of the respective macromove chords in the second series are executed during time periods T_1 , $T_2 - T_3$, $T_4 - T_5$ and $T_6 - T_7$, respectively.

The composite of the motions represented by FIGS. 26a and 26b is shown in FIG. 26c which is a plot of the points P_0 through P_7 having the X and Y coordinates shown in FIGS. 26a and 26b. Because of the greatly exaggerated scale, the plotted points P_0 through P_7 in FIG. 26c visibly deviate from a true circle. Thus, the chords drawn in FIGS. 26a and 26b appear significantly spaced from the arcs upon which they lie. Under normal cutting conditions, however, adjacent ones of the points P_0 through P_7 would be less than 0.01 inches apart and the points P_0 through P_7 in FIG. 26c would be extremely close to a true circle.

b. Assuming that Components of Only a Single Radius Vector are Given

The foregoing explanation of staggered circular interpolation given with reference to FIG. 25 was based on the assumption that the initially available data would include the inverse X component I_0 of the radius R_0 and the inverse Y component J_1 of the radius vector R_1 . This assumption was made in order to simplify the initial detailed explanation of staggered circular interpolation. Normally, however, the components I_0 and J_1 would not be available. Instead, it may be expected that the starting data will comprise both inverse components I_0 and J_0 of a single radius vector R_0 . In accordance with yet another specific aspect of the present invention, a process is provided for deriving from the inverse X and Y components I_0 and J_0 the inverse Y component J_1 , thus yielding the quantities which were assumed to have been given in the previous analysis.

Referring to FIG. 27, prior to time period T_1 first and second sets of signals respectively representing the inverse X and Y components I_0 and J_0 of the starting radius vector R_0 are produced. The signals representing I_0 are processed to derive therefrom signals numerically representing the quantity $I_0 \cdot \Delta D/R$. From FIG. 27 it may be seen that $I_0 \cdot \Delta D/R$ represents twice the Y component ΔY_{01} of the chord shown in FIG. 27 to extend over a distance ΔD , from the point P_{-1} to the point P_1 on the circle.

As the next step in deriving J_1 , the signals representing $2 \Delta Y_{01}$ and J_0 are processed so as to derive from them signals numerically representing the quantity $J_0 - \Delta Y_{01}$. The resulting signals represent to a very close approximation the desired Y axis component J_1 . A slight error is involved in this process due to the fact that ΔY_{01} would have to be subtracted from J_0 , which is the Y coordinate distance from the center of the chord in FIG. 27 to the center C of the circle, in order that the result be exactly J_1 . By subtracting ΔY_1 from J_0 instead, an error which is equal to $J_0 - j_0$ is introduced. This error, however, is negligible and is visible in FIG. 27 only because of its greatly exaggerated scale.

2. Staggered Circular Interpolation as Carried Out by the System of FIG. 9

a. In General:

The numerical control system of FIG. 9 is operative in the mode to be described hereinafter to cause relative motion between two machine members along a circular path. The elements of the system which are of particular significance in carrying out this function include first and second storage means, shown as the ΔX register 109 and the ΔY register 107. By means which include the digital computer 53, a first series of numbers are stored in the first storage means at regularly spaced intervals ΔT . Respective ones of these numbers represent the X coordinate distances between a first series of equidistant points P_0 , P_2 and P_4 on a circle as shown in FIGS. 25a-f. Additional means also including the computer 53, are provided for storing in the ΔY register 107 a second series of numbers at regularly spaced intervals $\Delta T'$ which are staggered symmetrically with, but are of the same duration as the time intervals ΔT . Respective ones of the latter numbers represent the Y coordinate distances between a second series of equidistant points P_1 , P_3 and P_5 on the circle. As seen in FIGS. 25a-f, the points P_1 , P_3 and P_5 are staggered symmetrically with, but spaced the same distance apart as the points P_0 , P_2 and P_4 .

The object of generating the two series of numbers is, of

course, to provide successive macropositions which are to be reached along the X and Y axes during the respective time intervals ΔT and $\Delta T'$. Toward this end, means are provided which are responsive to the numbers stored in the ΔX register 109 for causing relative motion between the machine members during successive ones of the time intervals ΔT corresponding to respective ones of the X coordinate distances represented by the successive numbers stored in the ΔX register. Correspondingly, means are also provided which, in response to the numbers stored in the ΔY register 107 causes relative motion between the machine members during successive ones of the time intervals $\Delta T'$ corresponding to respective ones of the Y coordinate distances represented by the successive numbers stored in the ΔY register.

b. The Form of Starting Data for a Circular Path Segment

Before going into further detail as to the system, consideration will be given first to the form of the data which will be assumed to have been provided by means of the punched holes on the tape 37. Referring to FIG. 3, the tape will include in the form of punched holes a pair of numbers representing the X and Y coordinates of the desired end point of the circular arc that is to be executed by the system. These coordinates are referred to as XEP and YEP when they are initially stored in the buffer registers 47X and 47Y and are referred to as XCEP and YCEP when subsequently stored in the active registers 95 and 97. Thus, the first two numbers in a data block given for a circular path segment have the same significance and are identified by the same symbols as the corresponding numbers in a data block for a linear path segment.

The next two numbers in a data block for a circular path segment are the I and J numbers. Although they are analogous to the corresponding numbers for a linear block in that they represent X and Y coordinate distances, the I and J numbers for a circular block are different in an important respect. Thus, the I and J numbers in a circular data block represent the X and Y coordinate distances from a given point on the circle to which they relate to the center of that circle. In actual practice successive path segments follow one another as shown in FIG. 2. Consequently, the starting point of a given segment will be the same as the end point of the previous segment. From this it follows that the I and J values of the starting radius vector of a circular path segment, that is, the X and Y coordinate distances from the initial point on the path segment to the center of the circle may be given in the form of the X and Y coordinates of the center of the circle. In this event, the system would be required to perform the additional step of deriving from the end point coordinate of the previous path segment and from the coordinates of the center of the circular path segment, the I and J values of the starting radius vector for that path segment. However, in order to simplify this description, it will be assumed, as in the case of a linear block, that the I and J values of the starting radius vector are given.

Concluding the data block for a circular path segment, the fifth and sixth numbers are the path designation "G" and the macromove quotient "Q" as in the case of a linear data block. As explained in a previous section, the G number will be 02 for a clockwise circular path, 03 for a counterclockwise circular path.

The significance of the macromove quotient Q was explained in detail with reference to FIG. 6 and is expressed by the equation (8a) there shown. In physical terms, the macromove quotient Q is equal to the ratio of a macromove chord ΔD of a circular arc (see FIGS. 6a-6d) to the length of its radius vector. Stated as a very simple equation,

$$Q = \Delta D / R \quad (27)$$

c. Generating Successive I, J, ΔX and ΔY Numbers

Assuming that a circular path segment is to be executed next by the system, the numbers included in the data block representing that segment will have been transferred from the tape 37 to the buffer registers 47 well before the need for that data arose. Toward the end of the execution of the path segment immediately preceding the circular path segment, the data related to the circular path segment is transferred from the buffer registers 47 to the active registers 57. In the exem-

plary system shown in FIG. 9, the I register 99 serves as a means for initially storing the number I_0 representing the X coordinate distance from the starting point P_0 on the circle to the center of the circle. Similarly, the J register 101 serves as the means for storing the corresponding J_0 number.

As stated previously, the ΔX and ΔY registers 109 and 107 serve to receive and store the ΔX and ΔY numbers representing desired excursions along the X and Y axes. To derive the J_1 component in the manner explained with reference to FIG. 27, first control means are provided for causing the computer 53 to multiply the initially stored number I_0 by $\pm Q$ (+ if G_{02} , - if G_{03}). The result represents twice the desired initial extension of the circle along the Y axis and is stored in the ΔY register 107 as the first ΔY number.

Second control means are provided for causing the computer 53 to replace the initially stored J_0 number with a new number J_1 which is equal to the remainder of J_0 less one-half of the stored $2 \Delta Y_{01}$ number. The second control means also serves to cause the computer to multiply the resulting sum by $\pm Q$ (- if G_{02} , + if G_{03}) and to store the product, representing ΔX_{02} , the first desired extension of the circle along the X axis, in the ΔX register 109.

With the numbers I_0 and J_1 in the I and J registers 99 and 101, the starting conditions assumed in FIG. 25a have been attained. Ignoring initially the execution of the distances represented by the numbers ΔY_{01} and ΔX_{02} , third control means are additionally provided for causing the computer 53 to replace the initially stored number I_0 with a new number I_2 , which is equal to the remainder of I_0 less the stored first number ΔX . This third control means is also operative to cause the computer to multiply the resulting sum by $\pm Q$ (+ if CW, - if CCW) and to store newly (that is, as the new ΔY number) the resulting product (after having picked up the previous remainder YR), representing the next desired extension of the circle along the Y axis, in place of the stored first number ΔY . The operations initiated by the third control means are represented by the vector diagrams and equations appearing in FIGS. 25c and 25d.

To replace the stored number J_1 in the J register 101 with the remainder of that number less the newly stored number ΔY , to multiply this remainder by Q , and to store newly the resulting product (allowing for remainders), representing the next desired extension of the circle along the X axis, in place of the stored first number ΔX in the ΔX register 109, fourth control means are provided, which are effective to cause the computer 53 to carry out each of the necessary operations. Furthermore, each of the third and fourth control means is operative to repeat its assigned series of functions a predetermined time after the other has carried out its assigned series of functions, so that ΔX and ΔY numbers are newly stored alternately and repeatedly.

The succession of ΔX and ΔY numbers are utilized in the generation of the circle by means operative to extend the circle along the Y axis by amounts corresponding to one-half the first ΔY number and successive ones of the newly stored ΔY numbers and along the X axis by amounts corresponding to the first ΔX number and successive ones of the newly stored ΔX numbers. The extension of the circle is physical but may be either in the form of electrical position command signals or in the form of mechanical movement either of a single machine component along two axes relative to a stationary machine component or in the form of relative movement between two components moved at right angles to one another. In the exemplary system, extension of the circle is executed both electrically, in the form of macroposition signals developed by the computer 53 and stored in the XCP register 117 and YCP register 115 and mechanically, by movement of the machine element 14 by means of the servo drives 75 and 77.

The foregoing description of some of the concepts underlying staggered circular interpolation and of the manner in which these concepts are carried out by the system of FIG. 9 was based on the diagrams of FIGS. 25a-f which show a circular path which is generated in a counterclockwise direction.

The following detailed description will be patterned on the assumption that it is the path segment 27c shown in FIG. 2 which is being generated. In effect, the ensuing description will be a continuation of the previous detailed description relating to the generation and execution of the linear path segment 27b. That description had progressed, in section G. 2. d (8), to the end of the linear block which was reached either at the end of time period $\Delta T1000$ or $\Delta T1001$. Let it be assumed that the end point was reached at the end of time period $\Delta T1000$. If it were also assumed that the path segment following path segment 27b is also to be linear, then the iteration shown in FIG. 17 would be performed during $\Delta T1000$. That is, during the time period $\Delta T1000$ the system would operate in mode 4 and would execute, in 10 substantially equal micromoves, the last of the ten adjusted macromoves which were computed during time period $\Delta T990$. In the present case, however, where the path segment 27c is to be circular, the iteration carried out after the last of the FIG. 22 type of iterations is different from that shown in FIG. 17 and appears in FIG. 28.

The iterations carried out in accordance with FIGS. 17 and 28 are very similar. Both of them are directed to executing the last adjusted macromoves along the X and Y axes for the preceding linear path segment. They also resemble one another in that during both iterations data is transferred from buffer registers 47 to the active registers 57 in order to permit their use in computing the macromoves ΔX and ΔY for the following path segment; linear in the case of FIG. 17, and circular in the case of FIG. 28.

The FIG. 28 type iteration differs from the FIG. 17 type iteration in that, the computations shown in the last two columns of FIG. 28 are for deriving the initial macromoves for a circular path in contrast to the corresponding operations shown in the last two columns of FIG. 17, which represent operations for deriving the initial macromoves for a linear path.

As shown in FIG. 20, a FIG. 28 type of iteration may follow either a series of upwardly adjusted macromoves, which has been assumed to be the case, or a series of downwardly adjusted macromoves which are represented by the lower series of FIG. 22 iterations, the last of which occurs during time period $\Delta T1000$.

The steps involving transfer of data from buffer to active, shown to occur during column 4600 of FIG. 28, is the same as explained previously with reference to Table I and need not be repeated here. The first series of operations which are characteristic of FIG. 28 are those which occur during column 4800. These are shown in detail in Table II of section G. 2. a. (2)(a) to which reference shall again be made.

d. Data Processing Steps Performed Prior to the Generation of Signals Representing a Circular Path

The first series of steps 4817 ($G_{02} + G_{03}$) through 4864 ($G_{02} + G_{03}$) is directed to deriving the number $2 \Delta Y_{01}$ (see FIG. 27). This is obtained by multiplying I by Q (where $Q = \Delta D/R$) and storing the product in the ΔY register 107 and in the Y remainder register 111. Ignoring the sign of the product I·Q initially, it may be confirmed by comparison of the first and third columns of Table II that the steps performed during steps 17 through 64 of column 4800 ($G_{02} + G_{03}$) are essentially the same as the corresponding steps performed during column 4800- G_{01} . The steps 4817-4864 ($G_{02} + G_{03}$) associated with circular interpolation differ from the steps 4817-LIN - 4864-LIN principally in that the multiplicand is read from the I register 99 rather than from the J register 101. This difference is reflected by condition 9 in the PGA 400 associated with the I register 99, showing that a READ gating signal will be applied to that register during step 4818, provided that either a G_{02} or a G_{03} signal appears at the outputs of the G buffer decoder 221 in FIG. 9k. The corresponding condition 9 for PGA 391 of the J register 101 is not realized because it requires the presence of a G_{01} signal at the output of the decoder 221, which occurs only when the following path segment is to be linear.

The product of the multiplication performed during steps 17 through 64 in column 4800 ($G_{02} + G_{03}$) is written into the ΔY register 107 and the YR register 111 during steps 68 through 71 in exactly the same manner as described previously with reference to column 4800- G_{01} . Briefly, the product of I·Q is split into two parts, with the four lowest order digits being stored in the YR register 111 and the other four digits being stored in the ΔY register 107. Again, the same gating and control signals are utilized to carry out these steps as were used for performing the corresponding steps for a linear path segment and need not be referred to again. The following operation, however, does not find its counterpart in the computations performed for a linear path. It relates to computing $-0.5 \Delta Y$ in order to derive the required starting component J_1 in the manner explained previously with reference to FIG. 27. From FIG. 27 it will be recalled that the initially computed ΔY macromove number is referred to as $2 \Delta Y_{01}$, so that, in effect, the operation amounts to advancing the initially provided J_0 component by $0.5 (2 \Delta Y_{01})$.

The first step in the process is to clear the Answer Register 53a during step 4873 ($G_{02} + G_{03}$) and this is in response to an appropriate control signal from the PGA 351 (condition 19). During the same program step and during the following four program steps 4874-4877 ($G_{02} + G_{03}$) the $2 \Delta Y_{01}$ number stored in the ΔY register 107 is subtractively read into the computer 53 five times by means of five READ gating signals applied to the register by its associated PGA 415 (condition 16) and by means of five subtract control signals applied to the computer 53 by the SX PGA 479 (condition 10).

During the following step 4878 ($G_{02} + G_{03}$) the contents of the Answer Register 53a, representing $-5(2 \Delta Y_{01})$ are shifted one position to the right in response to a control signal from the PGA 411 (condition 30). This effectively causes the number in the Answer Register to appear as $-0.5(2 \Delta Y_{01})$. The J_0 number stored in the J register 101 is read into the computer 53 during the next time period 4879 ($G_{02} + G_{03}$) in response to a READ gating signal from PGA 391 (condition 3) and is automatically added by the computer to the $-0.5(2 \Delta Y_{01})$ stored therein. The result, $J_0 - 0.5 (2 \Delta Y_{01}) = J_0 - \Delta Y_{01}$, appears in the register 53a during program step 4880 ($G_{02} + G_{03}$) and is written into the J register as the number J_1 by a WRITE gating signal from its PGA 357 (condition 3).

Summarizing the operations performed thus far, following the entering of a circular block into the active registers 57, the first ΔY number, representing twice the desired initial extension along the Y axis, was derived from I_0 and Q and using the ΔY number and the initially provided number J_0 , the number J_1 was calculated and stored in the J register 101. As explained previously with reference to FIG. 25b, the number J_1 is required to derive the first X axis macromove number ΔX_{02} and the steps for achieving this are shown in column 4900 ($G_{02} + G_{03}$) of Table II.

The initial steps carried out during steps 4918-4971 ($G_{02} + G_{03}$) correspond almost exactly to the steps performed during the corresponding steps 4918-4971 G_{01} in preparation for computing the ΔX macromove number for the linear path segment 27b. The principal difference is that, instead of I the multiplicand is J so that, to transfer the multiplicand to the computer 53 a gating signal is applied to the J register 101 during step 4918 ($G_{02} + G_{03}$) from the PGA 391 (condition 10). The number thus transferred is J_1 and is multiplied by Q, with the product appearing in the Answer Register 53a after the program step 4964 ($G_{02} + G_{03}$). The product is then split into two parts and stored in the ΔX register 109 and the XR register 113. There are now stored in the ΔX register 109 and the ΔY register 107, appropriate macromove numbers for initiating generation of the first pair of macroposition numbers corresponding to the X and Y coordinates of the point P2.

Reference will now be made to a further provision which differentiates circular interpolation from linear interpolation. It is directed to selecting the proper sign for the multiplicand I or J depending upon whether the desired circular path is to

progress clockwise or counterclockwise. The rule governing the selection of sign may be easily derived by referring to FIGS. 29a and 29b. They are vector diagrams showing the signs of I, J, ΔX and ΔY in all four quadrants during the generation of a circle either clockwise (FIG. 29a) or counterclockwise (FIG. 29b).

As shown in FIG. 29a, when a circular path is generated clockwise, the sign of J is always opposite the sign of ΔX . On the other hand, the sign of I is seen to be always the same as the sign of ΔY . Consequently, in view of equations (18) and (19), whenever the circular path is to progress clockwise, $-J$ and $+I$ should be used in order that ΔX and ΔY have the proper sign.

The relative signs of I and ΔY are the exact opposite and the relative signs of J and ΔX are the same when the circular path to be generated is to progress counterclockwise, and this is shown in FIG. 29b. Thus, during all four quadrants, I and ΔY are of opposite sign, while J and ΔX are of the same sign. Therefore, when performing equations (18) and (19) the number J is used without a modification, but the number I is provided with a minus ($-$) sign. However, the I and J numbers still carry their own signs as part of the numbers.

In other portions of this specification and in certain of the claims the process of giving the I or J number a negative significance before multiplying it by Q is expressed in terms of multiplying the I or J number by $-Q$ for the sake of brevity. These of course are fully equivalent expressions, since giving a number a negative significance is equivalent to multiplying it by -1 .

The foregoing rules for selecting the sign of the multiplicand I or J are also indicated in FIG. 28 and require no further comment. The appropriate control signals for implementing these rules are indicated by conditions 13 and 14 in the Program Gate Array 479 as shown in FIG. 9c. Thus, when the I multiplicand number is read into the computer during step 4818, a control signal directing the computer 53 to attach a negative significance to that number will be applied to the computer if a G_{03} signal is produced by the GBUF decoder 221 in FIG. 9k; in other words, if the path to be interpolated is counterclockwise. Similarly, when the J multiplicand is read into the computer 53 during step 4918, a control signal is applied to the computer 53 directing it to attach a negative significance to that number if a G_{02} signal is produced by the decoder 221; that is, if the path to be interpolated is clockwise.

With the numbers ΔX_{02} , $2 \Delta Y_{01}$, I_0 and J_1 stored in the registers 109, 107, 99 and 101 and with the G number transferred to the G active register 105 during steps 4973-4974 ($G_{02} + G_{03}$) the tape reader may be started during program step 4995 ($G_{02} + G_{03}$) in order to transfer to the buffer registers 47 the data relating to the next path segment 27d in FIG. 2. Following this, the End of Block Counter 186 is advanced from its count state S14 which it had reached during the last of the FIG. 22 iterations through count states S15, S16 into count state S1 in the manner explained previously with reference to the corresponding program steps 4997- G_{01} - 4999- G_{01} in Table II. The last change in the count state of the End of Block counter 186 causes the system to switch over from its mode 4 operation into mode 1 in which the AND gates 175 of the Multi Mode Time Base Logic 67 in FIG. 9d are enabled. The operations comprising the iteration of the system in mode 1 when performing circular interpolation are shown in FIG. 31. Before turning to FIG. 31, however, reference will first be made to FIG. 30 to illustrate the physical significance of the steps performed thus far.

e. A Physical Representation of Macromoves and Micromoves Generated During the Initial Portion of a Clockwise Circular Path Segment

FIGS. 30a and 30b show in detail the macromoves and micromoves executed during the initial iterations of the circular path segment 27c. Thus, it will be understood that the initial point P_0 in FIGS. 30a-b corresponds to the point P_2 of the circular path segment 27c in FIG. 2. Furthermore, each of the

time periods ΔT associated with a complete iteration such as that which appears in FIG. 31 is shown as two time periods, such as T_1 and T_2 in FIGS. 30a and 30b. Thus, the first iteration $\Delta T1001$ associated with the execution of the path segment 27c is shown in FIGS. 30a and 30b as the time periods T_1 and T_2 .

From the starting data, comprising the I_0 and J_0 components of the starting radius R_0 , there was computed the macromove quantity $2 \Delta Y_{01}$ which was subsequently used to derive the J_1 component of the radius vector R_1 . The J_1 component was then used to compute during the steps shown in column 4900 of FIG. 28 the first macromove number ΔX_{02} which appears in FIG. 30b. Thus, at the end of the iteration shown in FIG. 28, which corresponds to time period $\Delta T1000$ in FIG. 20, and just prior to the time period T_1 shown in FIGS. 30a-b, there are stored in the ΔX and ΔY registers the numbers representing the required excursion along the Y axis to reach the Y coordinate of the point P_1 and the required excursion along the X axis to reach the X coordinate of the point P_2 . Stored in the J register 101 is the J_1 number and stored in the I register 99 is the number I_0 .

This, then, represents the state of the data at the start of time period $\Delta T1001$. Physically, the preceding linear path segment 27b has just reached P_0 in FIGS. 30a-b and the next type of iteration, illustrated generally in FIG. 31 performed first during time period $\Delta T1001$ and repeated many times during the following time periods, is ready to begin.

f. Data Processing Steps and Operations Performed by the System in Mode 1 While Executing a Circular Path Segment

1. Y Axis Data Used for Updating YSC Register Changes Midway Through an Iteration Due to Staggering

A comparison between FIG. 31, representing a typical iteration during staggered circular interpolation, with FIG. 18 representing the corresponding iteration during simultaneous linear interpolation reveals certain organizational similarities. First, as in FIG. 18, two strips extend horizontally across the top of FIG. 31, representing the updating of the XSC and YSC registers 121 and 119. These operations are identical with those performed during linear interpolation.

Although the updating operations during circular interpolation are the same as those performed during linear interpolation, there is a difference, due to the staggered nature of circular interpolation according to the invention, in the data which is being processed. From the detailed explanation of linear interpolation with reference to FIG. 18, it will be recalled that the updating of the XSC register 121 was done on the basis of the ΔX macromove number which was generated before the start of the iteration and which was valid for the entire iteration shown in FIG. 18. This is also the case with staggered circular interpolation. Thus, the ΔX_{02} number which was computed during the 4900 column of FIG. 28 is truly the macromove to be executed during the time periods T_1 and T_2 shown in FIG. 30b to which the time period $\Delta T1001$ of the first of many FIG. 31 iterations corresponds.

It was seen then, that the macromove data used to update the XSC register 121 remains the same for an entire circular iteration, just as in the case of linear interpolation. The same is not true, however, for the data used for updating the YSC register 119. This is so because the initially computed macromove number $2 \Delta Y_{01}$ is only valid for the Y axis up to the Y coordinate of point B_1 which is to be reached during the first half of the iteration shown in FIG. 31. Indeed, only half of this number, ΔY_{01} is utilized, this half representing the desired Y coordinate distance from the point P_0 to the point P_1 . Therefore, a new ΔY macromove number must be computed for the updating of the YSC register 121 during the column 1500 of FIG. 31 and for the following nine updatings of that register. This "new" ΔY macromove number is then used as the basis for updating the YSC register 121 during the second half of the FIG. 31 iteration in which it is computed and for the first half of the following FIG. 31 iteration. This staggered arrangement is reflected in FIG. 31 by using two update strips side-by-side, each extending across five columns of the FIG. 31 chart under

the single strip representing the updating of the XSC register 119 and extending across all 10 columns of the FIG. 31 chart.

The generation of the first pair of micropositions for the X and Y axes, as well as the subsequent micropositions during the FIG. 31 iteration, may be explained with reference to Table III. Each of the steps performed in accordance with that table, to which detailed reference was made in section G. 2. b. are also performed during staggered circular interpolation. Thus, during the first five steps of column 1000 of the FIG. 31 iteration, the XSC register 121 is updated by dividing the number ΔX_{02} by 10 and adding the resulting quotient, $0.1 \Delta X_{02}$, to the microposition number then in the XSC register (X_0). To update the YSC register 119, the number $2 \Delta Y_{01}$ in the ΔY register 107 is divided by 10 during the following five program steps, and the result, $1/5 \Delta Y_{01}$, is added to the number (Y_0) in the YSC register. Since the number (Y_0) represents the Y coordinate of the Point P_0 , the updated microposition number, $Y_0 + 1/5 \Delta Y_{01}$ represents the Y coordinate of the first microposition to be reached during the FIG. 31 iteration, and is so identified in FIG. 30a. During each of the subsequent columns 1100-1800 of the FIG. 31 iteration, the XSC microposition number is updated by $0.1 \Delta X_{02}$ and during the last column 1900 it is conformed to the upgraded XCP number which is $X_0 + \Delta X_{02}$. Similarly, during each of the four columns 1100-1400 the YSC microposition number is updated by $0.2 \Delta Y_{01}$, the last such update yielding a YSC microposition number which is $Y_0 + \Delta Y_{01}$.

2. Upgraded YCP Number Modified During Iteration Using New ΔY Macromove Number

Another significant difference between the iterations shown in FIGS. 31 and 18 lies in the manner in which the upgrading of the XCP and YCP registers 117 and 115 is carried out. In the case of linear interpolation, it will be recalled from FIG. 18 that the XCP and YCP registers are upgraded substantially concurrently, near the beginning of the iteration, with numbers which represent the macroposition which is to be reached at the end of the iteration. In the case of linear interpolation, such upgrading operations were seen to have a dual purpose. First, the upgraded XCP and YCP numbers are used as a preliminary step in the look-ahead process, whereby a prediction is made of the point expected to be reached at the end of the 11th macromove following the iteration during which the test is made. A second purpose of the upgrading operation was seen to be to provide numbers accurately representing the macroposition to be reached at the end of the iteration, numbers which can be used in the last updating of the XSC and YSC registers.

The upgrading of the XCP and YCP registers 117 and 115 in carrying out circular interpolation have a similar two-fold purpose. The XCP and YCP upgrading in the FIG. 31 iteration starts out in exactly the same manner as that carried out in a FIG. 18 operation for a linear interpolation. That is, the XCP register 117 is upgraded during steps 10-12 of the 1000 column and the YCP register 115 is upgraded during the steps 12-14. These steps are shown in detail in Table III. The

number $XCP_0 + \Delta X_{02}$ which is stored in the XCP register after the upgrading accurately represents the X coordinate X_2 of the macroposition P_2 which will be reached at the end of the iteration. The same is not true, however, of the number which will be stored in the YCP register after it has been upgraded. The reason for this is that, to upgrade the number in the YCP register, the macromove number $2 \Delta Y_{01}$ stored in the ΔY register 107 is used; that is, the macromove number $2 \Delta Y_{01}$ is added to the previous macroposition number Y_0 stored in the YCP register. However, the ΔY macromove number so used applies only up to the fifth microposition that is generated during the FIG. 31 iteration, that one which is produced during column 1400. For the following five micropositions of that iteration and for the first five micropositions of the following iteration, a new ΔY number representing the Y axis macromove ΔY_{13} must be computed which, it will be seen, is done during column 1400 of the iteration after the updating of the XSC and YSC registers.

The upgraded number $Y_0 + 2 \Delta Y_{01}$ which is in the YCP register 115 during column 1000 of the first FIG. 31 iteration is sufficiently accurate for purposes of the look-ahead process performed subsequently during that column. It is not sufficiently accurate, however, for deriving the macroposition number during column 1900 of the iteration. Consequently, means are provided for correcting the upgraded macroposition number in the YCP register 115 as soon as the new ΔY macromove number ΔY_{13} becomes available during the FIG. 31 iteration under consideration.

The manner in which this is carried out will be understood more readily if the upgrading of the number Y_0 in the YCP register during the column 1000 is thought of as projecting the number Y_0 presently in the YCP register ahead by the full macromove $2 \Delta Y_{01}$. Then, recognizing that only the first half of the projection $2 \Delta Y_{01}$ is correct and that the second half is in error, it may be seen that a correction can be effected by retracting the point which had been projected forward by the full macromove $2 \Delta Y_{01}$ by one-half of that macromove; that is, ΔY_{01} , and by then projecting forward again by one-half of the newly computed macromove ΔY_{13} which is truly applicable to the second half of the FIG. 31 iteration.

To retract the point which had been projected forward by a full macromove $2 \Delta Y_{01}$ during column 1000, the number in the upgraded YCP register is "downgraded" by one-half ($2 \Delta Y_{01}$) during column 1300; that is, one-half of the "old" ΔY number $2 \Delta Y_{01}$ is subtracted therefrom. Then, after the "new" number ΔY_{13} has been computed during column 1400 of the FIG. 31 iteration, one-half of that number is added to the "downgraded" YCP number, thereby again upgrading it by one-half ΔY truly to represent the Y coordinate ($Y_0 + \Delta Y_{01} + \frac{1}{2} \Delta Y_{13}$) of the macroposition P_2 to be reached at the end of the iteration (see FIG. 30a).

Reference will now be made to Tables VI and VI A which show in detail all of the steps characteristic of the FIG. 31 iteration with the exception of the operations performed during the first 13 steps in each of the 10 columns comprising the iteration, these being shown in Table III.

TABLE VI
Steps performed in Mode 1 circular

	1000-CIRC	1300-CIRC	1400-CIRC	1500-CIRC	1900-CIRC
0 to 13.	See Table III				
14... Advance E.O.B. ? S2					
15... R/J		R/YCP	R/I	R/YCP	R/J
16... CX		SAL	R/ ΔX , SX W/I, CAR	SAL	R/ ΔY , SX W/J, CAR
17... C/FAX ? $\geq$ CAR, FP		R/YCPR	R/Q R/ $\pm I$ and multiply by Q	R/YCPR	R/ $\pm J$ and multiply by Q
18... R/J ? FAX R/I ? FAX SAR, C/UBFF		R/Y Δ , SX		R/ ΔY	
19... C/LBFF R/YCP ? FAX		R/Y Δ , SX		R/ ΔY	
20... R/XCP ? FAX		R/Y Δ , SX		R/ ΔY	
21... SAR CX		R/Y Δ , SX		R/ ΔY	

TABLE VI - Continued
Steps performed in Mode 1 circular

	1000-CIRC	1300-CIRC	1400-CIRC	1500-CIRC	1900-CIRC
0 to 13...	See Table III				
22...	R/YCEP ? FAX R/XCEP ? FAX	R/YΔ, SX		R/ΔY	
23...	P/UBFF ? > CAR, FP, SX	W/YCPR		W/YCPR	
24...	R/J ? FAX R/I ? FAX	SAR		SAR	
25...	SAR R/YCP ? FAX	W/YCP	Carry out ±IXQ multiplication	W/YCP	Carry out ±JXQ multiplication
26...	R/XCP ? FAX				
27...	SAR				
28...	CX R/YCEP ? FAX				
29...	R/XCEP ? FAX P/LBFF ? < CAR				
30...	R/ΔX ? FAX R/ΔY ? FAX				
31...	W/ACNX R/XCP ? FAX				
32...	R/YCP ? FAX				
33...	SAR				
34...	SX R/XCEP ? FAX R/YCEP ? FAX				

TABLE VI A

	1000-CIRC	1300-CIRC	1400-CIRC	1500-CIRC	1900-CIRC
35.....	See Note 1				
36.....	SAL				
37.....	SAL				
38.....	SX R/ΔY ? FAX		Carry out IXQ multiplication		Carry out JXQ multiplication
39.....	R/ΔX ? FAX				
40.....	R/ΔX ? FAX				
41.....	R/ΔX ? FAX				
42.....	R/ΔX ? FAX				
43.....	See Note 2				
44 to 64...					
45.....					
46.....					
47.....					
48.....					
49.....					
50.....					
51.....					
52.....					
53.....					
54.....					
55.....					
56.....					
57.....					
58.....					
59.....					
60.....					
61.....					
62.....					
63.....					
64.....					
65.....					
66.....					
67.....					
68.....					
69.....					
70.....					
71.....					
72 to 88...	No computations				
89.....				Advance	
90.....				E.O.B. ?S3	
91.....				E.O.B. S4	
92.....				E.O.B. M3	
93.....				E.O.B. M3	
94.....				E.O.B. M3	
95.....				E.O.B. M3	
96.....				E.O.B. M3	
97.....				E.O.B. M3	
98.....				E.O.B. M3	
99.....				E.O.B. M3	

Note 1: On step 1035 advance E.O.B. counter to S2 ? S1-SST-CIRC-UB-LB.

Note 2: On step 1043 advance E.O.B. counter to S3 ? S2-SST.

The first operation, which will be described with reference to Table VI will be the "downgrading" and subsequent upgrading of the YCP register during columns 1300 and 1500 of a FIG. 31 iteration. For a reason similar to that which caused a YR register 111 to be provided to store the four least significant digits of the ΔY macromove number a YCPR register 116 is provided to store the least significant digit of the upgraded YCP macroposition number, with the other five digits of the number being stored in the YCP register 115.

Continuing to assume that the FIG. 31 iteration under discussion is that which corresponds to the time periods T₁, T₂ in FIGS. 30a-b, the number in the YCP register 115 after its upgrading during column 1000 will represent the Y coordinate of the point P₀ projected forward; that is, upward by the macromove 2 ΔY₀₁. Algebraically, it is $YCP_0 + 2 \Delta Y_{01}$. The first step, then, to correct this number during column 1300 is to subtract one-half of the macromove number, ΔY₀₁, therefrom. Toward this end, during step 1315, the five highest order digits of the YCP number, which is $YCP_0 + 2 \Delta Y_{01}$ are transmitted from the YCP register 115 to the computer 53 in response to a READ gating signal from its associated PGA 425

(condition 5). These digits become stored in digit stages 2-8 of the Answer Register 53a during the following program step 1316: CIRC and are shifted one position to the left by a control signal from PGA 463 (condition 4). This frees stage 8 of the Answer Register 53a to receive the lowest order digit of the YCP number from the YCPR register 116 during program step 1317: CIRC. The appropriate READ gating signal is applied to the YCPR register 116 by a program gate 489 through an OR gate 491. It will be noted that the highest order digit of the YCP number, which normally resides in digit stage 2 of the Answer Register 53a, is now in digit stage 1 of the Answer Register so that, in effect, the number presently in the Answer Register 53a is $10(YCP_0 + 2 \Delta Y_{01})$.

In order to subtract half of the ΔY number 2 ΔY₀₁ from the inaccurately upgraded macroposition number $YCP_0 + 2 \Delta Y_{01}$, the entire number 2 ΔY₀₁ is subtracted five times from the number $10(YCP_0 + 2 \Delta Y_{01})$ in the Answer Register 53a, giving the result $10(YCP_0 + \Delta Y_{01})$. This result is then simply divided by 10, giving the "downgraded" macromove number $YCP_0 + \Delta Y_{01}$. Accordingly, the macromove number 2 ΔY₀₁ is subtractively read from the ΔY register 107 into the computer

53 five times, once during each of the program steps 1318-1322 CIRC. The READ gating signals required are represented by condition 14 of program gate 415 associated with the ΔY register 107 and by condition 5 of the SX PGA 479 in FIG. 9c. The result, $10(YCP_0 + \Delta Y_{01})$ appears in the Answer Register 53a during program step 1323 CIRC.

The lowest order digit of the result is transmitted from digit stage 8 of the Answer Register 53a to the YCPR register 116 during program step 1323 CIRC by a WRITE gating signal applied to the register 116 by a program gate 493 through an OR gate 495. The remaining five digits of the result are shifted toward the right from stages 1-7 to stages 2-8 during program step 1324 CIRC in response to an appropriate control signal from the PGA 411 (condition 24). Thus, the lowest order digit of the number $10(YCP_0 + \Delta Y_{01})$ has been stored in the YCPR register 116 and the remaining digits have been effectively divided by 10 so that what is stored in the Answer Register 53a during program step 1325 CIRC are the first five digits of the number $YCP_0 + \Delta Y_{01}$. These digits are transferred during the same program step to the YCP register 115 by a WRITE gating signal applied to the register from PGA 427 (condition 12). Thus, the number in the YCP register 115 has been "downgraded" to the middle of the FIG. 31 iteration and accurately represents the Y coordinate $YCP_0 + \Delta Y_{01}$ of the macroposition P_1 which is to be reached at the end of the first half of the FIG. 31 iteration; that is, at the end of the time period T_1 in FIG. 30.

During the following column 1400 of the FIG. 31 iteration, the new macromove number for the Y axis, ΔY_{13} , is derived and becomes available toward the end of column 1400. It is used during column 1500 to upgrade the "downgraded" macromove number in the YCP register to the value representing the Y coordinate of the point P_2 that is to be reached at the end of the FIG. 31 iteration. This is carried out by adding to the "downgraded" number $YCP_0 + \Delta Y_{01}$ in the YCP register 115 half of the new macromove number ΔY_{13} , producing the correct result, $YCP_0 + \Delta Y_{01} + 0.5 \Delta Y_{13}$. That this result is correct may be confirmed from FIG. 30a.

Turning to Table VI, the first step is to transmit the first five digits of the number $YCP_0 + \Delta Y_{01}$ from the YCP register 115 to the computer 53. This occurs during program step 1515 CIRC and is done by applying a READ gating signal to the YCP register 115 from the PGA 425 (condition 6). The seven digits are transferred from digit stages 2-8 to digit stages 1-7 of the Answer Register 53a during the following program step 1516 CIRC by a control signal from the PGA 463 (condition 5). The sixth and least significant digit of the number $YCP_0 + \Delta Y_{01}$ is transmitted from the YCPR register 116 to the computer 53 and, in particular, to its digit stage 8 during program step 1517 by a READ gating signal applied to the YCPR register from a program gate 497 through the OR gate 491. The number now stored in the Answer Register 53a is $10(YCP_0 + \Delta Y_{01})$. During the following five program steps 1518 CIRC - 1522 CIRC the new macromove number ΔY_{13} is added five times to the number in the Answer Register 53a by a succession of five READ gating signals represented by condition 15 of the PGA 415 associated with the ΔY register 107. The result, $10(YCP_0 + \Delta Y_{01} + 0.5 \Delta Y_{13})$, appears in the Answer Register 53a during program step 1523 CIRC. Its lowest order digit is transferred from digit stage 8 of the Answer Register 53a to the YCPR register 116 during the same program step by application of a WRITE gating signal to the register from its associated program gate 499 through the OR gate 495. The remaining seven digits are shifted one position to the right from digit stages 1-7 of the Answer Register 53a during the following program step 1524 CIRC by a shift control signal from PGA 411 (condition 25). Consequently, during program step 1525 CIRC the digits appearing in stages 2-8 of the Answer Register 53a represent the highest order digits of the number $YCP_0 + \Delta Y_{01} + 0.5 \Delta Y_{13}$. These digits are transmitted during the program step 1525 CIRC to the YCP register 115 by application of a WRITE gating signal to that register from its associated PGA 427 (condition 4). This completes the

second upgrading of the YCP number in the YCP register 115 so that it now represents correctly the Y coordinate of the point P_2 to be reached at the end of the FIG. 31 iteration (See also FIG. 30a).

3. Computing the New ΔY Macromove Number

Attention will now be directed to the specific steps carried out to compute the new Y axis macromove number ΔY_{13} . Referring first to FIG. 31, and in particular to column 1400 thereof which is taken to occur just before the time period T_2 in FIG. 30, the first task is to derive from the ΔX_{02} number in the ΔX register 109 and from the I_0 number stored in the I register 99 the I_2 component of the radius vector R_2 shown in FIG. 30a. As may be seen from FIG. 30a and recalling the previous discussion with reference to FIGS. 25c and 25d, the value I_2 will be required to compute the quantity ΔY_{13} . This operation is referred to in FIG. 31 as "upgrading I by subtracting ΔX ." Thus, it may be confirmed by reference to FIG. 29a that, when a circular path is to be generated clockwise, the macromove component ΔX must be subtracted from the I component in order that the change in the latter be in the right sense. Taking as an example the fourth quadrant, both I and ΔX are positive. Whatever the angular position of the radius vector with which a given I component is associated, as that radius vector is advanced clockwise, its I component must diminish. Clearly, then, subtraction of ΔX will yield the correct result. The same is true for a circular path which is to be generated counterclockwise, as may be seen from FIG. 29b. Taking the first quadrant as an example, the ΔX and I vectors are both negative. Again, regardless of the angular position of the radius vector, as it is rotated counterclockwise, its I component must shrink; that is, it must be a smaller negative number. Consequently, the subtraction of the negative ΔX number from the negative I number will result in a smaller negative number for the I component.

To subtract ΔX_{02} from I_0 , I_0 is transmitted from the I register 99 to the computer 53 during program step 1415 CIRC by a READ gating signal from PGA 400 (condition 6). During the following program step 1416 CIRC, the ΔX_{02} number is subtractively read into the computer 53 from the ΔX register 109. To this end, a READ gating signal is applied to the ΔX register 109 during program step 1416 CIRC from its associated PGA 409 (condition 13) and a subtract control signal is applied to the computer 53 from its associated PGA 479 (condition 6). The result, $I_0 - \Delta X_{02}$, appears in the Answer Register 53a during program step 1417 CIRC and is transmitted to the I register 99 as the upgraded or new I component number I_2 by the application of a WRITE gating signal to the I register 99 from its PGA 355 (condition 1).

With the desired I component number I_2 in the I register 99, the next operation is to derive the quantity $\Delta Y_{13} = I_2 \Delta D/R$. It will be recalled that $(\Delta D/R) = Q$, which is stored in the Q register 103. Consequently, the computation is simplified to $I_2 Q$ and is seen in FIG. 31 to occur during program steps 1417 CIRC - 1464 CIRC. Specifically, with reference to Table VI, the Q number is read into the computer 53 from the Q register 103 during program step 1417 CIRC by a gating signal from its PGA 389 (condition 2) after clearing of the Answer Register 53a by a control signal from the PGA 351 (condition 10). The upgraded I_2 number is transmitted to the appropriate storage register in the computer 53 from the I register 99 by a READ gating signal applied to the register from its PGA 400 (condition 7) during the following program step 1418 CIRC. During the same program step, a "multiply" control signal is applied to the computer 53 from its PGA 393 (condition 1). It should be noted that, since the path being generated is clockwise, the I_2 number used for the subsequent multiplication process is to be positive, in keeping with the rules developed with reference to FIG. 29. Had the illustrated path been progressing in a counterclockwise direction, the I number would have been read into the computer 53 with a sign reversal, caused by an SX control signal, as indicated by condition 11 in PGA 479 of FIG. 9c.

The operation $I_2 Q$ is completed by the computer 53 during

program step 1464 shown in Table VI-A and the result appears in the Answer Register 53a during program step 1465 CIRC.

The following operations which are carried out during steps 1465 CIRC - 1471 CIRC are the same as those shown to occur during steps 1865 LIN - 1871 LIN, as discussed previously with reference to FIG. 18 and Table IV-A. By means of these operations any number in the YR register 111 is added to the product $I_2 \cdot Q$, the first four digits of the sum are stored as the number ΔY_{13} in the ΔY register 107 and the four least significant digits in the YR register 111. The necessary control and gating signals are indicated by condition 1 of the SL4 PGA 347, condition 2 of the RYR PGA 399, condition 1 of an SR4 PGA 501 in FIG. 9c, condition 2 of the W/ ΔY PGA 395, condition 1 of the "ZR1 + ZR2" PGA 397, condition 4 of the SL4 PGA 347 and condition 1 of the W/YR PGA 398.

Having computed the macromove number ΔY_{13} during the 1400 column of the FIG. 31 iteration, that number is used for three purposes, some of which have already been explained. First, the macromove number ΔY_{13} is used during the next column 1500 and during each of the following four columns of the present FIG. 31 iteration and during the first four columns of the following FIG. 31 iteration to generate successive Y axis microposition numbers by repeated updatings of the YSC register 119. This is performed during steps 05, 06 and 10 of each of the columns 1500 - 1800 and 1000 - 1400 of the iteration as shown in Table III, by dividing the macromove number ΔY_{13} by 10 and successively adding the quotient to the number stored in the YSC register.

The second purpose for which the number ΔY_{13} is used is to upgrade the number in the YCP register 115 to the value which it is to have at the end of the FIG. 31 iteration. This was explained in the immediately preceding sections. The third purpose for which the number ΔY_{13} is used is to upgrade the component number J_1 which is presently in the J register 101 by subtracting from that number the macromove number ΔY_{13} . By inspection of FIG. 30a, it may be seen that the result will be J_3 ; that is, the Y component of the point P_3 . It is this component which is necessary to compute

$$\Delta X_{24} = \frac{-J_3 \cdot D}{R} = -J_3 \cdot Q.$$

The upgrading of the component number J_1 and the subsequent computation of the new macromove number ΔX_{24} required for the following FIG. 31 iteration is performed during the 1900 column of the current FIG. 31 iteration; that is, toward the end of time period ΔT_{1001} . The first step in the sequence, performed during program step 1915 CIRC is to transmit the number J_1 from the J register 101 to computer 53 by means of a READ gating signal from its associated PGA 391 (condition 1). The new macromove number ΔY_{13} is then read subtractively into the computer 53 from the ΔY register 107. The necessary gating and subtract signals are indicated by condition 17 of the R/ ΔY PGA 415 and condition 7 of the SX PGA 479. The result, $J_3 = J_1 - \Delta Y_{13}$ appears in the Answer Register 53a during the following program step 1917 CIRC and is transmitted to and stored in the J register 101 during that program step by application to the register of a WRITE gating signal from its associated PGA 357 (condition 1).

This completes the upgrading of the J vector component so that, in effect, there is now stored in the J register 101 the inverse Y component of the radius vector R_3 which has been "advanced" from P_1 to P_3 . Next, the Answer Register 53a is cleared during program step 1917 CIRC by a control signal from CAR PGA 351 (condition 11) and the multiplier Q is entered by a control signal from the R/Q PGA 389 (condition 4). The multiplicand J_3 is entered into the computer during program step 1918 CIRC from the J register 101 by a gating signal from its PGA 391 (condition 2). The circle being generated is clockwise; therefore, a $-J_3$ must be used and consequently the J_3 number is read into the computer 53 subtractively, as indicated by condition 12 of the SX PGA 479. With the multiplicand $-J_3$ just written into the appropriate storage

device in the computer 53 and with the multiplier Q stored in its associated storage device in the computer, the multiplication process $-J_3 \cdot Q$ is initiated during program step 1918 CIRC by a "multiply" control signal represented by condition 2 of the X PGA 393. The multiplication process $-J_3 \cdot Q$ is carried out during the ensuing program steps 1918 CIRC - 1964 CIRC indicated in Tables VI and VI-A. The product, $-J_3 \cdot Q$, appears in the Answer Register 53a during the next program step 1965 CIRC. During program steps 1965 CIRC - 1971 CIRC, any digits stored in the XR register 113 are added to the product. The resulting eight-digit number in the Answer Register 53a is split into two parts. These parts are stored in the ΔX register 109 as the macromove number ΔX_{24} , and in the XR register 113 as the "remainder" of ΔX_{24} in the same manner as described previously with reference to steps 1965 - 1971 of Table IV-A.

In executing the clockwise circular path segment 27c, the system repeats the FIG. 31 iteration many times. Thus, during the following time period ΔT_{1002} the I_2 component number is upgraded to $I_4 = I_2 - \Delta X_{24}$, and, using the new I_4 component number, the next Y axis macromove number $\Delta Y_{35} = I_4 \cdot Q$ is computed during column 1400, or, with reference to FIG. 30, just before the end of time period T_3 . The new macromove number ΔY_{35} is then executed at substantially uniform speed and in ten substantially uniform increments during time periods T_4, T_5 . Subsequently, during column 1900 of the second FIG. 31 iteration, the component J_3 is upgraded to $J_5 = J_3 - \Delta Y_{35}$, this occurring just before the end of the time period T_4 in FIG. 30. The new component number J_5 is then used to compute the next macromove number $\Delta X_{46} = -J_5 \cdot Q$ which is executed during time periods 5 and 6 shown in FIG. 30, again in ten equal increments due to the periodic updating of the XSC register.

g. The Particular Advantages of a Two-Level Interpolation Process in Generating a Circular Path Segment

It should be noted in connection with linear interpolation that the ΔX and ΔY macromove numbers need not be computed newly for each successive iteration as the linear path segment is being generated, provided there were the means for extremely fine resolution, i.e., the remainder is insignificant. Therefore, it would have been theoretically possible in the case of linear interpolation to use a single level interpolation in which the $\Delta X/10$ and the $\Delta Y/10$ micromove numbers would have been computed during the initial FIG. 17 iteration simply by carrying out the computations $J \cdot Q/10$ and $I \cdot Q/10$. In other words, linear interpolation could be carried out by computing micromoves and micropositions only and without the intermediate computation of macromoves and macropositions. In the case of circular interpolation this situation is significantly different. As the circular path is being generated, the macromove numbers ΔX and ΔY continually change. Thus, it is impossible initially to compute a pair of macromove numbers ΔX and ΔY or a pair of micromove numbers $\Delta X/10$ and $\Delta Y/10$ which can then be used for the entire process. To the contrary, it is apparent from FIGS. 30a and 30b that the distance to be covered during a given time period ΔT continually changes along both axes. But to compute each successive micromove and microposition by multiplying a radius vector component I or J by the number Q would put very heavy demands upon a computer. On the other hand, to compute only macromoves ΔX and ΔY and macropositions XCP and YCP would result in the generation of relatively widely spaced-apart points and a surface which would not be sufficiently smooth for all purposes. It is here then, that the two-level interpolation process is used to the best advantage. The relatively lengthy process of $J \cdot Q$ and $I \cdot Q$ is carried out only for deriving the macromove numbers ΔX and ΔY and the much simpler process of dividing each new macromove number ΔX and ΔY by 10 and adding the result to the microposition numbers stored in the XSC and YSC registers 121 and 119 is used for deriving the intermediate micropositions.

h. Terminating the Circular Path Segment

(1) Macromoves are not Adjusted and the Path Segment is not Necessarily Reached at the End of an Iteration

In several respects the manner in which a circular path segment is terminated in accordance with the present invention is similar to that discussed previously in connection with terminating a linear path segment. During each time period ΔT look-ahead signals are produced representing the projected fast axis coordinate of the point which is expected to be reached along the circular path segment during a following time period. These look-ahead signals are compared with signals representing the corresponding coordinate of the designated End Point of the path segment and a warning signal is produced during the first time period in which the comparison indicates that the fast axis coordinate represented by the look-ahead signals had passed the corresponding coordinate of the designated end point. Furthermore, as with a linear path segment, a determination is also made during each time period ΔT of the extent of the projected overrun. If the fast axis coordinate represented by the look-ahead signals is more than one-half macromove past the corresponding coordinate of the End Point of the path segment an auxiliary signal is produced. Movement along the circular path segment is terminated at the end of the time period ΔT preceding the particular time period for which an overrun has been predicted to occur if both a warning signal and an auxiliary signal were produced; that is, if the projected overrun was predicted to exceed one-half macromove. Movement is terminated at the end of the particular time period for which an overrun is predicted to occur if only the warning signal was produced; that is, if the projected overrun was predicted not to exceed one-half macromove.

When the path terminating process described briefly in the preceding paragraph is performed for a linear path in combination with the adjustment of macromoves in order to hit the End Point exactly, it becomes necessary to predict an overrun condition several time periods ΔT in advance in order to leave a sufficient number of such time periods in which to compute and execute the adjusted macromoves. In the termination process shown to be carried out by the exemplary system for a circular path segment, adjustment of macromoves is not used. Instead, the macromoves continue to be computed on the basis of the originally supplied data and an overrun or an underrun is permitted to occur; that is, if the path segment is terminated at the end of the time period for which an overrun was predicted to occur, that overrun will actually occur. Similarly, if the path segment is terminated during the immediately preceding time period, the path will have been terminated short of its designated End Point, thus causing an underrun. This may cause a small error to be introduced into the interpolation process performed for the following path segment which may in certain circumstances be tolerable. In this event, the following path segment is permitted to commence on the basis of the original data provided therefor. The resulting error will be in the nature of an offset.

In other circumstances, the offset which would result from terminating the circular path segment at the end of a time period ΔT without a correction may be intolerable. For use of the system in such situations an entirely different mode of correction is provided to compensate for failure of the preceding path segment to terminate on its prescribed End Point exactly. This correction, by which some of the data representing the following path segment is modified, is shown incorporated in the exemplary system and described in detail hereinafter in Sections H.2.k, for sake of completeness only. It is not claimed herein but is claimed, instead, in a separate application Ser. No. 762,836 filed by Arthur O. Fitzner for DATA PROCESSING METHOD FOR MINIMIZING POTENTIAL ERRORS OCCURRING DURING LINEAR AND CIRCULAR INTERPOLATION on Sept. 26, 1968

2 Proximity to the End Point Must be Tested for Both Axes

The operations connected with predicting a projected overrun and determining its magnitude are generally shown in FIG. 31 in column 1000. They correspond to a similar series of

operations performed during a typical macromove in the course of simultaneous linear interpolation and shown in column 1000 of FIG. 18. The first operation is to upgrade the numbers in the XCP and YCP registers 117 and 115 to the macroposition values which are to be reached at the end of the iteration represented by FIG. 31. This represents a prediction of the positions along the X and Y axes which will be reached by the path segment, or by an element which is moved along that path segment, by the end of the time period ΔT . Following this, a determination is made as to which of the X and Y axes is the fast axis so that the forward projection toward the End Point might be made along that axis. The test based upon comparison of a forward projected point along the fast axis with the corresponding coordinate of the designated End Point for the path segment is not sufficient, however, by itself, to give an unambiguous indication of a projected overrun of the designated End Point. The reason for this may be seen in FIG. 32. It shows movement along a path segment clockwise in the second and first quadrants toward a designated End Point whose X and Y coordinates are XCEP and YCEP. The End Point in FIG. 32 is in the top half of the 1st quadrant so that the X axis is the fast axis; that is, motion along that axis is at a higher rate than along the Y axis.

Consequently, the look-ahead signals are produced for the X axis and it is on the basis of a comparison of those signals with the signals representing the coordinate XCEP that a projected overrun is detected. In the case assumed in FIG. 32 the results of such a test would be correct because with motion along the path segment progressing clockwise and with the Y coordinate YCEP of the designated End Point being in the first quadrant, an overrun would, in fact, occur.

Let it be assumed instead, however, that the designated end point has the coordinates XCEP/YCEP' so that it falls in the fourth quadrant, as shown in FIG. 32. In such a case, a comparison of the look-ahead signals along the fast axis with the corresponding coordinate XCEP of the designated end point would, by itself, give a misleading result because it would indicate, (correctly) that the end point XCEP/YCEP' was about to be reached along the X axis, and this would be interpreted by the system (incorrectly) as an indication that the end point is about to be reached along the Y axis as well. To eliminate a false indication of proximity to the End Point, a preliminary test is performed by the exemplary system to determine whether or not the path segment has progressed along the slow axis to within the vicinity of the designated path End Point.

3 Testing for Proximity of the Path Segment to its Designated End Point Along the "Slow" Axis

In accordance with the related invention, to confirm that a circular path segment such as that shown in FIG. 32 has extended past its designated End Point rather than the mirror image of that point, a special test is performed. Specifically, along with each projection forward of a fast axis macroposition coordinate, a pair of slow axis coordinates are produced which "bracket" an adjacent slow axis macroposition coordinate. More particularly, in accordance with a feature of the present invention during each iteration a one-macromove look-ahead projection is made from the fast axis macroposition coordinate XCP toward the fast axis End Point coordinate XCEP. Additionally, in accordance with the related invention which is not claimed herein, and with particular reference to FIG. 32, a pair of points PL1 and PL2 are produced which are a small distance above and below the Y axis macroposition coordinate YCP for the iteration. The points PL1 and PL2 may be rapidly derived by respectively adding and subtracting the absolute value of 0.1J to and from the coordinate value YCP, yielding $YCP + |0.1J|$ for the point PL1 and $YCP - |0.1J|$ for the point PL2. In other words, the "size of the bracket" along the slow axis is chosen to be 0.2J. If YCEP, the slow axis End Point coordinate is within the bracketed range; that is, if it is less than $YCP + |0.1J|$ and greater than $YCP - |0.1J|$, the slow axis test is satisfied, indicating that the specified End Point is, in fact, being approached. Consequently, if the fast axis look-ahead test

indicates an overrun, it will be interpreted by the system as an actual projected overrun of the designated End Point. It may be said, therefore, that, in addition to looking ahead along the fast axis a probe is also cast to either side in order to determine whether an End Point is about to be reached.

4 Testing for a Projected Overrun by the Path Segment of its Designated End Point Along the "Fast" Axis

Following the proximity test for the slow axis, the previously predicted macroposition for the end of the current time period ΔT is projected forward by one macromove so that, in effect, the resulting signals represent the point which will be reached along the fast axis at the end of the following iteration. This point is then compared with the corresponding coordinate of the prescribed end point for the path segment by a "compute overrun" operation. If a projected overrun is in fact detected and if the prior check of the slow axis revealed that the path segment is in the vicinity of the slow axis coordinate of the prescribed End Point, the End of Block Counter 186 is advanced to its second, S2 state, as in the case of the linear path segment. The forward projection of the macroposition along the fast axis is then "retracted" by one-half macromove, again as in the case of linear path segment, and a second comparison is made with the corresponding coordinate of the prescribed End Point to determine whether or not a projected overrun condition still exists. If it does, it means that, were another FIG. 31 iteration to be performed during the following time period, the resulting overrun along the fast axis would exceed one-half macromove along that axis. In that event, the End of Block Counter 186 is advanced another step to its third count state.

(5) Typical Iterations for Terminating a Circular Path Segment

FIG. 31 represents a typical iteration performed during the generation and execution of signals representing points along a circular path segment before an overrun has been detected. Consequently, the blocks in column 1000 relating to the advancing of the End of Block Counter are crossed out, since the events which bring this about do not occur during a typical iteration. An iteration during which a large projected overrun is detected, causing the End of Block Counter 186 to be advanced from its first count state S1 through its second count state S2 to its third count state S3, is represented by FIG. 33 wherein the blocks in column 1000 showing these events appear without crosses therein.

In the situation shown by the FIG. 33 iteration; that is, where a large overrun has been detected, the path segment is to be terminated during the time period immediately preceding the time period for which such an overrun was predicted. Since the overrun predicted during the FIG. 33 iteration is for the following time period, the path segment must be terminated at the end of the FIG. 33 iteration and during the same iteration the basic computations required to convert the I, J and Q data for the next block into macromove numbers must be carried out. Consequently, during the FIG. 33 iteration the system must be changed from mode 1 to mode 4 operation and this is indicated by the designation of the last four columns of the FIG. 33 iteration as 4600, 4700, 4800 and 4900.

As will be described in greater detail, the set of calculations performed during the last four columns 4600 - 4900 of the FIG. 33 iteration are those required if the following path segment is to be linear. FIG. 34 shows an iteration identical to that shown in FIG. 33, except that the computations performed during its last four columns 4600 - 4900 are those that are required when the following path segment is circular.

FIG. 35 illustrates what happens during an iteration in which a small overrun is detected; that is, during which the End of Block Counter 186 is only advanced to its second count state as indicated by the fact that the last block in the 1000 column of the FIG. 35 iteration is crossed out. The only detail in which the FIG. 35 iteration (during which a small overrun is detected) varies from the FIG. 31 iteration (during which no overrun is detected) is in that, during the FIG. 35

iteration, the End of Block Counter is advanced from count state 1 to count state 2. All of the other computations which are performed during the FIG. 35 iteration are exactly the same as those carried out during the FIG. 31 iteration. In the flow chart of FIG. 20 each of the alternative iterations illustrated in FIGS. 33, 34, and 35 is shown to occur during time period ΔT 2500. This is simply an arbitrary time selected for sake of example and agrees generally with the length of the circular path segment 27c relative to the length of the preceding linear path segment 27b in FIG. 2.

The events which occur during the iteration which follows one in which a small overrun was detected, that is, a FIG. 35 iteration, are shown in FIG. 36. This iteration, shown in FIG. 20 to occur during time period ΔT 2051, is characterized by advancement of the End of Block Counter 186 during the 1000 column to its third count state S3 as a direct result of the advancement of the counter 186 to its second count state S2 during the preceding FIG. 35 iteration.

The consequence of stepping the counter 186 to its third count state S3 during the initial portion of the FIG. 36 iteration is seen to be the same as its advancement to that count state during a FIG. 33 iteration. Thus, the system is caused to switch over to its mode 4 operation at the end of the 1500 column of the iteration, as indicated by the designations 4600, 4700, 4800 and 4900 of the last four columns. During these four columns, set-up calculations are carried out which are identical to those performed during the corresponding columns of the FIG. 33 iteration, those required where the following path segment is to be a linear one. If, on the other hand, the next path segment is to be circular, the computations which are performed during columns 4600 - 4900 are those shown in FIG. 37, which is also shown in FIG. 20 to occur during time period ΔT 2501.

Summarizing the foregoing general discussion of typical iterations for terminating a circular path segment with reference to FIGS. 33 through 37, FIGS. 33 and 34 show what takes place during a time period in which a large overrun is detected. The set-up calculations performed during the last two columns, 4800 and 4900, of these iterations, are those required to begin execution of a linear and a circular path segment respectively. Thus, the computations shown to be carried out during the last two columns on the FIG. 33 iteration are exactly the same as those previously discussed with reference to the corresponding columns of FIG. 17. Similarly, the computations which are carried out during the last two columns of the FIG. 34 iteration are exactly the same as those explained previously with reference to FIG. 28.

FIG. 35 illustrates what happens during an iteration in which a small overrun is detected and, except for advancement of the End of Block Counter 186 from its count state S1 to its count state S2, all of the steps performed during this iteration are the same as those carried out during the FIG. 31 iteration. Finally, FIGS. 36 and 37 show the events which take place during the time period immediately following that in which a small overrun was detected. The two Figures differ from FIGS. 33 and 34 only in the manner in which the End of Block Counter 186 is advanced to its third count state in column 1000-CIRC. Otherwise, all of the computations carried out during the FIG. 36 iteration are exactly those performed during the FIG. 33 iteration and the same identity prevails between a FIG. 37 and a FIG. 34 iteration.

i. Data Processing Steps Directed to Detecting a Projected Overrun by Testing for Proximity to the End Point Along Both Axes

In the following discussion, reference will again be made to Tables VI and VI-A, which were referred to previously in Section H.2.f.(2).

The first columns of Tables VI and VI-A, taken one after the other and labeled 1000-CIRC, represent in detail the operations performed during column 1000 of each of FIGS. 31 through 37, with the small variations which exist between the corresponding columns of the several Figures being indicated in the Tables. It will be assumed initially that the steps

are performed as part of a FIG. 31 iteration; that is, that an overrun has not been detected.

1. Determining the Fast Axis

During the first 15 steps of the 1000-CIRC column of Table VI, the XSC and YSC registers 121 and 119 and the XCP and YCP registers 117 and 115 are updated and upgraded as explained previously with reference to Table III. The possible advancement of the End of Block Counter 186 during step 14 does not occur because the counter was not previously advanced to its second, S2, count state. During the following two program steps 1015-CIRC and 1016-CIRC, the J and I component numbers are transmitted to the computer 53 from the J and I registers 101 and 99 in order to determine which of them is larger; i.e., along which axis movement is to be at a higher rate. The READ gating signals causing the J and I numbers to be so transmitted are indicated by condition 4 of R/J PGA 391 and condition 3 of R/I PGA 400. Concurrently with transmission of the number I to the computer 53, a compare-instruct signal is applied to the computer from the "FP and CX" PGA 453 (condition 3). If the first of the two numbers transmitted to the computer 53 is greater than or equal to the second, that is, if J is greater than or equal to I, the FAX flip-flop 455 is preset, indicating by a logic 1 voltage level on its FAX output that motion along the X axis is to be equal to or faster than along the Y axis. If the contrary is true and J is less than I, the flip-flop is reset and a logic 1 voltage level on its FAX output indicates that the rate of motion along the Y axis is greater than the rate of motion along the X axis.

It will be observed that the order in which I and J are transmitted to the computer 53 is the reverse of that in which they were transmitted during a linear macromove. The reason for this is that in the case of circular interpolation the size of a macromove along the X axis is proportional to J and not to I as in the case of linear interpolation. Similarly, the Y axis macromove ΔY in the case of circular interpolation is proportional to I instead of J. Consequently, whereas in generating linear path segments movement on the X axis is faster than along the Y axis if I is greater than J, the reverse applies when the path segment being generated is circular.

2. Proximity Test for the Slow Axis

With the identity of the fast axis and, therefore, of the slow axis determined, the next operation is aimed at checking the slow axis for proximity to the End Point in accordance with the process described previously with reference to FIG. 32. In implementing this feature there are generated for a selected one of the X and Y axes first and second digital signals representing first and second spaced apart coordinates which are respectively farther from and nearer to the origin of the axes than the corresponding coordinate of the path segment; in the present instance, the upgraded XCP and YCP number. A determination is then made as to whether or not the corresponding coordinate of the designated End Point for the selected one of the axes, XCEP and YCEP, lies between the first and second spaced apart coordinates.

More specifically, the signals representing the first of the spaced apart coordinates along the selected one of the two axes are compared with the signals representing the corresponding End Point coordinate, and an upper bracket signal is produced and stored if the comparison indicates that the End Point coordinate lies between the origin of the two axes and the first coordinate. Furthermore, the digital signals representing the second of the pair of spaced apart points along the selected one of the two axes are also compared with the signals representing the corresponding coordinate of the End Point and a lower bracket signal is produced and stored if it is found that the second coordinate lies between the origin and the corresponding End Point coordinate. The proximity test along the slow axis is satisfied if both the upper and lower bracket signals have been stored.

To store the upper bracket and lower bracket signals, an upper bracket flip-flop UBFF 503 and a lower bracket flip-flop LBFF 505 are provided. They comprise the principal components of the Storages for Slow Axis End Point Proximity Test unit 71' in FIG. 9e.

The first step in checking the slow axis for proximity to the End Point is to clear the Answer Register 53a by means of a control signal from the CAR PGA 351 (condition 7). During the same time period the J number is transmitted to the computer 53 from the register 101 if the X axis is the fast axis and the I number is so transmitted if the X axis is not the fast axis. In other words, the inverse radius vector component along the slow axis is read into the computer. This is the first step in forming the absolute number $|0.1J|$ or $|0.1I|$. The means for applying the necessary READ gating signal to the appropriate one of the I and J registers 99 and 101 are indicated by condition 4 of the R/I PGA 400 and by condition 7 of the R/J PGA 391. In order to insure that the number read from one of the registers 99 and 101 is treated as an absolute value, i.e., so that its sign is disregarded, a Force Positive control signal is concurrently applied to the computer 53 from the "FP and CX" PGA 453 (condition 1).

As a result of the operations performed during step 1018-CIRC, at the end of the adder cycle associated with that step there is stored in the Answer Register 53a the quantity $|J|$ if the X axis is the fast axis and the quantity $|I|$ if the Y axis is the fast axis. In order to divide by 10 the absolute value stored in the Answer Register 53a, it is shifted one position to the right during the following step 1019 by a control signal applied to the computer from the SAR PGA 411 (condition 19). The number now stored on the Answer Register 53ab is therefore $|0.1J|$ or $|0.1I|$, depending upon the identity of the slow axis. In preparation for receiving the results of the slow axis proximity test the UBFF and LBFF flip-flops 503 and 505 are reset during the same program step 1019-CIRC by means of a common program gate 507.

To produce the signals representing the upper bracket coordinate, which is farther from the origin than the corresponding coordinate of the path segment, the absolute value $|0.1J|$ or $|0.1I|$ is added to the corresponding coordinate of the path segment, YCP or XCP, yielding as the end result $YCP + |0.1J|$ if the Y axis is the slow axis. Accordingly, during program step 1020-CIRC the macroposition number YCP is read into the computer 53 from the YCP register 115 if the X axis is the fast axis and the corresponding number from the XCP register 117 is so read if the Y axis is the fast axis. The necessary READ gating signals for these alternative operations appear in Program Gate Arrays 425 and 421 as the condition 4. The resulting coordinate number $YCP + |0.1J|$ or $XCP + |0.1I|$ is shifted one position to the right during program step 1021-CIRC by a control signal, represented in SAR PGA 411 as condition 20, in order to bring it into alignment with the corresponding coordinate of the designated End Point YCEP or XCEP with which the upper bracket coordinate number is to be compared.

The computer is placed in the compare mode during program step 1022-CIRC by a control signal from the "FP and CX" PGA 453 (condition 4) and during the same program step the End Point coordinate corresponding to that built up in the Answer Register 53a during the preceding steps is transmitted to the computer 53 for comparison therewith from one of the registers XCEP and YCEP 95 and 97, as indicated by condition 4 in each of the Program Gate Arrays 359 and 487.

Let it be assumed that the path segment and its designated End Point are as shown in FIG. 32, and that therefore movement along the Y axis is the slower. Accordingly, during program step 1022-CIRC the end point coordinate number YCEP is compared with the upper bracket coordinate $YCP + |0.1J|$, shown in FIG. 32 as the point PL1. Means are provided to produce an upper bracket signal if the End Point coordinate YCEP lies between the upper bracket coordinate $YCP + |0.1J|$ and the origin. Serving as a means for carrying out this function is the comparison circuitry of the computer 53 which produces a "greater than or equal to" signal at its output 339 if the first of the compared quantities, $YCP + |0.1J|$, is greater than or equal to the second of the compared quantities, YCEP. To store this signal the output line 339 of the computer 53 is connected to the J input of the UBFF flip-flop 503 through a program gate 509 whose other inputs are condi-

tioned to transmit the signal appearing on the line 339 during program step 1023-CIRC. Accordingly, if the first test for proximity along the slow axis is satisfied, the upper bracket signal causes the flip-flop 503 to be set and is thus stored therein.

Continuing with the assumption that the Y axis is the slow axis, during step 1024-CIRC the Answer Register 53a is cleared by a signal from the CAR PGA 351 (condition 8) in preparation for the following part of the slow axis proximity test. During the same program step the inverse component number from the J register 101 is transmitted to the computer 53 by a READ gating signal from the R/J PGA 391 (condition 8). During the steps which follow, the J number thus read into the computer 53 is divided by 10 and then subtracted from the macroposition number YCP in order to produce the lower bracket signal representing the lower bracket coordinate shown in FIG. 32 as the point PL2. Consequently, a "force positive" signal is applied to the computer 53 from the "FP and CX" PGA 453 (condition 2) in order to cause the computer to ignore the sign of the J number. Additionally, a "Subtract" control signal is also applied to the computer from the SX PGA 479 (condition 15) in order to cause the computer 53 to treat the J number as a negative number. Accordingly, when subsequently the J number is divided by 10 and added to the macroposition coordinate number YCP the result will be $YCP - |0.1J|$.

At the end of program step 1024-CIRC the number which is stored in the Answer Register 53a is $-|J|$ and during the following program step 1025-CIRC this number is shifted one position to the right and thereby divided by 10 by means of a "Shift Accumulator Right" control signal from the SAR PGA 411 (condition 21). The result is $-|0.1J|$. The YCP macroposition number is additively read into the computer 53 during the following step 1026-CIRC from the YCP register 115 by means of a READ gating signal from the R/YCP PGA 425 (condition 10). The result of this operation is $YCP - |0.1J|$; i.e., the lower bracket coordinate point PL2 in FIG. 32.

A test is next performed to determine whether or not the coordinate point $YCP - |0.1J|$ lies between the origin and the corresponding coordinate YCEP of the designated End Point of the path segment. Stated differently, a comparison is made between the lower bracket coordinate number $YCP - |0.1J|$ and the corresponding End Point coordinate YCEP, and if the first of the compared quantities is the lesser, the test is satisfied. This test is shown in Table VI to occur during program step 1028-CIRC. Preparatory to this comparison operation the lower bracket coordinate number $YCP - |0.1J|$ in the Answer Register 53ab is shifted one position to the right in order to bring its decimal point into alignment with the other of the compared quantities YCEP. This shift to the right occurs during program step 1027-CIRC in response to a control signal from the SAR PGA (condition 22).

During the next program step 1028-CIRC the computer 53 is placed in the compare mode by a control signal from the "FP and CX" PGA 453 (condition 5) and the YCEP End Point coordinate number is transmitted to the computer from the YCEP register 97 by a READ gating signal applied thereto by its associated R/YCEP PGA 487 (condition 5). If the "lower bracket" test represented by this comparison is satisfied, an output will appear at the "less than" output line 341 of computer 53 indicating that the lower bracket coordinate $YCP - |0.1J|$ is closer to the origin than the designated End Point coordinate YCEP. This signal is applied to the lower bracket flip-flop 505 through a program gate 511 whose other inputs are conditioned to transmit the "less than" signal to the J input of flip-flop during program step 1029-CIRC. The flip-flop thus becomes set at the end of program step 1029-CIRC, storing the lower bracket signal, if the lower bracket test is satisfied.

Summarizing, if both of the tests carried out for proximity of the path segment to the designated End Point along the slow axis are satisfied, both of the flip-flops 503 and 505 become set and maintain UB and LB signals at their Q outputs indica-

tive of their set condition. The system is now ready to perform the proximity test for the fast axis which has been assumed to be the X axis.

3. Proximity Test for the Fast Axis

5 The test which is performed during steps 1030-CIRC - 1043-CIRC during each of the iterations shown in FIG. 31-37 for detecting proximity to the designated End Point of a circular path segment along the fast axis is quite similar to the corresponding test performed during steps 1028-LIN - 1043-LIN during each of the iterations shown in FIGS. 18, 21, 23 and 24 for a linear path segment. In essence the only significant difference between the two tests is in the number of macromoves by which the position along the path segment is projected forward. For a linear path segment this forward projection was shown to be 11 macromoves. For a circular path segment the look-ahead is reduced to a single macromove. In view of these similarities reference should be made to the previously given detailed description of the two-step proximity test for the fast axis given in section G. 2. d(2) under the heading "Data Processing Steps Associated with Detecting a Projected Overrun." Briefly, with reference to Table VI and to FIG. 32, during step 1030-CIRC, Answer Register 53a is initially cleared by a signal from CAR PGA 351 (condition 9) and an X axis macromove number ΔX is entered into the computer 53 from the ΔX register 109 by a READ gating signal from the R/ ΔX PGA 409 (condition 12). From condition 12 of PGA 409 it is seen that the READ gating signal will issue only if the X axis is the fast axis. In the event that motion along the Y axis is faster, the ΔY number is read, as indicated by condition 12 of the R/ ΔY PGA 415.

During step 1031-CIRC a WRITE Gating Signal is applied to the CP input of the ACNX flip-flop 465 of the Fast Axis Projected Overrun Detector 71 in FIG. 9e. This conditions the flip-flop 465 to be set if the sign of the ΔX macromove number just entered into the computer 53 is negative and to be reset if the ΔX number is positive. Since it is being assumed that the circular path in FIG. 32 is being generated and that the path is approaching the End Point XCEP, YCEP in the first quadrant, ΔX is positive and ACNX flip-flop 465 is reset.

To produce the "look-ahead" number for the X axis, the upgraded macroposition number XCP is read into the computer 53 during program step 1032-CIRC from the XCP register 117 by a READ gating signal from the R/XCP PGA 421 (condition 3). Again it will be noted that condition 3 of PGA 421 includes the condition FAX so that the XCP number will only be read into the computer during step 1032-CIRC if the X axis is the fast axis. If this is not the case the number in the YCP register 115 is read instead, as indicated by condition three of the R/YCP PGA 425. The number stored at the end of the adder cycle associated with program step 1032-CIRC is $XCP + \Delta X$. Since XCP is the upgraded macroposition number which is to be reached at the end of the present iteration, the number $XCP + \Delta X$ is the position that is to be reached along the fast axis at the end of the following iteration. This number is shifted one position to the right in the Answer Register 53a by a control signal from the SAR PGA 411 (condition 23) in order to align the decimal point of the look-ahead number with the decimal point of XCEP, the X coordinate of the End Point of the circular path. During step 1034-CIRC the End Point coordinate XCEP is subtractively read into the computer 53 by a READ gating signal applied to the XCEP register 95 from its associated R/XCEP PGA 353 (condition 1).

65 The sign of the remainder of the subtraction operation performed during step 1034-CIRC is applied to the AND gates 467 and 469 of the fast axis projected overrun detector 71. As indicated by the legends on the inputs of the AND gate and as explained previously in section G. 2. d. (2) (b), if the sign of the remainder is the same as the sign of the macromove number which was read into the computer 53 during step 1030-CIRC, the forward projected look-ahead point has passed the corresponding coordinate of the path End Point and this condition is indicated by an SST ("Same Sign Twice") signal appearing at the output 182 of the detector 71.

The SST signal has the effect of advancing the End of Block Counter 186 by one count state during program step 1035-CIRC, but only if the prior proximity tests for the slow axis End Point yielded a positive result, as indicated by the UB and LB outputs from the UBFF and LBFF flip-flops 503 and 505, and (b) the Counter 186 was in state 1.

The means for producing a signal for advancing the End of Block Counter 186 under the aforesaid conditions is represented by condition 2 of the Program Gate Array 183. As indicated by the individual terms of the condition, such means would comprise coincidence gating means having eight inputs connected respectively to the 1000 output and the AND gate group 175 of the Multimode Time Base Logic 67 in FIG. 9d, the 30 output of the converter 169 and the 5 output of the converter 167 in the Time Base and Digital Sweep, the CIRC output of the Path Type Indicator 106, the UB and LB outputs of the flip-flops 503 and 505, the S1 output of the decoder 187 in the Time Base Mode Selector 69 and the SST output appearing on the output line 182 of the Fast Axis Projected Overrun Detector 71.

It will be understood that each listed condition in the PGA 183 represents such a coincidence gating means, also referred to herein as a "Program Gate," connected to receive the signals represented in that listed condition, and that the output of all of the coincidence gating means are connected to the respective inputs of a single OR gate in the manner shown in FIG. 15.

The next series of steps are directed to "retracting" the look-ahead point by one-half macromove in the same manner as explained previously in section G.2.d.(2)(c) with reference to a linear path. As described in that section, the number in the Answer Register 53a representing the look-ahead point is shifted to the left by two digit positions so as to give that number a "weight" of $10 + \Delta X$ relative to the macromove number in the ΔX register 109. This is caused to occur during program steps 1036-CIRC and 1037-CIRC by successive applications of Shift Left control signals from the SAL PGA 463 (conditions 2 and 3). During the following five program steps 1038-1042 the ΔX macromove number is read subtractively into the computer 53 from the ΔX register 109 by gating signals represented in the R/ ΔX PGA 409 by condition 14. The remainder produced in the Answer Register 53a during the adder cycle associated with program step 1042-CIRC is $10(XCP + 0.5\Delta X - XCEP)$, or 10 times the difference between the upgraded XCP macroposition projected forward by one-half macromove less the designated End Point coordinate XCEP for the path. In the context of the example shown in FIG. 32, if an overrun has occurred and it exceeded one-half macromove, the sign of the remainder in the Answer Register 53a will be positive. The sign of this remainder is applied to the AND gates 467 and 469 and if the sign is the same as that of the ΔX macromove, indicating a projected overrun even with only a one-half macromove forward projection, an SST signal indicative of this condition will appear on the output line 182 of the detector 71 during program step 1042-CIRC. By means of a program gate represented by condition three in the Program Gate Array 183 a signal is produced on the output line 184 of the array 183 in response to the SST signal during program step 1043-CIRC if the End of Block Counter 186 was previously advanced to its S2 (second) count state, as indicated by a signal on the S2 output of the decoder 187.

During a typical iteration for generating a circular path segment as shown in FIG. 31, no overrun is detected and the End of Block Counter 186 remains in its first count state so that a signal appears only on the S1 output of the decoder 187. Consequently, the blocks in column 1000 of FIG. 31 which represent advancement of the End of Block Counter 186 are crossed out. If a sufficient number of FIG. 31 type iterations have been carried out so that the path segment closely approaches its End Point XCEP, YCEP and if the first test for proximity to the fast axis End Point indicates an overrun but the second test does not, the counter 186 is stepped to its

second count state S2 but not its third, as shown in FIG. 35. In this case, the counter will be automatically advanced to its third count state, S3, during the following iteration during program step 1014-S2 by means of a stepping signal represented by condition 4 of the PGA 183. This situation is represented by FIGS. 36 and 37 which represent the last iteration for a circular path segment following an iteration in which a small overrun was detected.

If, on the other hand, both tests for the fast axis indicate an overrun condition, i.e., if the projected overrun is more than one-half macromove along the fast axis, the End of Block Counter 186 is stepped to its S3 (third) count state during the same iteration, as shown in FIGS. 33 and 34. In such event, set-up calculations for the following path segment are promptly carried out so that the current path segment may be terminated at the end of the iteration.

j. Data Processing Steps Directed to Changing Operation of the System From Mode 1 to Mode 4

To perform the setup calculation required to begin generation of the following path segment, the system must be placed in its fourth operating mode. It will be recalled that the exemplary embodiment of the system is placed in its fourth operating mode when the counter 186 reaches its count state S14. In the case of linear interpolation it was desirable to advance the End of Block Counter 186 by only one count state during each FIG. 22 iteration in order to carry out nine such iterations with adjusted macromoves. These intermediate iterations are not utilized with circular interpolation and, instead, it becomes desirable to switch the system over directly from mode 1 into its mode 4 operation during the iteration in which a large overrun is detected or during the iteration following that in which a small overrun has been detected. Consequently, means are provided for rapidly stepping the counter 186 from its S1 to its S14 count state during an iteration in which a large overrun has been detected, as shown in FIGS. 33 and 34 and to step it from its S2 to its S14 count state during the iteration following that in which a small overrun is detected, as shown in FIGS. 36 and 37. Means for rapidly advancing the End of Block Counter 186 to its fourteenth count state are indicated in the Program Gate Array 183 by conditions 5, 6 and 7-16.

To understand the manner in which the End of Block Counter 186 is rapidly stepped through several of its count states during a single iteration let it be assumed that a large projected overrun has been detected during time period ΔT_{2500} so that the iteration is of the type shown in FIG. 33 or 34. The End of Block Counter 186 will have advanced to its S3 count state during step 1043-CIRC. By means of a program gate represented by condition 5 of the Program Gate Array 183 a further stepping signal will be applied to the End of Block Counter 186 during program step 1589-CIRC causing it to be advanced to its S4 count state which now puts the system in Mode 2. By means of another program gate, (PGA 183, condition 6), during the following program step, now 2590-CIRC, yet another stepping signal is applied to the counter 186 as a result of it having reached its S4 count state during the preceding program step, so that at the end of program step 2590-CIRC the End of Block Counter 186 is stepped to its S5 count state. This results in a signal on the S5 output of the decoder 187 so that an M3 output signal is produced at the output of the OR gate 190, putting the system in Mode 3. This M3 output signal is used as an input to each of nine further program gates represented by conditions 8-16 of the PGA 183. Each of these nine program gates are also conditioned by the CIRC output of the Path Type Indicator 106, by the 3500 output of the group of AND gates 179 in the Multi Mode Time Base Logic 67, and the 90 output of the converter 169 in the Time Base and Digital Sweep 63. Finally, respective ones of the nine program gates represented by the conditions 8-16 receive outputs one through nine of the Units Decoder 167 of the Time Base and Digital Sweep 53 so that respective ones of them produce an output signal during successive ones of the program steps 3591-CIRC - 3599-CIRC, provided the OR gate 190 of the Multi Mode Time Base Logic 69 continues to produce an M3 signal.

The last of the stepping signals produced by the program gates represented by conditions eight to 16 places the End of Block Counter 186 at count state S14. The S14 signal from the decoder 187 is passed through and appears at the output of the OR gate 191 of the Time Base Mode Selector 69 as an M4 signal, which is applied to the AND gates 181 of the Multi Mode Time Base Logic 67, placing the system in its mode 4 operation.

If it were assumed instead, that only a small projected overrun of the End Point had been detected during a given iteration, the End of Block Counter 186 would be advanced during that iteration to its S2 count state only, as shown in FIG. 35. However, during the following iteration, which might be either of the type illustrated in FIG. 36 or of the type shown in FIG. 37, the End of Block Counter would be stepped to its S14 count state.

k. The Reason for and Manner of Modifying the Data Representing the Next Path Segment

The operation to be described in this section is related to the generation of path segments in accordance with the present invention, but as indicated previously, it is claimed in the patent application Ser. No. 762,836 filed by Arthur O. Fitzner on Sept. 26, 1968. In order to put this related feature in proper context certain features of the present invention will be summarized briefly. In accordance with the present invention a method and the system for performing that method have been described for moving a machine element or member at a controlled velocity along a series of path segments, any one of which may be either linear or circular. The starting point of each path segment is the same as the end point of the preceding path segment, i.e., the path segments are contiguous.

Each path segment is represented by a block of data from which there are initially derived the X and Y components I and J of the path segment if it is linear and the X and Y coordinate distances I and J from the starting point of the path segment to its prescribed center if it is circular. Each block of data also includes the composite velocity at which movement along its associated path segment is to occur, expressed as the "macromove quotient" $(V \cdot \Delta T) / \text{Dor} (V \cdot \Delta T) / R$.

From each block of data there are derived, during respective ones of a series of equal time periods ΔT , by a process called "interpolation," digital signals representing the X coordinates XCP of a series of points lying on the path segment and the Y coordinates YCP of a series of points lying on the path segment. During each of the time periods ΔT the machine element or member is moved along the X and Y axes to a new XCP, and a new YCP coordinate. The distance ΔX from a given XCP coordinate to the next XCP coordinate is so proportioned relative to the distance ΔY from the corresponding YCP coordinate to the next YCP coordinate that the composite motion of the element resulting from the respective motion through successive XCP and YCP coordinates will always be in the required direction to execute motion along the path segment. But there is no assurance that either coordinate of the end point of the path segment will coincide with the corresponding coordinate reached at the end of any one of the time periods ΔT . Indeed, without some special provision, it is almost certain that, were the element to continue to move, eventually, during one of the time periods ΔT one or both of the XCP and YCP coordinates to which the element is moved would lie past the corresponding coordinate of the end point of the path segment. Accordingly, movement of the element along the path segment is terminated at the end of that time period ΔT for which a selected one of the XCP and YCP coordinates falls closest to the corresponding coordinate of the end point. This may cause the end of the actually executed path segment to fall short of, or extend past, the starting point of the following path segment.

Therefore, were the next path segment to be derived from its associated data block as originally provided, its originally prescribed direction would remain unchanged, but its starting point would be different due to the underrun or overrun of the preceding path segment. In short, the new path segment, if it is

linear, or its center, if it is circular, would be offset by the amount of underrun or overrun of the preceding path segment.

In accordance with the related invention claimed in the application of Arthur O. Fitzner, the I and J values in the data block representing the following path segment are altered so as to change their composite vector until it extends between the actual end point of the preceding path segment and the prescribed end point or center of the following path segment, depending upon whether it is linear or circular. In the preferred method in accordance with this related feature, this is accomplished by modifying each of the I and J components of the next path segment by an amount which is equal to the underrun or overrun of the end point of the preceding path segment along the corresponding coordinate axis. For example, if the preceding path segment stops short of its prescribed end point by an amount ΔX along the X axis and by an amount ΔY along the Y axis, then the I and J values for the following path segment, whether it be linear or circular, are modified to $I + \Delta X$ and $J + \Delta Y$. If there is an overrun of ΔX and ΔY instead, the I and J values are modified to $I - \Delta X$ and $J - \Delta Y$.

Taking a specific example, let it be assumed that a large projected overrun was detected during time period $\Delta T2500$ and that the next path segment to be generated is linear. The iteration, therefore, is that shown in FIG. 33. Having detected during column 1000 of the iteration that the fast axis coordinate of the end point would be overrun by the end of the following iteration; i.e., during time period $\Delta T2501$, the system performs steps to terminate the current path segment at the end of the present iteration; that is, at the end of time period $\Delta T2500$, and to begin the following path segment at the start of time period $\Delta T2501$. The setup computations utilizing I and J to compute ΔY and ΔX for the following path segment are performed during column 4800 and 4900 of the FIG. 33 iteration, but since the current path segment has been stopped short of its designated End Point a correction is made to the I and J values and these corrections are made during column 4600 of the iteration.

1. Data Processing Steps Directed to Modifying the Data Representing the Next Path Segment

The first of the numbers to be corrected is I and, therefore, the first step in the correction process is to compute the correction to be made to I. This correction is $XCEP - XCP$, or the difference between the X coordinate of the path segment end point and the X coordinate of the path segment at the end of the last iteration for the path segment. The steps for carrying out this computation appear in column 4600-CIRC of Table I in section G.2.a.(1). The XCEP number is read into the computer 53 during program step 4615-CIRC from the XCEP register 95 by a READ gating signal applied to the register from the R/XCEP PGA 359 (condition 3). The number appears in the Answer Register 53a during the following program step 4616-CIRC and is shifted one position to the left by a control signal from the SAL PGA 463 (condition 10) in order to bring it into alignment with the XCP number which is to be subtracted therefrom during the next program step 4617-CIRC. During that program step the number XCP is transmitted to the computer 53 from the XCP register 117 by a gating signal from the R/XCP PGA 421 (condition 5) and in response to a subtract control signal from the SX PGA 479 (condition 8). The remainder, $XCEP - XCP$ appears in the Answer Register 53a during the next program step 4618-CIRC and represents the correction number which is to be added to the I number representing the next path segment and currently in the buffer register 471.

The I number is additively read into the computer 53 from the I buffer register 471 during program step 4618-CIRC by a READ gating signal from program gate 349. The result, $I + (XCEP - XCP)$, is the corrected I number for the following path segment and is entered into the I active register 99 during program step 4619-CIRC by a WRITE gating signal from the W/I PGA 355 (condition 2). Next, the J number in the J buffer register 47J is modified and in preparation the Answer Re-

gister 53a is cleared by a control signal from the CAR PGA 351 (condition 14). During the same program step, following the clearing of the Answer Register 53a, the YCEP number is entered into the computer 53 from the YCEP register 97 by a READ gating signal from the R/YCEP PGA 487 (condition 3). The number thus entered into the computer and lodged in its Answer Register 53a is shifted one position to the left during program step 4621-CIRC by a Shift Left control signal from the SAL PGA 463 (condition 11). Next, the YCP number is subtractively entered into the computer 53 from the YCP register 115 by a READ gating signal from the R/YCP PGA 425 (condition 9) and by a signal from the SX PGA 479 (condition 9). The result, appearing in the Answer Register 53a during the following program step 4623-CIRC, is YCEP - YCP, or the correction number for the J number in the buffer register 47J. That number is additively entered into the computer 53 during program step 4623-CIRC by a READ gating signal applied to the register from its associated program gate 353 and the J number is automatically added to the previously computed correction number resulting in the storage during program step 4624-CIRC in the Answer Register 53a of the corrected J number $J + (YCEP - YCP)$. This corrected J number is entered from the Answer Register 53a into the J active register 101 during program step 4624-CIRC by a WRITE gating signal from its W/J PGA 357 (condition 2).

The remaining operations carried out during column 4600-CIRC need not be described at this point since they are identical to the corresponding operations described with reference to the iteration shown in FIG. 17. They relate to the transfer from buffer to active of the X and Y components XCEP and YCEP of the next path segment, adding to them the necessary offset corrections stored in the XOFF and YOFF registers 365 and 367 and to the transfer of the Q number from buffer to active.

Similarly, the computational steps carried out during column 4800 for computing the ΔY number, and during column 4900 for computing the ΔX number are identical to those described in detail previously with reference to FIG. 17 and need not be considered again at this point.

m. Return of the System to Mode 1 Operation for Executing the Next Path Segment

After the ΔY and ΔX macromove numbers have been computed during columns 4800 and 4900 of the FIG. 33 iteration, the last number G in the data block, representing the following path segment 27d of FIG. 2, which is linear, is transferred from buffer to active during steps 4973 (G_{01}) - 4974 (G_{01}) shown in Table II, section G.2.a.(2) (a). Referring further to Table II, the tape reader 41 is started during step 4995 (G_{01}) in response to a signal applied to the J input of the flip-flop 257 in FIG. 9-1. This initiates reading from the tape into buffer of the data block representing the path segment which follows the path segment whose data block was just transferred from buffer to active. The only steps remaining to be performed during the FIG. 33 iteration prior to commencement of the execution of the following path segment is to switch the system into its mode 1 operation by advancing the End of Block Counter 186 through its S15 and S16 count states into its S1 count state by means of three successive signals produced by the Program Gate Array 183 during program steps 4997- G_{01} , 4998- G_{01} and 4999- G_{01} . The program gates required are represented in the Program Gate Array 183 by conditions 19, 20 and 21. With the End of Block Counter 186 in its S1 count state the signal appearing at the S1 output of the decoder 187 is applied to and produced on the output of the OR gate 189 of the Time Base Mode Selector 69 as the signal M1 enabling the AND gates 175 and placing the system in its mode 1 operation.

I. Detailed Description of a Servomechanism for Executing a Path Segment in Accordance with Interpolated Microposition Numbers

As noted previously, in accordance with the present invention a linear or a circular path segment is executed by physically extending the path segment. This extension may be

either in the form of electrical position signals or in the form of mechanical movement of a machine component such as the movable machine element 14 in FIG. 1. In the preceding sections a detailed description has been given of the generation of electrical position command signals. There will now be described in detail a pair of servomechanisms referred to previously with reference to FIG. 7a as elements 75 and 77 which are suitable for moving the machine element 14 in accordance with the X and Y axis macroposition command signals XCP and YCP and, more specifically, in accordance with the X and Y axis microposition command signals XSC and YSC.

Referring to FIG. 9, the numerical control system there shown includes a pair of servomechanisms 75 and 77 for translating the movable element 14 along X and Y axes according to the numerically defined command signals generated in accordance with the present invention. Since the servomechanisms may be identical, only one need be described. The movable element 14 is translated through distances and at velocities along the X axis corresponding to the numerically defined X axis commands which appear in the XSC register 121 by appropriate energization of the reversible servomotor 21 connected to turn the lead screw 20. Engaged with the lead screw through a nut (not shown) and fixed to the element 14 is the ram 16 (these elements having been described previously with reference to FIG. 1). The element 14 is moved with the ram in a -X or +X direction (toward the left or toward the right in FIG. 9a) and at speeds corresponding to the magnitude and polarity of a DC voltage applied to the servomotor 21 by a servoamplifier 521. The manner in which the signal for driving the servoamplifier 521 is generated will be described hereinafter.

Stated briefly, a comparison is made between the instantaneous position of the movable element 14 and the command signal. Based on this comparison an error signal is produced whose sense and magnitude correspond to the sign and extent of the error which exists between the instantaneous directed and actual positions of the element. In this manner, the velocity of the element is made to correspond to the rate of change in the commanded instantaneous position.

Provision is made to generate a feedback signal whose phase relative to a reference signal is indicative of the actual instantaneous position of the movable machine element 14. In the exemplary arrangement shown in FIG. 9a, the 500 cps output B of the sine wave shaper 181 (FIG. 9b) is used.

To provide an indication of the actual instantaneous position of the movable element 14, a resolver 523 is provided for producing a recurring feedback signal which is indicative by the sense and magnitude of its phase angle, relative to the reference signal B, of the sign and extent of the displacement of the movable element from a datum position. It comprises a stator with stator windings 525 and 527 associated with a rotor carrying a rotor winding 529. The resolver rotor is mechanically coupled through suitable gearing (not shown) to the lead screw 20, so that as the latter is turned by the servomotor 21, the resolver rotor turns with it. As a specific example, the resolver rotor may be geared to the lead screw 20 so that it makes one revolution for each 0.1 inch of travel by the movable element 14.

The resolver 523 is excited by the sinusoidal reference signal B fed to its rotor winding 529. The resulting pulsating magnetic field induces a pair of sinusoidal signals in the stator windings 525 and 527 with the amplitude of the output signals varying as sine and cosine functions of the angular position of the movable rotor winding 529 with respect to the stator windings.

By means of a phase shifter 531 the variable amplitude sinusoidal signals appearing across the resolver stator windings 525 and 527 are converted into a single sinusoidal signal C which is of constant amplitude but whose phase relative to that of the sinusoidal reference signal B shifts 360° with each complete rotation of the resolver rotor. In the illustrated example, for each 0.1 inch movement of the machine element

14, the resolver rotor winding 529 is rotated 360° so that with each such movement of the element the feedback signal C is shifted by 360° relative to the reference signal B. As the machine element 14 continues to move, the feedback signal C is brought into phase with the reference signal B at successive "datum" positions spaced 0.1 inches apart. When the element 14 is moved to the left or to the right of such a datum position, the sense and magnitude of the phase angle ϕ_2 between the mid-cycle points of the feedback signal C and the reference signal B reflects the sign and extent of the displacement of the element from that datum position.

The phase relationships just described are illustrated in FIGS. 38a-c. In FIG. 38a the feedback signal C lags the reference wave B by a small angle ϕ_{2a} indicating that the movable element 14 is on one side, say to the right, of one of the datum points. In FIG. 38b the feedback signal C lags the reference wave B by a larger angle ϕ_{2b} , indicating that the element 14 is further away from the datum position compared to the case illustrated by FIG. 38a. FIG. 38c illustrates the relative phases of the feedback signal C and the reference wave B when the movable element 14 is on the opposite side of a datum position from that illustrated in FIGS. 38a and 38b. In such a case the feedback signal C leads the reference wave B by an angle ϕ_{2c} .

As explained previously in section G.1.a.(1) with reference to FIGS. 9b and 10, provision is made in the system to produce signals digitally representing a reference number which during successive time intervals repeatedly and cyclically changes in small uniform increments from a first to a second predetermined value. The signaled reference number itself thus undergoes successive cycles of variation, relative to which a repeating signal may vary in phase.

More particularly, digital sweep generating means, shown in FIG. 9b as the cycling binary counters 131, 133, 135, are provided to produce binary coded decimal signals which change to signal a reference number that increases linearly in uniform steps at uniform time intervals during each cycle of the reference wave B. The reference number thus cyclically increases from a minimum value (e.g., 000) to a maximum value (e.g., 999) and then repeats, in synchronism with the reference wave B. In effect, therefore, each increment of the reference number corresponds to one time portion of a cycle of the reference wave, the latter being divided up into many small time intervals equal in number to the number of counts or increments in the reference number variation.

The phase relation here assumed between the reference square wave A, the reference wave B, and the number signaled by the cycling counters 131, 133 and 135, (as indicated by the positively sloped line D,) is shown in FIG. 38a. The vertical scale left of line D indicates that, during a complete cycle of the reference square wave A, the reference number D signaled by the cycling counters 131-135 increases from 000 to 999 at a uniform rate, and that the mid-point of the square wave A occurs at the instant when the cycling counters are just turning 500, the middle of their total range. The same relationships apply to the reference wave B, since it is assumed to be in phase with wave A.

Provision is made to produce a repeating signal F (called "compare signals") at each instant when the cyclically changing and digitally signaled reference number is equal to the periodically changing digitally signaled command number. In this way, the phase angle or phase modulation of the repeating signal F varies in sign and magnitude (relative to the cycling intervals of the reference number signaled by the digital sweep counters 131-135) according to the sense and extent of the changes in the microposition command number. Specifically, the repeating signal F is produced by comparing the cyclically changing reference number D, or digital sweep, against the dynamically changeable command number E, so that the phase angle of that repeating signal changes in sense and extent (relative to the recurring reference wave B, which is fixed in phase relative to the cycling intervals of the reference number) according to the sense and extent of the difference

between (a) the mid-point value (e.g., 500 as here described) of the cyclically changing reference number D and (b) the instantaneous value of the updated command number E. Thus, the digital intelligence of the dynamically changing command number E is converted into a corresponding phase analog variation, since the phase angle between the repeating signal F and the reference wave B changes by amounts, in a sense, and at an average rate which correspond respectively to the amount, sense and average rate of change of the command number.

As shown in FIG. 9d, this comparison of two digitally represented but changing numbers is effected by connecting the twelve output lines of the cycling counters 131-135 and the 12 output lines of the updated XSC register to the respective inputs of a fast-acting multi-digit comparing device 533. The comparing device operates to produce a compare signal F in the form of a short pulse on its output line 533a only when the reference number digitally represented in binary code on the output lines of the counters 131-135 is equal to the updated command number digitally represented in binary code on the output lines of the XSC register 121.

At some value of the reference number, during each cycle thereof (from 000 to 999) it will become equal to the digitally signaled XSC microposition command number because the latter, being a three-digit decimal number, will always fall within the range spanned by the cyclically changing reference number. Thus, a compare signal F will be produced at some instant during each cycle of the digital sweep counters 131-135 and the train of pulses F recurring at a nominal rate of 500 per second will by their variation in phase, relative to the reference wave B, represent the corresponding changes in the XSC command number.

The manner in which the compare signals F, i.e., the phase variable command signal, change in phase relative to the reference signal B is illustrated in FIGS. 38a, b, c. In these figures, the number held in the XSC register 121 is represented by the horizontal lines E, and the instant of equal comparison is indicated by their intersections with the lines D representing the cyclically changing reference number.

Since the reference number D increases from its minimum to its maximum value once during each cycle of the reference signal B, the instant, or phase, during the cycle of the reference signal B when the compare signal F occurs is unambiguously determined by the amount to which the reference number D has increased when it becomes equal to the command number E. Hence, the phase of the compare pulse F with reference to the phase of the reference wave B is determined by the magnitude of the command number E, so that the compare signal F is itself a time phase command signal.

To cause the movable element to advance at uniform velocity in small "micromove" increments in conformance with the larger generated "macromove" increments generated by the system, the information in the XSC storage register 121 is periodically updated in the manner explained in detail previously in section G.2.b.

FIGS. 38b and 38c show two examples of updating, both at the same instant t_u during the cycle of the reference signal B. In the example of FIG. 38b, the command number E is reduced by updating, while in the example of FIG. 38c the change is in the opposite direction. It is important to note that the instant of occurrence of the compare signal F is determined by the magnitude of the updated command number E.

The time phased command signal F and the variable phase feedback signal C are caused to operate a single bistate device whose outputs are then used to produce a polarized error signal which is indicative of the magnitude and the sense of the error phase angle between the phase command and feedback signals. The bistate device, shown as a flip-flop 535, has two stable states which will be referred to as "0" and "1" states. The flip-flop 535 is set into its "1" state by the compare signal F, fed to its "set" input terminal S from the output of the comparing device 533 at the instant when the reference number D becomes equal to the command number E.

Means are also provided for resetting the flip-flop 535 into

its "0" state at the instant of each positive-going zero crossing of the feedback signal C. The positive-going zero crossing signal for resetting the flip-flop is derived from the feedback signal C by means of a square wave shaping circuit 537. At one of its output terminals the square wave shaper 537 produces a first square wave G which is in phase with the feedback signal C. The square wave G is fed to a unipolar differentiating circuit 539 whose output is connected to the reset input terminal R of the flip-flop 535. In response to each positive-going wave front of the square wave G, and hence at the positive-going zero crossings of the feedback signal C, the differentiating circuit 539 produces a triggering pulse which resets the flip-flop.

The flip-flop 535 has a pair of output terminals labeled "1" and "0". When the flip-flop is in its "1" state, the "0" output produces a logic signal which is at the logical "0" level and the "1" output produces a logic signal at the logical "1" level. In the embodiment illustrated in FIG. 9a, the logical "0" level is positive relative to the logical "1" signal level. When the flip-flop is reset into its "0" state, the logic signals at its outputs are reversed with a logical "1" signal appearing at the "0" output and a logical "0" signal appearing at the "1" output.

At a second output terminal, the square wave shaper 537 produces a complement of the square wave G, such complement being labeled $\bar{G}$. Thus, at all times, the square wave shaper produces at its two outputs a pair of bi-valued logic signals, one signal $\bar{G}$ being at a relatively low, binary logic "1" level during each positive half cycle of the feedback signal C and at a binary "0" level during each negative half cycle of the feedback signal. The other square wave signal G is at a low, binary logic "1" level during each negative half cycle and at a binary "0" level during each positive half cycle of the feedback signal (FIG. 38a).

In order to produce a polarized DC signal which is indicative of the sign and magnitude of the error phase angle between the feedback signal C and the compare signal F, the complement output $\bar{G}$ of the square wave shaper 537 and the "1" output H of the flip-flop 535 are applied to a first AND gate 541, and the G output of the square wave shaper 537 and the "0" output H of the flip-flop 535 are applied to a second AND gate 543.

The outputs of the AND gates 541 and 543 are connected to control terminals 545 and 547 of two current generators respectively. The generators work into a common load resistor 553 but are oppositely poled so that when the negative current generator 549 is turned on, current flows from ground through the resistor 553 and through the generator 549 into a negative terminal 555, thus establishing a negative voltage K across the resistor. Conversely, when the positive current generator 551 is turned on, current flows from a positive terminal 557 through the generator 551 and the resistor 553 into ground, causing a positive voltage L to appear across the resistor. Regardless of which current generator is turned on, the size of the signal across the resistor 553 is kept at a predetermined magnitude by a bipolar Zener diode 559, having the same Zener voltage in both directions.

When the compare signal F leads the negative-going zero crossing point C₀ of the feedback signal C, the first AND gate 541, controlled by the $\bar{G}$ and H signal outputs of the wave shaper 537 and the flip-flop 535 produces a pulse, labeled in FIGS. 9a and 38b as "Enable K". The "Enable K" pulse turns on the "negative" current generator 549 so as to produce a negative signal K across the resistor 553 for a time interval which is directly proportional to the amount of the phase lead.

Similarly, when the compare signal F lags the point C₀, the second AND gate 543 under the control of the signals G and $\bar{H}$ produces a pulse shown as "Enable L" and turns on the "positive" current generator 551 to produce a positive signal L across the resistor 553, again for a time interval directly proportional to the amount of the phase lag.

The manner in which the square wave shaper 537, the flip-flop 535, the AND gates 541 and 543 and the current generators 549 and 551 cooperate to produce the bipolar K and L signals is shown by the various signals in FIGS. 38a-c, but will

not be described here. For a detailed explanation, reference should be made to my copending patent application for Apparatus for Controlling the Instantaneous Positions and Velocities of Movable Elements According to Digital Numerical Commands, Ser. No. 555,048, filed June 3, 1966 now U.S. Pat. No. 3,539,895. Suffice it to say that, with the compare signal F leading the negative-going zero crossing point C₀ of the feedback signal C, a negative signal, shown at K in FIG. 38b, is generated during a period which begins with the compare signal F and which ends with the next negative-going zero crossing of the feedback signal C.

If the compare signal F lags behind the same arbitrary point C₀ of the feedback signal C, a positive signal, shown at L in FIG. 38c, is generated during a period which commences with the negative-going zero crossing of the feedback signal C and ends with the occurrence of the compare signal F.

As the movable element 14 is being translated, the compare signal F representing the desired position and rate of change of position of the element 14 will lead or lag the negative-going zero crossing point C₀ of the feedback signal C depending upon whether the element is being driven in the +X or -X direction. To convert the variable width, constant amplitude pulses K or L into a smoothly variable control signal, e.g., a polarized variable DC voltage suitable for application to a servomotor and thence to the servomotor 21, they are averaged by a low-pass filter 561. The output of the low pass filter 561 is thus a polarized DC signal representative of the sense and magnitude of the average error phase angle between the compare signal F and the negative-going zero crossing C₀ of the feedback signal C over several cycles. After amplification by the servoamplifier 521, the DC error signal is applied to the servomotor 21 causing it to drive the movable element 14 in a direction and at a velocity which conforms with the direction and rate of change in the phase of the compare pulse F, that is, in a sense and at a rate directed by the sequence of XSC microposition command signals developed by the system. When the phase angle of the command phase pulse F begins to change, and the error phase angle departs from zero, the servomotor 21 begins to move, so that the phase of the feedback signal C is shifted by the resolver 523 in a direction to restore the error phase angle to zero. Thus, the instantaneous position error cannot exceed 0.05 inch and the single fine resolver 523 is sufficient to keep the system tracking the dynamically changing directed position along the X axis.

What has been described in the foregoing paragraphs is a particularly suitable filtering servo drive for moving the element 14 along the X axis in accordance with the digital command signal XSC developed by the system. The means for driving the movable element 14 along the Y axis in accordance with the Y axis microposition command number YSC include a second compare circuit 563, shown in FIG. 9d, which may be identical to the compare circuit 533 also shown in that figure. In response to the YSC command signal received over the trunk lines 81, the compare circuit produces a compare pulse shown as the "Y axis command" pulse. This pulse is applied to the Y axis servomechanism 77, shown in 9a as a single block. Its construction may be identical to that of the X axis servomechanism 75, as shown in FIG. 9a in detail. Thus, the Y axis servomechanism 77 includes the motor 25 (FIG. 1) which is operative to move the element 14 vertically with the ram 15 through the drive screw 24.

J. An Alternative System for Performing Linear and Circular Interpolation

1. Introduction

A second system for performing linear and circular iterations may be used as an alternative to that disclosed in FIG. 9a-l is shown in FIG. 40 a-m. The two systems are very similar and most of the components comprising the system shown in FIG. 9 have a corresponding counterpart in the system shown in FIG. 40. To permit a quick understanding of the similarities of the two systems, those components of the system shown in FIG. 40 which have a corresponding counterpart in the system shown in FIG. 9 are given the same reference numeral but

with a prime (') added.

With a few exceptions, the general block diagram shown in FIG. 7 applies to the second exemplary system of FIG. 40 as well as it does to the first exemplary system of FIG. 9. Some of the differences between the systems will be explained in the sections which follow.

2. Principal Structural Differences Between the First and Second Exemplary Systems

a. The Time Base Mode Selector

In the first exemplary system of FIG. 9 the Time Base Mode Selector 69, shown in FIGS. 9d and 9e, is arranged successively to place the system in four different operating modes. These changes in operating modes are brought about by successively stepping the End of Block Counter 186 through its count states, thereby placing a signal on successive ones of the sixteen outputs S1-S16 of its associated decoder 187. The decoder outputs S1-S16 are used for a single purpose: to control the advancement of the End of Block Counter 186 and to bring about the different operating modes of the system.

The second exemplary system of FIG. 40 has only three operating modes comparable to those of the system of FIG. 9. These modes are still controlled exclusively by the outputs of a decoder 187' corresponding to the decoder 187 of FIG. 9e. However, the relationship between the count states of the End of Block Counter 186' of FIG. 40e as signaled on the outputs of its associated decoder 187' and the count states of the system as signaled on lines M1', M2', and M3' in FIG. 40d is different from the relationship between the signals and count states of the corresponding components shown in FIGS. 9e and 9d. Specifically, the system of FIG. 40 is in its first mode so long as there is a signal on any of the outputs S1-S4 of its decoder 187'. These four outputs are connected to a single OR gate 570 whose output in turn is connected to the M1' input of the AND gates 175'. The system is placed in its second major operating mode when one of the outputs S6 and S7 of the decoder 187' carries a signal, these two outputs being connected through a second OR gate 571 to the M2' input of the second set of AND gates 177' of the Multi Mode Time Base Logic 67'. Finally, the system is placed in its third major mode of operation when there is a signal on the S8 output of the decoder 187', this output being directly connected to the M3' input line to the third group of AND gates 179' of the Multi Mode Time Base Logic 67'.

Another difference between the systems shown in FIGS. 9 and 40 is that in the second exemplary system the outputs of the decoder 187' are used to control the operation of the system in various ways other than by simply connecting them through a series of OR gates directly to the AND gates of the Multi Mode Time Base Logic, as illustrated in the first embodiment in FIG. 9. Thus, the signals appearing on the S5 and S6 outputs of the decoder 187' are applied to an OR gate 573 whose output is combined with the $G_{02} + G_{03}$ output of the G buffer decode unit 221' in FIG. 40i in an AND gate 575 whose output, represented by the logical expression $(S5 + S6) \cdot (G_{02} + G_{03})$ is inverted by an inverter 576. The output of the inverter 576, $(S5 + S6) \cdot (G_{02} + G_{03})$ is labeled as the output PX2 and is used throughout the system to control its operation in a manner which will be described in detail in the following section. Briefly, the function or signal PX2 is generated only so long as the end of the current path has not been detected if the next path segment is circular, but continues unaffected by the detection of the end of the path segment, if the next path segment is linear.

b. The Slow Axis End Point Storages

In the first exemplary embodiment which appears in FIG. 9, means were shown for testing proximity to the end point along the slow axis in the circular mode in order to avoid an ambiguity which is inherent when such proximity is tested along the fast axis only. The operation of this feature is described in Sections H.2.h.(2) and H.2.h.(3). As noted there, the feature of testing proximity to the end point along the slow axis is a related invention claimed in another application. In order to simplify the explanation of the manner in which the second ex-

emplary system of FIG. 40 operates, this feature is not shown as a part of that system. It will be, however, apparent to those skilled in the art how such a feature would be implemented in the second exemplary embodiment if this were desired.

Another minor difference is related to the fast axis projected overrun detector 71' of the second system of FIG. 40 which includes an additional component, an inverter 577, whose input is connected to the output of the OR gate 481' and whose output, used as the output of the detector 71', is SST. With the output SST of the inverter 577 being applied to the Program Gate Array 183' of the Time Base Mode Selector 69', the Program Gate Array is responsive to advance the count state of the End of Block Counter 186 each time a change of sign occurs. This is in contrast to the first system in which the Program Gate Array 183 is responsive to the repetition of the same sign twice on the COT (-) output of the computer 53.

c. The YCPR Register

For reasons which will be explained subsequently the YCPR register 116, as shown in FIG. 9g, and its associated gates 489, 491, 493, 495, 497 and 499 are not needed and do not appear in the second system.

d. Additional Registers XMP and YMP for Storing Macroposition Numbers in Machine Coordinates

In the first exemplary embodiment of FIG. 9 certain fixed and variable offsets, which are required to translate the numerical data from the programmer's coordinates to machine coordinates, are applied to the data initially when they are transferred from the buffer registers 47 to the active registers 57 and before any of the computations connected with the producing of macromoves and micromoves have been performed. (See Section G.2.a.(1)). The disadvantage inherent in this is that, when the data which is being computed is displayed for viewing by an operator or by personnel testing the system, the numbers which will appear will be less comprehensible to the viewer than they would be if they had been left in their original form, in which they conform to the dimensions appearing in the drawing of the part which is being machined. For this reason, in the second exemplary system the data are not converted into machine coordinates until the last computational sequence associated with carrying out a particular macromove, so that all of the results of previous computations will be displayed to the viewer in terms which are directly compatible with the numbers which were used to program the tape, and which appear on the part drawings.

It should also be noted that, in contrast to the first exemplary system, wherein a large fixed offset is added to the data to translate it from the four-quadrant programmer's coordinates into a single-quadrant coordinate framework, in the second exemplary embodiment no fixed offset is applied. Hence, the program gates 361, 363, 369, 379 and the OR gate 371 which are used in the first system for applying the fixed offset to that system do not appear in the second system of FIG. 40. Accordingly, in the system of FIG. 40 the machine's coordinate system has four quadrants, and its center is offset from the center of the programmer's coordinate system, shown in FIG. 2, only by the machinist's dialed-in offset.

More specifically, for most computations the data in the system, such as I, J, ΔX, ΔY, XCP, and YCP, are left unmodified. However, the microposition numbers which are periodically updated in the XSC and YSC registers 121' and 119' are always in machine coordinates. As previously explained, they are updated ten times during each iteration. To cause the numbers which are applied to them to be in machine coordinates, once during each iteration, just preceding the tenth updating of the XSC register 121', the number stored in the XCP register 117', which is in part coordinates, is modified by adding to it the X axis offset from the X OFF register 365' and writing the resulting number, referred to as XMP, into an additional register 579, as shown in FIG. 40g. Subsequently, the XMP number is read into the XSC register 121' as the tenth updated X axis microposition number. Similarly, for the Y axis, just prior to the tenth update of the

YSC register 119', the number stored in the YCP register 115' is transformed from part coordinates to machine coordinates by adding thereto the offset number stored in the Y OFF register 367' and depositing the result in a second additional register, the YMP register 581, shown in FIG. 40g below the XMP register 579. Subsequently, the YMP number is read into the YSC register 119' as the tenth updated Y axis microposition number.

The timing of the reading into and writing from the XMP register 579 is controlled by a pair of associated program gates 579a and 579b and, similarly, a pair of program gates 581a and 581b are provided to control the READ and WRITE gates of the YMP register 581.

e. An Additional "Scratch Pad" Register

In transferring data from the buffer registers to the active registers in the second exemplary system of FIG. 40, certain corrections are made to the data being transferred. To simplify the computations involved, a "scratch pad" register 582 is provided and is shown in FIG. 40j below the Q register 583a. During the computational sequence, data is written into the scratch pad register 582 from the output of the computer 53' under the control of a control signal W/SP from a Program Gate Array 582 and the signals stored in the register 582 are transmitted therefrom to the computer 53' in response to an R/SP signal from a Program Gate Array 585.

3. Principal Operational Differences Between the Two Embodiments

a. Linear Interpolation is also in a Staggered Mode

In contrast to the operation of the first exemplary system shown in FIG. 9 in which linear interpolation is performed in a simultaneous manner, in the second exemplary embodiment both circular and linear interpolation are carried out in the staggered mode. As will become apparent as this description proceeds, this similarity between the linear and circular interpolation modes greatly simplifies both the operation of the system and the manner in which it may be explained.

b. Neither Macromove Adjustment Nor Next Block Correction is Used in Going From One Path Segment to the Next

Two different modes for going from one path segment to the next were described with reference to the first exemplary system. In terminating a linear path segment exactly at the end of an iteration, a method was described whereby some of the macromoves are modified in size in order to cause the linear path segment to reach its designated end point exactly at the end of a given iteration. On the other hand, when a circular path segment is to be terminated, the macromove adjustment method was not shown to be used, but instead, the I and J numbers representing the following path segment were described as being modified in order to compensate for any errors which might result from the fact that the circular path segment being terminated at the end of one of the iterations might not end exactly at its designated end point.

Neither of these methods is employed in the second exemplary system shown in FIG. 40.

The reason for the difference is that the system of FIG. 40 operates in the staggered mode in generating both linear and circular path segments. Further, it may be assumed that in generating a succession of path segments in which a circular path segment follows a linear path segment, or a linear path segment follows a circular path segment or two circular path segments follow one another, successive ones of such path segments are tangential. Since the successive path segments are tangential and since all of them are interpolated in the staggered mode, a smooth transition may be made from one path segment to the following path segment without the need to terminate any of the path segments at the end of a given iteration. Instead, such end points are permitted to be reached during an iteration and the I and J components associated with the following path segment are initially upgraded by an amount which is determined by the proportion of the last iteration which was completed in accordance with the preceding path segment.

c. The Time Relationship of the Macromoves ΔX and ΔY Relative to the Time Period ΔT is Reversed

In the first exemplary system of FIG. 9 the ΔX macromove is computed at the end of an iteration and is executed during the entire following iteration, while the ΔY macromove is calculated toward the end of the first half of an iteration and is executed during the following half of that iteration and the first half of the following iteration. In the second exemplary embodiment of FIG. 40 the times for the computation and the execution of macromoves are exactly the reverse. A ΔY macromove is executed during a single iteration, and a ΔX macromove is executed during the second half of a given iteration and the first half of the following iteration. These time relationships are illustrated in the vector diagram of FIG. 41b for a linear path segment and in the vector diagram of FIG. 42 for a circular path segment. In both diagrams a time period corresponding to an iteration is indicated by the vector quantity ΔT , and a series of time periods which are staggered relative to the time periods ΔT , and which correspond to the time periods during which successive ΔX macromoves are executed, are indicated as the vector quantities $\Delta T'$.

d. Upgrading of XCP and YCP is Staggered

In the first exemplary system, discussed with reference to FIG. 9, the X axis macroposition XCP and the Y axis macroposition YCP are both upgraded during the first column of an iteration by an amount which is respectively ΔX and ΔY . The times for upgrading XCP and YCP are so close that, for practical purposes, they may be considered as being upgraded substantially at the same time, or in step. In contrast, in the second exemplary system illustrated in FIG. 40, the X axis macroposition is upgraded toward the end of the first half of an iteration and the Y axis macroposition YCP is upgraded toward the end of the same iteration, so that the times for upgrading XCP and YCP are spaced substantially half a time period ΔT apart. Thus, they are staggered with respect to one another by the same amount as are the times which separate the computations for deriving the X axis macromove ΔX and the Y axis macromove ΔY .

e. Steps Performed For Looking Ahead Along The Y Axis Differ From Those Performed For Looking Ahead Along The X Axis

The look-ahead operations performed by the system of FIG. 40 for detecting proximity to a designated end point are closely comparable to the look-ahead operations performed by the system of FIG. 9 when generating a circular path segment. Due to the staggered upgrading of the XCP and the YCP macroposition numbers in the system of FIG. 40, however, the details of the look-ahead operations performed by the latter system differ from those of the system of FIG. 9.

It will be recalled that in the FIG. 9 system, both the XCP macroposition number and the YCP macroposition number are upgraded at the beginning of an iteration to the value representing the positions to be reached at the end of that iteration along the X and Y axes. Shortly after these upgrading operations are performed, a forward projection or "look ahead" is made along the "fast" axis (the axis along which motion is the faster) by adding to the upgraded macroposition number associated with the fast axis a number equal to one macromove for that axis. This sequence is performed regardless of whether it is the X or the Y axis along which motion is the faster. Consequently, at the start of each iteration a number is produced, representing the position that will be reached along the fast axis at the end of the iteration following that in which the forward projection, or look-ahead calculation, was performed. This number is then compared with the corresponding coordinate of the designated end point to determine whether or not it will be reached during the following iteration.

In the second exemplary embodiment, shown in FIG. 40, the forward projection or look-ahead computation is also performed at the start of each iteration and the objective is still to determine whether or not the designated path segment end point will be reached along the "fast" axis during the iteration following that in which the look-ahead is performed. And, as in the case of the first exemplary system, the forward projection is again made with reference to a macroposition number

associated with the fast axis. However, as pointed out above, unlike the first exemplary system, the second exemplary system upgrades the XCP macroposition number not at the start of an iteration, but toward the end of the first half thereof and the upgrading of the YCP macroposition number occurs near the end of an iteration. Thus, when motion along the X axis is the faster, so that the forward projection must be made with reference to the SCP macroposition number, the magnitude of the forward projection for the look-ahead computation must be increased by one-half macromove because the number stored in the XCP register at the time of the look-ahead computation represents not the position at the beginning of the iteration during which the look-ahead is performed, but the position which was passed at the end of the first half of the previous iteration.

The foregoing remarks will be more readily understood with reference to FIGS. 43 and 44, which are vector diagrams illustrating the look-ahead operations along the X and Y axes respectively. Referring first to FIG. 43, a linear path segment 587 is shown as being extended during successive time periods $\Delta T1$, $\Delta T2$, $\Delta T3$, and $\Delta T4$ toward a designated path segment end point having the coordinates XCEP and YCEP. The last iteration associated with the linear path segment 587, performed during the time period $\Delta T4$, is shown as being divided into two equal zones A and B, and the end point of the path segment is shown as being reached during the second of the zones, zone B. The significance of this distinction is similar to the distinction between a small overrun and a large overrun which was made in connection with the first exemplary embodiment of FIG. 9, and will be further explained subsequently.

The first part in the look-ahead process performed during the time period $\Delta T2$ is to compute $XCP_1 - XCEP$, which is the X coordinate distance from the designated path segment end point to the mid-iteration target point $P1$ passed midway through the time period $\Delta T1$. This distance is shown as the vector 589 in FIG. 43, and, as is indicated by the direction of the vector, is a negative quantity. As the next step in the look-ahead process during the time period $\Delta T2$ there is added to the previously calculated result $XCP_1 - XCEP$ a number representing $2\Delta X$, which is the distance to be traveled along the X axis during two time periods ΔT , and which is indicated in FIG. 43 by the vector 591. If $XCP_1 - XCEP$ is less than $2\Delta X$, the sign of the quantity $(XCP_1 - XCEP) + 2\Delta X$ is opposite the sign of the quantity $XCP_1 - XCEP$ and this is used as an indication that the designated end point will be reached along the X axis during the first half of the time period $\Delta T3$.

Rearranging the terms of the last equation yields $(XCP_1 + 2\Delta X) - XCEP$ which is seen to be, in effect, a comparison of the X axis end point coordinate XCEP with the mid-iteration X axis macroposition coordinate SCP_1 projected forward by two macromoves. Thus, in effect, a two-macromove look-ahead is performed at the beginning of the iteration but is projected forward from a coordinate that was passed at the mid-point of the previous iteration.

As may be seen from FIG. 43, the two-macromove look-ahead performed during time period $\Delta T2$ does not produce a change in sign, i.e., the test produces a negative response. Therefore, it is known that the end point will not be reached during the first half of $\Delta T3$. During the same time period $\Delta T2$ a second test, which in effect extends the previous look-ahead by one-half macromove, is performed to determine whether or not the end point XCEP/YCEP will be reached during the second half of the iteration associated with the time period $\Delta T3$. This is done simply by adding one-half macromove, shown in FIG. 43 as the vector 592, to the previous result $(XCP_1 - XCEP) + 2\Delta X$, yielding the new sum $(XCP_1 - XCEP) + 2.5\Delta X$. If the sign of the resulting sum is different than that of the previous sum, this is taken as an indication that the end point XCEP/YCEP will be reached along the X axis during the second half of the time period $\Delta T3$.

Rearranging the terms of the last equation yields $(XCP_1 + 2.5\Delta X) - XCEP$ and, following the same reasoning as that advanced with respect to the 2 macromove look-ahead, it may

be seen that the second test is, in effect, a 2.5 macromove look-ahead projected forward from the point XCP_1 at the start of the time period $\Delta T2$. In the particular example shown in FIG. 43 the last result is also negative; that is, there is no sign reversal, since the end point is not reached during either half of the time period $\Delta T3$.

At the beginning of the next time period $\Delta T3$, the two look-ahead tests are again performed and the computations involved are indicated by the vectors 593, 595 and 597. The result of the first of the tests is again negative because the designated end point XCEP/YCEP is not reached during the first half of the time period $\Delta T4$. However, the response to the second test performed during the time period $\Delta T3$ and indicated by the vector 597 is positive. Thus, the sign of the quantity $XCP_3 - XCEP$ is negative, whereas the sign of the quantity $XCP_3 + 2.5\Delta X - XCEP$ is positive. The sign reversal following the second test, but not the first, thus reveals that the designated end point XCEP/YCEP will be reached during the time period $\Delta T4$ and that it will be reached only during the second half of that time period.

FIG. 44 illustrates the look-ahead computations which are performed for a path segment which is so oriented that, in executing it, movement along the Y axis is faster than movement along the X axis. The path segment represented by the vector 599 is shown to be linear and to extend from a point P_0 through a succession of points $P_1 - P_6$ through which it passes during time periods $T1 - \Delta T3$. The designated end point represented by the coordinates XCEP/YCEP is shown as being reached during the first half, or zone A, of the time period $\Delta T4$.

The first test which is illustrated in FIG. 44 is that which is performed during the time period $\Delta T1$ and whose purpose it is to predict whether or not the designated end point will be reached along the Y axis during the first half of the time period $\Delta T2$. The test is performed during the first portion of the time period $\Delta T1$ and, similarly to the test illustrated in FIG. 43, consists of calculating the Y coordinate distance from the end point XCEP/YCEP to the point P_0 on the linear path segment 599. This distance is represented in FIG. 44 by the vector 601. The next step in the process is to add to the vector 601, which is a negative quantity, the quantity $1.5\Delta Y$ or the Y coordinate distance to be traveled during the time period $\Delta T1$ and the first half of the time period $\Delta T2$. The result is $YCP_0 - YCEP + 1.5\Delta Y$. Its sign is the same as the sign of $YCP_0 - YCEP$ because $1.5\Delta Y$ is less than $YCP_0 - YCEP$, since the designated end point coordinate XCEP is not reached during the first half of the time period $\Delta T2$. A second test is then performed by adding the quantity $1/2\Delta Y$, represented in FIG. 44 by the vector 605, to the last result. The new result, $YCP_0 - YCEP + 2\Delta Y$ is again negative because the designated end point YCEP is not reached during the second half of the time period $\Delta T2$ either.

At the beginning of the following time period $\Delta T2$ another pair of look-ahead operations or tests are performed, these being represented in FIG. 44 by the vectors 607, 609 and 611. The response to both of these tests is negative due to the fact that the designated end point coordinate YCEP is still not reached during the time period $\Delta T3$. Finally, during the following time period $\Delta T3$ when, as part of the first test the quantity $1.5\Delta Y$, shown as the vector 613, is added to the quantity $YCP_4 - YCEP$, shown as the vector 615, the resulting sum, $YCP_4 - YCEP + 1.5\Delta Y$ is positive because the vector 613 is larger than the vector 615, signifying that the Y coordinate distance from the end point XCEP/YCEP to the point $P4$ is less than $1.5\Delta Y$. In other words, the end point XCEP, YCEP will be reached during the first half of the time period $\Delta T4$. Thus, a reversal in sign from negative to positive occurs after the first look-ahead test during time period $\Delta T3$, and this change in sign is used to indicate that the end point will be reached during the first half of the following time period.

4. Operation of the Second Exemplary System

a. Summary

The various operational sequences performed by the second exemplary system of FIG. 40 during a time period ΔT , other-

The computational steps performed by the system of FIG. 40 during a "standard" linear iteration, that is, during a time period ΔT in which a linear path segment is being generated and during which the designated end point of that path segment has not been detected, are shown in flow chart form in FIG. 46 and in detail in Tables VII, VIII and IX.

	X000	X100	X200	X300	X400	X500	X600	X700	X800	X900
00.....	Advance to S6 ? S5 Advance to S7 ? S6			Advance to S4 ? S3		Advance to S6 ? S5, Adv. to S7 ? S6			Advance to S3 ? S2	
01.....									Advance to S4 ? S3	
02.....					CAR					CAR
03.....					R/ Δ X					R/ Δ Y
04.....					R/XCP					R/YCP
05.....					W/XCP					W/YCP
06.....										
07.....					R/XOFF					R/YOFF
08.....										
09 to 014.....						None				
015.....					W/XMP				W/YMP	
016.....	CAR					R/XMP				
017.....	R/ Δ X						R/ Δ X			
018.....	SAR						SAR			
019.....	R/XSC						R/XSC			
020.....										
021.....	W/XSC, CAR									
022.....	R/ Δ Y								R/YMP	
023.....	SAR									
024.....	R/YSC									
025.....										
026.....	W/YSC									
027.....	CAR									
27 to 90.....						None				
01.....										Start Tape Reader ? S9
02.....										Advance to S10 ? S9
03.....										Advance to S11 ? S10
04.....										Advance to S12 ? S11
05.....										Advance to S13 ? S12
06.....										Advance to S14 ? S13
07.....										Advance to S15 ? S14
08.....										Advance to S16 ? S15
09.....										Advance to S1 ? S16
Adv to S5 ? S4									Adv to S5 ? S4	

TABLE VIII

	COLUMN 400A	COLUMN 400B	COLUMN 400C	COLUMN 400D	COLUMN 900A	COLUMN 900B	COLUMN 900C	COLUMN 900D
	X400-LIN-PX2	X400-CIRC	X400-S5- (G ₀₂ +G ₀₃)	X400-S8- (G ₀₂ +G ₀₃)	X900-LIN-PX2	X900-CIR	X900-S5- (G ₀₂ +G ₀₃)	X900-S8- (G ₀₂ +G ₀₃)
1 to 26	See Table VII							
27 to 34	No computations							
35	R/I	R/J	R/JBUF	R/J	R/J	R/I	R/IBUF	R/I
36								
37	R/Q	R/Q	R/QBUF	R/Q BUF	R/Q	R/Q	R/Q BUF	R/Q BUF
38	Multiply							
39	Carry Out	Carry Out	Carry Out	Carry Out	Carry Out	Carry Out	Carry Out	Carry Out
40	I×Q	±J×Q	±JBUF× QBUF	±J×QBUF	J×Q	±I×Q	±IBUF× QBUF	±I×QBUF
55	Multiplication	Multiplication	Multiplication	Multiplication	Multiplication	Multiplication	Multiplication	Multiplication
56	SL4							
57	R/X R				R/YR			
58	SR4							
59	W/ΔX				W/ΔY			
60	ZR1							
61	SL4							
62	WXR, CAR				W/YR, CAR			
63		R/I				R/J		
64		R/ΔX, SX				R/ΔY, SX		
65		W/I				W/J		
66	CAR							
67 to 99	See Table VII							

TABLE IX
Steps peculiar to Mode 1

	1000
1 to 32	None
33	R/L ? LIN; R/J ? CIRC
34	R/J ? LIN; R/I ? CIRC, CX
35	RF/FAX ? ≥; C/FAX ? <
36	FAR, R/XCP ? FAX; R/YCP ? FAX
37	FAR
38	SX, R/XCEP ? FAX
39	SX, R/YCEP ? FAX
40	SAL
41	R/ΔY ? FAX; R/ΔY ? FAX
42	R/ΔX ? FAX, SAL ? FAX
43	SAL ? FAX, R/ΔY ? FAX
44	R/ΔY ? FAX
45	R/ΔY ? FAX
46	R/ΔY ? FAX
47	ADVANCE to S2 ? S1-SST
48	ADVANCE to S3 ? S2
49	R/ΔX ? FAX, R/ΔY ? FAX
50	R/ΔX ? FAX, R/ΔY ? FAX
51	R/ΔX ? FAX, R/ΔY ? FAX
52	R/ΔX ? FAX, R/ΔY ? FAX
53	ADVANCE to S2 ? S1-SST
54 to 99	None

TABLE X
Steps peculiar to Mode 2

	2300	2800
1 to 69	None	
70	CAR, R/YCP	CAR, R/XCP
71	R/YCEP, SX	R/XCEP, SX
72	R/ΔY	R/ΔX
73	W/SP	W/SP
74	CAR, R/JBUF	CAR, R/IBUF
75	R/SP, SX	R/SP, SX
76	W/I ? (G ₀₂ + G ₀₃), CAR	W/I ? (G ₀₂ + G ₀₃), CAR
77	R/QBUF ? S7	
78	W/Q ? S7, CAR	
79	R/GBUF ? S7	
80	W/G ? S7, CAR	
81	R/IBUF ? S7-LIN	
82	W/I ? S7-LIN	
83	CAR	
84	R/JBUF ? S7-LIN	
85	W/J ? S7-LIN	
86	CAR	
87	ADVANCE to S8 ? S7	
88 to 99	None	

Table VII, organized in 10 columns of 100 steps each, shows those operations which are always performed during a given program step regardless of the mode in which the system operates.

Table VIII shows four different series of operations for each of the columns 400 and 900 of an iteration. The first pair, labeled 400A and 900A, applies during a standard linear iteration. The second pair, labeled 400B and 900B, applies to a

TABLE XI
Steps peculiar to Mode 3

	3300	3800
1 to 34	None	
35	R/XEP	
36	W/XCEP	
37	CAR	
38	R/YEP	
39	W/YCEP	
40	CAR	
41 to 78	None	
79	ADVANCE to S9 ? on S8	
80 to 99	None	

standard circular iteration. The third pair, labeled 400C and 900C, applies to certain types of linear to circular transitional iterations, and the fourth pair, labeled 400D and 900D, is applicable to another type of linear to circular transitional iteration.

Table IX represents those computational steps which are peculiar to Mode 1 of the system of FIG. 40, the mode in which the system is placed in order to perform a standard linear or a standard circular iteration.

Turning now to the detailed description of the steps carried out by the system of FIG. 40 in performing a standard linear iteration, it may be assumed that the E.O.B. counter 186' has been placed in its S1 count state, and that a signal therefore appears on the S1 output of its associated decoder 187'. This signal is gated through the OR gate 570 of the Time Base Mode Selector 69' on to line M1' of the Multi-Mode Time Base Logic 67' in FIG. 40d. As explained with reference to the first exemplary embodiment of FIG. 9, the signal on the line M1' enables the first group of AND gates 175' so that, as the hundreds BCD to D converter 171' of the Time Base and Digital Sweep 63' (FIG. 40b) is stepped through its 10 count states 000 through 900, its output signals appear on the 1000 through 1900 output of the group of AND gates 175'. This set of signals is used to provide the gating necessary to place the system in its first principal mode of operation.

The operations which are performed during a standard linear iteration will be described with particular reference to FIG. 58 which is a vector diagram illustrating the last staggered linear iteration in which no overrun has been detected, this iteration occurring during time period ΔT998 and being also shown in flow chart form in FIG. 46. It will be understood that, during the time periods preceding the particular time period ΔT998, many iterations of the type shown in FIG. 46 will have been performed by the system in executing a linear path segment. With reference to FIG. 46, it will be noted that the microposition numbers stored in the XSC register 121' and in the YSC register 119' are updated during each of the 10 sub-periods, or columns, of the iteration during the pro-

gram steps 16 through 26 of the respective columns. It will also be noted that the XCP and YCP macroposition registers 117' and 115' are upgraded during the initial program steps of the X400 and X900 columns of the iteration. The updating and upgrading operations will be explained subsequently and attention will now be directed to the specific steps involved in performing the look-ahead process which is shown to be carried out during the initial column of the iteration.

2. The Two Stage Look-Ahead Along the X and Y Axes

The look-ahead process was discussed in principle with reference to FIGS. 43 and 44. Its details are shown in Table IX. The first test is to determine the identity of the fast axis. This is achieved in the same manner in which it was performed in the first exemplary system, namely, by comparison of the I and J components of the path segment being generated. Accordingly, the I and J numbers are transmitted to the computer 53' during program steps 1033 and 1034 from the I and J registers 99' and 101' in response to R/I and R/J signals applied to the respective registers by their associated Program Gate Arrays 400' and 391' respectively. The requisite gating for producing the signals R/I and R/J are indicated in PGA 400' as condition 5 and in PGA 391' as condition 6.

Concurrently with the transmission of the number from the J register 101', a compare-instruct signal is sent to the computer 53' from its associated "FP" and "CX" PGA 453' (condition 1). During the following step 1035 the FAX flip-flop 455' which is connected to the compare outputs of the computer 53' is preset, thereby indicating by a logic 1 voltage level on its FAX output that motion along the X axis is faster than it is along the Y axis. On the other hand, if J is greater than I, the flip-flop is reset and this fact is indicated by a logic 1 voltage level on its FAX output.

The steps which follow comprise a two-stage look-ahead along the fast axis, as described in principle with reference to FIG. 43 and are quite similar to the look-ahead computations discussed in connection with the first embodiment of FIG. 9 in detail with reference to Tables VI and VII in Section H.2.h.(4).

During program steps 1036-1046, calculations are performed to determine the X coordinate distance remaining from the mid-point of the previous iteration occurring during $\Delta T997$ and the designated end point XCEP/YCEP of the linear path segment illustrated in FIG. 58. Steps are then performed for comparing this remaining X coordinate distance with the quantity $2 \Delta X$, so as to determine thereby whether or not the designated end point will be reached during the first half of the following iteration $\Delta T999$. Specifically, during program steps 1036 the Accumulator Register 53' is cleared in response to a signal from the CAR PGA 351 (condition 10). During the same program step the number stored in the XCP register 117' representing the X coordinate of the point along the path segment reached at mid-point of the previous iteration $\Delta T997$ is transmitted from the register to the computer 53' in response to a signal from the R/XCP PGA 421' (condition 2). During the following program step 1037 the contents of the Answer Register 53a', representing the just transmitted XCP number, are shifted one position to the right in order to bring it in alignment with the number stored in the XCEP register 95' which is then transmitted to the computer 53' during the following program step 1038 in response to a signal applied to the XCEP register 95' from its associated R/XCEP PGA 359' (condition 1).

In order to subtract the XCEP number from the XCP number, a subtract control signal is applied to the computer 53 from the SX PGA 479' (condition 3). The result, Xcp - XCEP, appearing in the Answer Register 53a' during program step 1039 is shifted one position to the left therein in order to bring it into alignment with the macromove number ΔX stored in the ΔX register 109' with which it is to be compared next (SAL PGA condition 1).

During the following two program steps 1040 and 1041 the macromove number stored in the ΔX register 109' is read into the computer 53' twice, additively, in response to successive

READ control signals applied to the ΔX register 109' by its associated R/ ΔX PGA 409' (conditions 12 and 13). The number appearing in the Answer Register 53a' is XCP - XCEP + $2 \Delta X$ and its sign will be the same as that of the number XCP - XCEP, provided that XCP - XCEP is not less than $2 \Delta X$, or, in other words, that the X coordinate distance remaining from the mid-point of the iteration $\Delta T997$ to the designated end point XCEP/YCEP of the linear path segment illustrated in FIG. 58 is not less than $2 \Delta X$. If this is not the case, there will be a sign reversal. To detect this sign reversal, the sign of the number XCP - XCEP appearing in the Answer Register 53a' during program step 1039 is stored in the ACNX sign storage flip-flop 465' during that program step, as previously explained in detail in Section G.2.d.(2)(b) with reference to the exemplary system of FIG. 9. This sign is then effectively compared with the sign of the number XCP - XCEP + $2 \Delta X$ which appears in the Answer Register 53a' during program step 1042. In particular, if the sign of the number appearing in the Answer Register 53a' during the program step 1047 is different from the sign of the number which had appeared in the Register during program step 1039 and which was stored in the ACNX flip-flop 465' during that program step, a signal represented by the symbol SST is produced during program step 1047 by the Fast Axis Projected Overrun Detector 71' and in response to that signal the Program Gate Array 183' of the Time Base Mode Selector 69' applies a pulse to the E.O.B. counter 186' (condition 1) so as to advance the Counter to its S2 count state. Advancement of the Counter 186' into its S2 count state therefore signifies that an overrun has been detected and that it will occur during the first half of the following iteration.

As explained previously with reference to FIG. 43, the test just described is, in effect, a 2 macromove look-ahead projected forward at the beginning of an iteration from a point along the X axis which was passed during the mid-point of the previous iteration. It will be readily seen that, as of the time when the projection is made, it is, in effect, a 1.5 macromove projection of position along the X axis from the position which is to be passed along the X axis at that time. It is in this sense that the term "Determine remaining distance and compare with 1.5 macromoves" is used in the first column of FIG. 46 and it is for this reason that the first projection forward is illustrated in FIG. 58 as the vector 615 which is 1.5 ΔX in length and which is projected forward from the point P₀ which is passed at the beginning of the iteration $\Delta T998$.

To perform the second look-ahead operation during the iteration $\Delta T998$, the $\frac{1}{2} \Delta X$ is added to the previously accumulated result so as to increase, in effect, the forward projection from 1.5 ΔX , represented by the vector 615 in FIG. 58, to 2 ΔX , indicated in FIG. 58 as the vector 617. To accomplish this, during the program step 1048-1052 the macromove number stored in the ΔX register 109' is additively read into the computer 53' five times in response to READ gating signals from the R/ ΔX PGA 409' (condition 14). To give these operations the effect of adding 0.5 ΔX to the previously accumulated sum in the Answer Register 53a', that sum is shifted one position to the left during a previous program step, 1042, in response to an appropriate control signal from the SAL PGA 469' (condition 3). The result, appearing in the Answer Register 53a' during the program step 1053, is XCP - XCEP + $2.5 \Delta X$, where XCP is the macroposition number representing the X coordinate of the point passed during the mid-point of the previous iteration, $\Delta T997$. With reference to FIG. 58 it will be seen that the result may be restated as XCP₀ - XCEP + $2 \Delta X$. Means are provided for advancing the E.O.B. counter 186', provided it remained in its count state S1 after the first test and provided also that, as a result of the second test a reversal in the sign of the number stored in the Answer Register 53a' occurred and was signaled as an SST signal to the PGA 183' of the Time Base Mode Selector 69'. The means for so advancing the E.O.B. counter 186' is represented by condition 3 of the PGA 183'.

In the foregoing paragraphs the particular computational steps involved in performing the look-ahead operations for the $\Delta T998$ iteration have been set out. The results of the look-ahead tests performed during the $\Delta T998$ iteration are seen to be negative, since both the 1.5 ΔX look-ahead vector 615 and the 2 ΔX look-ahead vector 617 are smaller than the vector 614, representing the distance remaining to the end point coordinate XCEP. Consequently, the E.O.B. counter 186' will not have been advanced to its S3 count state during program step 1048, nor will it have been advanced to its S2 count state during program step 1053. These facts are indicated in FIG. 46 by crossing out the blocks which refer to the advancing of the E.O.B. counter 186'.

Before continuing with the remaining operations performed during the $\Delta T998$ iteration which is under discussion the look-ahead computation which would have been carried out had the Y axis been the fast axis will be briefly considered.

If it be assumed that motion along the Y axis is to progress faster than along the X axis, the situation which is depicted by FIG. 44, the comparison of J and I during steps 1033 and 1034 would reveal during the following program step 1035 that I is less than J, thereby causing the FAX flip-flop 455' to be cleared by resetting it. The identity of the fast axis would then be indicated by a logic 1 signal on the FAX output of the flip-flop 455'.

In the following three program steps, 1036-1038, the distance remaining along the Y axis to the designated end point XCEP/YCEP would be computed by subtracting YCEP from the macroposition number then stored in the YCP register 115', which, in the example illustrated in FIG. 44, is YCP₀. Specifically, during program step 1036 the macroposition number YCP₀ would be read from the YCP register 115' in response to a READ gating signal from its associated R/YCP PGA 425' (condition 2). Next, during program step 1037, the YCP₀ number would be shifted one position to the right in the Answer Register 53a' in response to a control signal from the SAR PGA 411' (condition 19) in order to bring the YCP number into alignment with the YCEP number which is to be subtracted therefrom during the following program step. During that program step, No. 1038, the YCEP end point coordinate is read into the computer 53' from the YCEP register 97' by a signal applied to that register from its associated R/YCEP PGA 487' (condition 1) and is read into the computer subtractively by the concurrent application thereto of a subtract control signal from the SX PGA 579' (condition 3). The result, appearing in the Answer Register 53a' during the program step 1039 is YCP₀ - YCEP and is shifted one position to the left in response to a control signal from the SAL PGA 463' (condition 1) in order to bring it into alignment with the ΔY macromove number which is to be added thereto during subsequent program steps, as will now be explained.

During program step 1040 the macromove number ΔY is read into the computer 53' from the ΔY register 107' in response to a gating signal from the R/ ΔY PGA 415' (condition 12). The sum, which appears during the following program step in Answer Register 53a', is YCP₀ - YCEP + ΔY . To this result there is added during the program steps 1042-1046 the quantity $\frac{1}{2} \Delta Y$ in order to complete the 1.5 ΔY look-ahead calculation, YCP₀ - YCEP + 1.5 ΔY . This is achieved by first shifting the previously accumulated sum in the Answer Register 53a', YCP₀ - YCEP + ΔY one position to the left during program step 1041 in response to a Shift Left signal from the SAL PGA 463' (condition 2) and by reading the macromove number ΔY into the computer 53' five times from the ΔY register 107' during program steps 1042-1046 in response to control signals from the R/ ΔY PGA 415' (condition 13). Due to the shifting to the left of the sum YCP₀ - YCEP + ΔY the effect of additively reading the macromove number ΔY five times into the computer in its normal field is effectively to add $\frac{1}{2} \Delta Y$ to the aforementioned sum. Consequently, the result stored in the Answer Register 53a' during program step 1047 is YCP₀ - YCEP + 1.5 ΔY . In the manner explained previously

with reference to the test for the X axis, if the sign of the sum stored in the Answer Register 53a' during program step 1047 differs from the sign of the sum previously stored in that register during program step 1039 an SST signal is applied by the Fast Axis Projected Overrun Detector 71' to the Time Base Mode Selector 69' whose Program Gate Array 183' is conditioned to advance the E.O.B. counter 186' to its S2 count state (condition 1).

To form the second look-ahead test during the time period $\Delta T1$, indicated by the $\frac{1}{2} \Delta Y$ vector 605 in FIG. 44, the macromove number ΔY is read from the ΔY register 107' into the computer 53' five more times during the program steps 1048-1052 by the application to the register of five additional READ control signals from its associated R/ ΔY PGA 415' (condition 14). For the same reasons advanced with reference to the program steps 1042-1048 the effect of these operations is to increase the sum stored in the Answer Register 53a' so that, during program step 1053 it is YCP₀ - YCEP + 2 ΔY . If the sign of this sum is different from the sign of the sum which had been stored in the Answer Register 53a' during the program step 1039, then, for the same reasons previously given with reference to the test for the X axis, the E.O.B. counter 186' is advanced from its S1 count state to its S2 count state by the application of an SST signal to the PGA 183' (condition 3).

3. Computing ΔX and ΔY Macromoves

Two other major computational sequences shown to be performed during a typical linear iteration in the flow chart of FIG. 46 are those directed to computing macromove numbers ΔX and ΔY . The computational sequence by which the ΔX macromove number is derived is shown in FIG. 46 to occur during the program steps 435-462 and the computations for deriving the macromove number ΔY are shown as being carried out during the program steps 935-962. The computational steps for deriving ΔX and ΔY are identical to the corresponding computational sequences for deriving those quantities as disclosed with reference to the first exemplary system, in FIG. 18 generally, and in Tables IV and IV-A in detail. The steps involved in completing the computation J-Q are shown in FIG. 18 to occur during program steps 1817-1864. Similarly, the multiplication of I-Q to compute ΔX is shown in FIG. 18 to occur during program steps 1917-1964. Another operation, that of storing the four least significant digits of the product in the remainder registers XR and YR, which in FIG. 46 is shown to be performed during program steps 460-462 and 960-962 are indicated in FIG. 18 as being performed during program steps 1869-1871 and 1969-1971. These operations are explained in detail in Section G.2. d.(4) under the heading "Data Processing Steps Associated with Calculating Adjusted Macromoves." For an explanation of the reasons why these computations are made, reference should be made to the cited section. At this point, only the computations themselves and the control signals related to them will be described.

Referring first to the computations shown in FIG. 46 to be performed in Column X400, the detailed steps associated therewith are shown in Table VIII, Column 400A. This is so because the system is generating a linear path segment and therefore an enabling signal appears on the LIN line of the path type indicator 106'. Furthermore, since the iteration under discussion is one during which a designated end point has not yet been detected, an enabling signal is also produced by the inverter 577 on its PX2 output line (FIG. 40d). As indicated at the heading of Column 400A, it is these conditions which are required in order for that column of computations to be performed. The first step in the sequence is to read the number I into the computer 53' from the I register 99', and this is done during program step 435 in response to a READ gating signal from the R/I PGA (condition 3) associated with the register.

With the multiplicand I in the computer 53', the multiplier Q is transmitted from the Q register 103' during program step 437 in response to a READ gating signal from the R/Q PGA 389' (condition 1). During the following program steps

439-455 the multiplication of I-Q is carried out by the computer 53', this operation being initiated by a multiply control signal applied to the computer 53' during program step 437 from the PGA 393' (condition 1). The product I-Q appears in the Answer Register 53a' during program step 456 and is shifted four positions to the left by a Shift Left control signal from the SL4 PGA 347' (condition 1) so as to bring the four least significant digits of the product from the Overhanging Register 53b' into the Answer Register 53a'. The remainder digits from the previous multiplication of I-Q are next read additively into the computer 53' from the XR register 113' by a READ control signal from the R/XR PGA 400' during the program step 457, and the resulting sum is then returned to its original position by shifting it four positions to the right by means of Shift Right control signals from the SR4 PGA 501' (condition 1). With the four least significant digits of the product again out of the Answer Register 53a', the remaining digits of the product are transmitted from the Answer Register to the ΔX register 109' by a WRITE control signal applied to the register by its associated W/ ΔX PGA 401' during program step 459. The digits which were just transmitted from the Answer Register 53a' but which had not been removed therefrom by that operation are next eliminated during program step 460 by a control signal from the ZR1 PGA 397' (condition 1). With the Answer Register 53a' thus cleared, the four least significant digits of the product I-Q, as modified by the corresponding digits of the previous product I-Q, are shifted from the Overhanging Register 53b' into the Answer Register 53a' in response to Shift Left control signals from the SL4 PGA 347' (condition 3) and are transmitted to the XR register 113' during the following program step 462 by a WRITE control signal applied to the register by its associated W/XR PGA 403'. Finally, during the same program step, but following the transmission of the XR remainder digits, the entire Accumulator Register 53ab' is cleared by a CLEAR control signal from the CAR PGA 351' (condition 6).

Turning now to the steps involved in the computation of ΔY which are shown to occur in FIG. 46 during its Column 900, the steps involved are those detailed in Table VIII, Column 900A for the same reason as those cited in connection with Column 400A of that Table. Briefly, the number J is read into the computer 53 during program step 935 by a READ signal from the R/J PGA 391' and the number Q is read from the Q register 103' during program step 437 by a control signal from the R/Q PGA 389' (condition 2). In response to a multiply control signal applied to computer 53' from the X PGA 393' (condition 2), the multiplication of J-Q is performed during the program steps 939 - 955. The product is shifted four positions to the left in the Accumulator Register 53ab' during program step 956 by a signal from the SL4 PGA 347' (condition 2) and the remainder digits of the previously computed product of J-Q are additively read into the computer 53' from the YR register 111' by a READ control signal from the R/YR PGA 399'. During the following program step 958 the corrected product is shifted four positions to the right in the Accumulator Register 53ab' by a Shift Right control signal from the SR4 PGA 501' (condition 2). The number remaining in the Answer Register 53a' is transferred during the next program step 959 to the ΔY register 107' by a WRITE control signal from its associated PGA 395'. During the next two program steps 960 and 961 the Answer Register portion 53a' of the Accumulator Register 53ab' is cleared by a signal from the ZR1 PGA 397' (condition 2) and the digits remaining stored in the Overhanging Register portion 53b' are transferred into the just cleared Answer Register portion 53a' by Shift Left control signals from the SL4 PGA 347' (condition 4). As the final part of the computation, the digits just transferred into the Answer Register portion 53a' are transmitted therefrom into the YR remainder register 111' by application of a WRITE control signal to the register by its associated W/YR PGA 398'. The Register 53ab' is then cleared (CAR PGA 351' condition 7).

4. Updating the XSC and YSC Microposition Registers

The third major type of operation which is shown in 46 as being performed during a standard linear iteration is that of updating the XSC and YSC registers 121' and 119'. These operations are almost identical to the corresponding operations performed by the first exemplary system of FIG. 9 and discussed with reference to Table III in Section G.2.b. The most apparent difference is that the updating operations are performed slightly later during the iteration, during the band of program steps 16-26, in contrast to the band of program steps 00-10 during which the corresponding operations were shown in Table III to be performed in the first exemplary system.

Referring again to Table VII, the first step in updating the XSC register 121' is to clear the Answer Register 53a' by a CLEAR signal applied to computer 53' from its CAR PGA 351' (condition 3). During the same program step, but a short time later, the macromove number in the ΔX register 109' is read into the computer 53' by a READ control signal applied to the register from its R/ ΔX PGA 409' (condition 2). The ΔX number appearing in the Answer Register 53a' during the next program step 017 is shifted to the right one position in order to divide it by 10. This is in response to a Shift Right control signal from the SAR PGA 411' (condition 1). Thus, when the current microposition number is read into the computer from the XSC register 121' during the following program step 018 by a control signal from its associated R/XSC PGA 412' (condition 1), the number in the Answer Register 53a' to which that microposition number is added is 0.1 ΔX . The resulting sum which appears in the Answer Register 53a' during the program step 019 is $XSC + 0.1 \Delta X$, or the updated XCS microposition number. This number is then transmitted to the XSC register 121' during program step 021 by a WRITE control signal applied to the register by its control gate 413'.

The YSC register 119' is updated during the following seven program steps and in preparation thereto the Answer Register 53a' is cleared by a CLEAR control signal from the CAR PGA 351' (condition 4). After the Answer Register 53a' has been cleared, but during the same program step 021, the Y axis macromove number is read into the computer 53' from the ΔY register 107' by a READ control signal from the R/ ΔY PGA 415' (condition 2). The ΔY macromove number thus stored in the Answer Register 53a' is divided by 10 by shifting it one position to the right in the register during the program step 022 in response to a Shift Right control pulse from the SAR PGA 411' (condition 10). As a result, the number in the Answer Register 53a' during program step 023 is 0.1 ΔY to which the Y axis macroposition number is added during the program step 023 by transmitting that number from the YSC register 119' by application of a READ control signal thereto from its associated R/YSC PGA 417' (condition 1.) The resulting updated Y axis macroposition number $YSC + 0.1 \Delta Y$ is entered into the YSC register 119' during a subsequent program step 026 by a WRITE control pulse applied to the register by the W/YSC program gate 419'. With the updating sequence thus completed the Accumulator Register 53ab' is cleared by a control signal from the CAR PGA 351' (condition 5).

The foregoing sequence for updating the number stored in the XSC register 121' is repeated during each of the columns or sub-periods X100, X200 and X300 of each iteration performed by the system of FIG. 40. This fact is indicated in Table VII by the arrows which extend horizontally from the computational steps listed in its first column, X000, related to the updating of the XSC register, through the following three Columns X100, X200 and X300. Furthermore, the operations associated with the updating of the YSC register 119' are performed in the same manner during each of the following eight columns X100 - X800 of each iteration performed by the system, as indicated in Table VII by the extension of horizontal arrows in that Table from the steps listed in the first Column X000 related to the updating of the YSC register 119' through the following eight columns of the Tables, X100 - X800.

The manner in which the X axis microposition number in the XCS register 121' is updated during the fifth Column X400 of each iteration differs from that in which the register is updated during the preceding four columns. Generally, during the Column X400, and prior to the updating of the XSC register 121', the macroposition number in the XCP register 117' is upgraded to a new value, that number is translated into machine coordinates, and the translated XCP number, referred to as the XMP number, is written into the XSC register 121' as the updated X axis macroposition number. The exact manner for upgrading the XCP register 117' and for translating the upgraded number in it into machine coordinates will be described subsequently. At this point it will suffice to say that there is stored in the XMP register 579 during program step 415 the new upgraded XMP number representing, in machine coordinates, the position along the X axis which should be reached during the sub-period of time represented by that column. The number stored in the XMP register 579 is then entered into the computer 53' during program step 416 and this number is then written into the XSC register 121' by a WRITE control signal applied to the register by its program gate 413' during program step 421.

During each of the following five sub-periods of time represented by Columns X500 - X900 of Table VII, the number in the XSC register 121' is updated by $0.1 \Delta X$ in the same manner as was described with reference to the first four columns X000 - X300 of Table VII. Thus, the appropriate READ control signals for entering the macromove number from the ΔX register 109' into the computer 53' are indicated in the R/ ΔX PGA 409' as conditions 6-10. Similarly, the Shift Right control signals for dividing the ΔX number by 10 are indicated in the SAR PGA 411' as conditions 5-9. Finally, the control signals for entering the microposition number to be updated, from the XSC register 121' into computer 53', are represented by conditions 5-9 in the R/XSC PGA 412'.

To summarize briefly, it is seen that the X axis microposition number stored in the XSC register 121' is updated once during each iteration by writing into the register the upgraded macroposition number for the X axis in machine coordinates, and that the XSC register is updated nine more times simply by adding $0.1 \Delta X$ to it each time. The same sequence is also followed in updating of the YSC register 119' except that it is updated with the upgraded Y axis macroposition number during the last sub-period of the iteration, represented by Column X900 in Table VII. Thus, the upgraded macroposition number YMP in machine coordinates is read into the YSC register 119' during Column X900, and $0.1 \Delta Y$ is added to that number during each of the first nine sub-periods, or Columns X000-X800 of the following iteration.

Specifically, in a manner similar to that applicable to the XCP register 117' and to be described subsequently, the number in the YCP register 115' is upgraded and translated into machine coordinates, and as so upgraded and translated, is stored in the YMP register 581 during program step 915. It is then read into the computer 53' from the YMP register 581 during program step 921 and is written into the YSC register 119' during program step 926 in response to a WRITE control signal applied to that register by its W/YSC PGA 419', as also indicated by the horizontal arrow extending from the corresponding step shown in Column X000.

5. Upgrading the XCP and YCP Macroposition Registers

As explained previously with reference to FIG. 46, the macroposition registers XCP and YCP 117' and 115' are upgraded in a staggered manner during Column X400 and X900 of the standard linear iteration. Referring further to FIG. 46, and in particular to the upgrading of the XCP register 117', the detailed steps involved appear in Table VII. In preparation for the upgrading of XCP register 117', the Accumulator Register 53ab' is cleared during program step 403 (CAR PGA 351' condition 1). Subsequently, during program step 404 the macromove number in the ΔX register 109' is read into the computer 53' (R/ ΔX PGA 409' condition 1) and during the following program step 405 the macroposition number then in

the XCP register 117' is additively read into the computer 53' (R/XCP PGA 421' condition 1). The result, $XCP + \Delta X$, appearing in the Answer Register 53a' during program step 406, represents the target coordinate for the mid-point of this iteration. Thus, in the example illustrated in FIG. 46, which corresponds to the time period ΔT 998 illustrated in FIG. 58, the new macroposition number appearing in the Answer Register 53a' during program step 406 is XCP_1 , shown in FIG. 58 as the X coordinate to be reached halfway through the present iteration ΔT 998.

During the same program step the upgraded XCP number is transmitted to the XCP register 117' (W/XCP PGA). Similarly, to upgrade the YCP register 115', the Accumulator Register 53ab' is again cleared during program step 903 and, during the following program step 904, the macromove number ΔY is read into the computer from the ΔY register 107' by a READ control signal from the R/ ΔY PGA 415' (condition 1). The existing macroposition number is read into the computer 53' from the YCP register 115' by a control signal from the R/YCP PGA 425' (condition 1).

The resulting sum, appearing in the Answer Register 53a' during the following program step 906, representing the upgraded Y axis macroposition coordinate identified in FIG. 58 as YCP_2 , is written into the YCP register 115' during that program step by a WRITE signal from the W/YCP PGA 427'.

To convert the upgraded macroposition numbers stored in the XCP and YCP registers 117' and 115' from part coordinates into machine coordinates, each of the upgrading operations just described is followed by an operation whereby the offsets stored in the X OFF and Y OFF registers 365' and 367' are added to the macroposition numbers. Referring first to the upgraded XCP macroposition number, and with particular reference to Table VII, the number stored in the X OFF register 365' is read into the computer 53' during program step 408 in response to a READ control signal from the program gate 373' associated with that register. The number thus read into the computer is automatically added to the upgraded XCP number which, although transmitted from the computer to the XCP register 117' during the previous program step 406, remained in the Answer Register 53a' in the absence of an intervening CLEAR signal. Accordingly, there appears in the Answer Register 53a' during the next program step 409 a number which is, in effect, the XCP number in machine coordinates and which will be referred to as the XMP macroposition number. It is entered into the XMP register 579 by a WRITE control signal from its associated W/XMP program gate 579b during program step 415.

The sole reason for producing the XMP macroposition number is because that number is used for the 10th updating of the microposition coordinate number in the XSC register 121'. Toward this end, as explained previously, the number just stored in the XMP register 579 is read into the computer 53' during program step 416 by a READ control signal applied to the register by its associated R/XMP program gate 579a.

Subsequently, during program step 421 when the contents of the Answer Register 53a' are transferred to the XSC register 121', as indicated by the first horizontally extending line in the 021 program step row in Table VII, the XMP number is automatically entered into the XSC register 121'.

Similarly, with reference to the tenth updating of the YSC register 119', with the upgraded Y axis macroposition number YCP still in the Answer Register 53a', the Y axis offset number is read into the Answer Register from the Y OFF register 367' in response to a READ control signal from its associated program gate 381'. The result, which is the YCP number in machine coordinates, or YMP appears in the Answer Register 53a' during program step 909 and is written into the YMP register 581 during program step 915 by a WRITE control signal applied to the register by its associated W/YMP program gate 581b. This number is subsequently utilized to effect the 10th update of the YSC register 119' by again transferring it from the YMP register 581 to the com-

puter 53' by means of a READ control signal applied to the register by its associated control gate 581a during program step 921. The number thus lodged in the Answer Register 53a' is automatically written into the YSC register 119' during program step 926 by a WRITE control signal produced by the W/YSC program gate 419' and represented by the first line extending toward the right in program step row 026 of Table VII.

c. Generating A Circular Path Segment

A general flow chart of the operations which are performed by the second exemplary system of FIG. 40 during a "standard" circular iteration appears in FIG. 49. FIG. 49 relates to a specific time period ΔT_{2498} which is also identified specifically in the vector diagram comprising FIG. 60. The significance of this time period is that it is the last staggered circular iteration during which no overrun has been detected. It will be understood that typically, iterations of the type shown generally in FIG. 49 will have been repetitively performed many times during previous time periods ΔT in generating the circular path segment. Further reference will be made to FIG. 60 in explaining the manner in which the circular path segment there shown is terminated and continued as a linear path segment. It will be sufficient to note at this point that, (a) the path segment which is executed during the time period ΔT_{2498} corresponding to the iteration shown in FIG. 49 extends between the points P_0 and P_2 , (b) the total excursion along the Y axis executed during that time period appears in FIG. 60 as the vector ΔY_{02} and finally (c) a vector quantity ΔX_{13} is computed shortly before the point P_1 is reached, this quantity representing the X axis macromove executed during the second half of the iteration ΔT_{2498} and the first half of the following iteration ΔT_{2499} .

The computations performed during a standard linear iteration (FIG. 46) and a standard circular iteration correspond in several respects. First the manner in which the look-ahead operations are performed in the standard circular mode is identical to the manner in which those operations are performed in the standard linear mode, except for the sequence in which I and J are read to determine the fast axis. As explained in connection with the first exemplary system, (Section G. 2. d. (2)(a)), the sequence is reversed for circular interpolation, so that J is read first (R/J PGA condition 5) and I is read second (R/I PGA condition 5). For this reason, except for the determination of the fast axis, the steps which are listed in Table IX and which relate to the look-ahead tests performed during the first part of an iteration are the same for a circular and a linear iteration. As in the case of FIG. 46, which relates to a linear iteration during which no overrun has been detected, those blocks in FIG. 49 which refer to the advancing of the E.O.B. counter 186' are crossed out.

Equally significantly, the updating and upgrading operations shown in FIG. 49 in relation to a circular iteration are identical to the corresponding operations shown in FIG. 46 with reference to a linear iteration. Again, since the operation of the second embodiment in updating and upgrading the microposition and macroposition registers respectively is the same in the linear and circular modes, those steps in Table VII which relate to these operations apply equally to both the linear and the circular mode of operation of the system.

Turning finally to the computation of ΔX and ΔY in the circular mode, shown to occur in the X400 and the X900 columns of FIG. 49, these necessarily differ from the corresponding operations performed in the linear mode for reasons which have been given previously in discussing the basic concept of linear and circular interpolation. It will be helpful to observe, however, that the computational sequences performed by the second exemplary embodiment of FIG. 40 in deriving ΔX and ΔY for a standard circular iteration are exactly the same as those which are carried out by the first exemplary embodiment of FIG. 9 in deriving the same quantities. This may be confirmed by comparing the operations which are shown to occur during steps 935-962 of FIG. 49 with the steps shown to be performed during the program

steps 1417-1471 of FIG. 31. From that comparison it will be seen that the computations for deriving ΔY are the same in both of the exemplary embodiments. Similarly, a comparison of the operations carried out by the second exemplary system during steps 435-462 with the steps performed by the first exemplary embodiment during program steps 1917-1971 will reveal that the computations for deriving ΔX are also identical in both exemplary embodiments.

The specific steps taken to derive ΔX and ΔY are those given in Table VIII, Columns 400B and 900B. With reference to Column 400B, in preparation for performing the multiplication of J-Q, the multiplicand J is read into the computer 53' from the J register 101' in response to a READ control signal from its associated R/J PGA 391' (condition 1). Next, the multiplier Q is read into the computer from the Q register 103' during program step 437 by a READ control signal from its R/Q PGA 389' (condition 3). It will be noted that both of the aforesaid conditions depend on an output on the CIRC line of the path type indicator 106', a signal which exists only when the system is to generate a circular path segment. The computer 53' is then instructed to multiply J-Q, this instruction originating from the X PGA 393' (condition 1). The product appears in the Accumulator Register 53ab' during program step 456 and is operated upon during the program steps 456-462 in the same manner and in response to the same control signals as described previously with reference to Column 400A of Table VIII. As a result of these operations all but the four least significant digits of the product are stored in the ΔX register 109' and the four least significant digits of the product are lodged in the XR register 113'.

A similar sequence is carried out during steps listed in Column 900B to derive the macromove component number ΔY . The multiplicand I and the multiplier Q are successively read into the computer 53' from the registers 99' and 103' during program steps 935 and 937 in response to READ control signals applied to the registers by their respective Program Gate Arrays 400' and 389' (conditions 1 and 4 respectively). The numbers thus transferred are multiplied in the computer during program steps 939-955 in response to a multiply control signal applied to the computer by the X PGA 393' (condition 2) with the result appearing in the Accumulator Register 53ab' of the computer during program step 956. This product is split into two parts during the program steps 957-962 and stored in the ΔY and YR registers 107' and 111' in the same manner and in response to the same control signals as described previously with reference to Column 900A of Table VIII in relation to performing a linear iteration.

In principle at least, the upgrading of I by subtracting ΔX and the upgrading of J by subtracting ΔY , as indicated in FIG. 49 with reference to a staggered circular iteration, are the same as the corresponding operations shown in FIG. 31 to be performed by the first exemplary system of FIG. 9 during a staggered circular iteration. There is a difference between the two, but it is only one of timing. Specifically, in the first exemplary embodiment shown in FIG. 9 and referred to with reference to FIG. 31 the I and J vector component numbers are upgraded by subtracting ΔX and ΔY from the respectively, but this does not occur until immediately prior to the time when the upgraded I and J vector component numbers become needed for the subsequent computation of I-Q and J-Q respectively. In contrast, in the second embodiment of FIG. 40, as indicated by the flow chart of FIG. 49, the I vector component number is upgraded by subtracting ΔX from it immediately after that ΔX number becomes available, that is, immediately after the ΔX number has been computed by multiplying J-Q. The upgraded I number, while available shortly before the second half of the iteration, i.e., toward the end of the 400 column of the iteration, is not utilized until half an iteration later when it is multiplied by Q to derive the next macromove number. Similarly, the J vector component number is computed by subtracting from it the macromove number ΔY immediately after that macromove number has been computed by multiplying I-Q. Again, the upgraded J vec-

tor component number simply remains stored and is not utilized until half an iteration later, at which time it is multiplied by Q to derive the next ΔX macromove number.

The above difference between the first and second exemplary embodiments in timing the upgrading of I and J is of no consequence to the operation of the system. It does illustrate, however, the many equivalent alternatives between which a designer may choose without departing from the scope of the present invention.

The particular steps for upgrading of I and J are listed in Table VIII, Columns 400B and 900B. Referring first to the upgrading of I, it is performed in exactly the same manner and for exactly the same reason as described previously with reference to Tables VI and VI-A in Sections H.2.f.(1) and H.2.f.(3). Thus, the I number is entered into computer 53' from the I register 99' in response to a signal from the R/I PGA 400' (condition 4) and the macromove number ΔX is subtracted therefrom during the following program step 463 by reading the number ΔX from the ΔX register 109' in response to a signal from the R/ ΔX PGA 409' (condition 11) while concurrently applying a subtract control signal to the computer 53' from the SX PGA 479' (condition 1). Finally, the remainder is transmitted from the Answer Register 53a' during program step 464 to the I register 99' as the upgraded I number by a control signal applied to the register by its associated W/I PGA 355' (condition 1). The sequence is terminated by clearing the entire Accumulator Register 53ab' by a clear control signal from the CAR PGA 351' (condition 8).

In a similar manner, to upgrade the J number, it is first transferred during program step 962 from the J register 101' by a signal from the R/J PGA 391' (condition 4). The just computed ΔY number is then subtracted from the J number during the following program step 963 by the application of a READ control signal to the ΔY register 107' from its R/ ΔY PGA 415' (condition 11) and by the application of a subtract control signal to the computer 53' by its SX PGA 479' (condition 2) during the same program step. The remainder, representing the upgraded J number, is written into the J register 101' during program step 964 by application of a WRITE control signal to the register from its W/J PGA 357' (condition 1).

d. Going From a Linear Path Segment To a Circular Path Segment

1. Assuming That the Desired Transition Is In Zone A

Reference will now be made to FIG. 58 to explain the manner in which a transition is made in the second exemplary system of FIG. 40 in going from a linear path segment to a circular path segment. The linear path segment is represented by the vector 621, and its X and Y components are shown by the vectors I_{LIN} and J_{LIN} . It will be understood that the generation of the illustrated linear path segment 621 will have begun during the iteration $\Delta T000$ and that a FIG. 46 type standard linear iteration will have been performed during that time period and during each of the following time periods up to and including $\Delta T998$. However, in order to reduce the size of FIG. 58, only the last of these iterations is shown.

Progress along the linear path segment 621 extends through points P0, P1, P2, P3, and P4 during time periods $\Delta T998$ and $\Delta T999$. The desired transition point between the linear path segment 621 and the following tangential circular path segment 623 is XCEP/YCEP and is crossed during the first half of the next iteration, $\Delta T1000$. Hence, it is termed a "Zone A" transition.

During iteration $\Delta T1000$ the path which is actually executed is neither purely linear nor purely circular. Therefore, the point P5 which falls on the extension of the linear path segment 621 is not exactly on the actual path which is executed during the transitional time period $\Delta T1000$. It is, however, a close approximation. Subsequent points which are passed at the beginning and midpoint of successive iterations along the circular path segment 623 are shown as the points P6, P7, P8, and P9.

The block of data representing the circular path segment 623 typically includes the inverse radius vector components I_{BLF} and J_{BLF} of the starting radius vector extending from the center C of the path segment 623 to the desired transition point XCEP/YCEP. Stated differently, I_{BLF} and J_{BLF} represent the X and Y coordinate distance from the transition point XCEP/YCEP to the center C of the circular path segment 623. As will be explained in detail, the transition from the linear path segment 621 to the circular path segment 623 will be made by changing over from the linear mode to the circular mode at the beginning of iteration $\Delta T1000$ for the Y axis and at the start of the second half of the iteration $\Delta T1000$ for the X axis.

Turning now to the description of the entire sequence of transition from the linear path segment 621 to the circular path segment 623, let it be assumed that the initial series of iterations $\Delta T1000 - \Delta T997$ have been performed so that the FIG. 46 type of iteration to be carried out during the time period $\Delta T998$ is about to begin. As explained previously with reference to FIG. 46, neither of the look-ahead operations performed during this iteration reveals a proximity to the end point XCEP/YCEP and the path segment 621 is extended along the Y axis by the distance ΔY_{02} from the point P0 to the point P2 during iteration $\Delta T998$, and along the X axis by the distance ΔX_{13} from the point P1 to the point P3, during the second half of iteration $\Delta T998$ and during the first half of the following $\Delta T999$.

At the beginning of the iteration $\Delta T999$ another pair of look-ahead operations are performed, as indicated by the vectors 625, 627 and 629. Since the designated end point XCEP/YCEP falls in Zone A of the time period $\Delta T1000$, proximity to it is revealed by the first, or 1.5 ΔX , look-ahead test, causing the counter 186' to be advanced from its S1 to its S3 count state as a result. The operations which are performed during the iteration $\Delta T999$ are shown in FIG. 47 and it may be observed there that the particular block in the first column which refers to advancing of the counter 186' to its count state S3 is not crossed out. The advancing of the End of Block Counter 186' occurs during program steps 1047 and 1048, as shown in Table IX, and the necessary program gates for permitting a program clock pulse to be applied to the counter during those program steps are represented by conditions 1 and 2 of PGA 183' of the Time Base Mode Selector 69'.

For reasons which need not be explained in this description of the second exemplary system, it is desirable to use the J_{BLF} and Q_{BLF} numbers in calculating the ΔX macromove component during the first of the linear to circular Zone A type transitional iterations, i.e., the iteration shown in FIG. 47. Thus, the J and Q numbers which are associated with the following circular path segment 623 are used to compute ΔX_{35} , the X coordinate distance from the point P3 to the point P5, and the equation used is $\pm J_{BLF} \cdot Q_{BLF} = \Delta X_{35}$, which applies for a circular path segment. Insofar as the actual path generated thereby is concerned, however, the result will be the same as that which would be produced had ΔX_{35} been produced by multiplying I' Q, using the numbers associated with the linear path segment 621. This is so because the two path segments 621 and 623 are tangential at the transition point XCEP/YCEP. FIG. 58 is shown with an exaggerated, enlarged scale. Thus, in actual fact or with a negligible error, the X coordinate distance from the point P3 to the point P5 will be the same for the circular path segment 623, produced by the computation $\pm J_{BLF} \cdot Q_{BLF}$ as it would be for the linear path segment 621, computed as I' Q.

As indicated in the heading of Table VIII, Column 400C, one of the necessary conditions for transferring J_{BLF} and Q_{BLF} from buffer to active and for performing the computation $\Delta X_{35} = \pm J_{BLF} \cdot Q_{BLF}$ is that the counter 186' be in its S5 count state. Accordingly, means are provided to advance the counter 186' to its S5 count state in response to its previous advancement to its S3 count state. These means are program gates which are part of the PGA 183' and are represented therein by conditions 6 and 7. As shown in Column X300 of

Table VII, the counter 186' is advanced to its S4 count state during step 00 provided that it is found in its S3 count state during that program step. The counter 186' is further advanced automatically during step 397 near the end of the same Column X300. As a result, prior to the commencement of the following column of the iteration, the necessary conditions $X400 \cdot S_3 \cdot (G_{02} + G_{03})$ for Column 400 C of Table VIII are met, since, with the following path segment being a circular one, a signal will also be produced on the $G_{02} + G_{03}$ output of the G BUFFER decode unit 221'. Referring to Table VIII, Column 400 C, the updating and upgrading operations are performed in the same manner as described with reference to Table VII and the standard linear iteration as shown in FIG. 46.

Turning to the computation of $\pm J_{BUF} \cdot Q_{BUF}$, the J number associated with the circular path segment 623 is transmitted from the J_{BUF} register 471' to the computer 53' during program step 435 (RD/J BUF PGA 353' condition 1). The Q number, also associated with the following path segment 623, is read from the Q BUF register 470' during program step 437 into the computer 53' (RD/Q BUF PGA 385' condition 1). The multiplication $\pm J_{BUF} \cdot Q_{BUF}$ is then carried out by the computer 53' during the steps 439-455 (X PGA 393' condition 1). The product is handled in the same manner as described previously with reference to Column 400 A of Table VIII, with the four least significant digits being stored in the XR register 113' and the higher order digits being stored in the ΔX register 109'. It will be noted that the computation performed during program steps 439-455 is in accordance with basic equation (8a), discussed in detail with reference to FIG. 6d, and that the resulting macromove number is represented in FIG. 58 as the vector ΔX_{35} (LIN). The reason for characterizing ΔX_{35} as a linear vector even though it is the result of a computation typical of a circular path segment is that, as explained previously, it is substantially the same as that which would have resulted had it been computed on the basis of I-Q using the numbers associated with the linear path segment 621. The macromove number ΔX_{35} (LIN) is executed in ten substantially equal micromoves during the second half of the time period $\Delta T999$ and the first half of the following time period $\Delta T1000$.

The next major computation to be performed during time period $\Delta T999$ is that directed to deriving the Y axis macromove ΔY_{46} which will be required during the following time period $\Delta T1000$ to execute the required movement along the Y axis from point P4 to P6. In making this computation there is first derived the inverse X component I_x of a radius vector R_1 which extends from the center C of the path segment 623 to the point P5. Once this quantity has been derived, it is used to compute, in the manner described previously with reference to FIG. 6, the Y component of the chord D_1 which extends between the points P4 and P6 and which is intersected at its midpoint by the radius vector R_1 . The number which is presently in the I BUF register 471' is the inverse X component of the "starting" radius vector which extends from the center C of the path segment 623 to the transition point XCEP/YCEP. Thus, in order to derive the inverse X component I_x of the radius vector R_1 , the starting radius vector whose inverse X component is presently available in the I BUF register 471' must be "projected ahead" until it extends to the point P5. The calculations for deriving the inverse X component I_x of the forward projected radius vector are labeled "Prorate Calculations for I" in FIG. 47 and are performed during Column 800 of the iteration, with the system in Mode 2.

In order to place the system in its second mode of operation, the End of Block Counter 186' is advanced to its S6 count state during program step 500 (see Table VII, Column X500). The program gate for causing the counter thus to be advanced is represented in PGA 183' by condition 10. With the counter 186' producing a signal on its S6 output, an enabling Logic 1 voltage level is passed through the OR gate 571 of the Time Base Mode Selector 69' to the M2' input line of the Multi-Mode Time Base Logic 67', thereby placing the system into its second operating mode. Consequently, the operations which

are performed during the ninth sub-period of the time period $\Delta T999$ are those which are listed in Column 2800 of Table X. The "Prorate Calculations for I" are performed during program steps 2870-2876. The first operation in the process, executed during program steps 2870-2873, is to compute $X_A = XCP_3 - XCEP$, representing the X coordinate distance from the designated end point XCEP/YCEP to the point P3 and shown by the similarly labeled vector in FIG. 58. During program step 2870 the Accumulator Register 53ab' is cleared and the number XCP_3 is read into the computer 53' from the XCP register 117' (CAR PGA 351' condition 12 and R/XCP PGA 421' condition 3). To subtract XCEP from XCP_3 it is subtractively read from the XCEP register 95' into the computer 53' during program step 2871 (R/XCEP 359' condition 2 and SX PGA 479' condition 5). The next step in the "Prorate Calculations for I" is to compute $X_B = X_A + \Delta X$. The number X_A is the X coordinate distance from the designated end point XCEP/YCEP to the point P5 and is also represented in FIG. 58 by a correspondingly labeled vector. To compute X_B , the number ΔX_{35} is additively read into the computer 53' from the ΔX register 109' during program step 2872 (R/ ΔX PGA 409' condition 15) and the result, $XCP_3 - XCEP + \Delta X_{35} = X_B$ is written into the Scratch Pad register 582 during the following program step 2873 (W/SP PGA 583 condition 2).

The "Prorate Calculations for I" are completed by the computation of $I_x = I_{BUF} - X_B$. From FIG. 58 it may be observed that subtraction of the X_B vector from the I_{BUF} vector results in the vector labeled " I_x ", which is the desired X coordinate distance from the point P5 to the center C of the circle, or, stated differently, the inverse X component of the desired radius vector R_1 .

To compute $I_x = I_{BUF} - X_B$, the Accumulator Register 53ab' is first cleared and then the contents of the I_{BUF} register 471' are read into the computer 53', both during program step 2874 (RD/I BUF PGA 349' condition 2 and CAR PGA 351' condition 14). With the minuend thus stored in the computer 53', the subtrahend X_B is next read subtractively into the computer from the Scratch Pad register 582 (R/SP PGA 585 condition 2 and SX PGA 479' condition 7). The result, $I_x = I_{BUF} - X_B$, appearing in the Answer Register 53a' during program step 2876, is written into the I Active Register 99' during the same program step (W/I PGA 355' condition 2). Subsequently, but still during the same program step, the Prorate Calculations for I_x are completed by clearing the Answer Register (CAR PGA 351 condition 16). The remaining operations, listed by way of horizontal arrows extending into Column 2800 of Table X, are all conditioned on the presence of an output signal on the S7 output of the decoder 187' and consequently are not performed during the iteration represented by FIG. 47.

The I_x which was computed and stored in the I register 99' as a result of the Prorate Calculations just described is utilized during the final column or sub-period of the iteration in computing ΔY_{46} . The computation, $\pm I_x \cdot Q_{BUF}$, is carried out during program steps 935-962. The specific steps performed as part of the computation are those listed in Column 900 D in Table VIII. By comparing Column 900D with Column 900B, the column which relates to a standard circular iteration, it may be seen that the steps which are performed as part of the computation $\pm I_x \cdot Q_{BUF}$ are exactly the same, with the only exceptions being that the source of the multiplier is the Q BUF register 470' (RD/Q BUF PGA 385' condition 2), and that the reading of I is in response to condition 2 of R/I PGA 400'. All of the other operations represented by the program steps in Column 900D are identical to those previously explained with reference to Column 900B and need not be described here again. Suffice it to say that the four least significant digits of the product are stored in the YR register 111' for subsequent use and that the higher order digits of the product, comprising the desired macromove number ΔY_{46} , are stored in the ΔY register 107'.

The foregoing computation of ΔY_{46} completes the operations performed during $\Delta T999$. The following iteration, occurring during time period $\Delta T1000$, is shown generally in the flow

chart of FIG. 48. As in the case of the previous iteration, the upgrading and updating operations are performed in the same manner as in the standard iteration, these steps having been explained previously with reference to Table VII. A principal feature of the iteration illustrated in FIG. 48 is the performance of prorate calculations for J and the subsequent transfer of the numbers from the Q and G Buffer Registers, Q BUF and G BUF. The latter transfer operations are conditioned on the presence of a signal on the S7 output of the counter 186'. Consequently, in response to the advancement of the counter 186' during the preceding iteration to its S6 count state, the counter is advanced to its S7 count state during the first program step 000 of the FIG. 48 iteration (PGA 183' condition 11).

The reasons behind the prorate calculations for J are similar to those considered previously with reference to the prorate calculations for I. Specifically, it will be recalled that the desired X component of motion was last computed as the macromove ΔX_{35} and that this macromove was, in effect, the X coordinate distance from the point P3 to the point P5 on the linear path segment 621. The object of the prorate calculations for J is to permit the computation of the X component of a chord D_2 extending from the point P5 to the point P7 and subtending the initial portion of the arc 623. As will be recalled from the basic analysis of circular interpolation with reference to FIG. 6, in order to compute the macromove component ΔX_{57} , it will be necessary first to derive the inverse Y component of the radius vector R_2 which extends from the center C of the arc to the point P₆, which is midway between points P₅ and P₇ on the arc. Thus, in effect, the starting radius vector extending from the center C to the transition point XCEP/YCEP must be "projected forward" by more than one-half macromove and the Y component J_- of this forward projected radius vector R2 must be computed.

The prorate calculations involve three steps again, as in the case of the prorate calculation for I, with the first step comprising the calculation $YCP_4 - YCEP$. The result, Y_A , is seen in FIG. 58 to be the Y coordinate distance from the desired transition point XCEP/YCEP to the point P₄. The necessary operations are listed in Column 2300 to Table X and include clearing of the Accumulator Register 53ab' during program step 2370 and the transmission of the number YCP₄ from the YCP register 115' to the computer 53' during the same program step *CAR PGA 351' condition 11 and R/YCP PGA 425' condition 3). The end point coordinate number YCEP is then subtracted from the number YCP₄ during program step 2371 by subtractively reading it into the computer 53' during program step 2371 (R/YCEP PGA 487' condition 2 and SX PGA 479' condition 4).

The second step of the prorate calculations is to add to the remainder Y_A stored in the Answer Register 53a' the macromove number ΔT_{46} , which, it will be recalled, was just computed as a result of the computation performed during Column X900 of the preceding iteration during time period ΔT_{999} . The significance of this step will be understood when it is observed from FIG. 58 that the macromove number ΔY_{46} is positive, whereas the previously computed remainder Y_A is negative. Thus, when Y_A is added to ΔY_{46} the effect is to subtract Y_A from ΔY_{46} so as to produce a remainder shown in FIG. 58 as the vector Y_B . This remainder represents the Y coordinate distance from the point XCEP/YCEP to the point P₆.

The operations involved are performed during program steps 2372-2373. During these program steps the macromove number ΔY_{46} is transmitted from the ΔY register 107' to the computer 53' (R/ ΔY PGA 415' condition 15) and the result $Y_B = \Delta Y_{46} + Y_A$, appearing in the Answer Register 53a' during the following program step 2373, is transmitted therefrom to the Scratch Pad Register 582 during that program step for temporary storage (W/SP PGA 583 condition 1). It will now be readily seen from FIG. 58 that by subtracting the positive number Y_B from the negative inverse radius vector component number J_{BUF} , the desired inverse component number

J_- of the forward projected radius vector R2, representing the Y coordinate distance from the point P6 to the center C will be obtained. This is the third of the prorate calculations for J and is performed during program steps 2374-2375. After first clearing the Accumulator Register 53ab' (CAR PGA 351' condition 13) the J_{BUF} component number is read from the J BUF register 47J' and is additively applied to the computer 53' (RD/J BUF PGA 353' condition 2). To subtract the number Y_B , which is presently in the Scratch Pad Register 582, from the number Y_{BUF} which is in the computer 53', the number Y_B is subtractively read from the Scratch Pad Register 582 into the computer 53' during program step 2375 (R/SP PGA 585 condition 1 and SX PGA 479' condition 6). The result, $J_- = J_{BUF} - Y_B$, stored in the Answer Register 53a' during program step 2376, is transferred during that program step to the J active register 101' and the computation is closed by clearing the Answer Register 53a' (W/J PGA 357' condition 2 and CAR PGA 351' condition 15).

The J_- number, just calculated, remains in the J active register 101' until the following sub-period for Column X400 of the ΔT_{1000} iteration, at which time it is used as the multiplicand of the computation $J_- \cdot Q$. Before this computation is performed, however, the Q and G numbers associated with following circular path segment are transferred from buffer to active, these operations being performed during program steps 2377-2379, listed in Column 2300 of Table X. From Table X it will be observed that these transfer operations are conditioned on the counter 186' being in its S7 count state, a condition which was brought about by stepping the counter 186' into that count state at the beginning of the iteration ΔT_{1000} . The new Q number is read into the computer 53' from the Q BUF register 47Q' during program step 2377 (R/Q BUF PGA 385' condition 3) and is re-transmitted from the Answer Register 53a' during the following program step 2378 into the Q active register 103' (W/Q PGA 387' condition 1). During the same program step the G number associated with the following circular path segment is transmitted from the G BUF Register 47G' to the computer 53' (RD/G BUF PGA 251G' condition 1) and is again transmitted from the Answer Register 53a' during the program step 2379 into the G active register 105' (W/G PGA 409' condition 1).

Following the transfer of Q and G from buffer to active, the counter 186' is once again advanced, this time from its count state S7 to its count state S8 (PGA 183' condition 13). As a result, the M3' input line of the group of AND gates 179' forming part of the Multi-Mode Time Base Logic 67' is enabled, thereby placing the system in its third operating mode. This does not have any effect on the operation performed by the system until the next to the last sub-period of the iteration because none of the operations prior to that sub-period, or column, in FIG. 48, are programmed to take place only when the system is in its third operating mode. During the next to the last column of the iteration, however, labeled "3800," the designated end point coordinates XEP and YEP associated with the following circular path segment 623 are transferred as a direct result of the system having been placed in its third operating mode during that iteration.

Returning now to the computations directed to deriving the X axis macromove ΔX_{57} , these computations follow the equation $\Delta X_{57} = \pm J_- \cdot Q$, where Q is the new macromove quotient which was just transferred into the active Q register 103'. The broad outlines of the computation of ΔX_{57} appear in Column X400 of FIG. 48. Its details are listed in Column 400B of Table VIII.

It should be noted that the computation $\pm J_- \cdot Q$ is simply the standard equation for circular interpolation which was previously referred to with reference to FIG. 6 as the equation (18). Thus, the number J_- is simply the required J component of a radius vector and the only way in which it differs from the J number shown, for example, in Column X400 of FIG. 49, is the manner in which it was derived. Thus, the computations which are performed during program steps 435-464 of the ΔT_{1000} iteration shown in FIG. 48 follow exactly the same

sequence as those which are performed during Column X400 of a standard circular iteration such as the one shown in FIG. 49 and described with reference to that Figure. Consequently, this series of operations need not be described again, since their sequence and the control signals which bring them about are precisely the same and are represented by the same program gate conditions as those appearing in Column X400 of FIG. 49. Thus, the first full circular macromove, ΔX_{57} , becomes available during Column X400 of the iteration $\Delta T1000$ and is subsequently executed during the second half of that iteration and the first half of the following iteration $\Delta T1001$ through the process of updating, as explained previously with reference to a standard circular iteration.

As mentioned earlier, the designated end point coordinate numbers XEP and YEP are transferred during Column 3800 of the FIG. 48 iteration $\Delta T1000$ during program steps 3835-3837. The detailed steps appear in Table XI and involve the transmission of the XEP number during program step 3835 from the XEP register 47X' to the computer 53' (RD/XEP BUF PGA 359' condition 2) and the subsequent re-transmission of the XEP number from the Answer Register 53a' during program step 3836 to the XCEP active register 95' (W/XCEP PGA 375' condition 2). The Accumulator Register 53ab' is then cleared (CAR PGA 351' condition 26) and the YEP end point coordinate number is transmitted to the computer 53' from the YEP BUF Register 47Y' (RD/YEP BUF PGA 377' condition 2). The number appears in the Answer Register 53ab' during the following program step 3837 and is transmitted during that program step to the YCEP active register 97' (W/YCEP PGA 383' condition 2). During the same program step the entire Accumulator Register 53ab' is cleared (CAR PGA 351' condition 28).

Referring again to FIG. 58, the next macromove to be computed is ΔY_{68} for the Y coordinate distance from the point P6 to the point P8. Because the points P6 and P8 happen to have approximately the same Y coordinates in the vector diagram of FIG. 58, the macromove ΔY_{68} cannot be shown there. Nevertheless, whatever its size, it must be determined or computed before the commencement of the iteration $\Delta T1001$. In fact, the computation for deriving ΔY_{68} takes place during Column X900 of the iteration $\Delta T1000$ appearing in FIG. 48, and it is seen to conform to the previously described equation (19) appearing in FIG. 6 and also in FIG. 49. Thus, the multiplicand I used in the computation $\pm I \cdot Q$ during Column X900 is that which was derived during Column X400 by subtracting ΔX_{57} from the previous value of I, this previous I number having been stored in the I register 99' and representing the X coordinate distance from P5 to the center C of the circular path segment 623. Thus, the upgraded I number which is used in the computation of $\pm I \cdot Q$ during Column X900 in FIG. 48 is the X coordinate distance from the point P7 to the center C, i.e., the inverse I component of the radius vector extending from the center C of the path segment 623 to the point P7 thereof. The computation of $\pm I \cdot Q$ is then performed during program steps 935-962 in exactly the same manner as it is during a standard circular iteration such as that shown and described with reference to FIG. 49. Following that computation, the product is split into two parts with its four least significant digits being stored in the YR register 111' and with its higher order digits being stored in the ΔY register 107' as the macromove number ΔY_{68} .

Following the computation of the ΔY_{68} number, that number is utilized immediately to upgrade the J number being held in the J active register 101' by subtracting ΔY_{68} therefrom. The upgrading of J is performed during program steps 963-964 in the same manner and in response to the same signals as described previously with reference to the corresponding operation in FIG. 49. The resulting upgraded J number, representing the Y coordinate distance from the point P8 to the center C of the circular path segment 623, will be used during the following iteration $\Delta T1001$, the standard circular iteration shown in FIG. 49, to compute the X axis macromove ΔX_{79} .

Having transferred all of the numbers associated with the path segment 623 from buffer to active, the tape reader 41' will now be started in order to cause the data block comprising the numbers associated with the path segment which follows the circular path segment 623 to be transferred from the tape to the buffer registers. The starting of the tape reader 41' is conditioned on the counter 186' being in its S9 count state and also on the Time Base and Digital Sweep 63' producing the number 991. Consequently, as a first part of initiating the starting of the tape reader 41' the counter 186' is advanced from its S8 count state to its S9 count state during program step 3879 (PGA 183' condition 16). Subsequently, during Column X900 and in particular during program step 991 an enabling signal is applied to the tape reader start control flip-flop 257' by the program gate 269' whose inputs are connected to the 900, 90, and 1 outputs of the Time Base and Digital Sweep 63' (FIG. 40b).

The iteration to be performed during program step $\Delta T1001$ is a standard circular iteration as shown in FIG. 49. Consequently, the system must be returned to its Mode 1 during the final steps of the iteration $\Delta T1000$. Accordingly, the counter 186' is rapidly stepped from its S9 count state through its count states S10-S16 to its S1 count state during program steps 992-999. These steps are shown in detail in Table VII, Column X900, and the appropriate program gates are represented in PGA 183' by the conditions 17-24. As a result, the system is placed in its Mode 1 during program step 999 of iteration $\Delta T1000$. Consequently, the following iteration $\Delta T1001$ is a standard circular iteration according to FIG. 49 and this type of iteration is repeated many times in generating the circular path segment 623. The flow chart appearing in FIG. 45 is based on the assumption that the FIG. 49-type standard circular iterations are repeated until time period $\Delta T2498$ which, in the example illustrated in FIG. 45, is followed by a series of transitional iterations leading into a linear path segment. This circular to linear transition will be explained subsequently with reference to FIG. 60, but first there will be described another type of linear to circular iteration, one in which the desired transition point is in Zone B, as shown in FIG. 59.

2. Assuming the Desired Transition is in Zone B

The transition from a linear path segment to a circular path segment under circumstances in which the desired transition occurs in Zone B is illustrated in FIG. 59. As might be expected, the operations which are performed in the latter case are for the most part analogous to those which are performed under the circumstances illustrated in FIG. 58. To stress the similarities, the vector diagrams comprising FIGS. 58 and 59 are drawn to the same scale and are identical in many respects. Thus, in both FIGS. 58 and 59 the initial iterations performed during time periods $\Delta T1000 - \Delta T997$ in generating the linear path segment 621 are foreshortened to conserve space. The first iteration shown fully is, in both cases, that performed during time period $\Delta T998$. In executing the final portion of the linear path segment, the points P0, P1, P2, P3 and P4 are traversed during iterations $\Delta T998$ and $\Delta T999$ and an additional point P5 is traversed before the designated end point XCEP/YCEP is reached during the second half of the iteration $\Delta T1000$.

The iteration performed during the time period $\Delta T998$ is again the standard linear iteration explained in detail previously with reference to FIG. 46 and need not be discussed further. Neither of the look-ahead operations, projected forward from the point P0 during that iteration, yields a positive result, since the end point XCEP/YCEP is more than 2 macromoves away from the point P0. During the following time period $\Delta T999$, proximity to the end point XCEP/YCEP is detected during the second of the look-ahead operations. This requires a different response by the system than that indicated in the circumstances depicted in FIG. 58 where a proximity to the end point XCEP/YCEP was detected as a result of the first look-ahead operation performed during the time period $\Delta T999$. The iteration performed during the time period

$\Delta T999$ under the circumstances depicted in FIG. 59 is shown in FIG. 50. It will be noted that the third block in the first column of the FIG. 50 iteration is crossed out, indicating that the counter 186' was not advanced to its S3 count state, since the remaining distance to the designated end point XCEP/YCEP was not detected to be less than 1.5 macromoves. It will also be noted, however, that the fourth block in the first column of the iteration is not crossed out, thus indicating that the End of Block Counter 186' was advanced to its S2 count state as a result of the second, 2 macromove, look-ahead test which revealed that the remaining distance from the point P2 to the designated end point XCEP/YCEP along the X axis was between 1.5 and 2 macromoves.

The manner in which the look-ahead operations are performed have been explained previously. It need only be noted that the counter 186' is advanced to its S2 count state during program step 1053, as shown in Table IX, in response to an SST signal produced during that program step by the Fast Axis Projected Overrun Detector 71' (FIG. 40e). Because the counter 186' was not advanced to its S3 count at the beginning of time period $\Delta T999$, the computation of ΔX_{36} during column X400 of the iteration shown in FIG. 50 is performed exactly as it would be during a standard linear iteration, such as that shown in FIG. 46. The first significant effect of advancing the counter 186' to its S2 count state at the beginning of time period $\Delta T999$ is to cause the numbers which are then stored in the I BUF and Q BUF registers 471 and 470', and which are associated with the following circular path segment 623, to be used during column X900 of the iteration to compute the new macromove number ΔY_{46} . As explained previously, the reason why it is desirable to utilize these numbers which are associated with the following path segment need not be understood by the reader. It is sufficient to understand that, as in the case of the corresponding macromove computation discussed with reference to FIG. 58, $\pm I_{BUF} \cdot Q_{BUF}$ yields essentially the same result for ΔY_{46} as if the linear path computation $J \cdot Q$ had been used, because the linear path segment 621 is tangential to the circular path segment 623. Thus, in effect, ΔY_{46} , computed as $\pm I_{BUF} \cdot Q_{BUF}$, is the Y component of that portion of the linear path segment 621 which extends from the point P4 to the point P6. Thus is seen from FIG. 59 to be appropriate, since the transition point XCEP/YCEP does not occur until after the point P5 has been passed.

The operations necessary to compute $\pm T_{BUF} \cdot Q_{BUF}$ are listed in Column 900C of Table VIII. As may be seen from the heading of that column, counter 186' must be in its S5 count state in order that the listed operations be performed by the system. Accordingly, during Column X800 of the FIG. 50 iteration the counter is advanced first from its S2 count state to its S4 count state during program steps 800 and 801 in response to the counter having been previously advanced to its S2 count state as a result of the second overrun test performed at the beginning of that iteration. At the end of the same column, X800, during program step 897, counter 186' is advanced to its S5 count state. The program gates for causing the counter 186' to be thus advanced are represented in the Program Gate Array 183' by the conditions 4, 5 and 8. Additionally, the steps are listed in Table VII, Column X800.

The multiplicand I_{BUF} and the multiplier Q_{BUF} are read from the buffer registers 471' and 470' into the computer 53' during program steps 935 and 937 respectively (RD/I BUF PGA 349' condition 1 and RD/Q BUF PGA 385' condition 2). With respect to RD/I BUF PGA 349' condition 1, it will be noted that it calls for the presence of a signal on the $G_{02} + G_{03}$ output of the G buffer decode unit 221' and that such a signal is indeed present at that output, since the following path segment is to be a circular one. Furthermore, with respect to the RD/Q BUF PGA 385' condition 1, it will be observed that it requires a PX2 signal at the output of the AND gate 575 of the Time Base Mode Selector 69'. That signal is produced under the existing conditions by the presence of the $G_{02} + G_{03}$ signal at one

of the inputs of the AND gate 575 and by the concurrent presence at the other input of the AND gate 575 of the signal produced at the S5 output of the decoder 187', the latter signal being passed to that input of the AND gate 575 through the OR gate 573.

The multiplication of $\pm I_{BUF} \cdot Q_{BUF}$ is performed in the same manner as was explained with reference to the Column 900A of Table VIII except that, of course, the sources of the multiplier and of the multiplicand are different. As in the case of Column 900A, the multiplication process is initiated by a multiply control signal from the X PGA 393' (condition 2). The product of the multiplication is split into two parts, with its four least significant digits being stored in the YR register 111' and with its higher order digits being stored in the ΔY register 107' as the macromove number ΔY_{46} . The macromove number ΔY_{46} thus computed is executed during the following iteration $\Delta T1000$ in 10 micromoves.

The operations which are performed during time period $\Delta T1000$ are shown in FIG. 51, which illustrates the second of the linear to circular transitional iterations performed when the desired transition point is in Zone B. During the first half of the time period $\Delta T1000$ the execution of the macromove number ΔX_{36} , which was computed during the previous iteration $\Delta T999$, will be completed and the new macromove number ΔX_{37} will have to be computed during the Column X400 of the current iteration in order to provide the necessary data for executing movement along the X axis past the point P5. Of the total movement which is to be executed between the points P5 and P7 during the second half of time period $\Delta T1000$ and the first half of the following time period $\Delta T1001$, only the initial portion, that which extends from the point P5 to the point XCEP/YCEP is to be linear. The remaining and larger portion of the path segment, extending from the point XCEP/YCEP to the point P7 is seen to be part of the circular path segment 623. Consequently, the macromove ΔX_{37} is better computed through the process of circular interpolation as the X component of the chord D1 which extends from the point P5 to the point P7. In order to perform this computation it will first be necessary to derive the Y component J_y of the radius vector R1 extending from the center C of the arc 623 to the point P6.

As may be seen from FIG. 59, the radius vector R1 intersects the chord D1 at right angles and is the appropriate radius vector for computing the X component of that chord. As in the case of the transition explained with reference to FIG. 58, however, neither component of the radius vector R1 is initially given. It must, instead, be derived from the initially available Y component J_{BUF} of the starting radius vector (not shown in FIG. 59) which extends from the center point C to the transition point XCEP/YCEP. The calculations necessary to derive J_y from J_{BUF} are listed as the "Prorate Calculations for J" in Column 2300 of the iteration shown in FIG. 51. These operations are analogous to the prorate calculations for I, which were explained previously with reference to FIGS. 47 and 58. An understanding of the efficacy of the prorate calculations for J will be readily gained by comparing them with the corresponding vectors in FIG. 59. Very briefly, $Y_A = YCP_A - YC_{EP}$ is first computed and the negative number Y_A is added to the positive macromove number ΔY_{46} , thereby yielding the number Y_B representing the Y coordinate distance from the point XCEP/YCEP to the point P6, which, it will be recalled, is on the linear path segment 621. Finally, the Y_B number which is positive, is subtracted from the originally given number J_{BUF} , which is negative, resulting in the larger negative number J_y , representing the Y coordinate distance from the point P6 to the center point C of the path segment 623.

The particular steps which form part of the prorate calculations for J are listed in Table X in Column 2300. To place the system in its second operating mode the counter 186' is advanced at the beginning of the FIG. 51 iteration from its S5 count state to its S6 count state. From Table VII it is seen that this occurs during the first program step of the iteration in response to the counter having been advanced to its S5 count

state during the previous iteration. The appropriate program gate is listed as condition 9 in the PGA 183'. To compute $Y_A = YCP_4 - YCEP$ the Accumulator Register 53ab' is cleared and the number stored in the YCP register 115' is entered in the computer 53' during program step 2370 (CAR PGA 351' condition 11 and R/YCP PGA 425' condition 3). YCEP is subtracted from YCP₄ during program step 2371 (R/YCEP PGA 487' condition 2 and SX PGA 479' condition 4). The next computation, $Y_B = Y_A + \Delta Y_{46}$ is performed during the next program step 2372 by reading the ΔY_{46} macromove number from the ΔY register 107' (R/ ΔY PGA 415' condition 15) and the result is written into the Scratch Pad register 582 during program step 2373 from the Answer Register 53a' (W/SP PGA 583 condition 1).

The third computation, $J_- = J_{BUF} - Y_B$, is initiated by clearing the Accumulator Register 53ab' and by reading the numbers contained in the J BUF register 47J' into the computer 53' during program step 2374 (CAR PGA 351' condition 13 and RD/J BUF PGA 353' condition 2). The Y_B number being held in the Scratch Pad register 582 is subtracted during program step 2375 (R/SP PGA 585 condition 1 and SX PGA 479' condition 6) from J_{BUF} which is in the computer 53'. The result J_- appears in the Answer Register 53a' during program step 2376 and is written into the J Active Register 101' (W/J PGA 357' condition 2). The entire Accumulator Register 53ab' is then cleared (CAR PGA 351' condition 15).

The J_- stored in the J Active Register 101' as a result of the last operation is utilized during the following sub-period, or column, X400 of the FIG. 51 iteration to compute $\Delta X_{57} = \pm J_- \cdot Q_{BUF}$. The specific steps involved are listed in Column 400D of Table VIII. From the heading of that column it may be seen that each of the necessary conditions, X400, S6, and ($G_{02} + G_{03}$) are present. First, the multiplicand J_- is read from the J Active Register 101' into the computer 53' (R/J PGA 391' condition 2) and then the multiplier Q_{BUF} is read into the computer from the Q BUF register 47Q' (RD/Q BUF PGA 385' condition 1). With reference to the latter condition it will be noted that a PX2 signal is still present at the output of the AND gate 575 of the Time Base Mode Selector 69' due to the presence of a signal on the S6 output of the decoder 187', as well as the presence of a signal on the $G_{02} + G_{03}$ output of the G buffer decode 221'. The multiplication $\pm J_- \cdot Q_{BUF}$ is carried out subsequently during program steps 439-455 in response to a multiply control signal (X PGA 393' condition 1). Again, the four lowest order digits of the product are stored in the XR register 113' and the remaining digits of the product are stored in the ΔX register 109' as the ΔX_{57} macromove number.

The first half of the just calculated macromove ΔX_{57} is executed during the second half of the FIG. 51 iteration during time period $\Delta T1000$ in 5 micromoves. During the same time, movement along the Y axis is in accordance with the macromove number ΔY_{46} whose execution is completed at the end of the FIG. 51 iteration during time period $\Delta T1000$. Thus, it becomes necessary before the beginning of the following time period $\Delta T1001$ to compute a new Y axis macromove number. As seen in FIG. 59, movement during time period $\Delta T1001$ is from the point P6 to the point P8 along the circular path segment 623. Since the desired movement is along a circular path segment during the entire time period $\Delta T1001$, the macromove number ΔY_{46} is computed through the method of circular interpolation in accordance with equation (19). Thus, referring further to FIG. 59, the macromove number ΔY_{46} is derived by computing the X component of the chord D2 connecting the points P6 and P8 on the arc 623. The appropriate radius vector for this computation is R_2 which extends from the center C of the path 623 to the point P7 lying midway between the points P6 and P8 along the path segment 623. Therefore, in accordance with equation (19), the X component I_- of the radius vector R2 must first be derived from the initially available X component I_{BUF} of the starting radius vector (not shown) extending from the center point C to the transition point XCEP/YCEP. The necessary prorated calculations

for I are listed in Column 2800 of Table X. The geometrical basis for these calculations may be readily seen from FIG. 59, where each of the quantities X_A , X_B and I_- are represented by vectors. Furthermore, the operations which are performed to carry out the prorated calculations for I are exactly the same as those described in connection with the FIG. 47 iteration with reference to FIG. 58 and Table X. It will thus be sufficient to note that, as a result of the prorated calculations for I, the desired I_- number representing the X component of the radius vector R_2 is stored in the I active register 99' prior to the commencement of the final Column X900 of the FIG. 51 iteration.

Following the computation of I_- and during Column 2800 of the FIG. 51 iteration, the Q and the G numbers associated with the following circular path segment 623 are transferred from buffer to active. The specific operations involved are those indicated in Column 2800 of Table X by the first six of the horizontally extending arrows extending into that column. All of the steps associated with the transferring of Q and G from buffer to active are conditioned on the presence of a signal on the S7 output of the decoder 187'. Consequently, during Column X500 the counter 186' is advanced from its S6 count state to its S7 count state in response to a signal produced by a program gate represented in the PGA 183' by condition 12. The transfer of G and Q is effected in the same manner as explained previously with reference to Column 2300 of Table X in response to control signals represented by RD/Q BUF PGA 385' condition 4, W/Q PGA 387' condition 2, RD/G BUF PGA 251' condition 2 and W/G PGA 409' condition 2.

Following the transfer of Q and G from buffer to active the counter 186' is stepped from its count state S7 to its count state S8 during program step 2882 (PGA 183' condition 14). As a result, the M3' input line to the third group of AND gates 179' becomes enabled, thereby placing the system in its third operating mode in which the output of the hundreds BCD to D converter 171' of the Time Base and Digital Sweep 63' is gated through and appears at the outputs of the AND gates 179'. These outputs are not utilized until the fourth column of the following iteration, shown in FIG. 52 and performed during time period $\Delta T1001$.

The new Y axis macromove number ΔY_{46} required for the following time period $\Delta T1001$ is computed during the last, X900 column of the FIG. 51 iteration. The multiplicand is the I_- number which was just computed and which is stored in the I active register 99', and the multiplier is the Q number associated with the following path segment 623 and which was just transferred to the Q register 103' during the previous Column 2800.

The specific steps for computing $\pm I_- \cdot Q$ are those which are characteristic of circular interpolation and are listed in Column 900B of Table VIII. The program steps during which the multiplicand is read from the I Active Register 99' and the multiplier Q is read from the Q Active Register 103', as well as the time for performing the computation $\pm I_- \cdot Q$, are apparent from Table VIII, and the conditions representing the program gates for generating the appropriate READ, WRITE and MULTIPLY control signals are listed in the appropriate Program Gate Arrays. They are, R/I PGA 400' condition 1, R/Q PGA 389' condition 4, and X PGA 393' condition 2. It will be noted that the transfer of the numbers in the I and Q Active Registers 99' and 103' prior to their multiplication is conditioned on the presence of a signal on the CIRC output of the Path Type Indicator 106' rather than on an output from the Time Base Mode Selector 69'. Consequently, the advancement of the counter 186' to its S8 count state and the possible removal of a signal from the $G_{02} + G_{03}$ output of the G Buffer Decode Unit 221' has no effect on the transfer of the numbers from the I and Q registers. Stated differently, the reading of the numbers from the I and Q registers 99' and 103' is still controlled by the G number which is associated with the circular path segment 623, but that G number is now stored in the G Active Register 105' and is manifested by a signal on the

CIRC output of the Path Type Indicator 106' associated with the G Active Register.

The last operation performed during the FIG. 51 iteration is that of upgrading J_{-} , which is still stored in the J Active Register 101', so as to produce a number representing the inverse Y component of the radius vector (not shown) which extends from the center C of the path segment 623 to the point P8 thereon. This new J number will then be used during the next iteration, shown in FIG. 52 and performed during time period $\Delta T1001$, to compute, as the next macromove ΔX_{78} , the X component of the chord (not shown) which extends from the point P7 to the point P9 on the arc. To upgrade the J_{-} the newly computed macromove number ΔY_{88} is subtracted from it. The operations involved in upgrading J_{-} are listed as occurring during program steps 962-964 of Column 900B in Table VIII. The manner in which these steps are performed and the control signals required will not be repeated here, since they were explained previously with reference to the standard circular iteration shown in FIG. 49. From that description it will be recalled that the upgraded J number is stored in the J Active Register 101' during program step 964, following which the Accumulator Register 53ab' is cleared. This completes the operations which comprise the second of the Zone B type linear to circular transitional iterations, as shown in FIG. 51.

The third of the Zone B type linear to circular transitional iterations, performed during time period $\Delta T1001$ and illustrated generally in FIG. 52, is almost identical to a standard circular iteration of the type shown in FIG. 49. The FIG. 52 iteration is distinguished by the operations listed in its fourth column and in its last column, during which further numbers are transferred from buffer to active and during which also the transfer of data from the tape to the buffer register is initiated. Specifically, the changeover of the system to its third operating mode is first manifested during the fourth column of the FIG. 52 iteration, labeled appropriately "3300." During this column the XEP and YEP numbers, representing the designated end point of the circular path segment 623, are transferred from the buffer registers 47X' and 47Y' to the Active Registers 95' and 97' respectively. The operations in transferring the end point coordinate numbers from buffer to active are listed in Column 3300 of Table XI and are seen to correspond to the previously described transfer of those numbers from buffer to active with reference to the iteration shown in FIG. 48, and shown there to occur during Column 3800. Thus, the XCP number is transferred from buffer to active during program steps 3335 and 3336 (RD/XEP BUF PGA 359' condition 1, and W/XCEP PGA 375' condition 1).

After clearing the Accumulator Register 53ab' (CAR PGA 351' condition 25) the YEP number is transferred from buffer to active during program steps 3336 and 3337 (RD/YEP BUF PGA 377' condition 1, and W/YCEP PGA 383' condition 1). Then the Accumulator Register is again cleared (CAR PGA 351' condition 27).

For reasons which will appear subsequently, counter 186' is advanced from its S8 count state to its S9 count state during program step 3379, also shown in Table XI (PGA 183' condition 15). During the following Column X400, ΔX_{78} is computed by multiplying $\pm J_{-}$ Q in exactly the same manner as explained previously with reference to Column 400B of Table VIII and Column X400 of FIG. 49. Furthermore, following the computation of the new macromove number ΔX_{78} , the number in the I Active Register 99' is upgraded exactly as described with reference to Table VIII, Column 400B, and FIG. 49, Column X400, resulting in the storage of the upgraded I number, representing the inverse X component of the radius vector (not shown) extending from the center C of the circle to the point P9 on the arc 623. This upgraded I number is then used during Column X900 of the iteration to compute the Y axis macromove number ΔY_{8-10} , followed by the upgrading of the J number then in the J Active Register 101' by subtracting the new ΔY_{8-10} number therefrom. The tape reader 41' is started next in order to initiate the transfer of data associated with the path segment following the circular

path segment 623 into the Buffer Registers. The manner in which this occurs and the control signals which bring it about are the same as those described previously with reference to the second transitional Zone A type iteration shown in FIG. 48. As described there, the tape reader is started in response to the previous advancement of the counter 186' to its S9 count state during program step 991. Following this, again in the same manner as described with reference to FIG. 48 previously, the counter 186' is stepped through its count states S10-S16 to its count state S1 during program steps 992-999, thereby placing the system in its first operating mode in preparation for performance of a standard circular iteration during the next time period $\Delta T1002$. From this point on until the approach of the end of the path segment 623 the standard circular iteration of the type shown in FIG. 49 is performed repetitively during successive time periods, beginning with the time period $\Delta T1002$.

e. Going From a Circular Path Segment To a Linear Path Segment

1. Assuming the Desired Transition Is In Zone B

Let it be assumed that the circular path segment 623 shown to have been begun in FIG. 58 during the time period $\Delta T1001$ was continued for another approximately 1500 time periods ΔT and that it was to be followed by another linear path segment. This is the situation illustrated in FIG. 60, and depicted by the lowermost branch of FIG. 45. The circular path segment 623 in FIG. 60 has been rotated clockwise relative to its orientation in FIG. 58 for convenience in viewing the drawing.

As in the case of FIGS. 58 and 59, the initial portion of the first path segment illustrated is abbreviated to conserve space. Only the final portion of the path segment 623 is shown fully and is seen to extend from the point P0 through the points P1-P5 to the designated transition point XCEP/YCEP where the following linear path segment 635 begins. The linear path segment 635 has an X component shown in FIG. 60 as the vector I_{BLF} and a Y component shown as the vector J_{BLF} . These vectors are represented as numbers in the I and J Buffer Registers 47I' and 47J' along with the number Q_{BLF} stored in the Buffer Register 47Q' and representing the desired velocity along the path segment 635. Many standard circular iterations of the type shown in FIG. 49 will have occurred during the time periods $\Delta T1001 - \Delta T2497$ before the particular iteration performed during time period $\Delta T2498$ is carried out. During the last of these iterations, performed during time period $\Delta T2497$, the Y axis macromove number ΔY_{02} will have been computed.

The iteration performed during time period $\Delta T2498$ is the last standard circular iteration of the type shown in FIG. 49 during which proximity to the designated transition point XCEP/YCEP had not been detected. There follow during the time periods $\Delta T2499 - \Delta T2501$ three transitional iterations, shown in FIGS. 55, 56, and 57, during which a conversion is made from the circular path segment 623 to the linear path segment 635. Following the last of the transitional iterations a standard linear iteration of the type shown in FIG. 46 is performed during time period $\Delta T2502$, and this type of iteration is repeated during each of the subsequent time periods $\Delta T2503$ on until the end of the linear path segment 635 is approached.

It will be noted that the look-ahead operations illustrated in FIG. 60 are projected along the Y axis because, as drawn in FIG. 60, the path segment 623 has a larger macromove component along the Y axis than it has along the X axis. The operations that take place during the time period $\Delta T2498$ in extending the path segment 623 to the point P2 are those which have been explained previously with reference to FIG. 49. Proximity within $2\Delta Y$ macromoves of the designated end point XCEP/YCEP is detected by the second look-ahead test which is performed during the time period $\Delta T2499$ as illustrated in FIG. 55. As a result, the End of Block Counter 186' is advanced to its S2 count state. This has no effect upon the computation of ΔX_{33} during Column X400 of the FIG. 55 iteration, which is performed in accordance with the circular

interpolation equation $\Delta X = \pm J \cdot Q$. Thus, the operations performed during Column X400 shown in FIG. 55 are exactly the same as those performed during a standard circular iteration as shown in FIG. 49.

During Column X800 of the FIG. 55 iteration counter 186' is further advanced, first to its S4 count state during program steps 800 and 801, and then to its S5 count state during program step 897, as explained previously with reference to the corresponding column of FIG. 50. Even though the counter 186' has been advanced to its S5 count state, the next macromove computation performed during Column X900 of FIG. 55 remains the same as it would be in the case of a standard circular iteration as shown in FIG. 49. Thus, except for the advancement of the End of Block Counter from its count state S1 to its count state S5, all of the other operations of the system performed during time period $\Delta T2499$, as illustrated in FIG. 55, are exactly the same as they would be during the standard circular iteration as shown in FIG. 49. At the beginning of the second of the circular to linear transitional iterations, as shown in FIG. 56, the counter 186' is advanced one step further to its S6 count state (PGA 183' condition 9). The first macromove computation performed during Column X400 of the FIG. 56 iteration is still carried out in accordance with the circular interpolation equation $\Delta X = \pm J \cdot Q$. The result of this computation is ΔX_{57} , which would be the desired X axis excursion if the circular path segment 623 were to be continued to the mid-point of the following time period $\Delta T2501$. From FIG. 60 it may be seen that this is not the case and that, actually, the circular path segment is to be terminated during the second half of the previous iteration $\Delta T2500$. However, the error introduced by using a circular interpolation computation is much less than that which would appear to be the case from FIG. 60. In actual practice the resulting error will be negligible because the contiguous path segments 623 and 635 are mutually tangential. It appears significant only because of the large scale of FIG. 60.

In examining FIG. 56 there will be noted the presence of a crossed-out block labeled "Carry Out Prorate Calculations for J: Do Not Use Result." This simply means that the system is permitted to carry out the prorate calculations for J which it is conditioned to do in response to the counter 186' being in its S6 count state, as explained previously with reference to FIG. 51, but that the result is not used. The reason that the result is not used is that there is no need for "advancing" a radius vector as there was in the case where the following path segment represented by that radius vector was circular. Stated differently, where the path segment into which a transition is made is a linear one, as shown in FIG. 60, the I and J numbers associated with that path segment can be used in calculating macromoves without any changes being made to them. The same also applies to the prorate computation represented by the crossed-out block in Column 2800.

The ΔY_{88} macromove number representing the Y axis macromove to be executed during the following time period $\Delta T2501$ is computed during the last Column X900 of the FIG. 56 iteration during the time period $\Delta T2500$. This computation is in accordance with the linear interpolation equation $\Delta Y = J \cdot Q$ and both the multiplier and the multiplicand involved are those associated with the new path segment 635. Therefore, it becomes necessary prior to the computation of ΔY_{88} to transfer the data associated with the following path segment 635 from buffer to active. This is performed during the preceding column labeled "2800" in FIG. 56. As implied by the label "2800", the transfer of data from buffer to active is performed with the system in its second operating mode and this in turn is brought about by advancing the End of Block Counter to its S7 count state during program step 501 (see Table VII, Column X500, and PGA 183' condition 12). This places the system in Mode 2. In this mode, the Q and G numbers are transferred from buffer to active during program steps 2877-2879 in the manner and in response to the control signals described previously with reference to FIG. 51 (Section J. 4. d.(2)). Furthermore, the new I and J numbers are

similarly transferred during program steps 2879-2881 (RD/I BUF PGA 349' condition 4, W/I PGA 355' condition 4, RD/J BUF PGA 353' condition 4 and W/J PGA 357' condition 4). It will be understood, of course, that in the process of transferring a number through the Accumulator Register 53ab' from buffer to active, the Register is cleared following the transmission from it of each number to its respective active register so as to be ready to receive the following number which is to be so transferred. (With respect to FIG. 54, see CAR PGA 351' conditions 17, 19, 21, and 23. With respect to FIG. 56, see CAR PGA 351' conditions 18, 20, 22, and 24).

As an incidental result of placing the system in Mode 2 the prorate calculations for I are also performed during Column 2800 of FIG. 56. However, the result of these computations is not written into the I active register 99' because neither signal G02 or G03 is present; instead the I number in the buffer register 47I' is read into the I active register 99' on program steps 2879 and 2880. Similarly, the results of the "prorate J" operations shown crossed out in FIG. 56 are not written into the J active register 101', but instead the J number from the J buffer register 47J' is transferred to the J active register 101' during program steps 2880 and 2881. After these transfers, the End Of Block Counter 186' is advanced from state S7 to state S8 (see FIG. 56 and condition (14) in PGA 183').

As stated previously, the J and Q numbers just transferred from buffer to active are utilized during the following Column X900 of the FIG. 56 iteration to compute ΔY_{88} which is subsequently executed during the following, third transitional iteration, shown in FIG. 57 and carried out during time period $\Delta T2501$.

Referring to FIG. 57, the third of the circular to linear Zone B type transitional iterations shown there will be recognized as being basically a standard linear iteration of the type shown in FIG. 46. As in the case of all of the iterations, the upgrading and updating operations are the same as those shown in FIG. 46. Additionally, the computations for deriving ΔX_{79} during Column X400 and ΔY_{8-10} during Column X900 are the same as those shown in the corresponding columns of FIG. 46. The first point of departure in FIG. 57 occurs in the fourth column, labeled "3300," during which the end point coordinates for the next path segment are transferred from the XEP and YEP registers to the XCEP and the YCEP registers respectively. These transfer operations are identical to those shown in the corresponding column of FIG. 52, which shows the third of the Zone B type linear to circular transitional iterations. Similarly, as in Column 3300 of FIG. 52, in the corresponding column of FIG. 57 the transfer of the end point coordinates is followed by advancement of the End of Block Counter 186' from its S8 count state to its S9 count state during program step 3379.

The second point of departure in FIG. 57 appears in Column X900 after the computation of ΔY_{8-10} and involves the starting of the tape reader 41' and the advancement of the counter 186' into its S1 count state in order to place the system into its Mode 1 operation for the following iteration. These last operations are also identical to the operations which occur during the corresponding program steps shown in FIG. 52 and are listed in detail in Table VII, Column X900, program steps 991-999.

Agreeably with the shape of the path segment 635 which is to be generated, the iteration performed during the following time period $\Delta T2502$ is a standard linear iteration of the type shown in FIG. 46. During this time period, therefore, the second half of the macromove ΔX_{79} computed during the previous time period $\Delta T2501$ is completed and the ΔY_{8-10} macromove, also computed during that time period, is carried out. A further pair of macromove numbers for the X and Y axes are computed during the time period $\Delta T2502$, as explained with reference to FIG. 46, and many more iterations of the FIG. 46 type are repeated until the end of path segment 635 is approached.

2. Assuming the Desired Transition is in Zone A

In view of the detailed explanation given for the process of transition from a linear path segment to a circular path segment, and vice versa, with reference to FIGS. 58, 59 and 60, the steps for making a transition from a circular path segment to a linear path segment during Zone A of an iteration will be described only briefly. It should be noted that, while the explanation which will be offered will be brief, the flow charts and Tables provide a complete understanding of this type of a transition.

Let it be assumed that, in generating the circular path segment 623 a succession of standard circular iterations have been performed beginning with time period $\Delta T1001$ and that the last of these occurred during time period $\Delta T2498$ as shown in FIG. 60. Let it be assumed further that, during the following time period $\Delta T2499$, proximity to the end point XCEP/YCEP is detected as a result of the first look-ahead operation, thus revealing that the point falls in Zone A of the iteration which will be performed during time period $\Delta T2500$. In this event, the iterations which will be performed during time periods $\Delta T2499$ and $\Delta T2500$ respectively will be those illustrated in FIG. 53 and 54. Following the second of the transitional iterations, the system will be able to revert to a standard linear iteration of the type shown in FIG. 46. This sequence of iterations, from a FIG. 49 standard circular iteration through a FIG. 53 and a FIG. 54 transitional iteration to a FIG. 46 standard linear iteration is shown in the flow chart comprising FIG. 45.

The operations represented by FIG. 53 and 54 may be most readily understood by settings FIGS. 53 and 54 side-by-side on the one hand, arranging FIGS. 55, 56 and 57 similarly on the other hand, and comparing the two groups of Figures. Referring first to FIGS. 55, 56 and 57 and also to FIG. 60, it will be recalled that, because the desired transition point XCEP/YCEP occurred in Zone B, i.e., during the second half of the time period $\Delta T2500$, it was not until the end of the iteration during which the transition point was to occur that the first linear macromove, ΔY_{60} , was computed. Indeed, during the first half of the following time period $\Delta T2501$ the X axis movement was still in accordance with the circular macromove number ΔX_{57} which was computed during the previous transitional iteration performed during time period $\Delta T2500$. Finally, it was only at the end of the time period $\Delta T2501$, that following the time period during which the transition point XCEP/YCEP was passed, that the tape reader 41' was started.

In contrast, in the case of a linear to circular transition where the desired transition point is passed during the first half of the time period, it becomes desirable to compute both of the macromove numbers ΔX and ΔY in accordance with the linear interpolation equations $\Delta X = I \cdot Q$ and $\Delta Y = J \cdot Q$ during the time period in which the iteration is to occur. Accordingly, during the FIG. 54 iteration performed in time period $\Delta T2500$, the transfer of data from buffer to active which, in the case of the FIG. 56 iteration is not performed until Column 2800, is carried out during Column 2300. Similarly, the transfer of the end point coordinates which did not occur until Column 3300 in the case of the FIG. 57 iteration performed during the time period $\Delta T2501$ for a Zone B transition, is performed during Column 3800 of the time period $\Delta T2500$, as shown in FIG. 54.

Since the transfer of data from buffer to active is to occur half a time period ΔT earlier for a Zone A transition, depicted in FIGS. 53 and 54, than for a Zone B transition, depicted in FIGS. 55, 56 and 57, the End of Block Counter 186' must also be stepped through its count states earlier. This, too, may be confirmed by comparing FIGS. 53 and 54 with FIGS. 55, 56 and 57. Thus, for a Zone A linear to circular transition the counter 186' is stepped through its count states S3, S4, S5, S6 into count state S7 during Columns X300 and X500 of the FIG. 53 iteration performed in time period $\Delta T2499$ and during Column X000 of the FIG. 54 iteration taking place during time period $\Delta T2500$. Consequently, the condition necessary for the transfer of data from buffer to active is present during

the period represented by the fourth column labeled "2300" of FIG. 54, in contrast to the FIG. 56 iteration where this condition was not present until the period represented by the column labeled "2800" of that iteration.

As a result of the transfer of the G number associated with the linear path segment 635 from buffer to active, a signal is on the LIN output of the path-type indicator 106' associated with the G Active Register 105', causing the computation of ΔX during Column X400 of the iteration $\Delta T2500$ to be in accordance with the linear interpolation computation $\Delta X = I \cdot Q$. Furthermore, due to the transfer of I and Q from buffer to active the numbers which will be used in this computation will be those associated with the linear path segment 635. The same also applies to the computation of ΔY during the last column of the iteration.

To cause the end point coordinates XEP and YEP to be transferred from buffer to active during time period $\Delta T2500$, the counter 186' is advanced from its S7 count state to its S8 count state during Column 2300 of the FIG. 54 iteration. Appearance of a signal on the S8 output of the decoder 187' places the system in its Mode 3 operation and as a result, as indicated in Table XI, the operations necessary to transfer the end point coordinates from buffer to active are carried out during program steps 3835 through 3837. Following this, the counter is once again advanced from its count state S8 to its count state S9. In response to the presence of a signal on the S9 output of the decoder 187', the tape reader is started during program step 991 (see Table VII, Column X900) and during the following program steps, again in response to the presence of a signal on the S9 output of the decoder 187', the sequence by which the counter is advanced through its count states S10-S16 into its count state S1 is begun. It follows, therefore, that a standard linear iteration in accordance with FIG. 46 is performed during the following time period $\Delta T2501$.

It will have been noted that, both in the case of a Zone A and a Zone B circular to linear transition, the look-ahead operations are performed only during the first of the transitional iterations. Thus, with reference to a Zone B transition, the look-ahead operations are not a part of the FIG. 56 and FIG. 57 iterations, and, for a Zone A transition, the look-ahead operations are omitted from the FIG. 54 transitional iteration. The reason for omitting these operations is simply that, once proximity to the designated transition point XCEP/YCEP has been detected during the preceding transitional iteration, performance of the look-ahead operations during the following transitional iterations becomes unnecessary. The cause of the interruption of the look-ahead operation during all but the first of the transitional iterations is that for both a Zone A and a Zone B circular to linear transition the system is switched out of Mode 1 operation during the first transitional iteration, when the counter 186' is advanced to its S5 count state (FIGS. 53 and 55). It will be recalled that, in order for the system to be in its Mode 1 operation the counter must be in one of its three initial count states S1-S3. It will also be recalled from the previous explanation of the look-ahead process with reference to Table IX that, in order for the look-ahead operations to be performed, the system must be in its first operating mode. It follows, therefore, that the look-ahead operations are only performed during the first transitional iteration and are not resumed again until the first standard linear iteration following the transitional iteration.

It may be confirmed from FIGS. 48, 51, and 52 that the above comments concerning the look-ahead operations apply also to the linear to circular transitional iterations.

K. Possible Modifications to Both of the Exemplary Systems

In the foregoing description of two exemplary systems for performing interpolation according to the invention, as well as in the claims which follow, the assumption has been that, in order to make a tool cut a chosen path in a workpiece, the tool alone is moved. Clearly, this need not be so. The assumption had only been made to simplify both the specification and the claims. In practice, an equally valid assumption would be that

movement along one or more of the axes is by the workpiece instead of the tool. Naturally, the workpiece would have to be moved along its assigned axis (or axes) in a direction exactly opposite to that in which the tool would travel along that axis. This might be achieved in several ways. The data block on the tape could represent the excursions to be made by the tool as if it alone were to be moved to cut the desired path in a stationary workpiece. This data could then be processed exactly as described, but the servo associated with the axis along which the workpiece is to be moved would be so connected as to reverse the direction of motion from that which would be required if the tool were to be moved along that axis. In the first exemplary embodiment disclosed herein, this could be done simply by reversing the input leads to the phase shifter 531 in FIG. 9a so as to reverse the sense of the feedback signal and by reversing the connection of the armature winding of the servo motor 21 so as to reverse its direction of rotation. The same changes could also be made to the corresponding components of the second exemplary system shown in FIG. 40.

Alternatively, the computations performed by the system could be modified in a manner which will be apparent to those skilled in the art so as to reverse the sense of movement along selected axes without requiring any change to the servos. Indeed, it is common practice to provide numerical controls with "axis reverse" switches to permit making of a "left-hand" part from data programmed for a "right-hand" part.

L. Conclusion

What has been described is in part a method for deriving through a process of interpolation from blocks of data representing successive path segments a continuous, cyclic flow of numerical data representing an almost incredibly large number of extremely closely spaced points distributed along those path segments. Also disclosed have been two exemplary systems for carrying out this process of interpolation rapidly and automatically and featuring the use of a single time-shared computer made possible by the regular periodicity of the required computations.

The nature of the disclosed method and of the exemplary systems for performing it is illustrated by the data generated for a typical path segment. Such a path segment might be a straight line 6 inches long and represented on magnetic tape as a block of data comprising 6 digital numbers. If it be assumed that the desired speed at which movement along the 6-inch path segment is to occur is 1 inch per minute, the total time for executing a path segment will be 6 minutes. During this time, 180,000 successive positions will be generated for each of the axes along which movement along the path is to occur, or once every 2 milliseconds. The same applies in the case of a circular path segment, and the method and systems disclosed also permit a smooth transition from one path segment to the next, regardless of whether or not successive path segments are of the same type.

I claim as my invention:

1. The method of moving a member simultaneously along X and Y axes to produce resultant movement along a desired path segment which extends either substantially along or at right angles to a vector which is the sum of X and Y axis components d_x and d_y , comprising the steps of
 - a. computing and digitally signaling in a digital processing apparatus the numerical values of small incremental moves ΔX and ΔY each of which is proportional to a different one of the components d_x and d_y by the same factor of proportionality which is equal to $(V \cdot \Delta T)/D$ where V is a numerical value representing path velocity, ΔT is a numerical value representing a time period, and D is a numerical value representing the length of said vector,
 - b. measuring off successive, equal and actual time periods ΔT , and
 - c. during each such time period utilizing said digital signaling to cause movement of said member through the distance ΔX along said X axis and through the distance ΔY along the Y axis.

2. The method defined in claim 1 further including the step of changing the factor of proportionality relating the values of ΔX and ΔY to d_x and d_y by changing the numerical value V while keeping the numerical value ΔT constant, thereby to determine and control the velocity V of the movable member along the desired path segment.

3. The method defined in claim 1, further characterized in that said components d_x and d_y are the X and Y components I and J of a vector path segment having a length D and wherein said steps (b) and (c) are repeated until said member has moved along the desired path through a distance substantially equal to D .

4. The method defined in claim 3 wherein the incremental moves ΔX and ΔY are computed according to the expressions

$$\Delta X = \frac{V \cdot \Delta T}{D} \cdot I$$

$$\Delta Y = \frac{V \cdot \Delta T}{D} \cdot J$$

5. The method defined in claim 1, further characterized in that said components d_x and d_y are the X and Y axis components J and I of radius vectors R extending from the center of an arc to points on an arc forming the path segment, and the incremental moves ΔX and ΔY are the X and Y axis components of resultant vector increments ΔD extending substantially along chords of arcs which are bisected at right angles by said radius vectors.

6. The method defined in claim 5 wherein the incremental moves ΔX and ΔY are computed according to the expressions

$$\Delta X = \frac{V \cdot \Delta T}{R} \cdot J$$

$$\Delta Y = \frac{V \cdot \Delta T}{R} \cdot I$$

where V is the numerical value of a desired velocity of the member along the arc, ΔT is the numerical value of one of said time periods, and R is the length of said radius vector.

7. The method of moving a member simultaneously along X and Y axes to produce resultant movement along successive path segments D which are the vector sum of X and Y axis components I and J , comprising the steps of

1. producing within a digital processing apparatus, in accordance with successive sets of compiled data defining the successive path segments, successive sets of digital signals representing numerical values of (i) the x axis component length I , (ii) the Y axis component length J , and (iii) a factor of proportionality Q which is itself directly proportional to the desired path velocity of said member and inversely proportional to the path segment length D ,

- a. computing and digitally signaling in said apparatus, from said successive sets of signals, the numerical values of small incremental moves ΔX and ΔY which are respectively proportional to I and J by said factor of proportionality Q ,
- b. measuring off successive, equal time periods ΔT in actual time,
- c. during each such time period utilizing said digital signaling to cause movement of said member through the distance ΔX along the X axis and through the distance ΔY along the Y axis and
- d. changing said factor of proportionality Q as signaled in different ones of said successive sets of signals in order to change the path velocity at which said member is actually moved in response to said successive sets of compiled data.

8. The method defined in claim 7 in which said step (a) includes the multiplication of $I \times Q$ and $J \times Q$ in order to produce said digital signaling of the numerical values of said small incremental moves ΔX and ΔY .

9. The method defined in claim 8 wherein the factor of proportionality Q is equal to $(V \cdot \Delta T)/D$, where V is the desired

velocity along the path segment, D is the length of the path segment and ΔT numerically represents one of the said time periods; and wherein said incremental moves ΔX and ΔY are computed according to the respective expressions

$$\Delta X = \frac{V \cdot \Delta T}{D} \cdot I$$

$$\Delta Y = \frac{V \cdot \Delta T}{D} \cdot J$$

10. The method of moving a member simultaneously along X and Y axes to produce resultant motion substantially along successive arcuate path segments defined by radius vectors R extending from centers to points on the arcs and having X and Y axis components I and J , said method comprising the steps of

1. producing within a digital processing apparatus, in accordance with successive sets of compiled data defining the successive arcuate path segments, successive sets of digital signals representing numerical values of (i) the X axis component length I , (ii) the Y axis component length J , and (iii) a factor of proportionality Q which is itself directly proportional to the desired path velocity and inversely proportional to the length of the vector R ,
 - a. computing and digitally signaling in said apparatus, from said successive sets of signals, the numerical values of small incremental moves ΔX and ΔY which are respectively proportional to J and I by said factor of proportionality Q ,
 - b. measuring off at least one series of successive, equal and actual time periods ΔT ,
 - c. during each of the time periods in each of said series utilizing said digital signaling to cause movement of said member through at least the one of the distances ΔX and ΔY along the respective axes X and Y and
 - d. changing said factor of proportionality Q as signaled in different ones of said successive sets of signals in order to change the path velocity at which said member is actually moved in response to said successive sets of compiled data.

11. The method defined in claim 10 in which said step (a) includes the multiplication of $I \times Q$ and $J \times Q$ in order to produce said digital signaling of the numerical values of said small incremental moves ΔX and ΔY .

12. The method defined in claim 11 wherein the factor of proportionality Q is equal to $(V \cdot \Delta T)/R$, where V is the desired velocity along the arc, R is the length of said radius vector, and ΔT numerically represents one of said time periods, and wherein said incremental moves ΔX and ΔY are computed rapidly during each time period according to the expressions

$$\Delta X = \frac{V \cdot \Delta T}{R} \cdot J$$

$$\Delta Y = \frac{V \cdot \Delta T}{R} \cdot I$$

13. The method of moving a member simultaneously along X and Y axes to produce resultant composite motion at a desired vector velocity V , comprising the steps of

- a. computing and digitally signaling in a digital processing apparatus numerical values of ΔX and ΔY which are respectively equal to $V_x \cdot \Delta T$ and $V_y \cdot \Delta T$ where V_x and V_y are numbers representing the X and Y axis components of the desired vector velocity V and ΔT is a predetermined constant number representing time,
- b. measuring off successive, equal and actual time periods ΔT ,
- c. during each such actual time period causing said member in response to said digital signaling to move the distance ΔX along the X axis and the distance ΔY along the Y axis, and
- d. changing the velocity representing numbers V_x and V_y by equal proportions to modify the signaled values ΔX and

ΔY , thereby to modify the magnitude of the desired velocity V and the resultant velocity at which the member is moved.

14. The method moving a member simultaneously along X and Y axes to produce resultant composite motion at different desired velocities V and different vector angles, comprising the steps of

- a. computing in a digital processing apparatus a first number proportional to $V \cdot \cos \theta$, where V represents the desired velocity and θ represents the desired vector angle relative to the X axis,
- b. computing in said apparatus a second number proportional to $V \cdot \sin \theta$,
- c. computing and digitally signaling in said apparatus a third number ΔX equal to the product of the first number and a predetermined constant, time-representing number ΔT ,
- d. computing and digitally signaling in said apparatus a fourth number ΔY equal to the product of said second number and ΔT ,
- e. measuring off successive, equal and actual time periods ΔT , and
- f. during each such time period causing said member in response to the signals representing ΔX and ΔY to move the distance ΔX along the X axis and to move the distance ΔY along the Y axis.

15. The method defined in claim 14 further including the step of changing the quantity V to change the composite velocity at which said member is moved.

16. The method defined in claim 14 further including the step of changing the angle θ to change the direction in which said member is moved.

17. The method defined in claim 14 wherein the quantity $\cos \theta$ is determined by the ratio I/D and the quantity $\sin \theta$ is determined by the ratio J/D , where I and J are the X and Y axis component distances of a desired path segment having a length D , the quantities I and J normally being substantially greater than the quantities ΔX and ΔY .

18. The method defined in claim 14 wherein the quantity $\cos \theta$ is determined by the ratio J/R and the quantity $\sin \theta$ is determined by the ratio I/R , where I and J are the X and Y axis component distances of a radius length R extending from the center of a circle to a point on an arc of that circle which is to be traced by the movement of the member.

19. The method defined in claim 18 wherein the component distances I and J and the ratios I/R and J/R are changed at intervals of ΔT in actual time.

20. The method defined in claim 14 wherein the third and fourth numbers ΔX and ΔY are recomputed and re-signaled in said apparatus at intervals ΔT in actual time.

21. A method of generating a series of numerical command signals for moving a member displaceable along X and Y axes at velocity V in a straight line path at angle θ with the X axis comprising:

- a. measuring off successive equal periods of ΔT in actual time,
- b. producing in a digital processing apparatus first and second sets of signals respectively representing numerically unit moves $V \cdot \Delta T \cdot \sin \theta$ and $V \cdot \Delta T \cdot \cos \theta$,
- c. producing in said apparatus third and fourth sets of signals digitally representing respectively commanded positions XSC and YSC along the X and Y axes to which said member is to move, and
- d. changing said third and fourth sets of signals, once during each ΔT time period, by amounts equal to said unit moves $V \cdot \Delta T \cdot \sin \theta$ and $V \cdot \Delta T \cdot \cos \theta$, respectively.

22. The method of moving a member displaceable along X and Y axes at a selected angle in space and through a desired composite distance D at a desired composite velocity V which comprises

- a. representing by digital signals in a digital processing apparatus the X and Y components I and J of said composite distance,

- b. processing said digital signals in said apparatus to derive first and second sets of signals representing ΔX and ΔY numbers which are respectively equal to:
1. the product of I , a predetermined time-representing number ΔT and the composite velocity V , all divided by the composite distance D , and
 2. the product of J , said predetermined time-representing number ΔT , and the composite velocity V all divided by the composite distance D ;
- c. measuring off successive equal time periods of ΔT in actual time, and
- d. during each such time period utilizing said first and second sets of signals to drive said member simultaneously and substantially uniformly through the distance ΔX along the X axis and through the distance ΔY along the Y axis until the member has been moved substantially through the composite distance D .
23. The method of moving a member displaceable along X and Y axes through successive path segments D at successive selected angles in space with successive desired composite velocities V which comprises:
- a. representing numerically by digital electrical signals the X and Y components I and J of one path segment D ,
 - b. processing said signals to derive first and second sets of electrical signals numerically representing ΔX and ΔY distances which are respectively equal to:
 1. the product of I , a predetermined time-representing number ΔT , and the composite velocity V , all divided by the one path segment D , and
 2. the product of J , said predetermined time-representing number ΔT , and the composite velocity V all divided by the one path segment D ,
 - c. measuring off successive equal periods of ΔT in actual time,
 - d. during each such time period utilizing said first and second sets of signals to drive said member simultaneously and substantially uniformly through the distance ΔX along the X axis and through the distance ΔY along the Y axis until the member has been moved substantially through the one path segment D , and
 - e. during at least one of the last few periods ΔT prior to the complete execution of the one path segment D , representing numerically by digital electrical signals the X and Y components I and J of the next path segment D , and thereafter continuing the steps (b) through (d) defined above to effect execution of movement by the member through said next path segment D .
24. The method of moving a member displaceable along X and Y axes through a desired composite distance D at a selected angle in space and with a desired composite velocity V which comprises
- a. representing by first digital signals in a digital processing apparatus a numerical quantity Q which is equal to the quotient of the velocity V divided by the distance D , all multiplied by a number representing a predetermined time period ΔT ,
 - b. representing by second digital signals in said apparatus the X and Y components I and J of said composite distance D ,
 - c. processing said first and second digital signals to derive first and second sets of signals in said apparatus representing ΔX and ΔY numbers which are respectively equal to:
 1. the product of I times the value of Q , and
 2. the product of J times the value of Q ,
 - d. measuring off successive equal time periods of ΔT in actual time, and
 - e. during each such time period utilizing said first and second sets of signals to drive said member simultaneously and substantially uniformly through the distance ΔX along the X axis and through the distance ΔY along the Y axis.
25. The method defined in claim 1 further including the steps of

- d. computing and digitally signaling in said apparatus the numerical values of micromoves $\Delta X/N$ and $\Delta Y/N$, where N is a preselected constant,
 - e. measuring off N equal and successive actual time periods $\Delta T/N$ during each of said periods ΔT ,
 - f. and executing the said step (c) by utilizing the digital signaling of micromoves during each of said periods $\Delta T/N$ to cause movement of said member through the distance $\Delta X/N$ along said X axis and through the distance $\Delta Y/N$ along the Y axis.
26. The method of moving a member simultaneously along X and Y axes to produce resultant composite motion at a desired vector velocity V , comprising the steps of
- a. computing and digitally signaling in digital processing apparatus numerical values of ΔX and ΔY which are respectively equal to $V_x \cdot \Delta T$ and $V_y \cdot \Delta T$, where V_x and V_y are numbers representing the X and Y axis components of the desired velocity vector V and ΔT is a predetermined constant number representing time,
 - b. measuring off successive, equal and actual time periods $\Delta T/N$, where N is a predetermined constant,
 - c. creating X and Y axis command signals XSC and YSC representing the commanded positions of said member relative to reference points along the X and Y axes, respectively,
 - d. during each said time periods and in response to said digital signaling changing X and Y axis command signals by amounts equal to $\Delta X/N$ and $\Delta Y/N$, respectively, and
 - e. utilizing said changing command signals to move said member along the X and Y axes so as to keep the actual X and Y axis positions of said member substantially in agreement with the commanded positions represented by said changing command signals.
27. The method set forth in claim 26 wherein said X and Y axis command signals XSC and YSC are changed in said apparatus N times during each actual time period of ΔT duration, so that at the end of each ΔT period the member will have moved distances ΔX and ΔY along the X and Y axes, respectively.
28. The method set forth in claim 26 wherein said X and Y axis command signals are changed in said apparatus according to step (e) $N-1$ times during $N-1$ successive time periods $\Delta T/N$ and wherein during each N th time period $\Delta T/N$ said X and Y command signals are changed in said apparatus by whatever amounts may be necessary to make the total change therein for the preceding N periods equal to ΔX and ΔY , respectively.
29. A method of moving a member displaceable along X and Y axes at a selected angle θ in space at a desired composite velocity V comprising the steps of
- a. representing by first and second sets of digital signals in a digital processing apparatus numerical values of ΔX and ΔY respectively equaling $V \cdot \cos \theta \cdot \Delta T$ and $V \cdot \sin \theta \cdot \Delta T$, where ΔT numerically represents a predetermined time period,
 - b. deriving in said apparatus from said first and second sets of digital signals third and fourth sets of signals digitally representing $\Delta X/N$ and $\Delta Y/N$ where in N is a predetermined numerical constant,
 - c. measuring off successive periods $\Delta T/N$ in actual time,
 - d. creating from said third and fourth sets of signals digital command numbers XSC and YSC in said apparatus which are changed in steps during each such period $\Delta T/N$ by incremental amounts of $\Delta X/N$ and $\Delta Y/N$ and
 - e. driving said member along the X and Y axes at velocities respectively proportional to the average rates of change of said digital command numbers XSC and YSC .
30. The method of moving a member displaceable along X and Y axes through a desired composite distance D at a desired angle θ with a desired composite velocity V comprising the steps of
- a. representing in a digital processing apparatus by first digital signals the numerical values of the X and Y axis components I and J of said distance D ,

- b. representing in said apparatus by second digital signals the numerical values of macromove distances ΔX and ΔY which are equal to $(I \cdot V \cdot \Delta T)/D$ and $(J \cdot V \cdot \Delta T)/D$, where ΔT is a number representing a predetermined time interval,
- c. measuring off in actual time successive equal minor time periods each $\Delta T/N$ in duration, where N is a predetermined constant,
- d. generating in said apparatus third digital signals numerically representing commanded coordinate positions XSC and YSC of said member along the X and Y axes,
- e. deriving from said second digital signals fourth digital signals numerically representing micromove distances $\Delta X/N$ and $\Delta Y/N$,
- el. producing in said apparatus fifth and sixth digital signals representing numerically commanded positions XSC and YSC ,
- f. algebraically combining in said apparatus during each of said minor time periods $\Delta T/N$ said third and fourth digital signals to update to a new value said fifth and sixth digital signals such that

$$XSC_2 = XSC_1 + \frac{\Delta X}{N} \text{ and } YSC_2 = YSC_1 + \frac{\Delta Y}{N}$$

where XSC_1 and YSC_1 symbolize the value of XSC and YSC at the beginning of a given minor time period, and XSC_2 and YSC_2 symbolize the said new value, and

- g. moving said member along the X and Y axes by filtering digital servos which act to keep the actual positions of said member along the X and Y axes substantially equal to the changing numerical values of XSC and YSC .

31. A method of generating a series of numerical command signals for moving a member displaceable along X and Y axes at velocity V in a straight line path at angle θ with the X axis comprising

- a. measuring off successive equal major periods ΔT in actual time,
- b. measuring off successive equal minor periods $\Delta T/N$ in actual time so as to subdivide each of said major periods ΔT into N minor time periods $\Delta T/N$,
- c. producing in a digital processing apparatus first and second sets of signals respectively representing numerically macromove numbers $V \cdot \Delta T \cdot \cos \theta$ and $V \cdot \Delta T \cdot \sin \theta$,
- d. producing in said apparatus third and fourth sets of signals respectively representing numerically micromove numbers

$$\frac{V \cdot \Delta T}{N} \cdot \cos \theta \text{ and } \frac{V \cdot \Delta T}{N} \cdot \sin \theta$$

by dividing by a factor of N the respective ones of the signaled numbers $V \cdot \Delta T \cdot \cos \theta$ and $V \cdot \Delta T \cdot \sin \theta$,

- e. producing in said apparatus fifth and sixth sets of signals respectively representing commanded positions XSC and YSC along the X and Y axes to which said member is to move, and
- f. changing said fifth and sixth sets of signals during successive ones of said minor time periods by amounts equal to said micromove numbers

$$\frac{V \cdot \Delta T}{N} \cdot \cos \theta \text{ and } \frac{V \cdot \Delta T}{N} \cdot \sin \theta$$

respectively.

32. The method of moving a member simultaneously along X and Y axis to produce resultant movement along a desired path segment which extends substantially along or at right angles to a vector which is the sum of X and Y axis components d_x and d_y , comprising the steps of

- a. computing and digitally signaling in a digital processing apparatus the numerical values of small incremental moves ΔX and ΔY which are each proportional to a different one of the components d_x and d_y by the same factor of proportionality,
- b. measuring off a first series of immediately successive, equal and actual time periods ΔT_1 ,

- c. measuring off a second series of immediately successive, equal and actual time periods ΔT_2 each of which is equal in duration to a time period ΔT_1 and staggered to partially overlap one of the periods ΔT_1 ,

- d. during each time period ΔT_1 , utilizing the signaling of ΔX to cause movement of the member through the distance ΔX along the X axis, and
- e. during each time period ΔT_2 , utilizing the signaling of ΔY to cause movement of the member through the distance ΔY along the Y axis.

33. The method defined in claim 32 and further characterized in that said step (a) is performed repeatedly at intervals equal in duration to ΔT_1 and ΔT_2 , thereby to reflect in the computed values ΔX and ΔY any change in the components d_x and d_y or in the factor of proportionality.

34. The method of generating periodically updated numerical command signals for moving a member displaceable along X and Y axes at any desired angle θ relative to the X axis and at a composite velocity V , said method comprising the steps of

- a. measuring off a first series of immediately successive, equal and actual time periods ΔT_1 ,
- b. measuring off a second series of immediately successive, equal and actual time periods ΔT_2 each of which is equal in duration to a time period ΔT_1 and is staggered to partially overlap one of the periods ΔT_1 ,
- c. computing and digitally signaling in a digital processing apparatus the numerical values of incremental moves ΔX and ΔY which are respectively proportional to $V \cdot \cos \theta \cdot \Delta T$ and $V \cdot \sin \theta \cdot \Delta T$, where ΔT numerically represents the duration of each period ΔT_1 or ΔT_2 ,
- d. utilizing, during each actual time period ΔT_1 , the digital signaling of ΔX to cause a change in an X axis command signal by the amount ΔX , and
- e. utilizing, during each actual time period ΔT_2 , the digital signaling of ΔY to cause a change in a Y axis command signal by the amount ΔY .

35. The method set forth in claim 34, and wherein said computing and digital signaling of the moves ΔX and ΔY is repeated at a rate of once per period ΔT of time, so as to modify the composite velocity or the angle represented by the changing X and Y axis command signals if the quantities V or θ change.

36. A method of generating a series of numerical commands for moving a member displaceable along X and Y axes at velocity V in a straight line path at angle θ with the X axis comprising

- a. generating in a digital processing apparatus first, second, third and fourth sets of time base signals, the first recurring at equal time periods of ΔT_1 , the second recurring at equal time periods ΔT_2 , the third recurring at equal time periods of $\Delta T_1/N$, N times during each time period ΔT_1 , and the fourth recurring at equal time periods $\Delta T_2/N$, N times during each of the time periods ΔT_2 , said time periods ΔT_1 and ΔT_2 being of equal duration ΔT but staggered and interleaved relative to one another so that said time periods $\Delta T_1/N$ and $\Delta T_2/N$ are coextensive in time,
- b. producing in said apparatus fifth and sixth sets of signals respectively representing commanded points XSC and YSC along the X and Y axes to which said member is to move,
- c. producing in said apparatus a seventh set of signals digitally representing the quantity $V \cdot \Delta T \cdot \sin \theta$,
- d. producing in said apparatus an eighth set of signals digitally representing the quantity $V \cdot \Delta T \cdot \cos \theta$,
- e. deriving from said seventh set of signals a ninth set of signals in said apparatus digitally representing the quantity

$$\frac{V \cdot \Delta T}{N} \cdot \sin \theta$$

- f. deriving from said eighth set of signals a tenth set of signals in said apparatus digitally representing the quantity

$$\frac{V \cdot \Delta T}{N} \cdot \cos \theta$$

- g. during successive ones of said time periods $\Delta T1/N$ changing the quantity represented by said fifth set of signals by the quantity represented by said ninth set of signals, and
- h. during successive ones of said time periods $\Delta T2/N$ changing the quantity represented by said sixth set of signals by the quantity represented by said tenth set of signals.
37. The method set forth in claim 36, further characterized in that the said step (c) is executed once during each time period $\Delta T1$, and said step (d) is executed once during each time period $\Delta T2$.
38. The method of moving a member simultaneously along X and Y axes at a resultant velocity V through a series of points distributed along an arc of a circle, comprising the steps of
- measuring off a succession of equal actual time intervals ΔT ,
 - representing by digital signals in a digital processing apparatus a number XSC designating the X axis coordinate of a point on the arc,
 - representing by digital signals in said apparatus a number YSC designating the Y axis coordinate of a point on the arc,
 - computing in said apparatus at successive intervals ΔT the X axis displacement ΔX between successive points on the arc which are separated by a linear chord distance ΔD ,
 - computing in said apparatus at successive intervals ΔT the Y axis displacement ΔY between successive points on the arc which are separated by a linear chord distance ΔD , where the distance ΔD for steps (d) and (e) is equal to $V \cdot \Delta T$,
 - changing the representation of the number XSC by the computed quantity ΔX at successive intervals ΔT ,
 - changing the representation of the number YSC by the computed quantity ΔY at successive intervals ΔT ,
 - moving said member along the X axis through distances and at a velocity corresponding respectively to the changes and average rate of change of the represented number XSC, and
 - moving said member along the Y axis through distances and at a velocity corresponding respectively to the changes and the average rate of change of the represented number YSC.
39. A method of moving a member displaceable along X and Y axes on a circular arc containing an initial point P0 and a succession of points P1, P2, P3, P4, P5...Pn, each of said points P0 through Pn being a linear distance ΔD from the adjacent one of said points comprising
- measuring off successive, equal and actual time periods T1, T2, T3, T4, T5...Tn,
 - prior to the respective time periods T1, T3, T5, ... producing in a digital processing apparatus signals respectively representing numerically the X axis distances ΔX_{02} , ΔX_{24} , ΔX_{46} ... between the points P0 to P2, P2 to P4, P4 to P6 ...
 - during respective pairs of said series of time periods T1-T2, T3-T4, T5-T6 ... driving said member under control of said signals through said distances ΔX_{02} , ΔX_{24} , ΔX_{46} ... along the X axis,
 - prior to the respective time periods T2, T4, T6 ... producing in said apparatus signals respectively representing numerically the Y axis distances Y13, Y35, Y57 ... between the points P1 to P3, P3 to P5, P5 to P7 ..., and
 - during respective pairs of said time periods T2-T3, T4-T5, T6-T7 ... driving said member along the Y axis through said distances ΔY_{13} , ΔY_{35} , ΔY_{57} ... with the speed of said member along both axes being kept substantially uniform during each of said pairs T1-T2, T3-T4, T5-T6 ... and T2-T3, T4-T5, T6-T7 ... of time periods.
40. A method of moving a member displaceable along X and Y axes on a circular arc generated by a radius vector of length R which passes from an initial point P0 on said arc

through a series of points P1, P2, P3, P4, P5 on said arc, each of said points P0 through P5 being a linear distance ΔD from the adjacent one of said points, comprising

- measuring off successive, equal and actual time periods T1, T2, T3, T4, T5 ...
- prior to time period T1:
 - producing in a digital processing apparatus first and second sets of signals respectively representing numerically the inverse X component I0 of said radius vector in its initial position at point P0 and the inverse Y component J1 of said radius vector at point P1,
 - processing said J1 signals to derive therefrom in said apparatus signals numerically representing the quantity

$$\frac{J_1 \cdot \Delta D}{R} = \Delta X_{02}$$

and storing said ΔX_{02} signals,

- during time period T1:
 - processing said I0 and ΔX_{02} signals to derive therefrom in said apparatus signals numerically representing the quantity $I_0 - \Delta X_{02} = I_2$ and storing said I2 signals,
 - processing said I2 signals to derive therefrom in said apparatus signals numerically representing the quantity

$$\frac{I_2 \cdot \Delta D}{R} = \Delta Y_{13}$$

and storing said ΔY_{13} signals,

- during time periods T1 and T2 utilizing said ΔX_{02} signals to move said member at uniform speed along the X axis through a distance ΔX_{02} ,
- during time period T2:
 - processing said J1 and ΔY_{13} signals to derive therefrom in said apparatus signals numerically representing the quantity $J_1 - \Delta Y_{13} = J_3$ and storing said J3 signals,
 - processing in said apparatus said J3 signals to derive therefrom signals numerically representing the quantity

$$\frac{J_3 \cdot \Delta D}{R} = \Delta X_{24}$$

and storing said ΔX_{24} signals,

- during time periods T2 and T3 utilizing said ΔY_{13} signals to move said member at uniform speed along the Y axis through a distance ΔY_{13} ,
- during time period T3:
 - processing in said apparatus said I2 and ΔX_{24} signals to derive therefrom signals representing the quantity $I_2 - \Delta X_{24} = I_4$ and storing said I4 signals,
 - processing in said apparatus said I4 signals to derive therefrom signals numerically representing the quantity

$$\frac{I_4 \cdot \Delta D}{R} = \Delta Y_{35}$$

and storing said ΔY_{35} signals,

- during time periods T3 and T4 utilizing said ΔX_{24} signals to move said member at uniform speed along the X axis through a distance ΔX_{24} , and
- during time periods T4 and T5 utilizing said ΔY_{35} signals to move said member at a uniform speed along the Y axis through a distance ΔY_{35} .

41. A method of moving a member displaceable along X and Y on a circular arc generated by a radius vector of length R which passes from an initial point P0 on said arc through a series of points P1, P2, P3, P4, P5 on said arc, each of said points P0 through P5 being a distance ΔD from the adjacent one of said points, comprising

- measuring off successive, equal and actual time periods T1, T2, T3, T4, T5 ... and measuring off successive shorter time intervals t each of which is in duration equal to two of said periods divided by N, where N is an integer,
- prior to time period T1:
 - producing in a digital processing apparatus first and second sets of signals respectively representing numerically the inverse X component I0 of said radius vector in

its initial position and the inverse Y component J_1 of said radius vector at point P_1 ,

2. processing in said apparatus the signals J_1 to derive therefrom signals numerically representing the quantity

$$\frac{J_1 \cdot \Delta D}{R} = \Delta X_{02}$$

and storing said ΔX_{02} signals,

- c. during time period T_1 :

1. processing in said apparatus said I_0 and ΔX_{02} signals to derive therefrom signals numerically representing the quantity $I_0 - \Delta X_{02} = I_2$ and storing said I_2 signals,
2. processing in said apparatus said I_2 signals to derive therefrom signals numerically representing the quantity

$$\frac{I_2 \cdot \Delta D}{R} = \Delta Y_{13}$$

and storing said ΔY_{13} signals,

- d. during time periods T_1 and T_2 :

1. deriving in said apparatus from said ΔX_{02} signals, signals numerically representing the quantity $\Delta X_{02}/N$, and
2. utilizing said last derived signals to move said member along the X axis through a distance of $\Delta X_{02}/N$ during each of N successive time intervals t ,

- e. during time period T_2 :

1. processing in said apparatus said J_1 signals and said ΔY_{13} signals to derive therefrom signals numerically representing the quantity $J_1 - \Delta Y_{13} = J_3$ and storing said J_3 signals,
2. processing in said apparatus said signal J_3 to derive therefrom signals numerically representing the quantity

$$\frac{J_3 \cdot \Delta D}{R} = \Delta X_{24}$$

and storing said ΔX_{24} signals,

- f. during time periods T_2 and T_3 :

1. deriving in said apparatus from said ΔY_{13} signals, signals numerically representing the quantity $\Delta Y_{13}/N$, and
2. utilizing said last derived signals to move said member along the Y axis through a distance $\Delta Y_{13}/N$ during each of N successive time intervals t ,

- g. during time period T_3 :

1. processing in said apparatus said I_2 signals and said ΔX_{24} signals to derive therefrom signals representing the quantity $I_2 - \Delta X_{24} = I_4$ and storing said I_4 signals,
2. processing in said apparatus said I_4 signals to derive therefrom signals numerically representing the quantity

$$\frac{I_4 \cdot \Delta D}{R} = \Delta Y_{35}$$

signals,

- h. during time periods T_3 and T_4 :

1. deriving in said apparatus from said signals ΔX_{24} signals numerically representing the quantity $\Delta X_{24}/N$, and
2. utilizing said last derived signals to move said member along the X axis through a distance $\Delta X_{24}/N$ during each of N successive time intervals t ,

- i. during time periods T_4 and T_5 :

1. deriving in said apparatus from said ΔY_{35} signals, signals numerically representing the quantity $\Delta Y_{35}/N$, and
2. utilizing said last derived signals to move said member along the Y axis through a distance $\Delta Y_{35}/N$ during each of N successive time intervals t ,

42. The method of moving a member simultaneously along X and Y axes to produce resultant motion along a desired path to reach a commanded end point having position coordinates X_2, Y_2 , comprising the steps of

- a. computing and signaling by first and second sets of digital signals in a digital processing apparatus the numerical values of incremental moves ΔX and ΔY which are

respectively proportional to $\cos \Theta$ and $\sin \Theta$, where Θ is the angle at which said path lies relative to the X axis,

- b. measuring off successive, equal and actual time periods ΔT ,

- c. during each of said periods utilizing said first and second sets of signals to cause said member to move the distance ΔX along the X axis and to move the distance ΔY along the Y axis,

- d. detecting and signaling in said apparatus a forecasted overrun during that particular one of said time periods ΔT when one of the coordinates X_p or Y_p which will be reached by said member at the end of the next M periods ΔT lies beyond the corresponding end point coordinate X_2 or Y_2 on the path, where M is an integer greater than zero,

- e. in response to such detection, adjusting said first and second signals in said apparatus to represent modified values $\Delta X'$ and $\Delta Y'$ of incremental moves which will bring the member to actual coordinates X_a and Y_a substantially equal to the end point coordinates X_2 and Y_2 .

43. The method defined in claim 42 and wherein said integer M is 11.

44. The method defined in claim 42 and wherein said integer M is 1.

45. The method defined in claim 42 and wherein the modified value $\Delta X'$ is made equal to $1/M$ th of the difference $(X_2 - X'_a)$ and the modified value $\Delta Y'$ is made equal to $1/M$ th of the difference $(Y_2 - Y'_a)$, where X'_a and Y'_a are the coordinates to be reached by said member during said particular one of said time periods.

46. The method defined in claim 43 and wherein said step (d) includes the sub-steps of

- d1. producing in said apparatus an indicator signal indicative of particular axis (X or Y) along which the member's velocity is the greatest,

- d2. in response to said indicator signal and during each of said time periods ΔT , computing and digitally signaling in said apparatus the numerical value of the predicted coordinate (X_p or Y_p) which will be reached by said member along the axis of greatest velocity (X or Y) at the end of the next M periods ΔT ,

- d3. comparing in said apparatus said signaled value (X_p or Y_p) during each of said time periods ΔT with the corresponding end point coordinate (X_2 or Y_2) on the axis of greatest velocity, and

- d4. executing said step (e) when said comparing indicates that the said predicted coordinate (X_p or Y_p) lies beyond the end point coordinate (X_2 or Y_2) in the direction of travel along the axis of greatest velocity.

47. The method of moving a member displaceable along X and Y axes at a selected angle in space and through a desired composite distance D at a desired composite velocity V to a desired end point coordinates X_2, Y_2 which comprises

- a. representing by digital signals in a digital processing apparatus the X and Y components I and J of said composite distance,

- b. processing in said apparatus said digital signals to derive first and second sets of signals representing ΔX and ΔY numbers which are respectively equal to:

1. the product of I , a predetermined time-representing number ΔT and the composite velocity V , all divided by the composite distance D , and
2. the product of J , said predetermined time-representing number ΔT , and the composite velocity V all divided by the composite distance D ,

- c. measuring off successive equal periods of ΔT in actual time,

- d. during each such time period utilizing said first and second sets of signals to drive said member simultaneously and substantially uniformly through the distance ΔX along the X axis and through the distance ΔY along the Y axis, and

e. adjusting the said first and second sets of signals in said apparatus to represent modified values of the ΔX and ΔY numbers during at least one of said periods to make the position reached by said member at the end of a given time period ΔT agree closely with the desired end point coordinates X_1, Y_2 .

48. The method of moving a member displaceable along X and Y axes through successive composite distance components D at successive selected angles in space with successive desired composite velocities V which comprises:

- representing numerically by digital electrical signals in a digital processing apparatus the X and Y components I and J of a first composite distance D ,
- processing said signals in said apparatus to derive first and second sets of signals numerically representing ΔX and ΔY distances which are respectively equal to:
 - the products of I , a predetermined time-representing number ΔT , and the composite velocity V , all divided by the first composite distance D , and
 - the product of J , said predetermined time-representing number ΔT , and the composite velocity V , all divided by the first composite distance D ,
- measuring off successive equal time periods of ΔT in actual time,
- during each of said time periods ΔT utilizing said first and second sets of signals to drive said members simultaneously and substantially uniformly through the distance ΔX along the X axis and through the distance ΔY along the Y axis,
- adjusting said first and second sets of signals in said apparatus to represent modified values $\Delta X'$ and $\Delta Y'$ of said ΔX and ΔY distances during at least one of said periods to make the composite movement executed by said member at the end of one of said time periods ΔT agree closely with the desired first composite distance D , and
- during at least one of the last few periods ΔT prior to the complete execution of the composite distance D representing numerically in said apparatus by digital electrical signals the X and Y components I and J of the next composite distance D and processing such latter signals as defined in step (b) above, and immediately after said one time period repeating the steps (c) through (e) defined above to effect execution of movement by the member of said next composite distance D .

49. For use with a numerical director wherein, during each of a series of equal time periods $\Delta T_1, \Delta T_2 \dots \Delta T_n$ digital XCP signals are generated in a digital processing apparatus to represent points $XCP_1, XCP_2, \dots XCP_n$ which are spaced along the X axis ΔX distance apart and which are to be successively reached by a controlled member in order to reach the X axis coordinate $XCEP$ of the desired end point of a path segment at a desired velocity, a method of predicting the particular time period during which said member would go past said end point, comprising the steps of

- producing in said apparatus XCEP signals representing said XCEP coordinate,
- during each of said time periods producing in said apparatus look-ahead signals representing the projected X axis coordinate of the point which said member will be directed to reach during the following M time periods, where M is a preselected integer,
- during each of said time periods comparing in said apparatus said look-ahead signals with said XCEP signals, and
- producing in said apparatus a warning signal during the first time period in which said comparison indicates that the X axis coordinate represented by said look-ahead signals is beyond the X axis coordinate represented by said XCEP signals.

50. The method of claim 49 further characterized by the additional step of terminating movement along said path segment at the end of the time period preceding said particular time period.

51. The method of claim 49 further characterized by the additional steps of

- producing in said apparatus an auxiliary signal during said first time period if the amount by which the X axis coordinate represented by said look-ahead signals is past the X axis coordinate represented by said XCEP signals exceeds approximately one-half the X axis distance ΔX to be traversed by said member during the particular time period in which said overrun is predicted to occur,
- terminating movement along said path segment at the end of the time period preceding said particular time period if both of said warning signal and said auxiliary signal are produced during said first time period, and
- terminating movement along said path segment at the end of said particular time period if only said warning signal is produced during said first time period.

52. The method of claim 49 further characterized by the additional step of generating in said apparatus modified signals XCP' in place of at least one of said XCP signals which normally would occur after said first time period, said modified signals XCP' representing points which are spaced along the X axis $\Delta X'$ distance apart and the last one of which coincides with said XCEP coordinate.

53. The method of claim 51 further characterized by the additional steps of

- generating in said apparatus modified signals XCP' in place of each of M successive ones of said signals in response to the production of both said warning signal and said auxiliary signal during said first time period, said modified XCP signals representing points which are spaced on the X axis by a distance which is greater than ΔX , and
- generating in said apparatus modified XCP signals in place of each of M successive XCP signals in response to the production of only said warning signal during said first time period, said modified XCP signals representing points which are spaced along the X axis by a distance which is less than ΔX .

54. A method of generating a series of numerical command signals for moving a member displaceable along X and Y axes to a target point X_2, Y_2 along a straight line path at a velocity V and at an angle θ to the X axis comprising

- generating time base signals cyclically recurring at, and defining a succession of, equal time periods ΔT ,
- producing in a digital processing apparatus first and second sets of signals respectively representing, in the form of macromove numbers, distances $V \cdot \Delta T \cdot \cos \theta$ and $V \cdot \Delta T \cdot \sin \theta$ through which said member is to move over an elapsed time of ΔT duration along the X and Y axes respectively,
- producing in said apparatus third and fourth sets of signals respectively representing points XCP, YCP along the X and Y axes which said member is to reach,
- periodically upgrading said third and fourth sets of signals by amounts equal to said macromove numbers $V \cdot \Delta T \cdot \sin \theta$ and $V \cdot \Delta T \cdot \cos \theta$ respectively,
- following each upgrading of one of said sets of signals producing in said apparatus signals representing the projected distance which said member will be directed to traverse along a selected one of said X and Y axes during the following M time periods of ΔT duration where M is a predetermined integer other than zero, and
- determining in said apparatus whether or not the position to be reached by the member after it has traversed said projected distance will be beyond its target point for said selected axis.

55. The method of claim 54 further characterized by the additional step of

- in response to a determination that said member would reach its target point along said selected axis within M time periods of ΔT duration, changing said macromove numbers $V \cdot \Delta T \cdot \sin \theta$ and $V \cdot \Delta T \cdot \cos \theta$ for subsequent time periods ΔT so as to cause said upgraded third and fourth

sets of signals in said apparatus to represent coordinate points XCP and YCP which are substantially equal to the target coordinates X2 and Y2 after upgrading during one of the time periods.

56. The method of claim 54 further characterized by the additional steps of

- a. during each time period ΔT producing in said apparatus first and second sets of signals representing the amount, if any, by which said member would be directed past its target point along said selected axis during the following M and M-1/2 time periods,
- b. if it is determined that said member would be directed to travel past its target point along said selected axis within M-1/2 time periods, increasing by equal proportions the signaled macromove numbers $V \cdot \Delta T \cos \theta$ and $V \cdot \Delta T \sin \theta$ to be traveled along said X and Y axes during each of the M-1 time periods ΔT which precede the specific time period ΔT in which said member would travel past its target point, so that said member will actually reach its target points X2, Y2 substantially at the end of the last one of said preceding M-1 time periods ΔT , and
- c. if it is determined that said member would be directed to travel past its target point along said selected axis within M time periods ΔT but not within M-1/2 time periods ΔT , reducing the signaled macromove numbers $V \cdot \Delta T \cos \theta$ and $V \cdot \Delta T \sin \theta$ representing the distances to be traversed by said member during the specific time period ΔT in which the element would otherwise travel past its target point along said selected axis and also during the M-2 time periods ΔT prior thereto by equal proportions so that the element will actually reach its target points x2, Y2 substantially at the end of said specific time period ΔT .

57. In a system for moving a member simultaneously along X and Y axes to produce resultant movement along a desired path segment which extends substantially along or at right angles to a vector which is the sum of X and Y axis components d_x and d_y , the combination comprising

- a. means for digitally signaling the numerical values of the components d_x and d_y ,
- b. a time shared arithmetic computer having a cyclic time base generator repeatedly signaling recurring successive and equal time periods ΔT ,
- c. means for feeding the signals representing d_x and d_y to said computer during at least a portion of some of said periods ΔT along with signals representing a factor of proportionality Q to obtain output signals from the computer representing small, incremental moves ΔX and ΔY which are respectively proportional to d_x and d_y by said factor of proportionality Q , said factor of proportionality Q itself being directly proportional to the desired path velocity and inversely proportional to the length of said vector,
- d. and means responsive to said output signals from said computer for driving said member through the distance ΔX along the X axis during each of said time periods ΔT , and for driving said member through the distance ΔY along the Y axis during each of said time periods ΔT .

58. In a system for moving a member simultaneously along X and Y axes to produce resultant movement along a path segment D having X and Y axis components I and J, the combination comprising

- a. means for digitally signaling the numerical values of the components I and J,
- b. a time shared arithmetic computer having a time base generator repeatedly signaling the lapse of successive, equal time periods ΔT ,
- c. means feeding the signals representing I and J to said computer during at least a portion of some of said time periods together with signals representing a factor of proportionality to obtain output signals from the computer representing small incremental moves ΔX and ΔY which are respectively proportional to I and J, said factor of proportionality Q itself being directly proportional to the

desired path velocity and inversely proportional to the length of said path segment,

- d. and means responsive to said output signals for driving said member during each period ΔT through the distance ΔX along the X axis and the distance ΔY along the Y axis.

59. The combination set forth in claim 58, further characterized in that said means (d) includes

- d1. accumulator registers adapted to receive and digitally store signals representing command coordinate numbers XSC and YSC,
- d2. means controlling said computer to cause the latter once during each time period ΔT to (i) add the number ΔX to the number XSC and store the result in said registers and
- ii. add the number ΔY to said number YSC and store the result in said registers,
- d3. and servo drive means responsive to the XSC and YSC signals to drive said member along the X and Y axes through distances and at velocities proportional to the extents and the average rate of change of the numbers XSC and YSC, respectively.

60. The combination set forth in claim 58, further characterized in that said factor of proportionality Q is equal to $(V \cdot \Delta T)/D$, where V represents a desired velocity along the path segment, D represents the length of the path segment, and ΔT numerically represents the duration of each of said periods ΔT , the said output signals ΔX and ΔY thereby respectively representing the quantities $(V \cdot \Delta T \cdot I)/D$ and $(V \cdot \Delta T \cdot J)/D$.

61. In a numerical control system a director for producing first and second sets of digital command signals each changed at time intervals of ΔT to represent successive designated X and Y coordinates of a controlled member which is to move at a velocity V from a pair of initial coordinates X_i, Y_i along a linear path D having an X component I and a Y component J comprising

- a. a digital computer,
- b. a plurality of digital signal storage means for storing digital signals;
- c. means for storing in respective ones of a first and a second of said signal storage means, signals numerically representing I and J,
- d. means for causing said computer successively to process the signals stored in said first and second storage means and to produce as a result signals numerically representing the quantities

$$\Delta X = I \cdot \frac{V \cdot \Delta T}{D} \text{ and } \Delta Y = J \cdot \frac{V \cdot \Delta T}{D}$$

- e. means for storing the signals representing ΔX and ΔY in respective ones of a third and a fourth of said signal storage means,
- f. said plurality including fifth and sixth storage means for holding first and second command signals numerically representing commanded coordinates XCP and YCP,
- g. means for initially supplying into said fifth and sixth storage means signals which make the stored representation of XCP and YCP equal to the initial coordinates X_i and Y_i , and
- h. means for causing said computer to operate during equal, successive time periods ΔT on the signals stored in said fifth storage means so as to change the coordinate value XCP represented thereby the quantity ΔX and on the signals stored in said sixth storage means so as to change the coordinate value YCP represented thereby by the quantity ΔY .

62. In a numerical control system a director for producing first and second sets of digital command signals changed at periods of ΔT to represent successive designated X and Y coordinates of a controlled member which is to move at a velocity V from a point having coordinates X_0, Y_0 along a linear path D having an X component I and a Y component J comprising

- a. a digital computer,

- b. a plurality of digital signal storage means for storing digital signals,
- c. means for storing, in respective ones of a first and a second of said signal storage means, signals numerically representing I and J ,
- d. means for causing said computer successively to process the signals stored in said first and second storage means and to produce as a result signals numerically representing the quantities

$$\Delta X = I \cdot \frac{V \cdot \Delta T}{D} \text{ and } \Delta Y = J \cdot \frac{V \cdot \Delta T}{D}$$

- e. means for storing the signals representing ΔX and ΔY in respective ones of a third and a fourth of said signal storage means,
- f. means for initially storing in respective ones of a fifth and a sixth of said plurality of storage means signals XSC and YSC representing the coordinates X_0 and Y_0 , and
- g. means for causing said computer to operate at intervals of $\Delta T/N$, where N is a predetermined constant, on the signals XSC stored in said fifth storage means so as to change the coordinate value represented thereby by the quantity $\Delta X/N$, and on the signals YSC stored in said sixth storage means so as to change the coordinate value represented thereby by the quantity $\Delta Y/N$.

63. A numerical control system for moving a controlled member in an X - Y coordinate system at a velocity V from a pair of initial coordinates X_i, Y_i along a linear path D having an X component I and a Y component J comprising

- a. a digital computer,
- b. a plurality of digital signal storage means for storing digital signals,
- c. means for storing in respective ones of a first and a second of said signal storage means signals numerically representing I and J ,
- d. means for causing said computer successively to process the signals stored in said first and second storage means and to produce as a result signals numerically representing the quantities

$$\Delta X = I \cdot \frac{V \cdot \Delta T}{D} \text{ and } \Delta Y = J \cdot \frac{V \cdot \Delta T}{D}$$

where ΔT is a predetermined time interval;

- e. means for storing the signals representing ΔX and ΔY in respective ones of a third and a fourth of said signal storage means,
- f. said plurality including fifth and sixth storage means for holding first and second command signals numerically representing X and Y commanded coordinates XCP and YCP,
- g. means for initially setting said fifth and sixth storage means to make the stored representation of XCP and YCP equal to the initial coordinates X_0 and Y_0 ,
- h. means for feeding to said computer
 - 1. the signals stored in said third and fifth storage means, respectively representing ΔX and XCP, and
 - 2. the signals stored in said fourth and sixth storage means, respectively representing ΔY and YCP,
- i. means for causing said computer to produce in response to the signals from said third and fifth storage means signals representing $XCP_{new} = XCP + \Delta X$ and in response to the signals from said fourth and sixth storage means signals representing $YCP_{new} = YCP + \Delta Y$,
- j. means for feeding said XCP_{new} and YCP_{new} signals from said computer to said fifth and sixth storage means respectively, and
- k. first and second filtering servos respectively connected to respond to said changing XCP and YCP signals and to keep the positions of said controlled member along the X and Y axes substantially equal to the numerical values of XCP_{new} and YCP_{new} .

64. In a numerical control system a director for producing first and second sets of digital command signals each changed

at time intervals of ΔT to represent successive designated X and Y coordinates of a controlled member which is to move at a velocity V from a pair of initial coordinates X_i, Y_i along a linear path D having an X component I and a Y component J comprising

- a. a digital computer
- b. a plurality of digital signal storage means for storing digital signals,
- c. means for storing, in respective ones of a first and a second of said plurality, signals numerically representing I and J ,
- d. means for storing, in a third one of said plurality, signals representing Q which is equal to said velocity V times one of said time intervals ΔT , all divided by said path D ,
- e. means for causing said computer to utilize the signals stored in said first and third storage means to produce signals numerically representing the quantity I times Q , and for storing said I times Q signals in a fourth one of said plurality to represent an X increment ΔX ,
- f. means for causing said computer to utilize the signals stored in said second and third storage means to produce signals numerically representing the quantity J times Q and for storing said J times Q signals in a fifth one of said plurality to represent a Y increment ΔY ,
- g. said plurality including sixth and seventh storage means for holding first and second command signals numerically representing commanded coordinates XCP and YCP,
- h. means for initially supplying into said sixth and seventh storage means signals which make the stored representation of XCP and YCP equal to the initial coordinates X_i and Y_i ,
- i. means for causing said computer to utilize the signals in said fourth storage means to change at time intervals at ΔT the coordinate value XCP represented by the signals stored in said sixth storage means by the quantity ΔX , and
- j. means for causing said computer to utilize the signals in said fifth storage means to change at time intervals of ΔT the coordinate value YCP represented by the signals stored in said sixth storage means by the quantity ΔY .

65. In a numerical control system a director for producing first and second sets of digital command signals changed at periods of ΔT to represent successive designated X and Y coordinates of a controlled member which is to move at a velocity V from a point having coordinates X_0 and Y_0 along a linear path D having an X component I and a Y component J to an end point having coordinates X_1, Y_1 comprising

- a. a digital computer,
- b. a plurality of digital signal storage means,
- c. means for storing digital signals for storing in respective ones of a first and a second of said signal storage means signals numerically representing I and J ,
- d. means for causing said computer successively to process the signals stored in said first and second storage means and to produce as a result signals numerically representing the quantities

$$\Delta X = I \cdot \frac{V \cdot \Delta T}{D} \text{ and } \Delta Y = J \cdot \frac{V \cdot \Delta T}{D}$$

- e. means for storing the signals representing ΔX and ΔY in respective ones of a third and a fourth of said signal storage means,
- f. said plurality including fifth and sixth storage means for holding first and second command signals numerically representing commanded coordinates XCP and YCP,
- g. means for initially supplying into said fifth and sixth storage means signals which make the stored representation of XCP and YCP equal to the coordinates X_0 and Y_0 ,
- h. means for causing said computer to operate over a given period of time at intervals of ΔT on the signals stored in said fifth storage means so as to change the coordinate value XCP represented thereby by the quantity ΔX and on the signals stored in said sixth storage means so as to change the coordinate value YCP represented thereby by the quantity ΔY , and

- i. means for causing said computer to operate over another given period of time at intervals of ΔT on the signals stored in said fifth and sixth storage means to change the coordinate values XCP and YCP represented thereby by quantities $\Delta X'$ and $\Delta Y'$, said quantities $\Delta X'$ and $\Delta Y'$ having values such that after one of said time intervals ΔT during said another given time period the coordinate values of XCP and YCP will closely agree with the coordinates X_1, Y_1 of said end point.

66. In a numerical control system a director for producing first and second sets of digital command signals changed at periods of ΔT to represent successive designated X and Y coordinates of a controlled member which is to move at a velocity V from a point having coordinates X_0 and Y_0 along successive linear paths D , each having an X component I and a Y component J comprising

- a digital computer,
- a plurality of digital signal storage means for storing digital signals,
- means for storing in respective ones of a first and a second of said signal storage means signals numerically representing I and J ,
- means for causing said computer successively to process the signals stored in said first and second storage means and to produce as a result signals numerically representing the quantities

$$\Delta X = I \cdot \frac{V \cdot \Delta T}{D} \text{ and } \Delta Y = J \cdot \frac{V \cdot \Delta T}{D}$$

- means for storing the signals representing ΔX and ΔY in respective ones of a third and a fourth of said signal storage means,
 - said plurality including fifth and sixth storage means for holding first and second command signals numerically representing commanded coordinates XCP and YCP,
 - means for initially supplying into said fifth and sixth storage means signals which make the stored representation of XCP and YCP equal to the coordinates X_0 and Y_0 ,
 - means for causing said computer to utilize the signals stored in said third storage means to change by an amount ΔX at intervals of ΔT the coordinate value XCP represented by the signals stored in said fifth storage means,
67. The director of claim 66 further characterized by means for producing third and fourth sets of command signals XSC and YSC changed at periods of $\Delta T/N$, where N is a predetermined constant, to represent successive designated X and Y coordinates which are intermediate the X and Y coordinates XCP and YCP represented by said first and second command signals, said means for producing third and fourth sets of command signals including
- seventh and eighth storage means for storing digital signals,
 - means for initially storing in respective ones of said seventh and eighth storage means XSC and YSC signals representing the coordinates X_0 and Y_0 ,
 - means for causing said digital computer during each successive period $\Delta T/N$ to process the signals stored in said third storage means to derive therefrom the quantity $\Delta X/N$ and thereafter to operate upon the signals stored in said seventh storage means so as to change the XSC coordinate value represented thereby by the quantity $\Delta X/N$, and
 - means for causing said digital computer during each period $\Delta T/N$ to process the signals stored in said fourth storage means so as to derive therefrom the quantity $\Delta Y/N$ and thereafter to operate upon the signals stored in said eighth storage means so as to change the YSC coordinate value represented thereby by the quantity $\Delta Y/N$,
 - means for causing said computer to utilize the signal stored in said fourth storage means to change by an amount ΔY at intervals of ΔT the coordinate value YCP represented by the signals stored in said sixth storage means,

- means responsive to the coordinate represented by the signals stored in at least one of said fifth and sixth storage means reaching within a predetermined distance of the end of the first of said linear paths D for entering signals numerically representing the I and J values of the next one of said paths D in the respective ones of said first and second storage means, said means (d) through (h) being operative to repeat their defined functions thereafter.

68. The director of claim 66 further characterized by means for predicting the specific time period ΔT during which the command signal XCP will be changed to represent X coordinates which are past a desired X axis end point coordinate XCEP, said means for predicting including

- ninth storage means and means for storing in it the signals representing said end point coordinate XCEP, and
- means for causing said computer to process during each said time period ΔT the signals stored in said third, fifth, and ninth storage means to produce a signal representing the comparative magnitudes of
 - the quantity stored in said ninth storage means, and
 - the sum of the quantity stored in said fifth storage means and M times the quantity stored in said third storage means, where M is a preselected integer greater than zero, and
- means responsive to said signal representing comparative magnitudes for producing a block end warning signal during the first time period in which the initially lesser of the compared quantities becomes the greater.

69. The director of claim 68 further characterized by means for producing, during the same time period ΔT in which said block end warning signal appears, a signal indicative of whether or not the command signals scheduled to be produced during said specific time period ΔT represent an X coordinate which differs from said end point coordinate XCEP by more than one-half ΔX .

70. The director of claim 68 further characterized by means responsive to said block end warning signal for changing, for a predetermined number of time periods ΔT prior to said specific time period ΔT , the signals stored in said third and fourth storage means to represent modified values of $\Delta X'$ and $\Delta Y'$ which, when repetitively added said predetermined number of times to the coordinate quantities represented by the signals XCP stored in said fifth storage means, will cause said last named signals to represent the X and Y axis coordinate values which essentially equal said end point coordinate value XCEP.

71. In a straight line generator for extending a straight line through a distance D at a resultant velocity V during successive time periods ΔT along the Y axis and during successive time periods $\Delta T'$ along the X axis, said time periods ΔT and $\Delta T'$ being of equal duration but symmetrically staggered, the combination comprising

- computing means for performing arithmetic computations,
- means for storing a number I and a number J respectively representing the X and Y component distances of the distance D ,
- means for receiving and storing a number ΔX ,
- means for receiving and storing a number ΔY ,
- first control means for causing said computing means to multiply said number I by a factor of $(V \cdot \Delta T)/D$ and to store the resulting product in said means for receiving and storing the number ΔX ,
- second control means for causing said computing means to multiply said number J by a factor of $V \cdot \Delta T/D$ and to store the resulting product in said means for receiving and storing the number ΔY ,
- means for extending said straight line along the X axis during each of said periods ΔT by an amount represented by said number ΔX , and
- means for extending said straight line along the Y axis during each of said periods $\Delta T'$ by an amount represented by said number ΔY .

72. In a numerical control system a director for producing X and Y axis command signals numerically representing motion along a straight line through a distance D at a resultant velocity V during successive time periods ΔT along the Y axis and during successive time periods $\Delta T'$ along the X axis said time periods ΔT and $\Delta T'$ being of equal duration but symmetrically staggered, the combination comprising

- a. computing means for performing arithmetic computations,
- b. means for storing a number I and a number J respectively representing the X and Y coordinate distances from a first point P_0 to a second point P_e ,
- c. means for receiving and storing a number ΔX and a number ΔY respectively representing desired motion along the X and Y axes during said time periods $\Delta T'$ and ΔT respectively,
- d. first control means for causing said computing means to multiply said number I by a factor of $V\Delta T/D$ and to store the resulting product in said means for receiving and storing the number ΔX ,
- e. a second control means for causing said computing means to multiply said number J by a factor of $V\Delta T/D$ and to store the resulting product in said means for receiving and storing the number ΔY ,
- f. a first updatable storage register and means for entering into it a number X_i representing the X coordinate of said point P_0 ,
- g. a second updatable storage register and means for entering into it a number Y_i representing the Y coordinate of a point P_i on a straight line between said points P_0 and P_e and at a distance of $(V\Delta T)/2$ from said point P_0 ,
- h. third control means for causing said computing means to change the number X_i stored in said first register N times during each of said time periods ΔT , each time by an amount substantially equal to $\Delta X/N$, and
- i. fourth control means for causing said computing means to change the number Y_i stored in said second register N times during each of said time periods $\Delta T'$, each time by an amount substantially equal to $\Delta Y/N$.

73. A numerical control for causing relative motion between two machine members along a circular path, comprising

- a. first and second storage means for storing digital signals,
- b. means including a digital computer for storing in said first storage means a first series of numbers at regularly spaced time intervals ΔT , respective ones of said first series of numbers representing the X axis distances between a first series of equidistant points P_0, P_2, P_4 on said circular path,
- c. means including said digital computer for storing in said second storage means a second series of numbers at regularly spaced time intervals $\Delta T'$ which are staggered symmetrically with but are of the same duration as said time intervals ΔT , respective ones of said second series of numbers representing the Y axis distances between a second series of equidistant points P_1, P_3, P_5 on said circular path staggered symmetrically with but spaced the same distance apart as said points P_0, P_2, P_4 ,
- d. means responsive to the numbers stored in said first storage means for causing relative motion between said machine members corresponding to respective ones of said X axis distances during successive ones of said time intervals ΔT , and
- e. means responsive to the numbers stored in said second storage means for causing relative motion between said machine members corresponding to respective ones of said Y axis distances during successive ones of said time intervals $\Delta T'$.

74. a circle generator comprising

- a. first and second storage means for storing digital signals,
- b. a digital computer,
- c. first control means for causing said computer to calculate and store in said first storage means a first series of num-

bers at regularly spaced time intervals ΔT , respective ones of said first series of numbers representing the X axis distances between a first series of equidistant points P_0, P_2, P_4 on said circle,

- d. second control means for causing said digital computer to calculate and store in said second storage means a second series of numbers at regularly spaced time intervals $\Delta T'$ which are staggered symmetrically with but are of the same duration as said time intervals ΔT , respective ones of said second series of numbers representing the Y axis distances between a second series of equidistant points P_1, P_3, P_5 on said circle staggered symmetrically with but spaced the same distance apart as said points P_0, P_2, P_4 ,
- e. third and fourth storage means respectively for storing numbers representing the X coordinate of said point P_0 and the Y coordinate of said point P_1 ,
- f. means for causing said digital computer successively to change the number stored in said third storage means N times during each of said time intervals ΔT , each time by an amount equal to $1/N$ times the number then stored in said first storage means, and
- g. means for causing said digital computer successively to change the number stored in said fourth storage means N times during each of said time intervals $\Delta T'$ each time by an amount equal to $1/N$ times the number then stored in said second storage means.

75. A numerical control for causing relative motion between two machine members along a circular path, comprising

- a. computing means for performing arithmetic computations,
- b. means for storing a number I representing the X coordinate distance from a first point P_0 of said circular path to the center of said circular path, a number J representing the Y coordinate distance of a second point P_1 on said circular path to the center of said circular path, a number ΔY representing a desired extension of said circular path along the Y axis toward said point P_1 , and a number ΔX representing a desired extension of said circular path along the X axis from the X coordinate of said point P_0 to the X coordinate of a point P_2 on said circular path, said point P_1 being midway between said points P_0 and P_2 ,
- c. first control means for causing said computing means to replace the stored number I with the remainder of said number I less the stored number ΔX , then to multiply said remainder by a factor $-Q$, where Q is the ratio of the distance between said points P_0 and P_2 and the distance between said point P_0 and the center of said circular path, and finally to store the resulting product as a newly stored number ΔY , representing the next desired extension of said circular path along the Y axis, in place of the previously stored number ΔY ,
- d. second control means for causing said computing means to replace the stored number J with the remainder of said number J less the newly stored number ΔY , then to multiply said last remainder by said ratio Q , and finally to store the resulting product as a newly stored number ΔX , representing the next desired extension of said circular path along the X axis, in place of said previously stored number ΔX , each of said first and second control means repeating its assigned series of functions a predetermined time after the other of said control means has carried out its assigned series of functions, and
- e. means for causing relative motion between said machine members along the X and Y axes by amounts corresponding to successive ones of said newly stored ΔX and ΔY numbers.

76. In a circular curve generator the combination comprising

- a. computing means for performing arithmetic computations,
- b. means for storing a number I representing the X coordinate distance from a first point P_0 of said circle to the

- center of said circle, a number J representing the Y coordinate of a second point P_1 on said circle to the center of said circle, a number ΔY representing a desired extension of said curve along the Y axis toward said point P_1 , and a number ΔX representing a desired extension of said curve along the X axis from the X coordinate of said point P_0 to the X coordinate of a point P_2 on said circle, said point P_1 being midway between said points P_0 and P_2 ,
- c. first control means for causing said computing means to replace the stored number I with the remainder of said number I less the stored number ΔX , then to multiply said remainder by a factor $-Q$, where Q is the ratio of the distance between said points P_0 and P_2 and the distance between said point P_0 and the center of said circle, and finally to store the resulting product as a newly stored number ΔY , representing the next desired extension of said curve along the Y axis, in place of the stored number ΔY ,
- d. second control means for causing said computing means to replace the stored number J with the remainder of said number J and the newly stored number ΔY , then to multiply said last remainder by said ratio Q , and finally to store the resulting product as a newly stored number ΔX , representing the next desired extension of said curve along the X axis, in place of said stored number ΔX , each of said first and second control means repeating its assigned series of functions a predetermined time after the other of said control means has carried out its assigned series of functions,
- e. means for storing a pair of numbers respectively representing the X coordinate of said point P_0 , and the Y coordinate of said point P_1 , and
- f. third and fourth control means respectively for causing said computing means to change the stored P_0 X coordinate and the P_1 Y coordinate numbers by increments corresponding to respective successions of said newly stored ΔX and ΔY numbers, each said incremental change being made by said computing means in N substantially equal steps.
77. In a circle generator, the combination comprising
- a. arithmetic computing means for performing arithmetic computations,
- b. means for initially storing a number I_0 representing the X coordinate distance from a first point P_0 on said circle to the center of said circle and a number J_0 representing the Y coordinate distance of said point P_0 to the center of said circle,
- c. storage means for receiving from said computing means and storing a number ΔY representing a distance along the Y axis and a number ΔX representing a distance along the X axis,
- d. first control means for causing said computing means to multiply said initially stored number I_0 by a factor $-Q$, where Q is the ratio of the distance between said point P_0 and a further point P_2 on said circle and the distance between said point P_0 and the center of said circle, and to store the product, representing twice the desired initial extension of said circle along the Y axis, in said storage means as the first number ΔY ,
- e. second control means for causing said computing means to replace the initially stored number J_0 with a new number J_1 which is equal to the remainder of said number J_0 less one-half of said stored first number ΔY , to multiply said sum by said ratio Q and finally to store the product, representing the first desired extension of said circle along the X axis, in said storage means as the first number ΔX ,
- f. third control means for causing said computing means to replace the initially stored number I_0 with a new number I_2 which is equal to the remainder of said number I_0 less the stored first number ΔX , then to multiply said last-named remainder by a factor of $-Q$, and finally to store newly the resulting product, representing the next desired

- extension of said circle along the Y axis in place of the said stored first number ΔY ,
- g. fourth control means for causing said computing means to replace the stored number J_1 with the remainder of said number J_1 less the newly stored number ΔY , then to multiply said last remainder by said ratio Q , and finally to store newly the resulting product, representing the next desired extension of said circle along the X axis, in place of said stored first number ΔX , each of said third and fourth control means repeating its assigned series of functions a predetermined time after the other of said third and fourth control means has carried out its assigned series of functions, and
- h. means for extending said circle by amount corresponding to one-half said first number ΔY and successive ones of said newly stored numbers ΔY along the Y axis, and by amounts corresponding to said first number ΔX and successive ones of said newly stored numbers ΔX along the X axis.
78. A circle generator comprising
- a. first and second storage means for storing digital signals,
- b. a digital computer,
- c. first control means for causing said digital computer to calculate and store in said first storage means a first series of numbers at regularly spaced time intervals ΔT , respective ones of said numbers representing the X coordinate distances between a first series of equidistant points P_0, P_2, P_4 on said circle,
- d. second control means for causing said digital computer to calculate and store in said second storage means a second series of numbers at regularly spaced intervals $\Delta T'$ which are staggered symmetrically with but are of the same duration as said time intervals ΔT , respective ones of said second series of numbers representing the Y coordinate distances between a second series of equidistant points P_1, P_3, P_5 on said circle staggered symmetrically with but spaced the same distance apart as said points P_0, P_2, P_4 ,
- e. third and fourth storage means respectively for storing numbers representing the X coordinate of said point P_0 and the Y coordinate of said point P_1 ,
- f. means for causing said digital computer to change the numbers stored in said third storage means during successive ones of said time intervals ΔT by amounts equal to the numbers stored in said first storage means,
- g. means for causing said digital computer to change the numbers stored in said fourth storage means during successive ones of said intervals $\Delta T'$ by amounts equal to the numbers stored in said second storage means,
- h. fifth storage means for storing a number X_1 representing the X coordinate of a target point on said circle,
- i. control means for causing said computer to compare during each said time interval ΔT the sum of (1) the changed number stored in said third storage means, and (2) an integral multiple of the number stored in said first storage means with (3) the number stored in said fifth storage means, and to produce a signal indicating which of the compared quantities is the lesser; and
- j. means responsive to said signal for producing a block end warning signal during the first time period ΔT in which the lesser of the compared quantities becomes the greater.
79. In a numerical control system a director for producing first and second sets of digital command signals changed at time periods of ΔT to represent successive designated X and Y coordinates of a controlled member which is to move at a velocity V from a point having coordinates X_0, Y_0 along a linear path to an end point having coordinates X_1, Y_1 , the combination comprising
- a. an arithmetic digital computer,
- b. a plurality of digital signal storage means for storing digital signals,
- c. means for storing in respective ones of a first and a second of said signal storage means signals numerically representing quantities ΔX and ΔY .

- d. means for initially storing in respective ones of a third and a fourth of said plurality of storage means signals representing the coordinates X_0, Y_0 ,
- e. means for causing said computer to operate at intervals of ΔT on the signals stored in said third storage means so as to vary the coordinate value represented thereby by the quantity ΔX and on the signals stored in said fourth storage means so as to vary the coordinate values represented thereby by the quantity ΔY , the values represented by the resulting signals in said third and fourth storage means being designated XCP and YCP respectively,
- f. means for comparing the quantities $XCP + M\Delta X$ and $XCP + M\Delta X - \frac{1}{2}\Delta X$ during each of said time periods ΔT with the value of said X axis end point coordinate X_1 and giving a first indication if the first of said quantities has been changed to go through the value of said coordinate X_1 and giving a second indication if the second of said quantities has also been changed to go through the value of said coordinate X_1 ,
- g. means responsive to said first indication to cause said digital computer to replace, during the following time period ΔT , signals stored in said first and second storage means with signals respectively representing the quantities

$$\frac{X_1 - XCP}{M-1} \text{ and } \frac{Y_1 - YCP}{M-1}$$

and

- h. means responsive to said second indication to cause said digital computer to replace during the same time period ΔT in which said second indication is given the signals stored in said first and second storage means with signals representing the quantities

$$\frac{X_1 - XCP}{M-1} \text{ and } \frac{Y_1 - YCP}{M-1}$$

80. A method for reducing the handling capacity required for storage in arithmetic elements of a numerical director used to generate command signals for moving a member displaceable along X and Y axes by continuously computing in the form of macromove numbers the distances which said member is to traverse along the X and Y axes during each of a succession of equal time periods and by further processing the computed macromove numbers to produce successive X and Y axis target points to be reached by said member after successive ones of said time periods comprising

1. measuring off in actual time successive, equal time periods ΔT ,
- a. computing and signaling in digital processing apparatus during each time period to N significant figures the distance to be traversed by said member along one of said axes during the following time period,
- b. adding in said apparatus to the result the last M digits of the distance required for said one axis during the previous time period,
- c. storing in said apparatus the first $N-M$ digits of the sum of said addition in a first register and the last M digits of

said sum in a second register, and

- d. processing in said apparatus only the digits stored in said first register to produce signals representing target points along said one axis, whereby the number of digits required to be processed by said director in the process of computing said target points is reduced by M without causing an error to accumulate.

81. A method of advancing the X and Y axis servo drives of a member driven by said servo drives along the X and Y axes through a selected path comprising

- a. measuring off a first series of time intervals ΔT and a second series of time intervals $\Delta T'$, said intervals $\Delta T'$ being of the same duration as, but starting midway through respective ones of, said intervals ΔT ,
- b. individually computing and signaling, in a digital processing apparatus, for each of said time intervals ΔT the X axis distance to be traversed by said member during said time interval ΔT and for each of said time intervals $\Delta T'$ the Y axis distance to be traversed by said member during said time interval $\Delta T'$,
- c. during each of said time intervals ΔT advancing the X axis servo drive in a plurality of substantially equal increments the sum of which equals the X axis distance computed for a said time interval ΔT , and
- d. during each of said time intervals $\Delta T'$ advancing the Y axis servo drive in a plurality of substantially equal increments the sum of which equals the Y axis distance computed for a said time interval $\Delta T'$.

82. A method of moving a member displaceable along X and Y axes through successive path segments which comprises

- a. representing in a digital processing apparatus a first path segment by a pair of digitally signaled path numbers I and J ,
- b. measuring off a first series of time intervals ΔT and a second series of time intervals $\Delta T'$, said intervals $\Delta T'$ being of the same duration as, but starting midway through respective ones of said intervals ΔT ,
- c. for each of said time intervals ΔT computing in said apparatus from said path numbers I and J the Y coordinate distance to be traversed by said member during said time interval ΔT and for each of said time intervals $\Delta T'$ computing in said apparatus from said path numbers I and J the X coordinate distance to be traversed by said member during said time interval $\Delta T'$,
- d. during each of said intervals, ΔT moving said member along the Y axis in a plurality of substantially equal increments the sum of which equals the Y coordinate distance computed for said time interval ΔT , and during each of said time intervals $\Delta T'$ moving said member along the X axis in a plurality of substantially equal increments the sum of which equals the X coordinate distance computed for said time interval $\Delta T'$, and
- e. during at least one of the last few periods ΔT prior to the complete execution of said first path segment representing in said apparatus a second path segment by a second pair of digitally signaled path numbers I and J and thereafter repeating steps (c) and (d) to cause said member to execute said second path segment.

* * * * *

UNITED STATES PATENT OFFICE
CERTIFICATE OF CORRECTION

Patent No. 3,656,124 Dated April 11, 1972

Inventor(s) John K. McGee Page 1 of 9

It is certified that error appears in the above-identified patent and that said Letters Patent are hereby corrected as shown below:

Column 7, line 16, "distance" is changed to --distances--.

Column 11, line 31, erase "P0".

Column 21, line 22, " $\bar{T}$ " is erased, and --T-- is substituted.

Line 23, "T" is erased, and -- $\bar{T}$ -- is substituted.

Line 43, "Flip-flop" is erased, and --flip-flop-- is substituted.

Column 28, line 69, "in" (second occurrence) is erased.

Column 31, line 47, before "lines" insert -- - -- (a minus symbol).

Column 35, line 2, "fliplops" is erased and --flip-flops-- is inserted.

Line 20, "stored" (first occurrence) is erased, and --an--
is substituted.

Column 41, in Table I, last column, "transferrinJ" is erased and
--transferring-- is substituted.

Column 43, line 55, "50.000" is erased and --50.0000-- is substituted.

UNITED STATES PATENT OFFICE
CERTIFICATE OF CORRECTION

Patent No. 3,656,124 Dated April 11, 1972

Inventor(s) John K. McGee Page 2 of 9

It is certified that error appears in the above-identified patent and that said Letters Patent are hereby corrected as shown below:

Column 47, line 70, " 4919^0G_{01} " is erased, and $--4919 \cdot G_{01}--$ is substituted.

Column 56, line 37, " $XCEP_2$ " is erased, and $--XCEP_2,--$ is substituted.

Column 60, line 7, " $\overline{FAX}$ " is erased, and $--FAX--$ is substituted.

Line 11, " $\overline{FAX}$ " is erased, and $\overline{FAX}--$ is substituted.

Column 61, line 60, after " XCP_1 ", the symbol $-- + --$ is inserted.

Column 62, line 6, after " XCP_n ", $-- +11 --$ is inserted.

Column 63, in Table IV A, at No. 89, "coutn" is erased, and $--count--$ is inserted.

Column 65, line 75, after "these", $--quantities could have been--$ is inserted.

Column 68, line 10, "9the" is erased, and $--9th--$ is substituted.

Column 71, line 16, after "25d" a period is inserted and a new paragraph begins.

Column 72, line 34, " $I_8 0$ " is erased, and $--I_0--$ is substituted.

UNITED STATES PATENT OFFICE
CERTIFICATE OF CORRECTION

Patent No. 3,656,124 Dated April 11, 1972

Inventor(s) John K. McGee Page 3 of 9

It is certified that error appears in the above-identified patent and that said Letters Patent are hereby corrected as shown below:

Column 84, line 52, " $(\Delta D/R)$ " is erased, and $--AD/R--$ is substituted.

Column 86, line 55, " ΔY " is erased, and $--\Delta X--$ is substituted.

Column 87, line 66, "lfor" is erased, and $--l\text{ for}--$ is substituted.

Column 88, line 74, "indicting" is erased, and $--indicating--$ is substituted.

Column 91, line 51, "and" is erased, and $--or--$ is substituted.

Line 54, "and" is erased, and $--or--$ is substituted.

Column 93, line 41, "YCEp" is erased, and $--YCEP--$ is substituted.

Line 46, "is" (second occurrence) is erased, and $--in--$ is substituted.

Column 95, line 35, " $10 + \Delta X$ " is erased, and $--10 (XCP + \Delta X)--$ is substituted.

Column 96, line 72, "53" is erased, and $--63--$ is substituted.

Column 97, line 40, " $(V \cdot \Delta T)$ " is erased, and $--V \cdot \Delta T--$ is substituted.

Same line, " $(V \cdot \Delta T)$ " is erased, and $--V \cdot \Delta T--$ is substituted.

Column 109, line 44, " ' 2" is erased, and $--+2--$ is substituted.

Line 51, " SCP_1 " is erased, and $--XCP_1--$ is substituted.

UNITED STATES PATENT OFFICE
CERTIFICATE OF CORRECTION

Patent No. 3,656,124 Dated April 11, 1972

Inventor(s) John K. McGee Page 3a of 9

It is certified that error appears in the above-identified patent and that said Letters Patent are hereby corrected as shown below:

Column 113, in Table IX, item 33, "R/L" is changed to --R/I--.

Item 35, "RF/FAX" is changed to --P/FAX--.

Item 36, "FAR" is changed to --CAR--.

Item 37, "FAR" is changed to --SAR--.

Item 38, "SX" (second occurrence) is erased.

Item 40, "R/ Δ Y" is changed to --R/ Δ X--.

Item 43, "FAX" is changed to --FAX--.

UNITED STATES PATENT OFFICE
CERTIFICATE OF CORRECTION

Patent No. 3,656,124 Dated April 11, 1972

Inventor(s) John K. McGee Page 4 of 9

It is certified that error appears in the above-identified patent and that said Letters Patent are hereby corrected as shown below:

Column 115, line 67, erase "Xcp" and substitute --XCP--.

Column 120, line 1, before "46", --FIG.-- is inserted.

Column 121, line 2, "XCS" is erased, and --XSC-- is substituted.

Column 124, line 59, "the" is erased, and --them-- is substituted.

Column 125, line 15, "(3." is erased, and --(3).-- is substituted.

Column 126, line 17, "ΔT1000" is erased, and --ΔT000-- is substituted.

Column 127, line 48, "I=" is erased, and --I*-- is substituted.

Line 59, "I=" is erased, and --I*-- is substituted.

Line 63, "I=" is erased, and --I*-- is substituted.

Column 128, line 6, "XCP3" is erased, and --XCP3-- is substituted.

Line 14, "CXEP" is erased, and --XCEP-- is substituted.

Line 26, "I=" is erased, and --I*-- is substituted.

Line 27, "vector from the IBUF" is erased, and --vector from the IBUF-- is substituted.

Line 28, "I=" is erased, and --I*-- is substituted.

UNITED STATES PATENT OFFICE
CERTIFICATE OF CORRECTION

Patent No. 3,656,124 Dated April 11, 1972

Inventor(s) John K. McGee Page 5 of 9

It is certified that error appears in the above-identified patent and that said Letters Patent are hereby corrected as shown below:

Line 33, "I=" is changed to --I*--. Same line, "-XB" is changed to -- -XB --.

Line 40, "I=" is changed to --I*--.

Line 45, "I=" is changed to --I*--.

Line 52, "I=" is changed to --I*--.

Line 55, "+I=" is changed to -- +I* --.

Column 129, line 34, "J=" is changed to --J*--.

Line 45, "*" is changed to --(--.

Line 53, " ΔT_{46} " is changed to -- ΔY_{46} --.

Column 130, line 1, "J=" is changed to --J*--.

Line 14, "J=" is changed to --J*--.

Line 19, "J=" is changed to --J*--.

Line 22, "J=" is changed to --J*--.

Line 61, " $\frac{+}{J} =$ " is changed to -- +J*--.

Line 67, "+J=" is changed to -- +J*--.

Line 70, "J=" is changed to --J* --.

UNITED STATES PATENT OFFICE
CERTIFICATE OF CORRECTION

Patent No. 3,656,124

Dated April 11, 1972

Inventor(s) John K. McGee

Page 6 of 9

It is certified that error appears in the above-identified patent and that said Letters Patent are hereby corrected as shown below:

Column 132, line 53, "ΔT1000" is changed to --ΔT000--.

Column 133, line 47, "+TBUF" is changed to --+I₊BUF--.

Column 134, line 40, "J=" is changed to --J*--.

Line 52, "J=" is changed to --J*--.

Line 67, "J=" is changed to --J*--.

Column 135, line 15, "J=" is changed to --J*--.

Line 23, "J=" is changed to --J*--.

Line 27, "J=" is changed to --J*--.

Line 29, "+J=" is changed to -- +J* --.

Line 34, "J=" is changed to --J*--.

Line 72, "I=" is changed to --I*--.

Column 136, line 3, "I=" is changed to --I*--.

Line 9, "I=" is changed to --I*--.

Line 13, "I=" is changed to --I*--.

Line 46, "I=" is changed to --I*--.

Line 51, "+I=" is changed to --I*--.

UNITED STATES PATENT OFFICE
CERTIFICATE OF CORRECTION

Patent No. 3,656,124 Dated April 11, 1972

Inventor(s) John K. McGee Page 7 of 9

It is certified that error appears in the above-identified patent and that said Letters Patent are hereby corrected as shown below:

Line 57, " $I_{\pm}$ " is changed to -- I_{*} --.

Column 137, line 4, " $J_{\pm}$ " is changed to -- J_{*} --.

Line 12, " $J_{\pm}$ " is changed to -- J_{*} --.

Line 14, " $J_{\pm}$ " is changed to -- J_{*} --.

Column 141, line 52, "iteration" is changed to --transition--.

Column 143, line 61, " d^x " is changed to -- d_x --.

Line 66, " $(V \cdot \Delta T)$ " is changed to -- $V \cdot \Delta T$ --.

Column 144, line 73, " $66Y$ " is changed to -- ΔY --.

Line 75, " $(V \cdot \Delta T)$ " is changed to -- $V \cdot \Delta T$ --.

Column 145, line 47, " $(V \cdot \Delta T)$ " is changed to -- $V \cdot \Delta T$ --.

Line 64, " $V_x \Delta T$ " is changed to -- $V_x \cdot \Delta T$ --.

Column 149, line 3, " $(I \cdot V \cdot \Delta T)$ " is changed to -- $I \cdot V \cdot \Delta T$ --. Same line,

$(J \cdot V \cdot \Delta T)$ is changed to -- $J \cdot V \cdot \Delta T$ --.

UNITED STATES PATENT OFFICE
CERTIFICATE OF CORRECTION

Patent No. 3,656,124 Dated April 11, 1972

Inventor(s) John K. McGee Page 8 of 9

It is certified that error appears in the above-identified patent and that said Letters Patent are hereby corrected as shown below:

Line 48, at the end of the equation $--\sin \theta---$ is inserted.

Line 60, at the end of the equation $--\sin \theta--$ is inserted.

Column 151, line 72, "T5-I6" is changed to $--T5-T6--$.

Column 152, line 63, after "Y", $--axes--$ is inserted.

Column 153, line 52, in the equation insert a minus (-) symbol before $--I_4--$.

Column 155, line 17, "products" is changed to $--product--$.

Column 156, line 27, before "signals" $--XCP--$ is inserted.

Column 158, line 23, "(V·A" is changed to $--V·A--$.

Line 24, "T)/D" is changed to $--T/D--$.

Line 28, "(V·ΔT·I)" is changed to $--V·ΔT·I--$. Same line, "(V·ΔT·J)" is changed to $--V·ΔT·J--$.

Column 159, line 19, "YO" is changed to $--Y0--$.

Column 160, line 49, after "means" $--for storing digital signals--$ is inserted.

Line 50, "digital signals for storing" is erased.

Columns 161 and 162, items "i" and "j" should be erased from claim 67

and should be placed at the end of claim 66.

UNITED STATES PATENT OFFICE
CERTIFICATE OF CORRECTION

Patent No. 3,656,124 Dated April 11, 1972

Inventor(s) John K. McGee Page 9 of 9

It is certified that error appears in the above-identified patent and that said Letters Patent are hereby corrected as shown below:

Column 162, line 63, " $V \cdot \Delta T$ " is changed to $--V \cdot \Delta T--$.

Column 163, line 32, " $V \cdot \Delta T$ " is changed to $-V \cdot \Delta T--$.

Signed and sealed this 10th day of April 1973.

(SEAL)
Attest:

EDWARD M. FLETCHER, JR.
Attesting Officer

ROBERT GOTTSCHALK
Commissioner of Patents