

(19)



Europäisches Patentamt
European Patent Office
Office européen des brevets



(11) Publication number:

0 175 179 B1

(12)

EUROPEAN PATENT SPECIFICATION

- (45) Date of publication of patent specification: **02.01.92** (51) Int. Cl.⁵: **B41B 19/00, B41B 19/01, G09G 1/14**
- (21) Application number: **85110751.6**
- (22) Date of filing: **27.08.85**

(54) **Method and apparatus for generating a set of signals representing a curve.**

(30) Priority: **10.09.84 US 649040**
10.09.84 US 649012

(43) Date of publication of application:
26.03.86 Bulletin 86/13

(45) Publication of the grant of the patent:
02.01.92 Bulletin 92/01

(84) Designated Contracting States:
AT CH DE GB LI

(56) References cited:
WO-A-83/02179
US-A- 4 029 947
US-A- 4 199 815
US-A- 4 331 955

COMPUTER, vol. 17, no. 5, May 1984, pages 40-48, IEEE, Long Beach, California, US; J. FLOWERS: "Digital type manufacture: an interactive approach"

J.D. FOLEY et al.: "Fundamentals of computer graphics", 1982, pages 514-523, Addison-Wesley Publ. Co., Reading, US

(73) Proprietor: **Linotype Company**
425 Oser Avenue
Hauppauge New York 11788(US)

(72) Inventor: **Schrieber, Ricky J.**
60 Nassau Drive
Great Neck New York 11021(US)

(74) Representative: **Schubert, Siegm, Dipl.-Ing. et al**
Patentanwälte Dr. V. Schmied-Kowarzik Dr. P. Weinhold Dr.-Ing. G. Dannenberg Dr. D. Gudel Dipl.-Ing. S. Schubert Dr. P. Barz
Grosse Eschenheimer Strasse 39
W-6000 Frankfurt am Main 1(DE)

Note: Within nine months from the publication of the mention of the grant of the European patent, any person may give notice to the European Patent Office of opposition to the European patent granted. Notice of opposition shall be filed in a written reasoned statement. It shall not be deemed to have been filed until the opposition fee has been paid (Art. 99(1) European patent convention).

DescriptionFIELD OF THE INVENTION

5 This invention relates to the field of encoding and storing data related to a variable size character for access and display, and particularly, to the encoding of display points projected in the shape of a continuous smooth curve. The field of this invention is the field of character and symbol generation with continuous and smooth outline curves.

10 DESCRIPTION OF THE PRIOR ART

The prior art contains many examples of character generating methods and systems. One such example is U.S. Patent 4,029,947 ('947) which generates characters from a single encoded master but uses straight line interpolation to approximate curves between a set of given points. Other encoding systems are shown in U.S. Patent No. 4,298,945 ('945) and U.S. Patent No. 4,199,815 ('815) which also show a system of encoding straight lines using the end points and then interpolating points on the locus of straight lines between the end points and about the outline of a character. A further patent is U.S. Patent No. 4,338,673 ('673), which stores a character in a single size by encoding straight line outlines of the character and then uses that encoding to generate points along straight line approximations of the outline at a desired size, as in the '945 patent and the '815 patent.

However, these and other curve generation techniques do not use a system whereby the coordinate points on the outline of a character are encoded in a single master and then, by utilizing a parametric cubic expression of a single variable, a series of signals describing nodes on the locus of a smooth continuous curve between those points, are generated for display of the character at any variable size.

25 Parametric cubic curves are known and shown in "Fundamentals of Interactive Computer Graphics", J.D. Foley and Andries Van Dam; Addison Wesley, Reading Mass, 1982. Shown therein are parametric cubic curves, as functions of a single variable, used to represent curve surfaces.

Further, T_EX and METAFONT, Donald E. Knuth; American Math Society, Digital Press, Bedford, Massachusetts, 1979, shows the use of a parametric cubic curve wherein, given the coordinate positions of a set of end points and the first derivative (slope) of a curve at the end points, a parametric cubic polynomial (a function of a single parameter t), can be used to generate the locus of points on a smooth continuous curve segment between those end points. As shown by Knuth, the locus of each curve segment depends on the location of the two end points of that segment, for example, Z₁ and Z₂, and the angle of the curve at Z₁, determined by the location of adjacent points Z₀, Z₂ and Z₃. In fitting the curve between two end points, the angles at Z₁ and Z₂ are predetermined by Knuth. For example, where the curve is from Z₀ to Z₁ to Z₂, Knuth assumes, as a rule, the direction the curve takes through Z₁ is the same as the direction of the arc of a circle from Z₀ to Z₁ to Z₂. However, not all curved outlines will satisfy this rule and, in many cases, Knuth requires a further adaptation of that foregoing process to produce the desired shape, namely of a curve representing the actual smooth curved outline of a character.

40 Knuth's process starts using the rule described above specifying a circle when fitting a curve between three given points Z₁, Z₂, Z₃. Knuth does show that a parametric cubic curve as a function of a single parameter t, as also shown in "Fundamentals of Interactive Computer Graphics", can be modified, as shown on page 20 of Knuth, to include "velocities" specified as "r" and "s" and which are functions of the entrance angle and exit angle of the curve segment at the given end points. The velocities' values determine how the resulting curve will vary as a function of t, either slowly describing a longer curve distance or more quickly describing a shorter curve distance. The r and s formulas are arbitrary and, in the case shown by Knuth, are chosen to provide excellent approximations to circles and ellipses when θ equals ϕ and $\theta + \phi$ equals 90 degrees. Additional properties chosen to be satisfied by Knuth's arbitrary formulas for r and s are further described in T_EX and METAFONT.

50 In addition to Knuth's internal requirement to specify an entrance and exit angle at the curve end points, where no previous angle or curve history is given, Knuth also must use the circle approximation rule, specified above, for each of the end points on the curve. This means that where the actual curve to be generated is not accurately produced by the Knuth circle approximation, an adjustment must be made. Knuth makes this adjustment by manipulating adjoining points. For example, where the curve is to be fit between points Z₁, Z₂ and Z₃. Knuth may manipulate the locations of Z₀ or Z₁ or Z₃ or Z₄ to obtain an accurate fit. A further problem in Knuth occurs when the sign of θ is the same as the sign of ϕ implying a) the curve entrance angle at point Z₁ is in the same quadrant as the curve exit angle at Z₂, b) a sine wave between Z₁ and Z₂ and c) an inflection point. Knuth must resort to a manipulation of point locations to

obtain a desired smooth curve around the inflection point.

In summary, the processes described above for producing outlines and showing the use of a parametric cubic curve expressed as a polynomial capable of generating a series of points along the locus of a curve between two given points can only do so with the disadvantages noted above.

Further, a method according to the first portion of claim 1 is known (COMPUTER, vol. 17, no. 5, May 1984, pages 40 - 48, IEEE, Long Beach, California, US), which method comprises the further steps of

- a. for said curve locus, defining the location and successive order of said knots thereon, and encoding as data on an encodable medium, signals representative of and corresponding to said knots,
- b. compiling data in an established compiler in accordance with a cubic parametric polynomial relationship and said knots at the said end points of said first curve segment and the locus of said first curve segment and
- c. applying said signals indicative of said locations of said first and second knots of said first curve segment to said compiler.

This prior art method differs from the method of the present invention in the same manner as is the case with the Knuth method. Since the known method likewise uses the circular arc method for calculation of the tangents (angles) at the knots. The method does not calculate or employ interknot angles or average interknot angles.

In the following summary of the invention, the method and system for using a parametric cubic curve which accurately reproduces a curve section between given points is shown and described and which method and system avoids the disadvantages discussed above.

SUMMARY OF THE INVENTION

This invention has as its object the generation, encoding and display of a series of points (nodes) along the locus of a curve segment between two given end points which are defined as "knots". It has been specifically developed to generate the locus of points (nodes) along the smooth curved outline of characters or symbols between given knots (i.e., Z_{a-1} , Z_a , Z_{a+1} . . . Z_{n-1} , Z_n), but it should be understood that its use is not necessarily limited to that purpose but may be used to generate and display smooth continuous curves between any end points, regardless of the application.

In the case of the applicant's preferred embodiment, the knots are coordinate points along the outline of a character, which may be an alphanumeric or any other character or symbol and which coordinate points may be representative of a master size encoded character at a normalized size, on a dimensionless normalized encoding grid. The coordinate points may be decoded for display at a predetermined normalized display size or at an expanded or reduced size. The knots may be along any outline or surface whether two or three dimensional, although the applicant's preferred embodiment is shown in a two-dimensional model for use in smooth continuous curve generation. Using the method and system of applicant, the nodes along a two-dimensional curve or three-dimensional surface locus may be generated. The system shown herein generates the nodes using the encoded knot positions and the slopes of the curve at the knots. The process generates signals representing the nodes' coordinate values and is based upon the length (Z) between the knots and the angular interrelationship (B) of each of the knots with respect to a reference angle on the encoding grid.

In the preferred embodiment, the knots' coordinates are encoded in a closed data loop and represent dimensionless coordinates about the closed outline loop of the character or symbol. The method described herein enters the closed data loop at a set of encoded coordinates representing a knot on the closed outline loop and initiates the node generating process by using the angular interrelationships of the knots about that entry knot. However, other methods may be used to initiate the process without departing from the inventive concept. In applicant's preferred embodiment, once having entered the encoded closed data loop, the analysis may proceed in a clockwise or counterclockwise direction around the closed data loop.

It should be understood that the principles of this invention are not limited to a closed outline loop or closed data loop, but may be applied to open outline loops and to open data loops.

The process may be used to generate either a smooth continuous curve between the knots or some other form of curve, forming a cusp at a knot for example, and which although continuous, it is not smooth at all locations but angular, as in the shape of a "K" or "G". In the preferred embodiment of the invention, the selection of a smooth continuous curve or merely a continuous curve, having a cusp for example, is made using stored rules which serve as default command codes which use the interknot angles formed between the knots, and the distance between the knots to produce a desired result. For example, where the angle formed by two knots is in excess of a predetermined threshold angle, a default command code may direct that a cusp be formed. Other rules such as override control codes may be similarly used to force a

cusps or smooth curve, regardless of the threshold conditions and which may be encoded in the closed data loop to override the default command codes as explained below.

Assuming, for the purpose of explaining the invention, it is desired to connect all knots in the closed outline loop with smooth continuous curves, the angles formed between an entrance knot at which the representative encoded data loop is entered and the first and second successive knots thereto in a chosen progression of knots in the closed outline loop are averaged to produce average angle values. As the curve formed between knots is a continuation of a curve which enters at the first of the knots (Z_a), and which exists at successive or at a second of the knots (Z_{a+1}), the tangent angles of the curve, which are the entrance angle at the first knot and exit angle at the second knot can be specified as average angles. The analysis continues by proceeding around the closed outline loop in the established progression of knots and the order thereof of the knots and examining the angles formed between the knots as described above.

As the knots form a master skeletal outline of the character or symbol when juxtaposed on a dimensionless encoding grid, arranged at a normalized size, the knots may be rotated or scaled relative to such a normalized encoding grid. Once the master encoded character or symbol (master encoded character) is scaled and positioned as desired, then the analysis described above, may continue wherein a) the encoded closed data loop of knots in the closed outline loop is entered at a first set of coordinates representing a first knot, b) the angular and spacial relationships of the knots on the normalized encoding grid are determined, using the representative encoded data and c) for each set of knots along the closed outline loop, assuming that each knot is to be connected by a continuation of a smooth curve, the average angles of the respective curve segments entering and exiting the respective knots are determined. Where a smooth continuous curve is desired, then the entrance angle of the curve segment at a knot would normally be the same as the exit angle of the curve from that same knot.

By using the foregoing method of analysis, namely, determining the average angles made by the curve passing through the knots, and using those average angles to represent the slope of the tangents to the particular curve segment at the respective knots, the disadvantages shown in the prior art are overcome.

In particular, in this inventive concept, there is no need to define a circle between the knots and then compute the locus of nodes on the circle as Knuth does. A faster process results using this invention which requires only the interknot angles of the knots relative to each other and to a reference angle be known. The process as stated above, uses a parametric cubic polynomial relationship of a single variable t to generate signals or data, indicative of and representing the coordinates of nodes on the locus of a smooth continuous curve segment between sets of respective knots. Where a greater resolution and greater number of nodes is needed to describe the locus, then the incremental value of the parameter t , may be decreased providing a greater number of discrete cumulative values of t and node coordinates. Where less resolution and fewer nodes are needed to define the locus, then the incremental value of t can be enlarged, producing fewer discrete cumulative values of t and nodes.

In the preferred embodiment, the dimensionless normalized master encoding grid represents an EM Square (M^2) of 864 by 864 Dimensionless Resolution Units (DRU's) in each of the X, Y axes. The M^2 is a measure used in the typesetting trade wherein a character is set within an M^2 and is shown herein in a manner consistent with the application of the inventive principles in the preferred embodiment. When the master encoded character is to be displayed, the master encoded character, at a normalized size on the normalized encoding grid is scaled, the knot coordinate positions at the scaled size are encoded and expressed in the appropriate display intercept units such as Raster Resolution Units (RRU's). These scaled encoded knots in the preferred embodiment are expressed in terms of the display RRU's and used to determine the incremental value of t . The incremental value of t is used to generate discrete cumulative values of t which are then applied to the cubic parametric polynomial to generate the node coordinates. Then through the cubic parametric polynomial relationship, signals indicative of the coordinates of the nodes are generated and stored as data. As defined by a parametric expression of a single parameter t , the resultant node's x, y coordinate values (X or Y are the axial directions in the preferred embodiment), vary separately as separate functions of t [i.e. $x(t)$, $y(t)$]. At the scaled size, the incremental value of t may be related to the reciprocal of the distance between knots expressed in RRU's (i.e. $Z_d = |Z_n - Z_{n-1}|$) or any other suitable method may be used to produce a value of t . Values representing the node coordinates are then generated using these incremental values of t , each value being added, to the previous cumulative values (with the second incremental value of t being added to the first incremental value of t and the third incremental value of t being added to the previous cumulative value of t and so on). In the preferred embodiment, the value of t is set to vary from 0 to 1.

The resulting series of signals, stored as encoded data, represent knots and nodes which define the locus of the smooth continuous curve between the knots and the outline of the character or symbol is a machine part ultimately used to control or modulate a display to form the desired character or symbol at the

desired size, in a visual image. The resultant data may be run length data, which is applied directly to a raster beam, to position the beam and energize the beam accordingly, or may be used to control a free running raster. Interpolation, rounding or truncating of value may be used to locate the nodes on the display intercepts corresponding to the raster line locations, where exact coincidence is lacking.

5 As previously stated, as characters or symbols are not always smooth curves but may contain cusps, a threshold test may be used, such as one based on the exterior angle formed by lines between knots. Where that exterior angle at a knot is greater than a predetermined threshold angle, for example, a cusp may be assumed. It being understood, however, that if the exterior angle is less than the threshold angle, the analysis previously described with regard for producing a locus of nodes to define a smooth continuous
10 curves would be used.

In the preferred embodiment, the master encoding grid is a Cartesian coordinate system. As the preferred embodiment is used in typesetting, the encoding grid relative to which the character is encoded is set within an M_2 , which in the preferred embodiment contains 864 by 864 Dimensionless Resolution Units (DRU's). The master encoded character, at its normal size is set over a portion of the available normalized
15 encoding grid area of the M^2 . As is known in the typesetting field, expansion areas in the M^2 are also provided for large size characters. The character may be scaled, rotated or projected by ordinary known techniques and new coordinates for the knots may be determined accordingly, as is known in the art. In the preferred embodiment, the scaling is done in increments of $1/1024$ DRU's. The new coordinate locations of the knots for the character at its scaled, rotated or projected positions are determined. As only integers are
20 used in the preferred embodiment, any fraction or equivalent thereof is discarded. In the process, the percent of reduction or enlargement is first calculated in relation to a desired character size in units of typesetter's points. The precision of the scaling is increased by an autoscaling linear interpolation increasing the resolution of the linear interpolation, as explained below. The result is a scaled coordinate point in RRU's without the need to utilize floating point arithmetic.

25 In the preferred embodiment and as stated above, override control codes are accessed responsive to data stored with the stored knot coordinates to reduce storage and the processing time.

The code 0 is used to indicate the end of all the loops.

A code 1 is used to indicate the movement in a relatively long direction on an axis, for example, the X axis. In this case, an X value is replaced with a new X coordinate value.

30 The code 2 indicates the same process as a code 1 for another axis, for example, the Y axis direction, where the Y value is replaced with a new Y coordinate value.

A code 3, as in codes 1 and 2, indicates X and Y are both replaced with new coordinate values.

A code 4 indicates the finish of a previous encoded loop and the start of a new loop.

Codes 5, 6 and 7 indicate that the X,Y or XY directions are respectively altered.

35 Codes 8 to 11 are editing commands forcing predetermined conditions for the curve at the respective knots as will be described.

The knots may be encoded in the closed data loop on a 4-bit memory boundary (nibble) and in the preferred embodiment, the first nibble value of a complete information set of nibbles is used to specify the number of nibbles used in the complete information set.

40 Additionally, the data is packed in a novel manner which can be interpreted as spacial information or control codes, as will be explained.

In summary, the inventive concept is a process and system for transforming a machine part, in the form of signals, encoded as data and representing a pattern of knots on the outline of a master size symbol, into a similar pattern at a reduced or enlarged size or transposed in space, by generating a series of encoded
45 data signals representing nodes which more definitely define the said pattern in the shape of smooth continuous curves or cusps and which data signals may be directly used to control a display process to visually display the pattern.

Accordingly, what is disclosed is a method of creating a data base on a storage medium in a computing device including the steps of:

- 50 (i) generating a series of signals representing nodes on a locus of a curve partially defined by a set of related knots, encoded as data on an encodable medium, with said knots defining end points of respective curve segments of said curve locus and with said knots being in a successive order in relation to said locus, and
- (ii) encoding said node signals as data stored on said storage medium for use when representing said
55 curve segments in a separate additional curve generating process responsive to the shape of said curve locus, as represented by said encoded node signals.

This method further comprises the steps of:

- (a) defining the locations and the successive order of said knots on said curve locus and encoding as

data on said storage medium, signals indicative of said knots,

(b) for a first knot (Z_a) representing a first end point of a first curve segment (21), deriving a first angle (α) measured with respect to an arbitrary reference line and indicative of the average of two interknot angles (B_{a-1} and B_a), wherein (Z_{a-1}) is the knot immediately preceding the first knot (Z_a), the interknot angle (B_{a-1}) is the angle formed between the reference line and a straight line connecting the said two knots (Z_a and Z_{a-1}), and wherein (Z_{a+1}) is the knot immediately succeeding knot (Z_a) in said successive order and inter-knot angle (B_a) is the angle formed between the reference line and a straight line connecting said knots (Z_a and Z_{a+1}), and encoding, as data on said storage medium, signals indicative of said first angle (α),

(c) at a second of said knots (Z_{a+1}), representing a second end point of said first curve segment (21), establishing a second angle (B_{a+1}) for said first curve segment, and encoding as data in said computing device, signals indicative of said second angle,

(d) establishing a compiler in a memory of a computing device, which compiler generates locus points for compiling data as a function of:

- i. a predetermined cubic parametric polynomial function of a parameter "t" incremented in steps;
- ii. said knots; and
- iii. said angles at said end points of said first curve segments;

(e) establishing a limited range "R" of values for said parameter "t" in the compiler,

(f) in said memory of said computing device, applying said signals indicative of said locations of said first and second knots of said first curve segment, to said compiler,

(g) in said memory of said computing device, applying said signals indicative of said first and second angles of said first curve segment to said compiler,

(h) generating a signal indicative of a distinct selected value of said parameter "t" within said range "R" by incremental signal generation, and applying said signal to said compiler to derive a signal indicative of a respective node location on said first curve segment according to said predetermined cubic parametric polynomial function,

(i) repeating step (h) by applying signals indicative of additional distinct selected values of said parameter "t", within said range "R", to derive a plurality of signals indicative of respective node locations on said locus of said first curve segment for respective distinct selected values of said parameter "t", and

(j) encoding said signals derived in steps (h) and (i), in said data base in said computing device memory for later sorting to represent said first curve segment for later generation of a visual representation of said curve by using said data signals directly to control a display process to visually display the curve.

Further disclosed is a method of encoding data and an encoded data system representing knots on an outline defined relative to a coordinate plane involving selecting sets of coordinates on said outline, to represent the knots, establishing a successive order of said knots, encoding said knots in a data order indicative of said knot order, and encoding by encoding a complete information set of data providing a control code indicative of either i) the coordinate locations of said knots or ii) a knot's direction relative to others of said knots or iii) a predetermined shape of said outline between a pair of said knots or iv) data indicative of the shape of said outline at a knot or v) encoding a complete information set providing the coordinate distances between adjacent knots.

Further disclosed is a method and system for encoding data representing knots on an outline defined relative to a coordinate plane and for generating a series of signals representing nodes on the locus of a curve partially defined by said set of knots, involving selecting sets of coordinates on said outline, to represent said knots, establishing a successive order of said knots, encoding said knots in a data order indicative of said knot order, and encoding a complete information set of data providing a code indicative of a predetermined shape of said outline between a pair of said knots or a complete information set providing the coordinate distances between adjacent knots.

Further disclosed is a method and system for generating a series of signals representing nodes on a locus of a curve partially defined by a set of related knots, encoded as data, with said knots defining the end points of respective segments of said curve locus and with said knots being in a successive order in relation to said locus, and for encoding and decoding said node signals as data and for use of said data when in an imaging process responsive to the shape for said curve segments as represented by said encoded data, involving defining the locations and the successive order of said knots on said curve locus and encoding as data, signals indicative of said knots, for a first knot, (Z_a), representing a first end point of a first curve segment, deriving a first angle, indicative of the average of the interknot angles between said first knot (Z_a), and selected related knots and encoding as data, signals indicative of said first angle, at a second of said knots (Z_b), representing a second end point of said first curve segment, establishing a second angle

for said first curve segment, and encoding as data, signals indicative of said second angle, establishing a compiler for compiling data according to a cubic parametric polynomial relationship between a parameter "t", said knots and angles at the said end points of a said curve segment and the locus of a said curve segment, establishing a range "R" of values for said parameter "t", applying said signals indicative of the
 5 said locations of said first and second knots of said first curve segment, to said compiler, applying said signals indicative of the said first and second angles of said first curve segment to said compiler, applying a signal indicative of a distinct selected value of said parameter "t" within said range "R", to said compiler to derive a signal indicative of a respective node location on said first curve segment, repeating the above by applying signals indicative of additional distinct selected values of said parameter "t", within said range "R",
 10 to derive a plurality of signals indicative of respective node locations on said locus of said first curve segment for respective distinct selected values of said parameter "t", encoding said signals derived above in a data base to represent said first curve segment accessing said data base signals, and controlling an imaging means responsive to said accessed signals to reproduce said curve.

Further disclosed is a method and system for linear interpolation of coordinate points between first and
 15 second end points, to produce coordinates on a straight line outline and where said coordinates are located on a coordinate system having a first coordinate direction and second coordinate direction and encoded in machine readable data words a radix "r", corresponding to the order of values for designated positions in said data words, involving encoding a first data word of "N" positions corresponding to the distance between the said first and second end points in the said first coordinate direction and placing said first data
 20 word into a first machine location, encoding a second data word of "M" bits corresponding to the distance between said first and second end points in said second coordinate direction and placing said second data word into a second machine location, determining the number of available positions, between the most significant position of said first data word and the most significant position of said first machine location, available for shifting said first data word in a first direction of the most significant positions of said first
 25 machine location, shifting said first data word by a maximum number of positions, equal to the said number of available positions in said first direction and the number of positions corresponding to the number of significant positions used to encode said second data word, and increasing the scale of said first data word by a scale factor related to the number of said positions shifted, deriving a third data word indicative of said second data word in said second machine location divided into said first data word shifted according to step
 30 d), encoding data words indicative of the coordinate of said straight line in said second coordinate direction, for respective ones of said data words encoded according to step f) encoding a multiple of said third data word, which are related to a respective coordinate in said first coordinate direction, on said straight line, reducing the scale of said multiples of said third data words above, to the scale of the first data word established prior to the above said shifting and encoding said third data words produced above with
 35 respective coordinates in said second coordinate direction to produce said coordinates on said straight line.

Further disclosed is encoding data representing knots on an outline defined relative to a coordinate plane and for decoding said encoding data for use in a display process to produce images of said outlines represented by said encoded data by selecting sets of coordinates on said outline, to represent said knots, establishing a successive order of said knots, encoding said knots in a data order indicative of said knot
 40 order, the encoding including encoding a complete information set of data providing a control code indicative of either i) the coordinate locations of said knots or ii) a knot's direction relative to others of said knots or iii) a predetermined shape of said outline between a pair of said knots or iv) data indicative of the shape of said outline at a knot, or v) providing data indicative of the coordinate distances between adjacent knots, decoding said complete information sets in a decoding order related to said data order responsive to
 45 said complete information set being indicative of the coordinate distances between adjacent knots, producing an image of a smooth continuous curved outline or a straight line between said adjacent knots or, responsive to said complete information set being indicative of a said control code, as set forth in i), ii), iii), or iv), producing an image of a smooth continuous outline or a straight line according to the said coordinate locations of said knots relative to adjacent knots in said successive knot order or producing an image of
 50 said outline being smooth at respective knots or being sharp and forming cusps at respective knots.

Further disclosed is encoding data representing knots on an outline loop defined relative to a coordinate plane, for producing a display image of said outline and decoding responsive to the interrelationship of said knots on said outline loop, and imaging said outline loop responsive to said decoded data involving selecting sets of coordinates on said outline loop, to represent said knots, establishing a successive order of
 55 said knots, encoding said knots in a data order indicative of said knot order, encoding a complete information set of i) data indicative of the coordinate distances and interknot angles between adjacent knots, comparing the relative positions of successive knots to at least a first interknot criterion, responsive to said step d) of comparing, i) producing a first indication that a set of said successive knots is within said

criterion, or ii) producing a second indication that a set of said successive knots is outside said criterion, and i) responsive to said first indication imaging said outline loop in the form of a smooth continuous curve, or ii) responsive to said second indication, imaging said outline loop in the form of a straight line, between said set of successive knots.

Further disclosed is encoding and accessing a single data set of solution values functionally related to and representing the solution set to at least two domains of a variable, involving defining a single data set of solution values functionally related to a first domain of a variable and to a second domain of a variable, arranging said data set of solution values in an order related to said first domain and said second domain, accessing said data set of solution values relative to respective values in said first domain to derive at least a part of said solution set to said first domain, and accessing said data set of solution values relative to respective values of said second domain to derive at least a part of said solution set to at least said second domain.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1a shows a set of knots (i.e. Z_a , Z_{a+1} , Z_{a+2} , . . .) forming the skeletal outline of a master encoded character.

FIG. 1b shows the closed outline loops with reference to a character.

FIG. 1c shows curve segments Z_{a-1} , Z_a , Z_a , Z_{a+1} , and curve Z_{a+1} , Z_{a+2} and the relationships of the deviance angles θ and ϕ , at knot Z_a and Z_{a+1} .

FIG. 1d and FIG. 1e shows in greater detail the deviance angles of the curve segment Z_a , Z_{a+1} , at the respective entrance and exit knots, as functions of the interknot angle (B) and the entrance and exit angles, and .

FIG. 2 shows the relationships of the knots, tangent angles and angles and to a curve, for the purpose of explaining the Hermite interpolator.

FIG. 3 shows a master encoding grid as may be used in the preferred embodiment, at a normalized size, for encoding a character or symbol such as the character G shown in FIG. 3, as a master encoded character, at a normalized size.

FIG. 4 shows the manner a look-up table may be arranged so access in one direction would provide a value equal to T_2 , as explained herein, and in the opposite direction, a value of T_3 , as explained herein.

FIG. 5 shows the angular relationships between a set of knots Z_a , and Z_{a+1} when a sharp knot or cusp is to be produced.

FIG. 6a shows the angular relationships at a set of knots when a knot between a straight line and a curve is desired to be smooth.

FIG. 6b shows the angular relationships at a smooth knot, and between a curve line and a straight line.

FIG. 7a shows the angular relationships at a knot between a curve line and straight line, forced by a control code to override a default command code, which otherwise would produce the results shown in FIG. 7b.

FIG. 8 shows the angular relationships about a set of knots Z_a and Z_{a+1} when it is desired that a smooth continuous curve pass between the knots.

FIG. 9 shows a character as may be produced according to the principles of the invention.

FIG. 10 illustrates in block diagram a general type of system with which the method of this invention can be practiced

DESCRIPTION OF THE PREFERRED EMBODIMENT

As stated in the summary, the object of this invention is to generate a series of display signals, representing nodes describing the locus of a smooth continuous curve at a desired size, from a normalized curve encoded as a set of encoded knots. Normalized is used in its general and ordinary meaning to denote a norm or standard size. However, as would be apparent to those skilled in the art, the displayed curve, described in terms of display intercept coordinates (RRU's) is dependent upon the resolution of the display, and the rate or relationship between the number of raster resolution units (RRU's) at a given display resolution to the dimensionless resolution units (DRU's) in which the curve is encoded at its normalized size and on the normalized grid. The normalized encoded curve may also be thought of as a master which is encoded at a master size relative to the master encoding grid which may have any suitable coordinate system and which after scaling, may be used to generate the encoded data representing display intercepts in raster units, at a given raster resolution for any desired display size character. As the inventive concepts disclosed herein are used in the typesetting industry to produce typeset composition containing characters

having curved outlines in conformance to the highest graphic standards, scaling starts with a determination of the desired size of the character in any chosen system of measurement. In the preferred embodiment, character size is expressed in printer's points (351.282 micrometers/point or 0.01383 inches/point). It being understood, however, that the invention can be used in connection with other units of measurements and with applications outside the printing or typesetting industry, without deviating from or changing the inventive concepts shown herein. The invention, as described herein is with reference to the printing industry, where the master encoding grid is made synonymous with a dimensionless encoding grid in the form of a typesetting M^2 . This is the application of the preferred embodiment of the invention and discloses the best mode of using the invention.

The problem solved by this invention may be best considered by viewing FIG. 1a which shows a series of data encoded knots (Z_{a-1} , Z_a , Z_{a+1} , Z_{a+2} . . .) on a dimensionless encoding grid having coordinates in the X and Y axial directions, with the X direction coinciding with a zero degree (0°) reference angle. It should be understood, however, that any system of coordinates can be used with any reference angle chosen without changing the manner in which the inventive concepts are used.

As shown in FIG. 1a, a number of knots Z_{a-1} , Z_a , Z_{a+1} , Z_{a+2} , Z_{a+3} inclusive to Z_{a-n} represent an outline loop which may be smooth and continuous over a series of such knots or continuous without being smooth over a series of such knots or a combination of the foregoing. As stated, the knots represent the skeletal outline of a predetermined normalized or master size character or symbol (hereinafter referred to as master size character), on a juxtaposed dimensionless encoding grid where the coordinates have dimensions of Dimensionless Resolution Units (DRU's). In the case of the encoded character, the master size is with reference to the normalized encoding grid and the area within the grid. The interknot angles between the knots, with reference to the reference angle, are shown as generally denoted by "B" and particularly as B_{a-1} , B_a , B_{a+1} and so on for the interknot angles at respective knots Z_{a-1} , Z_a , Z_{a+1} and so on. FIG. 1a shows a skeletal outline as may be typically used for symbol wherein all the knots Z_{a-1} through Z_{a-n} on the symbol outline are arranged on a closed outline loop. This relationship in its simplest form could be shown by the knots for the outline of an O or D, encoded with two such closed outline loops, for the exterior closed outline loops 11 and 15 and one for the interior closed outline loops 13 and 17 respectively, as shown in FIG. 1b.

As shown in FIG. 1a, a direction of progression of the curve outline is chosen with reference to an defined by the order of the knots around the outline. That progression of the curve locus and order of knots is shown by numeral 19 in FIG. 1a. The chosen order of knots defining the said progression, (i.e. Z_{a-1} , Z_a , Z_{a+2} Z_{a-n}) establishes an outline loop of knots. That outline loop may be a closed outline loop which ends upon itself, as shown in FIG. 1b, by closed outline loops 11, 13, 15 and 17. As will be explained below, a compiler functioning according to a parametric cubic polynomial is used to generate signals indicative of nodes which are the coordinates of locations on the locus of a smooth continuous curve between the knots. The order of the knots, defines an outline loop and the respective order of the nodes by their locations relative to the knots and to each other on the outline loop, as would be apparent to one skilled in the art. As explained below, the data representative or indicative of these knots and nodes are encoded in a data order indicative of the order of the knots and nodes on the outline loop. This data order establishes a data loop. As further explained below, the data loop may be designed to close upon itself so that the ending data location for the data loop is the starting data location where data was accessed therefrom in the encoding process and forming a closed data loop corresponding to the closed outline loop.

As stated above, the knots Z_a , Z_{a+1} , etc., may be encoded in the Cartesian coordinate system as X-Y points, using as a reference the normalized encoding grid, or say be encoded in any other system of coordinate points. The outline between the knots is not encoded initially as it will be represented by a generated series of nodes and which will represent the smooth continuous curve locus of the outline, according to the principle of the invention. The display intercept values for the nodes on the curve locus, at a predetermined display size, are related to the encoded knots on the dimensionless encoding grid shown in FIG. 1a by a parametric cubic polynomial relationship. Since the invention is used in a two-dimensional system, the parametric representation represents the curve locus of such nodes $Z(X,Y)$ independently as a third order polynomial function of a parameter "t" which is independent of the encoding grid coordinates. In the preferred embodiment, the parametric cubic polynomial is shown in a Hermite form, it being understood that those skilled in the art can use other forms for defining the polynomial such as the Bezier form, defined in "Fundamentals of Interactive Computer Graphics", referred to in the foregoing, and to which the improvements of this invention are applicable.

The parametric representation of a curve is one for which X and Y are represented as a third order (cubic) polynomial relationship of a parameter "t" where:

$$1.1 \ x(t) = a_x t^3 + b_x t^2 + c_x t + d$$

$$1.2 \ y(t) = a_y t^3 + b_y t^2 + c_y t + d$$

5 The Hermite representation of the parametric cubic polynomial uses the coordinate positions, of the knots, and the tangent angles at the knots, such as Z_a , Z_{a+1} , etc. The Bezier representation uses the positions of the curve's end points and two other points to define, indirectly, the tangents at the curve's end points. The improvements of this invention shown herein are applicable to any representation by the parametric cubic polynomial which uses the position of the curve's end points and the tangent angles of the
10 curve at the end points, directly or indirectly. For sake of explanation, only the Hermite form will be discussed.

As shown in FIG. 2, given end points P_1 , P_4 and the respective tangent vectors R_1 , R_4 at the two end points P_1 , P_4 and along the smooth continuous curve segment 18, a cubic parametric polynomial relationship between a parameter "t" and the locus of a curve segment 18, between a pair of end points, P_1 , and P_4 , may be represented as the following relationships below.
15

$$2.1 \ x(t) = P_{1x} + (3t^2 - 2t^3)(P_{4x} - P_{1x}) + t(t-1)^2 R_{1x} + t^2(t-1) R_{4x}$$

$$2.2 \ y(t) = P_{1y} + (3t^2 - 2t^3)(P_{4y} - P_{1y}) + t(t-1)^2 R_{1y} + t^2(t-1) R_{4y}$$

20 where (P_{1x}, P_{1y}) (P_{4x}, P_{4y}) are the coordinate values at P_1 and P_4 respectively, and (R_{1x}, R_{1y}) and (R_{4x}, R_{4y}) are the tangent values at P_1 and P_4 respectively, with respect to a straight line between P_1 and P_4 thereafter called entrance and exit angles, respectively.

Cubic curves as a minimum are used, as no lower order representation of curve segments can provide
25 continuity of position and slope at the end points where the curve segments meet, and at the same time can assure that the ends of the curve segment pass through specific points.

The derivation of the Hermite parametric cubic polynomials are shown in "Fundamentals of Interactive Computer Graphics" and the manner of using such parametric cubic polynomials to define the points along a curve are further discussed in $T_E X$ and METAFONT referred to in the foregoing.

30 The Hermite form of the parametric cubic polynomial is as stated is a series of curves as shown by Knuth in Chapter 2 in $T_E X$ and METAFONT and which is written in Euler notation as

$$3.1 \ Z(t) = Z_a + (3t^2 - 2t^3)(Z_{a+1} - Z_a) + rt(1-t)^2 S_1 - st^2(1-t) S_2$$

$$3.2 \ S_1 = e^{j\theta} (Z_{a+1} - Z_a); S_2 = e^{-j\theta} (Z_{a+1} - Z_a); 0 \leq t \leq 1$$

and, where r and s are positive real numbers. Equations 3.1, 3.2 define a curve having directions represented by the deviance angles θ and ϕ at Z_a , Z_{a+1} respectively.

The relationship shown in 3.1, 3.2 may be encoded into a compiler designed to process input data
40 related to the knot locations, the angles of the curve segment at its end points, and then for separate selected values of a parameter "t" to produce signals indicative of node locations on the locus of the curve segment. The compiler is shown at the end of the specification.

As stated above, the Hermite form of the parametric cubic polynomial is used in the preferred embodiment of the compiler but the principles of the invention can be used with any other cubic
45 polynomial using the directions of the curve at end points of the curve and the end point locations.

Assuming a curve direction for a given knot order from knot to knot, and in particular, from knot Z_a to knot Z_{a+1} , with knot Z_a being the entrance knot for curve segment Z_a , Z_{a+1} (hereafter curve segments will be defined by the respective segment and knots such as "curve Z_a Z_{a+1} ") and knot Z_{a+1} being the exit knot for curve Z_a , Z_{a+1} , then θ is the angular direction of the curve Z_a , Z_{a+1} at entrance knot Z_a , and ϕ is
50 the direction of curve Z_a , Z_{a+1} at exit knot Z_{a+1} . In the preferred embodiment, the interknot angle (B) made by the straight line from knot Z_a and Z_{a+1} , is defined in terms of a reference angle given for the normalized encoding grid. Also, the angles $\mathfrak{A}$ and $\mathfrak{B}$ of the curve are related to that same reference angle. Therefore, the interknot angle expressed below as B and the angles of the curve at the entrance knot, defined below as $\mathfrak{A}$ and the angle of the curve at the exit knot, defined below as $\mathfrak{B}$, are all defined with regard to a reference
55 angle.

As further explained below, according to the inventive principles, deviance angles θ and ϕ , shown used in equation 3.1 and 3.2, are the curve entrance and exit angles $\mathfrak{A}$, $\mathfrak{B}$, respectively, defined with regard to the interknot angles B and applied to the compiler in the form of the parametric cubic polynomial, as shown in

equation 3.1.

Angles θ and ϕ correspond to angles $\mathfrak{a}$ and $\mathfrak{g}$ respectively. The difference between θ and $\mathfrak{a}$ is that θ is measured using the line connecting knots Z_a and Z_{a+1} as the reference line whereas $\mathfrak{a}$ is measured using a different reference line, shown as "0° REFERENCE" in Figure 1c. ϕ and $\mathfrak{g}$ are related in a similar manner.

The angle θ is measured between the line tangent to Z_a and the reference line connecting Z_a and Z_{a+1} . The angle $\mathfrak{a}$ is measured between the line tangent to Z_a and the 0° reference line. The angle ϕ is measured between the line tangent to Z_{a+1} and the reference line connecting Z_a and Z_{a+1} . The angle $\mathfrak{g}$ is measured between the line tangent to Z_{a+1} and the 0° reference line.

In explaining the invention, $\mathfrak{a}$ and $\mathfrak{g}$ are used to identify the entrance angle of a curve segment at a first knot end point and the exit angle made by that same curve segment at a successive second knot end point, in the knot order. The first and second knots define the entrance and exit knots of the curve segment, with regard to the order of knots and the outline loop, as explained below. However, $\mathfrak{a}$ and $\mathfrak{g}$ are defined relative to a reference angle "q" on the master encoding coordinate grid. θ and ϕ are the entrance angle and exit angle in the cubic parametric polynomial of 3.1 and 3.2, defined relative to an interknot angle B between the entrance and exit knots (i.e. in the preferred embodiment, $\theta = \mathfrak{a} - B$; $\phi = B - \mathfrak{g}$). In Fig 1c, $\mathfrak{a}$ equals $\mathfrak{g}$ so that the curve exit angle is also designated $\mathfrak{a}$. The preferred embodiment uses a process of defining the entrance angle $\mathfrak{a}$ and exit angle $\mathfrak{g}$ in terms of the interknot angles on the outline loop, and with respect to a reference angle and uses the definition of the entrance and exit angles shown as θ and ϕ when applying the angles $\mathfrak{a}$ and $\mathfrak{g}$, at the respective entrance and exit knots, to the compiler, as based on and as required by the derivation of the cubic parametric polynomial, described herein. However, the angles θ and ϕ could be derived directly from the interknot angles B or precomputed from the relationship of the outline loop knots and accessed directly without first deriving $\mathfrak{a}$ and $\mathfrak{g}$ and without deviating from the principles of the invention.

The parameter "t" of 3.1 is allowed to vary over a defined range as further explained below, and for each discrete selected value of t, a discrete node on the locus of the curve between the entrance and exit knots and defined by the parametric cubic polynomial is generated by the compiler. The nodes would be the coordinate location of points on the locus of the curve described by equation 3.1 for each selected value of "t" and where knots Z_a and Z_{a+1} , and the angle of the curve at those knots, given as $\mathfrak{a}$ and $\mathfrak{g}$, derived above, was specified. The derivation of $\mathfrak{a}$, B, $\mathfrak{g}$, θ , and ϕ of "t" and their relationship according to the principles of the invention are further explained and shown below.

It should be noted, however, that the form of the Hermite interpolator used in the preferred embodiment imposes an opposite sign for ϕ than that shown in T_EX and METAFONT shown in Knuth.

It should also be noted that the deviance angles θ and ϕ representing the divergence between the straight line angle interknot angle (B), as used in the cubic parametric polynomial of 3.1, 3.2.

"r" and "s" affect the curve velocities, as described below and the length of the curve between its respective end points (i.e., curve Z_a , Z_{a+1} between Z_a , Z_{a+1}). "r" and "s" are velocities at Z_a , Z_{a+1} respectively, a large velocity value meaning the curve direction changes slowly while small values indicate the curve undergoing more pronounced directional changes [small values of r and s will then have less influence on the value of x(t) or y(t)]. The velocities, r and s, in T_EX and METAFONT are represented as

$$4.1 \quad r = \left| \frac{2 \cdot \sin \theta}{(1 + |\cos \theta|) \sin \theta} \right|$$

$$4.2 \quad s = \left| \frac{2 \cdot \sin \phi}{(1 + |\cos \phi|) \sin \phi} \right|$$

$$4.3 \quad \psi = (\theta + \phi) / 2$$

Considering the effect of r and s on the curve locus defined by the nodes, produced through the cubic polynomial relationship, the values of r and s may be limited to control the direction of the curve from the entrance knot to the exit knot. In the preferred embodiment, r and s are limited to the values of 0.1 to 4.0.

However, these values of r and s could be changed without departing from the principles of the invention.

As discussed in the foregoing, the system shown therein for using the relationship of the points along the smooth curve in T_EX and METAFONT have disadvantages which are eliminated by this invention which will become apparent by reading the following explanation.

As explained above, in using the parametric cubic polynomial relationship to define the nodes along a smooth curve, T_EX and METAFONT start with a circle approximation rule and then must make adjustments to produce a smooth continuous curve between given points. In this invention, those disadvantages are eliminated by the invention described in the following.

In the preferred embodiment, the sign of ϕ is reversed relative to the angle rotation used in T_EX and METAFONT.

The principles of this invention as applied to the preferred embodiment, may now be particularly seen with reference to FIG. 1a. As stated above, a skeletal outline is described by a progression of a series of knots in the order of $Z_a, Z_{a+1} \dots Z_n, Z_{a-1}$, which defines a closed outline loop about the outline of the character or symbol. The knots are encoded as coordinate points in a closed data loop, representing the closed outline loop. As stated, the object of the invention is to produce a series of nodes representing display coordinates on the locus of a curve between each of the knots, the curve being smooth and continuous. However, as will be shown, the principles of the invention can be modified so a series of points describing straight lines between knots can be generated and smooth continuous curves can be generated between knots which are interspersed with straight lines. In addition, cusps can be formed between smooth continuous curve and/or straight lines.

The smooth continuous curve is produced through the parametric cubic polynomial relationship described above. In using the invention, the following principles are applied. The data loop may be entered at the data values representing a knot, Z_a for example, and the angular relationship of the knot (Z_a) referenced to a preceeding knot (Z_{a-1}) and to a succeeding knot in the loop analyzed. The analysis may then proceed in a clockwise direction around the loop which may be considered in the forward direction. However, as stated, the analysis can proceed in a counter-clockwise or backward direction with the direction being used to denote or indicate other values, such as color, or other characteristics as may be needed. As will be understood, the directions used herein are chosen for explanation and do not limit the inventive principles.

FIG. 1a represents a series of knots $Z_a, Z_{a+1}, Z_{a+2} \dots Z_{a+n}, Z_{a-1}$ arranged in a closed outline loop, encoded relative to a master encoding grid and partially defining, in skeletal form, the points on a closed outline loop, such as outline 11, 13, 15, or 17, shown in FIG. 1b. The interknot angles, the angles from knot to knot (i.e. from knot Z_{a-1} to knot Z_a) are denoted generally by the letter "B" with a subscript reference indicating that the angle is formed by a straight line from a respective knot, B_{a-1} for example, to a successive knot B_a in the outline loop. The terms "preceding" and "successive" can be used with reference to the defined order of knots in the outline loop, i.e. either clockwise or counterclockwise, as the case may be. In FIG. 1a, the knot angles B are shown as B_a, B_{a+1}, B_{a+2} , and so on for respective knots.

The manner of applying the cubic parametric polynomial compiler in the preferred embodiment to define nodes between the knots along the locus of a smooth curve utilizes the deviance angle θ at the curve entrance knot and the deviance angle ϕ at the curve exit knot. The deviance angles θ and ϕ may also be referred to as entrance and exit angles.

According to the principles of the invention, and assuming a smooth and continuous curve segment is desired to be developed between a first knot Z_a and a second knot Z_{a+1} , for the curve Z_a, Z_{a+1} , shown as numeral 21, in FIG. 1c, then the inventive principles may be used to generate a series of signals representing the display coordinates for the nodes along the locus of the smooth continuous curve 21 as follows and similarly for the curves between others of respective pairs of knots. The data encoding of the knots representing each of the closed outline loops may also be thought of as a loop, or closed data loop, indicative of and representing the physical arrangement of the knots in the closed outline loop about the character or symbol outline. As stated, the outline loop and the data loop therefore, may be thought of as an order of knots in the loop path and related to a predetermined loop direction for such order (clockwise or counter-clockwise, for example). Within that progression, and for the respective outline loop direction, it can be easily seen that each knot (i.e. Z_a) represents an exit point for the curve segment of the outline loop in the knot order shown by the direction of arrow 20, from a preceding loop knot (i.e. Z_{a-1}) and at the same time, the entrance point for the curve segment from that knot (i.e. Z_a) to a successive knot (i.e. Z_{a+1}). The nodes, generated as explained below are given an order in the outline loop related to the order of knots on the outline loop and the encoding for the nodes is similarly arranged on the data loop in the order of the knots.

In explaining the invention, the convention used for identifying a curve part of the outline loop between

knots is to refer to it by its knot end points, (i.e. curve part 21 is curve Z_a, Z_{a+1}). Similarly, a convention is used to identify a curve entrance and exit angle as explained below.

The entrance angle α for curve Z_a, Z_{a+1} (between knots Z_a and Z_{a+1}), according to our convention, would be the angle represented by the first derivative of the curve Z_a, Z_{a+1} at knot Z_a and the exit angle ϑ would be the angle represented by the first derivative of the curve segment at the next successive knot Z_{a+1} .

As would be understood, where a smooth continuous curve is desired to pass through a knot, (i.e. Z_a) then the entrance angle α , at a knot would be the same as the exit angle ϑ at that same knot for the preceding curve segment Z_{a-1}, Z_a .

In explaining the principles of the invention, the entrance angles α and exit angles ϑ for curve segments starting and ending at knots are defined for the respective knots according to the following convention. The curve segments may be thought of as having an entrance angle α at the knot where the outline loop, in the defined progressive order of knots, enters a respective curve segment (i.e. curve Z_a, Z_{a+1}) forming an entrance angle α_a at that entering knot, and an exit angle ϑ at the next successive knot in the loop where the loop exits curve segment Z_a, Z_{a+1} forming exit angle ϑ_a at knot Z_{a+1} . In the convention chosen, the entrance and exit angles (α_a, ϑ_a) for a curve segment (Z_a, Z_{a+1}) of the loop are referenced to the entrance knot (Z_a). The entrance angle, α_a , and the exit angle, ϑ_a , therefore have a subscript reference to the entrance knot Z_a , but refer to the curve segment of the loop and the angle that curve segment makes at a first knot Z_a where the loop enters the curve Z_a, Z_{a+1} and the angle at a successive knot (Z_{a+1}) relative to the outline loop direction where it exits the curve Z_a, Z_{a+1} . The entrance angle, α_a , is then the angle the loop makes as it passes through knot Z_a , enters curve Z_a, Z_{a+1} , and continues on to knot Z_{a+1} . The exit angle, ϑ_a , is the angle the loop makes as it passes through knot Z_{a+1} , exits curve Z_a, Z_{a+1} , enters the next successive curve Z_{a+1}, Z_{a+2} and continues on to the next subsequent exit knot, Z_{a+2} . In summary, the entrance angle, α , and exit angle, ϑ , according to the convention chosen to explain the principles of this invention, refer to a curve segment of a loop outline between two knots such as Z_a and Z_{a+1} , the entrance angle α_a being the tangent angle or first derivative the outline loop makes as it enters curve Z_a, Z_{a+1} at preceding knot end point Z_a in the outline loop direction, and the exit angle, ϑ_a , the tangent or first derivative of the outline loop it makes as it exits that curve Z_a, Z_{a+1} at knot end point Z_{a+1} and enters the next successive curve Z_{a+1}, Z_{a+2} , in the outline loop direction. As can be seen, the exit angle, ϑ_a , for the curve Z_a, Z_{a+1} is the same angle as the entrance angle α_{a+1} , for the next successive curve Z_{a+1}, Z_{a+2} , in the outline loop direction. Similarly, the entrance angle, α_a is the exit angle, ϑ_{a-1} for the preceding curve Z_{a-1}, Z_a . It will be understood by those skilled in the art that these conventions can be changed without changing the principles of the invention.

Whereas Knuth uses the rule that a smooth continuous curve locus between points, such as from Z_{a-1} to Z_a to Z_{a+1} , must take the direction of a circle, the invention herein avoids that rule and the problems created thereby by using an average of the interknot angles (B) relative to a knot to define the respective knot entrance and exit angles, α and ϑ respectively, and thereby θ and ϕ respectively.

For example, in FIG. 1c, for the curve Z_a, Z_{a+1} , shown as numeral 21, the entrance angle α_a at knot Z_a of the curve Z_a, Z_{a+1} , for example, is related to the average of the interknot angles B_{a-1} (from the preceding knot Z_{a-1} to knot Z_a) and B_a (from Z_a to the succeeding knot Z_{a+1}). That average angle of the outline loop at any knot Z is α ; so expressed as α when referring to the entrance angle at a knot and into a curve segment and ϑ when referring to the exit angle from a knot and out of a curve segment. In the case of knot Z_a , the average entrance angle would be α_a for the curve Z_a, Z_{a+1} (shown by numeral 21), proceeding from Z_a to Z_{a+1} in the order shown by arrow 20 and ϑ_{a-1} for the average exit angle for the curve Z_{a-1}, Z_a (shown by numeral 23), proceeding from Z_{a-1} to Z_a in the order shown by arrow 20.

The angles α or ϑ are always referred to as entrance and exit angles respectively herein, but may or may not be average angles or not depending on the application of each, as explained herein.

The invention is explained with reference to the example of curve Z_a, Z_{a+1} , wherein Z_a is the entrance knot and Z_{a+1} is the exit knot, it being understood that the same analysis would apply to other curve segments either adjoining or further remote from curve Z_a, Z_{a+1} and having respective entrance and exit knots.

In using the compiler to generate the nodes along a smooth continuous curve, the encoded data loop representing the knots along the closed outline loop may be entered at any set of data representing any knot Z , for example, designated to be the entrance knot and the average angle determined by proceeding in the chosen loop direction using the interangular relationships of a knot to its related knots in the loop, to determine the average angles α and ϑ at such related knots. In the preferred embodiment, these related knots are adjacent knots to the next successive knot to the entrance knot.

In the preferred embodiment, the inventive principles of generating the nodes on the smooth continuous

curve locus, starts with a first average angle, determined for the knot coordinate successive to the data loop entrance knot, and which is represented by the data residing at the location in the closed data loop where that data loop was entered. The interknot angle for the entrance knot is then saved and used at the completion of the loop analysis to determine the average entrance and exit angles at that entrance knot when the outline loop analysis reaches that entrance knot in the closed outline loop knot. Where the data loop entrance knot is Z_{a-1} , its average angle would be determined using the preceding interknot angles B_{a-2} and B_{a-1} for related knots Z_{a-1} and Z_{a-2} .

Following the above, the average angle ϕ_a and exit angle ψ_{a-1} at knot Z_a is the average of B_{a-1} and B_a and expressed as

$$\phi_a = \frac{B_{a-1} + B_a}{2} = \psi_{a-1}$$

The average exit angle ψ_a and entrance angle ϕ_{a+1} at knot Z_{a+1} is the average of B_a and B_{a+1} and expressed as

$$\psi_a = \frac{B_a + B_{a+1}}{2} = \phi_{a+1}$$

As shown, the average entrance angle ϕ for any curve segment is the average of the interknot angles at the respective curve segment knot and at related knots, (i.e. the preceding knot in the outline loop) and the average exit angle ψ is the average of the interknot angles at the respective curve segment exit knot and at related knots (i.e. the next successive knot in the outline loop), as shown above.

The parametric cubic polynomial compiler may then be employed to relate the entrance and exit angles ϕ , ψ for the respective entrance and exit knots to a series of nodes describing the locus of a smooth curve between the respective entrance and exit knots (i.e. Z_a , Z_{a+1}). For the example above, the knots are Z_a , Z_{a+1} and the respective entrance and exit angles are ϕ_a and ψ_a for curve Z_a , Z_{a+1} .

The curve velocities r and s are related to the deviance angles θ and ϕ , as set forth above and as used in equation 4.1, 4.2, 4.3. As stated in the foregoing, θ is the deviance between the average entrance angle, for example, ϕ_a for curve Z_a , Z_{a+1} , at an entrance knot Z_a and the interknot angle B_a at that entrance knot Z_a . Similarly, ϕ is the deviance between the average exit angle, for example, at the exit knot, Z_{a+1} , for the curve Z_a , Z_{a+1} and the interknot angle B_a at that entrance knot Z_a .

The relationships between ϕ_a , B_a , ψ_a , θ_a and ϕ_a may be more clearly seen in FIGS. 1d and 1e for knots Z_a and Z_{a+1} respectively.

As a reminder, it should be remembered that the angles, θ and ϕ are the tangents to the locus of the curve represented by the nodes generated using the cubic parametric polynomial compiler with respect to the respective interknot angles.

The coordinate values of Z_a , Z_{a+1} , used through the parametric cubic polynomial compiler to generate the node values are the scaled intercept values given in RRU's, derived from the master encoded values of the knots, in DRUs. For example, assuming an encoding grid shown as an M^2 in FIG. 3 and defined as 864 X 864 DRU's between master encoding grid (x,y) coordinates 592, 736; 592, 1600; 1456, 1600 and 1456, 448 (wherein x,y points 592, 736 are defined as relative 0,0, relative to an x, y offset of 592, 736 respectively) and wherein the encoding grid is defined as 2047 by 2047 DRU's.

In the preferred embodiment, the universe of the master encoding grid comprising the M^2 is 32768 X 32768 DRU's and the offset which positions the origin of the M^2 can be positioned anywhere in that universe. In the preferred embodiment, the x and y offsets and shown in FIG. 3.

The x,y coordinates of the intercept locations for knots Z_a at the displayed size in RRU's may be derived prior to generating the nodes by scaling, using the following conversion factor (CF), for the preferred embodiment. Where:

"P" in the desired display size in points per M^2 (Pt/ M^2),

"Res" is the ratio of micrometers per point (uM/Pt) and serves to convert the display size from points to metric units;

"RRU/MM" is the display resolution in Raster Resolution Units per Micrometer;

M^2 /DRU is the inverse of the encoding grid resolution, in DRUs per M^2 ;

and

$$(P) \cdot (\text{Res}) \cdot (\text{RRU/MM}) \cdot (M^2/\text{DRU}) = (CF) \cdot (\text{RRU/DRU})$$

For Master Coordinate $Z_n(X_n, Y_n)$,

$(X_n \text{ DRUs}) (CF_x \text{ RRUs/DRU}) = X_n \text{ RRUs}$, and

5 $(Y_n \text{ DRUs}) (CF_y \text{ RRUs/DRU}) = Y_n \text{ RRUs}$

The value of the parameter t may be derived from the respective curve segment interknot distance Z_d
 $= |Z_{n+1} - Z_n|$ given after scaling in RRUs, as described below. The generated coordinates for the nodes
 are expressed in RRU's permitting run length encoding of the display outline coordinates. Where Z_d is a
 non-integer, the fractional value may be discarded or rounded with redundant values eliminated.

10 In the preferred embodiment of this invention, the incremental value of t is related to the inverted value
 of the absolute difference Z_d between the entrance and the exit knots,

$$\text{i.e. } t_{\text{inc}} = 1/Z_d = 1/|Z_{n+1} - Z_n|.$$

15

Z_d may be easily derived from the axial difference in the X direction, [i.e. $Z_d = [(X_{n+1} - X_n)/\text{Cosine}(B_n)]$,
 or the axial difference in the Y direction, [$Z_d = [(Y_{n+1} - Y_n)/\text{Sine}(B_n)]$]; Where $X_n, Y_n, X_{n+1}, Y_{n+1}$ are
 successive knots defining end points for a curve segment of the closed outline loop and B_n is the interknot
 angle there between).

20 In practice, it is better to use the larger axial value in the X or Y direction, to minimize error. In
 implementing the invention, the preferred embodiment limits increments of t to $1/1024$. Discrete values of
 $T_1 = 3t^2 - 2t^3$, $T_2 = t^2(1-t)$ and $T_3 = t(1-t)^2$ (see equation 3.1, 3.2) may be stored in a look-up table for
 1024 values of t in the range $1/1024$ to $1023/1024$.

In practice t is stored in the range of $1023/1024$ because the knot end point coordinate corresponding to
 25 $t = 1024/1024$ or 1 is saved, avoiding the need to calculate that knot end point and providing a more
 reliable result.

Further, because of the relationship of T_1 , T_2 and T_3 , a look-up table as shown in FIG. 4, having values
 of T_2 starting with $t=0$ and ending with a value of T_2 for $t=1$, may be accessed in reverse order for values
 of T_3 , as shown in FIG. 4.

30 The values, shown stored in a look-up table in FIG. 4, are for discrete values of t in discrete steps of
 $1/1024$ (i.e. $0/1024$ to $1023/1024$). However, t is shown being inclusive from 0 to 1 for the purpose of
 explanation, it being understood that in the preferred embodiment, varies between 0 and $1023/1024$ as the
 end point for T_2 $1024/1024$ is known.

Because the functions domain of the variable t for the solution sets $T_2 = t(1-t)^2$ and $T_3 = t^2(1-t)$
 35 overlap, a table of values may be accessed in one data direction with respect to a first domain of variables
 for which the value represents a solution set. Similarly, that solution set may be accessed in an opposite
 data direction for a second domain of a variable.

Also, because the function $|1 + \cos\psi| \sin\psi$ is symmetric about 45 degrees, it is only necessary to
 store values thereof for 0 to 45 degrees in half angle steps.

40 The solution set of values for the functions T_2 and T_3 may be accessed in a first data direction to
 provide the functional solution set values for the domain $t = 0$ to $t = 1$ and in reverse order for the domain
 $t = 1$ to $t = 0$. Values of Sine may be stored in half angle steps between 0 and 90 degrees. Further, any
 derived nodes may be stored and used again where the respective curve or corresponding symbol outline
 loop is to be duplicated.

45 By forming a cumulative value of t as a multiple of the incremental value of t and applying cumulative
 discrete selected values of t within the parametric cubic polynomial compiler, according to equation 3.1,
 display intercepts represented by the generated nodes are produced which may then be used as the
 display coordinates for the closed loop outline at the desired display size.

As stated above, the value of θ and ϕ used in the compiler are derived, according to the principles of
 50 the invention from the average entrance and exit angles, α and ϕ at a respective set of entrance and exit
 knots. The nodes which are generated using the incremental value of t to increment the cumulative value of
 t and by applying discrete selected cumulative incremented t values to the cubic parametric polynomial
 compiler to produce respective node coordinates for such discrete selected values of t .

The nodes, represent locations on the curve, which, as stated above, form an angle at the entrance knot
 55 of the curve as given by θ , and forms an angle at the exit knot given by ϕ . As stated, an order is chosen for
 the knots and the order of knots defines an outline loop of knots, the nodes being locations on the curve,
 between the knots also have an order defined by that knot order and by the defined outline loop. As would
 be understood by those skilled in the art, that order is the succession of knots and nodes encountered as

one progresses along the outline loop as specified. The data for the knots and nodes, as stated above, is encoded in a data loop in that same order of knots and nodes. The encoded knots and nodes are placed in the data loop, in that order corresponding to the outline loop. Then, as one progresses in the defined order along the data loop, corresponding to the outline loop, one would encounter the encoding for the knots and nodes in the data loop in an order, corresponding to the order one would encounter the respective knots and nodes on the outline loop represented by that said encoding.

In summary, given the knot locations, and the angles of the curve at an entrance and exit knot as defined above, and given an incremental value of t , then each discrete selected cumulative value of t , as described above, applied to the cubic polynomial parametric compiler would produce respective nodes, indicative of locations along the smooth continuous curve described by the cubic parametric polynomial. The encoding for the knots and nodes would be in the same order along a data loop as one would encounter the corresponding knots and nodes along the outline loop, when proceeding in a chosen order along the outline loop.

The outline loop, when closing upon itself as shown in FIG. 1b, for example, is called a closed outline loop. The data when encoded in a data loop which closes upon itself corresponding to the closed outline loop is called a closed data loop. In this way, accessing the closed data loop, in a direction corresponding to the chosen direction along the closed outline loop will return one to the initial or starting data loop entrance location. As described below, the data loop decoding is not complete until a determination is made that the accessed location for the closed data loop is starting location point and that accessing of all the encoded data in the closed data loop has been completed.

Of course, as one skilled in the art would realize, variations of the above closed outline loop and closed data loop can be made without departing from the principles of the invention.

These node coordinate values given in Raster Units may then be used in run length or other suitable data forms to modulate a display to produce the curve at the desired size, as is well known.

The compiler according to the foregoing is shown below.

ENCODING

The novel manner of encoding the data pack, according to the principles of the invention, is shown in the Table I below, wherein p is the value of dx , q is the value of dy and $0, 1, 2, \dots D, E, F$, are bit positions in the encoded data pack. In the preferred embodiment, the bit positions are valued according to the binary number system. The Control Codes, explained below, are shown in Table II. The encoding, shown for a maximum 12 binary words of 3 nibbles, in the preferred embodiment, is used to define a control code or coordinate position for the master encoding grid. It should be understood that the size of the word and the number of nibbles used, as described below would increase to accommodate a larger universe, commensurate with a larger size master encoding grid.

The binary words are encoded in a series of nibble length data words, arranged in the order of the data loop. In this way, as will be shown below, the values of selected data words with a nibble series may be used to define the number of nibbles in a complete information set and the initial nibble or bit positions in the next successive complete information set.

A complete information set (CIS) would be the set of data words necessary to completely define the next coordinate or a control code, as shown below.

TABLE I

Case Indi- cated by a 1st nibble	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	Case	Limits	
value of																			
1 - 3													0	0	q1	q0	.I.a	p=0 q=[1..3]	
0														0	0	0	.I.b	p=0 q=[0,4..18]	
4 - 7									q3	q2	q1	q0	0	1	q1	q0	.II.	p=[1..8] q=[1..8]	
8 - B							q3	q2	q1	q0	p3	p2	p1	p0	1	0	q4	p4	.III. p=[1..32] q=[1..32]
C - F		p6	q6	q5	q4	q3	q2	q1	q0	p3	p2	p1	p0	1	1	p5	p4	.IV. p=[1..128] q=[1..128]	

TABLE II

Case	Definition	Override Control Code	Total No. of Data Words Needed incl.	Purpose
Ib	end last	0	2	end of last loop
Ia	horizontal(x)	1	1+3 =4	horizontal motion
Ia	vertical(y)	2	1+3 =4	vertical motion
Ia	diagonal(x,y)	3	1+3+3 =7	diagonal motion
Ic	start loop	4	2	end/start new loop
Ib	x negate	5	2	invert x direction
Ib	y negate	6	2	invert y direction
Ib	xy negate	7	2	invert x & y direction
Ib	line sharp	8	2	linear interpolate sharp knot
Ib	line smooth	9	2	linear interpolate smooth knot
Ib	sharp	10	2	curve interpolate sharp knot
Ib	smooth	11	2	curve interpolate smooth knot

The data pack forming the closed data loop is used for encoding the axial distances in DRUs ($p=dx$, $q=dy$) between knots, and for the double purpose of identifying and accessing override control codes. These override control codes are used to speed the decoding process and reduce the requirement for encoded data describing coordinate points while offering the further option of edit commands.

The data pack override control codes may be used to override machine default command codes which would otherwise be responsive to parameters expressed within the data pack. As can be seen, the data pack offers the option of following default command codes, responsive to predetermined parameters, such as angles between knots or interknot distances, or override control codes for generating the desired series

of nodes and curve locus between knots.

In the preferred embodiment, the default command codes are selected responsive to the parameters of the closed outline loop described by the interknot distances expressed within the data pack. The process of the invention described above for generating a series of nodes forming a smooth continuous curve between a set of end points would normally be used except where the distance between those end points was greater than a predetermined distance, such as 128 units for the preferred embodiment. In that case, where the interknot coordinate distances were greater than 128 units, the default command code responsive thereto would direct straight line interpolation and the formation of a series of nodes to find the locus of that straight line between the knots defining that respective coordinate distance.

Similarly, where the exterior angle formed by a line from the entrance knot to the exit knot and by a line formed from the exit knot to a successive knot is greater than a predetermined angle, such as 40° in the preferred embodiment, then a default command code responsive thereto would direct that a cusp be formed at the exit knot.

Straight line interpolation is well-known in the art. However, within the inventive scheme herein is provided an auto-scaling interpolation which provides an innovative and simpler method of linear interpolation and at the same time increases its accuracy and the precision of the closed outline loop as defined by the series of generated nodes. The auto-scaling linear interpolator is described below.

The format of the data pack shown in Table 1 is based 4 bit boundaries defining data words of 4 bits each. Data from the data pack, representing discrete complete information sets is accessed in a series of data words in nibbles of 4 bits at a time with the significance of that complete information set (CIS) and the number of data words therein indicated by the value of the first nibble. The correspondence between the first data word value of a series of data words forming a CIS and the Case indicated by that first data word value is shown below.

First data word value Case Purpose		
1 - 3	Ia	Control Code
0	Ib	Control Code
4 - 7	II	Coordinate Distance
8 - B	III	Coordinate Distance
C - F	IV	Coordinate Distance

A series of data words may be a control code as in the Cases Ia and Ib or the coordinate distances between knots, represented by the x,y incremental values between such knots in the preferred embodiment, in Cases II, III and IV.

Where the data words are nibbles, in the preferred embodiment the first nibble value of a nibble series then indicates the Case, and directs the access of a predetermined number of successive nibbles or data words within the closed data loop from the CIS and to complete the override control code, as in Case Ia or Ib or to assemble the number of data words necessary to complete the values corresponding to the coordinate distance between the next knot in the closed outline loop, as in Cases II, III and IV.

The number of nibbles or data words accessed responsive to the first nibble or data word value are the nibble or data word series necessary to form the CIS for the particular Case, after the first nibble or data word, as follows:

In Case II, one nibble or a total of 2 nibbles are required. In Case III, two more nibbles after the first nibble, or a total of 3 nibbles are required, and in Case IV, three more nibbles after the first nibble, or a total of 4 nibbles are required to provide the value of the interknot coordinate distance.

In the Cases Ia and Ib, the number of successive nibbles which must be accessed, responsive to the value of the first nibble, is ordered responsively to the Case indicated and as explained in detail below. For example, Case I may require the accessing of 3 or 6 nibbles to complete the Case Ia control code.

At the end of the data word series of a CIS, accessed responsive to the first data word value, the next successive nibble in the closed data loop would then be considered the first data word value of the next data word series and corresponding CIS. That new first data word value of the next data word series would then in a similar manner indicate the number of successive data words to be accessed to complete the completed information set for the override control code or the value of the interknot coordinate distance, as the case may be. By encoding the closed data loop on 4 bit boundaries, the data with respect to control codes may be packed successively within the closed data loop with the data respecting the interknot coordinate distances and, as such, the closed data loop may serve the double function of providing override control codes as well as interknot distance data.

Table II provides a definition of the control codes, control code value, the number of nibbles including the first nibble to complete the control code instruction and the purpose of the control code.

Further, as shown in Table I, and as stated above, a first nibble value of 1 to 3 is indicative of Case Ia and with that first nibble value indicating the particular control code value for case Ia, either 1, 2, or 3, and the number of subsequent nibbles to be accessed for the CIS forming that particular Case Ia control code.

As shown in Table II, 3 additional nibbles are required for a total of 4 nibbles, including the first nibble, to complete the CIS responsible to a control code value 1, 3 additional nibbles for a total of 4 nibbles are needed to complete the CIS responsive to a control code value of 2 and 6 additional nibbles for a total of 7 nibbles are needed to complete the CIS responsive to a control code value of 3.

Where the first nibble value of a CIS is zero, then Case Ib is indicated, as shown in the Table I, and the access of the next successive nibble directed for completion of the Case Ib instruction. In Case Ib, 4 is added to that next nibble value to obtain the control code value for case Ib. The control code values for Case Ib are shown in table II with their corresponding purposes.

In summary a first nibble value of 0, or 1 to 3, indicates Case Ia or Case Ib as shown above, and directs the accessing of a predetermined number of nibbles in the closed data loop successive to the first nibble, to form the CIS for that Case and which is then used to determine the control code value which in turn directs the processor. The next nibble subsequent in the order of the closed data loop to the last nibble of the previous nibble series corresponding to a CIS then becomes a new first nibble value and is used to indicate either an override control code or interknot coordinate distance. Further, and as shown below, where the first nibble bit as shown above indicates a Case II, the ordering of one more nibble for a total of 2 nibbles is required to complete the CIS and to provide the incremental coordinate distance. For Case III, 2 more nibbles for a total of 3 nibbles are required to complete the CIS for the incremental coordinate distance; For Case IV, 3 more nibbles for a total of 4 nibbles are required to complete the CIS for the incremental coordinate distance for code 4. Then the next successive nibble in the closed data loop after the CIS would be the new first nibble value indicative of case Ia, Ib, II, III, or IV, as the case may be, leading to an indication of the number of successive nibbles needed to complete the CIS to complete the control code or incremental coordinate distance.

For the override control code Case Ia, where the next successive knot in the closed outline loop is to be defined by Case Ia, control code values 1, 2 or 3, as shown in Table II, then, the first nibble value will have a value of 1, 2 or 3 (thereby indicating override control code Case Ia), and with the control code value being indicated by the specified first nibble value (i.e. 1, 2, or 3). The number of successive nibbles to be accessed to form the CIS for that Case Ia control code are indicated by the value of the first nibble value. Where the value of that first nibble value is 1, then as shown in Table II, the next three nibbles in the closed data loop are accessed and used to denote horizontal or motion in the X direction with the next knot X coordinate given by the next three nibbles, completing the nibble series necessary to form the Completed Information Set for that control code.

Where the value of that 1st nibble is 2, then as shown in table II, a vertical motion in the Y direction is indicated with the next three nibbles in the closed data loop being accessed, to complete the nibble series necessary to form the CIS and indicating the Y coordinate value of the next knot.

Where the value of that 1st nibble is 3, then the next six nibbles in the closed data loop are accessed to complete the nibble series and necessary to form the CIS and indicating the next knot, diagonally related to the immediate knot, and with the X,Y coordinates therefore given in each of the next 3 nibbles.

In the preferred embodiment, the least significant bit of the control codes for Case Ia, namely control codes 1, 2, and 3 is used to provide a direction instruction for new X, Y or X,Y coordinate values relative to the preceding knot in the closed outline loop. That relative direction between the new knot position and the preceding knot is then followed when locating the positions of successive knots corresponding to the information in a Case II, III or III instruction indicating the incremental coordinate distance. As will be seen below, the direction may also be changed by a Case Ib control code 5, 6 or 7 which would negate the established X, Y or XY direction.

In summary, in the closed outline loop new knot coordinates as indicated by Case Ia, control code values 1, 2 or 3, shown in table II, would be located in a direction consistent with a previous X and Y direction instruction, unless the least significant bit of the first nibble value for the CIS indicates a change in direction, as in Case Ib, control code values 5, 6 and 7.

In summary, the first data word value is the value of the first data word of a series of data words forming the Completed Information Set (CIS) of the closed data loop, indicating a override control, code as in case Ia, for first nibble values 1, 2 and 3, or as in Case Ib, for value 0, or the incremental coordinate distances as for first nibble values 4-F, for cases II, III or IV.

Then, as stated above, if the 1st data word value, in a nibble size data word series is a zero, then

control code case Ib is denoted. Case Ib, directs the access of another nibble of 4 bits in the closed data loop and, if that nibble is not equal to 0, its value is added to value 3 to obtain the proper control code, as shown in Table II. In this way, a total value of control codes 0 and 4 to 18 may be defined. The meanings of each of the control codes having values 4-11 is shown in Table II.

5 If the control code value for the additional 4 bits is zero, the end of the last loop is indicated (i.e. bits 4-7 are 0 value).

Code 4 indicates the start of a loop.

Codes 5, 6 and 7 provide direction information for the next knot in the closed outline loop relative to the preceding knot.

10 Codes 8, 9, 10 and 11 are editing override control codes, which override the default command codes.

The editing override control codes are used to override the default command codes, which would normally be responsive to and result from a machine interpretation of the data pack values for cases II, III and IV.

Code 8 directs Linear Interpolation Sharp Knot. As shown in FIG. 5, it may be used to force the exit angle at a knot (i.e. ϕ_a at knot Z_{a+1}) for curve Z_a, Z_{a+1} , to be equal to the interknot angle B_a at the entrance knot, Z_a of that same curve Z_a, Z_{a+1} . As shown in FIG. 5, Code 8 forces the curve Z_a, Z_{a+1} , at knot Z_{a+1} to have the same exit angle at the exit knot (Z_{a+1}) as the interknot angle B_a at its entrance knot Z_a and produces a sharp cusp at Z_{a+1} . To complete the cusp at Z_{a+1} , the entrance angle ϕ_{a+1} for curve Z_{a+1}, Z_{a+2} would be forced equal to the interknot angle B_{a+1} at Z_{a+1} .

20 Code 9 denotes Linear Interpolation - Smooth Knot and may be used, for example, to force a smooth knot located at the end of a curve Z_a, Z_{a+1} which then becomes a straight line, or at the end of a straight line, which then becomes a smooth continuous curve.

The use of override control code 9 is shown in FIG. 6a where a knot joins a straight line curve Z_a, Z_{a+1} to a smooth continuous curve, Z_{a+1}, Z_{a+2} . In this case, ϕ_{a+1} is set equal to B_a .

25 Where a knot joins a curve section Z_a, Z_{a+1} to a straight line curve, Z_{a+1}, Z_{a+2} , as shown in FIG. 6b, then ϕ_a , the exit angle for curve Z_a, Z_{a+1} is made equal to B_{a+1} , the interknot angle between Z_{a+1}, Z_{a+2} .

It should be understood, however, that if a knot joins two straight lines, this rule shown with respect to FIG. 5 is used in the preferred embodiment.

Control code 10 directs a curve interpolation at a sharp knot and is used to form a cusp at the entrance knot or the exit knot of a smooth continuous curve segment joined by that knot to a straight line. Shown in FIG. 7a, are two examples, i.e. the curve Z_a, Z_{a+1} is formed with a sharp knot at the entrance knot Z_a and the curve Z_{a+1}, Z_{a+2} is formed with a sharp knot at the exit knot Z_{a+2} . This control code is useful in overriding a default command code which would otherwise require that the sharp knot at Z_a , for example have an entrance angle for curve Z_a, Z_{a+1} equal to B_a which would introduce a distortion in curve Z_a, Z_{a+1} at area 31, approximate the entrance knot Z_a for curve Z_a, Z_{a+1} , as shown in FIG. 7b. Similarly, a distortion would be introduced in a smooth continuous curve terminating in a sharp knot. For example, in curve Z_{a+1}, Z_{a+2} as shown by numeral 33 in FIG. 7b, approximate the exit knot Z_{a+2} , where ϕ_{a+1} would be forced to equal B_{a+1} to form a cusp. code 10 overrides the default command code and forces θ (i.e. θ_a) to be equal to ϕ (i.e. ϕ_a).

40 Where a cusp is to be formed at knot Z_a , the default command code would specify that the exit angle, ϕ_a at knot Z_a would be equal to the interknot angle B_{a-1} at the preceding knot Z_{a-1} , for curve Z_{a-1}, Z_a , and the entrance angle ϕ_a for curve Z_a, Z_{a+1} would be equal to the interknot angle B_a between the entrance knot Z_a for curve Z_a, Z_{a+1} and its exit knot Z_{a+1} . As stated above, this would produce the result shown in FIG. 7b, and a distortion of the smooth continuous curve Z_a, Z_{a+1} shown in FIG. 7a. With the result of FIG. 7b, the deviance angle θ_a between B_a and ϕ_a would be zero as stated above. In this case, to avoid the distortion shown by numerals 31, 33 in FIG. 7b and to produce the smooth continuous curve as shown in FIG. 7a, θ_a is forced equal to ϕ_a the deviance angle, at exit knot Z_{a+1} , as stated above. In the case shown in FIG. 7a, arranged, for the sake of explanation ϕ_a is equal to zero, ϕ_a is equal to both B_a and θ .

A similar result is forced by using a code 11 to control the shape of curve Z_{a+1}, Z_{a+2} at exit knot Z_{a+2} . As stated above, the default command code responsive to a cusp at Z_{a+2} for example, would direct the curve Z_{a+1}, Z_{a+2} , to the shape shown in FIG. 7b and particularly shown by numeral 33 approximate Z_{a+2} by directing that the exit angle ϕ_{a+1} at knot Z_{a+2} is equal to the interknot angle B_{a+1} at the entrance knot Z_{a+1} for that same curve. However, by using code 11 to direct that ϕ_{a+1} is equal to θ_{a+1} , then at exit knot Z_{a+2} for curve Z_{a+1}, Z_{a+2} ϕ_{a+1} is equal to $2 \cdot B_{a+1}$ ($\phi_a = 0, \phi_{a+1} = 0, \theta_{a+1} = \phi_{a+1} - B_{a+1}$).

55 For the sake of explanation, ϕ_{a+1} is equal to 0, for curve Z_{a+1}, Z_{a+2} .

The effect of code 10 is to produce a smooth continuous curve from or to a cusp such as at entrance knot Z_a or at exit knot Z_{a+2} and symmetrical about the respective curve midpoints, as shown by numeral 35 and numeral 37, respectively for curve Z_a, Z_{a+1} and for curve Z_{a+1}, Z_{a+2} .

Control code 10 is to override a default command code which would otherwise specify a cusp, such as where the exterior angle as described above was above a threshold such as 40° as shown in the preferred embodiment, or the interknot distance was greater than a predetermined distance, such as 120 units in the preferred embodiment. In this case, a smooth knot would be formed as described above, by taking the average of the interknot angles between a preceding knot and the subject knot and the subject knot and a successive knot in the outline loop direction of the closed outline loop. However, as explained above, if the absolute value of the interknot angles between the preceding knot and the subject knot and the subject knot and the successive knot is greater than 180° , then the supplemental average is used by supplementing the average angle by 180° . This is to orient the angle in the correct direction where the relationship of the average angle described above would result in a resultant angle 180° out of phase with its correct direction.

The application of the inventive principles will force the angular relationships of α , θ , θ and ϕ at the respective knots to conform to the rules described above, as necessary to produce the desired curve, straight line, cusp, or smooth knot, whether directed by the default command codes or by the override control codes. It should be noted, however, that where an override control code is used, it is used to force a result contrary to what would ordinarily be produced using the default command codes. For example, if the default command code would produce a cusp and a smooth knot was desired followed by a smooth continuous curve or preceded by a smooth continuous curve, then a code 10 would be used. If the default command code would have produced a smooth continuous curve, and a straight line joining a sharp knot is desired, then a code 8 would be selected. If a smooth knot is desired joined to a straight line, then a code 9 may be selected to override the default command control. All of the foregoing is shown in connection with FIGS. 5, 6, 6a, 7, 7a and 8 and in the text accompanying these figures.

A typical encoding for an A as shown in FIG. 9 is shown in the appendix. The compiler for decoding the packed encoded information is also shown below, and is used with a Motorola 68000 processor. As the desired output is a series of intercepts at the intersection of the locus of the character outline, and the display raster lines, these intercepts can be converted into modulating information for imaging the character on an imaging surface by any suitable well-known imaging device.

For the sake of explanation and in the way of an example, a closed data loop for the encoded A shown in FIG. 9, is set forth below and described.

CLOSED DATA LOOP FOR THE "A" OF FIG. 9 (Given in hexadecimal Notation)

```

83 98 3C 82 E8 5E 02 4C 01 5C 02 5E 16 49 00 5B
30 B2 54 D6 75 96 83 78 81 A7 D7 5A 37 1B 80 7C

57 3A 83 89 0E 85 E0 28 24 15 C0 20 49 77 96 88
7D 2F 30 60 62 F1 6D 0F 60 26 AD 28 20 C1 8A 88
06 50 10 06 95 99 7B 88 58 80 B0 82 08 01 3A 70
FF FF 00 10 79 06 50 37 31 19 7B 73 53 81 51 79

```

As stated above, the closed data loop is encoded on data words of 4 bit or nibble boundaries and the sequence of the nibbles is as given below. In accordance with the preferred embodiment, the first two bytes 5B of the closed data loop indicate the total number of bytes in the data loop for a closed outline loop. The first two bytes, 00 5b indicates that there are 91 bytes total in the data loop for the symbol A, describing outside closed outline loop 31 in the direction of arrows 31 and inside closed outline loop 35 in the direction of arrows 35.

In accordance with the preferred embodiment, the starting X and Y coordinates are given in three nibble packs of information for each respective X and Y location. Accordingly, the next three nibbles, 6 49 relates to the starting X location. In accordance with the principles of the invention and the preferred embodiment, the least significant bit of the data for a new X or a new Y location is a sign bit indicative of the direction. Accordingly, in processing the data in the preferred embodiment, a shift of one binary position is made to remove the sign bit, giving a decimal data value of 804. As described above in the preferred embodiment, since the origin of the M^2 is offset by 592 units, 592 must be subtracted from 804 to provide the correct X

coordinate with reference to the origin of the M^2 of 212. The sign of the X direction, whether positive or negative with regard to the origin of the M^2 is given by the sign bit and is positive if the sign bit is zero, in the preferred embodiment.

In accordance with the principles of the invention, the starting Y position given as a new Y position is indicated by the next nibble series of 3 nibbles or by 5E1, which is divided by 2 to remove the sign bit yielding the result of 750. In accordance with the offset of the M^2 at 736 units, 736 is subtracted from 752 to yield a Y coordinate of 16. The sign bit which is a 1, indicates a negative direction for Y. Accordingly for the "A" of FIG. 9, the start point x, y coordinates shown as numeral 39, is 212, 16 with the new direction being x,-y.

As stated in the preferred embodiment, the data pack is decoded by using the first two bytes to indicate the number of bytes in the closed data loop corresponding to the coordinate points around the closed outline loop, and by three nibbles each comprising twelve bits corresponding to the respective X and Y start locations. It should be noted that wherever the X coordinates are defined by new coordinate values, the least significant bit is used as a sign bit to indicate the coordinate direction. The process performs a divide by two which separates that bit to define the aforesaid direction signed. Additionally, the X position is referenced to the home or reference position in the M^2 which may be offset with regard to the origin of the master encoding grid by subtracting the offset from a coordinate position accordingly. With this in mind, the following process is described which produces the listing of coordinates shown below and in FIG. 9.

As stated, the start position is at XY coordinates 212, 0. As the last bit of data accessed from the data pack, to provide complete information for the preceding nibble series was the tenth nibble corresponding to hexadecimal number 5. The next nibble which follows a complete information series of nibbles is a first nibble value.

As shown, the first nibble value 2 indicates a case Ia and according to table II directs the access of the next three nibbles, 5C0 indicative of the new Y coordinate position. In accordance with the procedure above, a division by 2 removes the sign bit and a subtraction process is performed relative to the M^2 offset, to reference the new Y position to the M^2 home position or at zero. The first nibble value "1", following the nibble series for the completed information set of 5C02 then indicates a case Ia and a new X coordinate. In accordance with the process described above, a binary division of two is performed to isolate the sign bit, indicate the direction and a subtraction is performed to reference the new X position to the home position. Accordingly, the new X position is 16.

The next coordinate position of 16,16 is given by the complete information set 2E8, with the nibble series first nibble value 8 indicating of a Case III. In accordance with the format of the data pack as shown in Table I, $P = X$ is equal to 01110 and $Q = y$ is equal to 00010. This translates to 14 and 2 in decimal rotation respectively. In accordance with the preferred embodiment, as it would be redundant to use Cases II, III and IV, for a translation of 0, a 1 is added to the results of $p = x$ and $q = y$ to provide the new X and Y coordinate distances of 15 and 3 respectively. The p value is added to previous X value of 16 to provide a new X value of 31, and the q value is added to the previous Y value of 16 to provide a new Y value of 19 accordingly.

As a reminder, it should be understood, that this process of decoding the data pack being described is designed to decode the encoded knot coordinates only. After decoding the inventive principles described herein are applied to the knots to either produce a series of nodes describing a smooth continuous curve between the knots, as between knots 16, 16, and knots 31, 19 or a straight line as between knot 212, 0 and knot 16, 0. In accordance with the preferred embodiment, the direction of the knots indicated by the completed information set 2E8 relative to the previous knot is in accordance with the direction established by the last previous sign bit accessed for the respective X and Y directions as shown above, or by 567 as described below.

The next nibble following the nibble series for the completed information set above has the first nibble value of 8 which is a Case III, directing the access of the next two nibbles for providing the nibble series for complete information set of 3C8 corresponding to a $p = x$ value of 13 and a $q = y$ value of 4 in accordance with the format shown in Table II. In accordance with the preferred embodiment, the direction of the knot denoted by this complete information set follows the last previous direction indication given and is accordingly added to the previous coordinates of 31 and 19 to provide new coordinates of 44 and 23.

In accordance with the foregoing, the next nibble being a first nibble value is 8 which accordingly directs the access of the next two nibbles to provide a complete information set of 398 producing new coordinate values of 54, 27, accordingly.

The next nibble corresponding to a first nibble value of 8, as described above, indicates a case 8 and directs the access of the next two nibbles to form the nibble series corresponding to the complete information set of hexadecimal 7C8 and according to the processing described above decoding the new

coordinates of 67, 35.

As can be seen, the smooth continuous curve between knot end points 16,16 and passing through knots 31, 19 and 44, 23 and 54, 27 and, 67, 35 are formed according to the principles of the invention to form a smooth continuous curve.

5 The next first nibble value of the next complete information set is zero indicating Case Ib which directs the access of another nibble to form a two nibble series complete information set. As the next nibble was 8, for Case Ib, a value of 3 is added thereto forming a value of 11, indicative of control code 11 directing a smooth continuous curve be formed between the last knot having coordinate 67, 35 and the next knot which is to be indicated by the next complete information set accessed from the closed data loop.

10 As the first nibble value of the next complete information set is B indicating Case III, 3 nibbles are accessed forming a 4 nibble series describing the next complete information set and producing the new coordinates of 85, 59.

15 The next first nibble value of the next nibble series for the next complete information set is 3 indicating Case Ia, control code 3, shown in Table II, and which directs the access of a pair of three nibbles each which are indicative of the next X and Y positions and directions thereof relative to the previous knot. In accordance with the process described above, the new X and Y position is 349, 606 and as the distance between this new knot and the previous knot is greater than 128 units, the default command code directs linear interpolation. In

20 accordance with the above, the hexadecimal value 75A corresponds to the new X position and A7D corresponds to the new Y position. As stated above, the least significant bit of each of the above 3 nibbles corresponds to the relative direction of the new X and Y points.

The first nibble value of the next nibble series for the next complete information set is 1 indicating a Case Ia and directing the access of the next three nibbles of 7A8 to replace the X coordinate with 372 and a new set of coordinates 372, 606.

25 The next first nibble value for the next nibble series is a 3 corresponding to a case Ia, a control code 3 and the access of a pair of 3 nibbles, i.e. 968, corresponding to the new X position and 675, corresponding to the new Y positions and new X, Y coordinates of 612, 90 respectively.

30 The next first nibble value following the nibble series for the complete information set above is D, corresponding to Case IV which directs the access of the next three nibbles to form the 4 nibble series 254, and which according to the form shown in Table I provides a p = x value of 20 and a q = y value of 37. According to the process described above, a 1 is added to the p = x value and the q = y value to form decimal coordinate values of 21 and 38. Following the most recent x, y coordinate direction instructions given, the x incremental value is added to the previous coordinate 612 providing the new X coordinate of 633 and the Y incremental value is subtracted from the previous Y of 90 to provide a new Y coordinate

35 value of 52. The next first nibble value of the next nibble series for the next complete information set is a B corresponding to Case III which directs the access of two more nibbles to form a three nibble series, complete information set, and producing the next coordinate values of 650, 32 for X and Y respectively.

40 The next first nibble of the next nibble series for the next complete information set is 8 corresponding to a Case III, causing the access of the next two nibbles to produce a three nibble information set and new coordinate values of 659, 25 for X and Y respectively.

The next first nibble value is a 9 corresponding to case III and directing the access of the next two nibbles of 77 and corresponding to new X and Y coordinate 683, 17.

45 The next first nibble value is a 9 causing the access of the next two nibbles 04 and corresponding to the new X, Y coordinates with 704, 16.

The next new first value is 2 corresponding to Case Ia and directing the access of the next three nibbles corresponding to the 5C0 corresponding to a new Y value with the direction indicated by the first significant bit therein and producing new XY coordinates of 704, 0 respectively.

50 The next first nibble value of the next nibble series for the next complete information set is 1 corresponding to a Case Ia and directing the access of the next two nibbles forming a three nibble series complete information set 824 and corresponding to the new X, Y location of 450, with respect to the last previous direction given for the axial directions X and Y.

The next first nibble value 4, the next complete information set is 2 corresponding to case Ia and to complete information sets 5E0 corresponding to new XY coordinates 450, 16.

55 The next first nibble value is 8 corresponding to Case III and as described above corresponding to complete information set 0E8 and new XY coordinates 465, 17.

The next first nibble value is 9 corresponding to Case III and complete information set 389 and new XY coordinates 490, 21.

The next first nibble value is 8 corresponding to Case III and complete information set 38A and new XY coordinates 501,25.

The next first nibble value is 7 corresponding to Case II which directs the access of another nibble and according to the format shown in Table II provides a $p = x$ value of binary 101 corresponding to a decimal value of 5 to which one is added in accordance with the procedure above to provide a decimal value of 6 for the new X coordinate incremental distance. Similarly, the $q = y$ value is binary 011 which corresponds to a decimal value of 3, to which one is added in accordance with the process above to provide a new voice of 4 corresponding to the incremental Y coordinate distance. The new XY coordinates are therefore 507,29 accordingly to the directions given in the last previous direction instruction.

The next first nibble value is 8 corresponding to Case III and the access of two more nibbles to form the complete information set A88 and XY coordinates 516, 40.

The next first nibble value is 8 corresponding to Case III, the access of the next two nibbles form a complete information set C18 and new coordinates XY of 518, 53.

The next first nibble value is zero corresponding to case Ib which directs the access of the next nibble to, which 3 is added in accordance with procedures described above to form the control code 5. As shown in Table II is an X Negate which changes the X direction from its previous direction. The next first nibble value is an 8 corresponding to Case III and in accordance with the foregoing new X, Y coordinates 515, 67.

The next first nibble value is 8 corresponding to case III and new coordinates of X, Y of 508, 86, respectively in accordance with the above procedure.

The next first nibble value is a 0 corresponding to case Ib which directs the access of the next nibble 6 to which 3 is added according to the above procedure to provide a control code of 9. 9 as shown in Table II is a line smooth direction.

The next first nibble value is F corresponding to Case IV and in accordance with the procedure above causes the access of three more nibbles forming the complete information set 600F and new X, Y coordinates 459, 196 in accordance with directions as given.

The next first nibble value is 1 corresponding to Case Ia which directs the access of the next three nibbles, 62F indicative of an X coordinate point in accordance with the last previous directions given. Accordingly, the new X, y coordinates are 199, 196.

The next first nibble value is zero corresponding to Case Ib and directs in the access of the next nibble having a value of 6 to which 3 is added in accordance with the above procedure to produce a 9 corresponding to control code 9. A next first nibble value is zero once again corresponding to Case Ib which directs the access of a next nibble which is 3 to which 3 is added giving the control code value 6 which means negate Y.

The next first nibble value is F corresponding to Case IV and in accordance with the above procedure causes the access of the next three nibbles to complete the information set 7D2F and producing new X, Y coordinates of 148, 70 which is connected to the previous X, Y coordinates 199, 66, according to the override control code 9 and in the new Y direction.

The new first nibble value is zero corresponding to case Ib and directing the access of the next nibble having a value of 7 to which 3 is added to give it the control code 10.

The next first nibble value is an A corresponding to Case III and directing the access of the next two nibbles to give the complete information set 13A and new X and Y coordinates 144, 52 respectively.

The next first nibble value is zero corresponding to Case Ib which directs the access of another nibble having a value 8 to which 3 is added in accordance with the above procedure to provide the control code 11.

The next first nibble value is zero corresponding to Case Ib directing the access of another nibble having the value 2 to which 3 is added giving the control code 5 which negates the previous X direction.

The next first nibble value is A corresponding to Case III and the access of two additional nibbles giving the complete information set B08 and new XY coordinates 145, 40.

The next first nibble value is zero corresponding to case Ib in accordance with the above procedure control code 11.

The next first nibble value is 8 corresponding to Case III, complete information set 858 and new XY coordinates 151, 31.

The next first nibble value is 8 corresponding to Case III, complete information set 7B8 and new XY coordinates 163,23.

The next first nibble value is 9 corresponding to Case III, complete information set 599 and new XY coordinates 189,17. The next first nibble value is 9 corresponding to complete information set 069 and coordinates 212, 16.

The above XY coordinates 212,16 bring the traverse of the closed data loop bringing it back to the

starting point.

The next first nibble value is zero indicating Case Ib with the next access nibble I to which 3 is added producing the control code value 4 indicating the end of closed data loop.

In accordance with the preferred embodiment, a routine is added, not shown, which would be known to one skilled in the art to ensure that the closed data loop decoding closes upon the start location of the closed data loop and completes the closed outline loop by ending at the start point encoding of the closed data loop which represents the start of the closed outline loop.

The second closed outline loop, shown by numeral 35 and arrows 37, of the A now starts at the closed data loop coded hexadecimal numbers 790, 650 corresponding to new X and Y coordinate values 216, 232 and the X and Y directions, derived according to the process given above with the regard to the start points for the foregoing loop, 31.

The next first nibble value is 1 corresponding to Case Ia and directing the access of the next three nibbles 815 and new XY coordinates 442, 232.

The next first nibble value is a 3 corresponding to a Case Ia, directing the access of a pair of three nibbles 735, and 97B corresponding to new X and Y coordinates of 330 and 477 in the negative X and negative Y directions accordingly.

The next first nibble value is 1 corresponding to Case Ia directing the access of the next three nibbles 731 corresponding to X coordinate 328, Y coordinate 477 in the negative X and negative Y directions accordingly.

The next first nibble value is 3 corresponding to Case Ia directing the access of a pair of three nibbles each corresponding to hexadecimal value 690 and 790 and X and Y positions 216, 232, respectively. As 216, 232 are the start points, the closed data loop representing the closed outline loop 35 has been completed and the next first nibble value is zero indicating Case Ib with the last nibble accessed accordingly being zero indicating the end of loop. Once again, the routine described above is used to insure that the closed data loop ends at its start point.

The above is a representative encoding of a letter used in the process to derive the coordinate values around the outline of the character and any override control codes which may be used to replace default command codes. However, it should be understood that changes to the codings could be used consistent with the principles of this invention, as claimed herein, and that the invention should not be restricted to the coding or decoding process shown above, with respect to the preferred embodiment.

AUTOSCALING LINEAR INTERPOLATION

Linear interpolation is a well-known technique and is not claimed as an invention in this application. The autoscaling linear interpolation which is claimed and which is described below is a method of increasing the precision of a machine interpolation procedure which uses as a start point, a first set of coordinates and as an ending point a second set of coordinates. The coordinates are usually expressed in respective coordinate directions such as x and y, for example. In the preferred embodiment, the process of linear interpolation is for the purpose of producing coordinate points along a straight line between the first and second end points, which coincide with a second coordinate system such as the intercepts in a raster display. Each of the coordinates are found by determining the slope of the straight line between the two end points which is equal to the incremental distance in a first coordinate direction divided by the incremental distance in a second coordinate direction (i.e. $Y_2 - Y_1 / X_2 - X_1$). That first coordinate incremental distance is expressed in an encoded machine value as a first data word, (i.e. $Y_2 - Y_1$), in a first machine location. The second coordinate incremental distance between the two end points in the second coordinate direction (i.e. $X_2 - X_1$), is also expressed as a machine value and placed in a second machine location. The machine location limits the precision by which a data word may be expressed. For example, and as is well-known, each machine is based upon radix, which is a number base. The most common machine number base is the binary number base. Each data word accordingly, has a number of bit positions with each specified bit position being a specified power of that radix (i.e. $2^4, 2^3, 2^2, 2^1, 2^0, 2^{-1}, 2^{-2}$,). Accordingly, a shift of a data word in a direction of the most significant bits corresponding to higher exponential values or higher orders of the machine radix, results in an increase in the scale or value of the data word. Further, each shift in the direction of the most significant bits, results in an effective multiplication of the data word value by a scale factor of the machine radix raised to an exponential power corresponding to the number of bits shifted (i.e. 13 bit positions shifted is equal to a scale factor of 2^{13} in a binary machine or equivalent of 8196, in decimal notation. Conversely, each shift of a data word, towards the least significant bits, corresponding to lower exponential values or orders of that machine radix, corresponds to an effective division by the machine radix value and conversely to a reducing scale factor. Further, machine locations for storing data words are

limited in the number of bits available. As the precision of a data word is a function of the data space and the number of bits available for storing that data word, an increased precision for expressing a data word are obtainable by extending the number of bits or size of a machine location available for specifying a data word. For example, as is well-recognized, in a decimal system, the number 5.632498 is a more precise value than 5.3624 which lacks the last three significant places (i.e. .000098), of the former number. In binary, the number 10111.101 is a more precise expression than that number truncated or rounded to 10111.000, as the latter number is missing the bits .101 and is therefore less precise. However, the former binary expression requires a larger machine location for specifying all the relevant bits in that expression. The "point" in the above binary and decimal values are used to indicate the positions, according to the radix system used, to indicate position values equal to or greater than 1 (decimal) and less than 1 (decimal)- (i.e. integer and fractional values).

The effect of shifting to increase the scale of the Y increment is to eliminate the binary point in a third data word representing the slope or Y incremental value and thereby avoiding floating point arithmetic operations. The point separates the bit positions in the radix system selected, separating fractional from integer values in a data word having values equal to or greater than 1 and less than 1 (i. e. between the bit positions 2^0 and 2^{-1}). The binary point is equivalent to the decimal point in a radix 10 system and equivalent to a "point" between those machine positions having a value equal to or greater then 1 and less then 1 in any system and as stated, separating the fractional values from the integer values in the slope or Y increment value. By shifting a machine representation of a number in the direction of more significant bits, the scale of the number is increased, thereby moving the "point" effectively in the direction of the less significant bits. If a sufficient number of bit positions are shifted, the binary point is eliminated from the number. In this way, floating point arithmetic is avoided.

According to the invention, linear interpolation proceeds according to the known method by determining the coordinate distance in a first coordinate direction and in a second coordinate direction between a set of end points. The method is used in the machine having a radix "r" coresponding to the values of designated bit positions in the encoded words used within the machine. The process continues by encoding a first data word on "N" bits corresponding to the distance between the first and second end points in the first coordinate (i.e. $Y_2 - Y_1$), and then placing that first data word in a first machine location. To complete the process of calculation the slope, according to the interpolation method, a second data word of M bits corresponding to the distance between the first and second end points (i.e. $X_2 - X_1$) in a second coordinate direction is encoded and placed in a second location. The machine is now ready to use the data value in the first location and a data word value in the second location in a process of division to produce a third data word corresponding to the slope of the straight line between the first and second end points (i.e. $Y_2 - Y_1 / X_2 - X_1$). To increase the precision or resolution of the third data word according to the inventive process, the scale of the first data word (i.e. $Y_2 - Y_1$), is increased by determining the the number of available positions between the most significant position of the first data word and the most significant position of the first machine location. The first data word is then shifted in a first direction towards the most significant positions in that first machine location, utilizing those more significant positions which were unused, when expressing the first data word. The scale of the first data word is again increased by shifting a second time in the same first direction of the most significant positions by a number of positions coresponding to the number of bit positions used to encode the second data word (i.e. $X_2 - X_1$). In the preferred embodiment, the first machine location is expanded to accommodate this additional shift. It should be noted that the inventive principle is not limited to the size of available data space in any one machine or to any machine radix.

In the preferred embodiment, the second shifting operation described above occurs in an extension to the first machine location. The quotient corresponding to the slope (i.e. $Y_2 - Y_1 / X_2 - X_1$) produced as the third data word by division of the first data word by the second data word will reduce the first data word to a size coextensive with the first machine location. As a result then, the data word may be raised by a scale factor corresponding to a shift in the direction of the most significant positions corresponding to the number of significant positions used to express the denominator (i.e. $X_2 - X_1$) or second data word, as the next division step cancels that said shift of the first data value by the said number of bit positions in the second data word and coresponding reduces its position length to the size of the first machine location.

The third encoded data word (i.e. the Y incremental value for each X increment) is then stored. The first data word value is then changed to Y_1 and the second data word value is then changed to X_1 . The second data word is incremented by a coordinate unit value to produce X values between X_1 and X_2 which are then stored. The first data word is incremented by the value of the third data word per increment of the second data word to generate first data word values corresponding to positions between Y_1 and Y_2 . In Table III, are given the coordinate X, Y values generated.

In the iterative process, the third encoded data word at the higher resolution, produced, corresponding

to the slope produced according to the above, is used iteratively as a Y incremental value to derive a cumulative Y incremental value related to a respective X coordinate value in the first direction (i.e. Y) and stored as a cumulative Y. As no change has been made to the precision (i.e. the number of positions used to express the third data word), the precision of the third data word is the same as the first data word, produced by shifting the first data word and increasing its scale, as described.

As interpolation is an iterative process, the data word (i.e. the Y incremental value at the said higher precision), is used to produce the cumulative Y, related to the Y coordinate for a respective X coordinate.

As the purpose of producing data word values corresponding to points along a straight line, is to produce intercepts on a display coordinate system, the scale cumulative Y words are reduced to the scale used to express the end point coordinate values, by the step of truncating or rounding. Truncating may be accomplished by discarding a predetermined number of least significant bits or rounding may be used by a shift in the direction of the least significant bits, minus one, adding a bit corresponding to the rounding value such as .5 (decimal) and discarding any bits having values less than the least significant bit value of interest. In the preferred embodiment, truncating or rounding is to the position of the lowest integer value in the radix 2 system (i.e. to bit position 2^0).

In the foregoing manner, the scale of cumulative Y is then reduced to fit scale of the data words for the end point coordinates. In the preferred embodiment, the original precision is that of the coordinates on a display which are the display intercepts, shown as X, Y in Table III. The result of using this method is to avoid the incremental error introduced into each cumulative Y value by summing Y incremental at a lower scale.

The incremental value at the increased scale factor may be used to derive a cumulative Y for each respective coordinate X values by iteratively incrementing Y increment by Y increment to produce a first cumulative Y and then incrementing that first cumulative Y with the Y incremental value and so on to produce a series of distinct cumulative Y values at the higher scale factor for each X coordinate. The discrete cumulative Y values may then be reduced by the scale factor to the scale of the numerator (dx) before shifting and added to the initial Y coordinate value to produce the correct Y coordinate for each respective X coordinate. By producing the cumulative Y using Y increment at the increased scale factor, cumulative errors in the Y coordinate value due to an error in the cumulative Y value due to the lower scale factor of Y increments are avoided.

An example is shown in Table III. In the preferred embodiment, a radix 2 machine is used with locations of 16 bits each used and with the most significant or 16th bit being used for a sign bit. The Y register, therefore, corresponding to a first data word of 120 would be 0000 0000 0111 1000. As the leftmost or most significant bit of the register is not available for indicating the numerical value, as it is a sign bit in the preferred embodiment, then the first data word expressed in binary, may be shifted to the left in the direction of the most significant bits, a total of eight places corresponding to the eight zeros or unused binary bit positions to the left of "111 1000." The updated first data word, as shown in the register is now 0111100000000000 corresponding to a decimal value of 30,720.

The denominator corresponding to the distance between the end points in a second coordinate direction and to the second data word is the binary word 0000 0000 0011 0010 equal to a decimal value of 50. The quotient produced by the division of the dividend or first data word by the denominator or second data word produces a quotient having a number of bits related to the number of bits in the dividend reduced by the number of bits in the divisor or denominator. In the divisor, the most significant bit is in the 6th bit position corresponding to a decimal 2^5 or decimal of 32. The numerator corresponding to the first data word and expressed as the 16 bit binary word 0111 1000 0000 0000 may be raised by an order of the radix, (i.e. 2^5) corresponding to the order of the most significant bit position of the denominator, with the assurance that the quotient will fit within the register size containing the first data word after shifting above and without loss of any significant bits in the quotient, such as, for example, bit corresponding to the remainder.

To summarize the foregoing, the precision of data word for the numerator value, used in deriving the slope can be raised within the space allotted by a machine register by shifting the significant bits of that data word to the left or towards the most significant bit in the register equal to the number of unused bits in the register or in the example between the most significant bit position for the first data word shown, or seventh bit position and the most significant bit position available in the register. This causes the binary word corresponding to the numerator to be stated at its highest value thereby avoiding floating point arithmetic. The precision of the slope used in linear interpolation may be further increased by expanding the register space available for storing that binary word and shifting that first data word to the left, effectively raising it by a power of the radix used, corresponding to the number of significant bits in the denominator or divisor. As stated above, as the slope is the quotient produced by the division of the dividend or numerator (first data value), by the divisor or denominator (second data value), a shift of the numerator by the number

of significant bits in the denominator results in a quotient having a number of significant bits no greater than the number of bits in the dividend or numerator originally. In this way, the precision of the quotient or incremental value in the first coordinate direction is maintained equal to the precision of a numerator.

To continue with the example above, as the divisor contains 6 significant bits corresponding in binary notation to 2^5 or 32 then the numerator may be shifted to the left for a total of 13 places by increasing the scale, raising the value of the first data word by a scale factor of 2^{13} . As 2^{13} equals decimal 8192, the arithmetic accomplished in binary form and expressed in decimal is

$$\text{dec. } \frac{120 \times 8192}{50} = \frac{F0000}{32} \text{ HEX}$$

where

$$120 = Y_2 - Y_1;$$

$$50 = X_2 - X_1;$$

The first coordinate or Y incremental value is expressed in decimal as $19660/8192 = 2.399902$ or 4CCC in HEX. The actual incremental value defined by the coordinate end points is $120/50 = 2.4$. The difference between the Y incremental value, expressed as a third data word and the actual slope is $4 \times 10^{-3}\%$ error. As shown in Table III, the Y incremental produced at the higher precision is used to iteratively increment Y, which is then reduced in scale accordingly and added to the first end point in the first coordinate direction (Y direction) and that Y coordinate value is stored. In reducing the scale, the Y value may be truncated or rounded, to produce an intercept value, for example. As shown in Table III below, the intercept value is shown as produced by truncation and rounding.

Additionally, to increase the speed of the process, the value of the first data word or numerator may be compared to successive entries in an ordered set of values. By a comparison of the value of the first data word it can quickly be determined whether the coordinate difference (i.e. $Y_2 - Y_1$) is greater or lesser than a value in a particular position of the set of values. Accordingly, the set of values may be arranged in decreasing orders of the machine radix, (i.e. $2^{13}, 2^{12}, \dots, 2^0$). By a successive process of comparison in the decreasing order, it can easily be determined where in this procedure, the first data word in the first machine location is less than a value in the set of values. Then that particular value in the set of values could be referenced to an index value to indicate the number of bits available in the numerator register for shifting the first data word. It should be noted that according to the principles of the invention, it is not necessary where the denominator is a power of two, such as two or four, to divide, saving additional time, as shifting of the numerator accomplishes the same result.

Accordingly, the above process is equally valid for the reverse process where the set of values is in increasing order and the comparison is in the increasing order with the particular value of interest in the set being the first greater than the first data word.

In this way, it is possible to avoid floating point arithmetic by increasing the value of the numerator and corresponding to the number of unused bits available in the register for shifting while further increasing the value of the numerator corresponding to the number of significant bits in the denominator to maintain the power of two of the most significant bit of the quotient equal to the power of two of the most significant bit of the dividend and thereby increasing the precision of the quotient and the answer. Additionally, it is possible to use a set of values such as a look up table for example, to compare the value of the numerator to values in the denominator, selecting as an index number related to a value in said set of values to indicating the number of positions available for shifting the numerator by arranging the set of values in an order of values corresponding to decreasing or increasing orders of the radix and successively comparing the numerator value in the order of said decreasing or increasing powers of the radix until a particular value is found which is greater or less, respectively than the numerator value. That particular value can then correspond to an index value corresponding to the power of two available in the register for shifting the numerator and increasing its value by a power of two.

As would be apparent to those skilled in the art, the inventive principles can be applied to any radix system or to any coordinate system.

TABLE III

Start Point Delta_X=50
 X₁,Y₁ = 0, 0 Delta_Y=120 Y incremental = 19660 dec.;
End Point 4CCC Hex
 X₂,Y₂ = 50, 120 Scale Factor = 8196dec.

(truncated) (rounded)		Cumulative incremental	
X,Y	Y	Y per X coordinate	
1,2	[2]	Sum=	19660 = 4CCChex
2,4	[5]	Sum=	39320 = 9998hex
3,7	[7]	Sum=	58980 = E664hex
4,9	[10]	Sum=	78640 = 13330hex
5,11	[12]	Sum=	98300 = 17FFChex
6,14	[14]	Sum=	117960 = 1CCC8hex
7,16	[17]	Sum=	137620 = 21994hex
8,19	[19]	Sum=	157280 = 26660hex
9,21	[22]	Sum=	176940 = 2B32Chex
10,23	[24]	Sum=	196600 = 2FFF8hex
11,26	[26]	Sum=	216260 = 34CC4hex
12,28	[29]	Sum=	235920 = 39990hex
13,31	[31]	Sum=	255580 = 3E65Chex
14,33	[34]	Sum=	275240 = 43328hex
15,35	[36]	Sum=	294900 = 47FF4hex
16,38	[38]	Sum=	314560 = 4CCC0hex
17,40	[41]	Sum=	334220 = 5198Chex
18,43	[43]	Sum=	353880 = 56658hex
19,45	[46]	Sum=	373540 = 5B324hex
20,47	[48]	Sum=	393200 = 5FFF0hex
21,50	[50]	Sum=	412860 = 64CBChex
22,52	[53]	Sum=	432520 = 69988hex
23,55	[55]	Sum=	452180 = 6E654hex
24,57	[58]	Sum=	471840 = 73320hex
25,59	[60]	Sum=	491500 = 77FEChex

	26,62	[62]	Sum=	511160	=	7CCB8hex
	27,64	[65]	Sum=	530820	=	81984hex
5	28,67	[67]	Sum=	550480	=	86650hex
	29,69	[70]	Sum=	570140	=	8B31Chex
	30,71	[72]	Sum=	589800	=	8FFE8hex
10	31,74	[74]	Sum=	609460	=	94CB4hex
	32,76	[77]	Sum=	629120	=	99980hex
	33,79	[79]	Sum=	648780	=	9E64Chex
	34,81	[82]	Sum=	668440	=	A3318hex
15	35,83	[84]	Sum=	688100	=	A7FE4hex
	36,86	[86]	Sum=	707760	=	ACCB0hex
	37,88	[89]	Sum=	727420	=	B197Chex
20	38,91	[91]	Sum=	747080	=	B6648hex
	39,93	[94]	Sum=	766740	=	BB314hex
	40,95	[96]	Sum=	786400	=	BFFE0hex
25	41,98	[98]	Sum=	806060	=	C4CACHex
	42,100	[101]	Sum=	825720	=	C9978hex
	43,103	[103]	Sum=	845380	=	CE644hex
	44,105	[106]	Sum=	865040	=	D3310hex
30	45,107	[108]	Sum=	884700	=	D7FDChex
	46,110	[110]	Sum=	904360	=	DCCA8hex
	47,112	[113]	Sum=	924020	=	E1974hex
35	48,115	[115]	Sum=	943680	=	E6640hex
	49,117	[118]	Sum=	963340	=	EB30Chex
	50,120	[120]	Sum=	983000	=	EFFD8hex

40

The effect of shifting to increase the scale of the Y increment is to eliminate the binary point in a third data word representing the slope and Y increment and thereby avoiding floating point arithmetic operations. The binary point separates the bit locations separating fractional from integer values in a data word (i.e. having values equal to or greater than 1 and less than 1 (i.e. between the bit positions "2⁰" and "2⁻¹". The binary point is equivalent to the decimal point in a radix 10 system and equivalent to a "point" between those machine positions having a value equal to or greater than 1 and less than one in any system and separating the fractional values from the integer values in the slope or Y increment value.

45

By shifting in the direction of more significant bits, the scale factor of Y increment is increased, thereby moving the "point" effectively in the direction of the less significant bits. If a sufficient number of bit positions are shifted, the binary point is eliminated from Y increment. In this way, floating point arithmetic is avoided. The value of Y increment at the increased scale factor may be used to derive a cumulative Y for each respective coordinate X values by iteratively incrementing Y increment by Y increment to produce a first cumulative Y and then incrementing that first cumulative Y with Y increment and so on to produce a series of distinct cumulative Y values at the higher scale factor for each X coordinate. The discrete cumulative Y values may then be reduced by the scale factor to the scale of the numerator (dx) before shifting and added to the initial Y coordinate value to produce the correct Y coordinate for each respective X coordinate. By producing the cumulative Y using Y increment at the increased scale factor, errors in the Y coordinate value due to an error in the cumulative Y increment due to the lower scale factor of Y increment are

50

55

avoided.

The compiler used in the preferred embodiment to perform the method described herein is shown in the following program listings written in Motorola 68000 assembler language and contained within six modules.

5 The first module is the main program utilizing the control modules shown as modules 2-7. Its function is to access the pack data and interpret the knot locations and control codes therein.

Module 2 is a subset of module 1 and is used to evaluate each complete information set, identifying the Case and control code value thereof.

10 Module 3 is the compiler described earlier, which functions according to equations 3.1 and 3.2. and which compiles the knot locations, the angles of the curve at the knots, and the related values of the parameter t to produce the node locations on the curve segment locus.

Module 4 is the compiler which operates according to the method of auto-scaling linear interpolation shown herein.

15 Module 5 is a module which receives curve coordinate data, corresponding to the display coordinate system and sorts this data in the order of the raster lines on the display, so that the display data is accessed in timed relation to the generation of the raster lines, with data for display on any one particular raster line accessed in time with the location of the imaging beam at that raster line.

Module 6 is a general purpose memory allocation and release mechanism for buffer and raster line data.

20 Module 7 is an apparatus for performing general trigonometry.

As said, these modules are written in 68000 assembler code as used in the preferred embodiment to perform the invention as described. The language for the preferred embodiment is further compiled into machine object language for use on the Motorola 68000.

25 With respect to the system generally shown in Figure 10, there is illustrated a general type of apparatus with which the invention can be practiced. More particularly, in one aspect the CPU 1 is a microprocessor arrangement made up of, for example, a Motorola 68000 processor. The processor will be used to control the encoded information in data store 5. The data store can reside in part, for example, Magnetic Disc, CD ROM, Magnetic Tape, ROM, RAM, Networked Data Base Systems, other magnetic media or the like encodable media. It is to be understood that the above-identified specific elements are encompassed in
30 said data store 5. Control and initiation of the process can be effected, for example, by a keyboard 3 through which instructions can be given. The processor controls the imaging apparatus 7.

In a specific aspect, the arrangement described is especially adapted for use as a phototypesetter of the type using conventional laser raster scan technology. Alternatively, the system of Figure 10 can also correspond to a desktop computer with CRT display.

35

40

45

50

55

Copyright (c) 1983 Allied Corporation All Rights Reserved
Module -I-

38	9:00001A	4EB900000000	NEW_LOOP: EQU	*	GETNIB34
39	9:00001A	661A	JSR		ORG_OK
40	9:000020	4EB900000000	BNE.B		GETNIB34
41	9:000022	4EB900000000	JSR		D7,X_HOME(A6)
42	9:00002B	3D47FFFE	MOVE.W		GETNIB34
43	9:00002C	4EB900000000	JSR		D7,Y_HOME(A6)
44	9:000032	3D47FFFC	MOVE.W		GETNIB34
45	9:000036	4EB900000000	ORG_OK: EQU	*	
46	9:00003C	E24F	LSR.W		#1,D7
47	9:00003C	55EEFFFA	SCS		X_DIR(A6)
48	9:00003E	9E6EFFFF	SUB.W		X_HOME(A6),D7
49	9:000042	3007	MOVE.W		D7,D0
50	9:000046	4EB900000000	JSR		GETNIB34
51	9:000048	E24F	LSR.W		#1,D7
52	9:00004E	55EEFFF9	SCS		Y_DIR(A6)
53	9:000050	9E6EFFFF	SUB.W		Y_HOME(A6),D7
54	9:000054	3207	MOVE.W		D7,D1
55	9:000058	4B47	SWAP		D7
56	9:00005A	4EB900000000	MOVE.W		D0,D7
57	9:00005C	3E00	JSR		INIT_HERMIT
58	9:00005E	4EB900000000	MOVE.W		D0,D2
59	9:000064	3409	MOVE.W		D1,D3
60	9:000066	3601	MOVE.W		D0,D4
61	9:000068	7800	MOVE.W		D0,D4
62	9:00006A	4EB900000000	BASE_KNOT: EQU	*	GET_F0
63	9:000070	4A47	TST.W		D7
64	9:000072	674A	REQ.B		D0_CONTROL
65	9:000074	4A2EFFFF	TST.B		X_DIR(A6)
66	9:000078	6704	REQ.B		INC_X
67	9:00007A	9447	SUB.W		D7,D2
68	9:00007C	6902	BRA.B		RELY
69	9:00007E	D447	INC_X:		ADD.W
70	9:000080	4A2EFFFF	TST.B		Y_DIR(A6)
71	9:000084	6704	REQ.B		INC_Y
72	9:000086	9646	SUB.W		D6,D3
73	9:000088	6002	BRA.B		BEND_KNOT
74	9:00008A	D646	INC_Y:		ADD.W
75	9:00008C	3002	BASE_KNOT: EQU	*	
76	9:00008C	3002	MOVE.W		D2,D0
77	9:00008E	3203	MOVE.W		D3,D1
78	9:000090	3C04	MOVE.W		D4,D6
79	9:000092	3E01	MOVE.W		D1,D7
80	9:000094	4B47	SWAP		D7
81	9:000096	3E00	MOVE.W		D0,D7
82	9:000098	4EB900000000	JSR		HERMIT
83	9:00009E	60C4	BRA.B		NEXT_KNOT

Copyright (c) 1983 Allied Corporation All Rights Reserved
Module - I -

8	00000000	END_LAST: EQU *	
9	00000000	ADDQ.B #1, FINAL(A6)	
10	00000000	END_LOOP: EQU *	
11	00000000	JSR FINAL_HERMITE	
12	00000000	BNE.S LOOP_DONE	
13	00000000	TST.B FINAL(A6)	
14	00000000	BEQ NEW_LOOP	
15	00000000	LOOP_DONE: MOVE.W (SP)+, D0/D1/D2/D3/D4	
16	00000000	MOVE.L (SP)+, A6	
17	00000000	UNLK A6	
18	00000000	RTS	
19	00000000		
20	00000000	DO_CONTROL: EQU *	
21	00000000	MOVE.W D6, D7	
22	00000000	BMI.S CASE_DEFAULT	
23	00000000	CMPL.W #8, D7	
24	00000000	BLE.S CASE_OK	
25	00000000	CASE_DEFAULT: EQU *	
26	00000000	MOVED.L #8, D7	
27	00000000	CASE_OK: EQU *	
28	00000000	ADD.W D7, D7	
29	00000000	JMP CASE_TBL(PC, D7.W)	
30	00000000		
31	00000000	CASE_TBL:	
32	00000000	BRA.S END_LAST	
33	00000000	BRA.S HORIZONTAL	
34	00000000	BRA.S VERTICAL	
35	00000000	BRA.S DIAGONAL	
36	00000000	BRA.S END_LOOP	
37	00000000	BRA.S X_NEGATE	
38	00000000	BRA.S Y_NEGATE	
39	00000000	BRA.S XY_NEGATE	
40	00000000	BRA.S DEFAULT	
41	00000000		
42	00000000		
43	00000000		
44	00000000		
45	00000000		
46	00000000		
47	00000000		
48	00000000		
49	00000000		
50	00000000		
51	00000000		
52	00000000		
53	00000000		
54	00000000		
55	00000000		
56	00000000		
57	00000000		
58	00000000		
59	00000000		
60	00000000		
61	00000000		
62	00000000		
63	00000000		
64	00000000		
65	00000000		
66	00000000		
67	00000000		
68	00000000		
69	00000000		
70	00000000		
71	00000000		
72	00000000		
73	00000000		
74	00000000		
75	00000000		
76	00000000		
77	00000000		
78	00000000		
79	00000000		
80	00000000		
81	00000000		
82	00000000		
83	00000000		
84	00000000		
85	00000000		
86	00000000		
87	00000000		
88	00000000		
89	00000000		
90	00000000		
91	00000000		
92	00000000		
93	00000000		
94	00000000		
95	00000000		
96	00000000		
97	00000000		
98	00000000		
99	00000000		
100	00000000		
101	00000000		
102	00000000		
103	00000000		
104	00000000		
105	00000000		
106	00000000		
107	00000000		
108	00000000		
109	00000000		
110	00000000		
111	00000000		
112	00000000		
113	00000000		
114	00000000		
115	00000000		
116	00000000		
117	00000000		

Copyright (c) 1983 Allied Corporation All Rights Reserved

0:0000E2	0:0000E2	HORIZONTAL: EQU *
0:0000E2	0:0000E2	DIAGONAL: EQU *
0:0000E4	1006	MOVE.B D6,D0
0:0000EA	4EB900000000	GETNIB34
0:0000EC	F24F	LSR.W #1,D7
0:0000F0	55FFFFFFA	SCS X_DIR(A6)
0:0000F4	9F6EFFFE	SUB.W X_ROME(A6),D7
0:0000F6	3407	MOVE.W D7,D2
0:0000FB	5300	SUBQ.B #1,D0
0:0000FA	6792	BFX SEND_KNOT
0:0000FA	4FB900000000	VERTICAL: EQU *
0:000100	F24F	GETNIB34
0:000102	55FFFFFF9	LSR.W #1,D7
0:000106	9F6EFFFC	SCS Y_DIR(A6)
0:00010A	3607	SUB.W Y_ROME(A6),D7
0:00010C	600FF7E	MOVE.W D7,D3
0:000110	462FFFFA	BRA SEND_KNOT
0:000110	600FF54	X_NEGATE: EQU *
0:000114	462FFFFA	NOT.B X_DIR(A6)
0:000118	462FFFFA	BRA SAME_KNOT
0:00011B	462FFFFA	XY_NEGATE: EQU *
0:00011C	462FFFF9	NOT.B X_DIR(A6)
0:00011C	600FF48	Y_NEGATE: EQU *
0:000120	7001	NOT.B Y_DIR(A6)
0:000124	C006	BRA SAME_KNOT
0:000126	5204	DEFAULT: EQU *
0:00012B	E14C	MOVEQ.L #1,D4
0:00012A	02060002	AND.B D6,D4
0:00012C	F206	ADDQ.B #1,D4
0:000130	5206	LSL.W #8,D4
0:000132	1805	ANDI.B #2,D6
0:000134	600FF32	ASR.B #1,D6
0:000136		ADDQ.B #1,D6
		MOVE.B D6,D4
		BRA SAME_KNOT
		END

38

[illegible]

PT	3	N I B B L E S	+	2	C E T - N I B S
105	910000C6	7F02		107	GET_NIBS1: NOVEDAL #2, D7
106	910000C8	CE6D0028		108	AND.W NIB_CNT(AS), D7
109	910000CC	6700		109	BDQ.0 CNB3.1
110	910000CE	61A4		110	BSR GET_NIB1
111	910000D0	3F07		111	MOVE.W D7, -(SP)
112	910000D2	61CC		112	BSR GET_NIB2
113	910000D4	60EA		113	BNA CNB3.0
114	910000D6	61C8		114	BSR GET_NIB2
115	910000D8	3F07		115	MOVE.W D7, -(SP)
116	910000DA	6198		116	BSR GET_NIB1
117	910000DC	E14F		117	BSR GET_NIB1
118	910000DE	BE3F		118	LSL.W #8, D7
119	910000E0	4E75		119	OR.W (BP)+, D7
120				120	RTS
121				121	
122				122	C E T N I B S 4
123	910000E2	7Z01		123	CETNIB34: NOVEDAL #1, D7
124	910000E4	CE6D0028		124	AND.W NIB_CNT(AS), D7
125	910000E8	67DC		125	BDQ GET_NIB3
126				126	BNA GET_NIB4
127				127	
128				128	C E T - N I B 4
129				129	
130	910000EA	61B4		130	GET_NIB4: BSR GET_NIB2
131	910000EC	3F07		131	MOVE.W D7, -(SP)
132	910000EE	61D9		132	BSR GET_NIB2
133	910000F0	60EA		133	BNA CNB4.0
134				134	
135				135	END

Copyright (c) 1983 Allied Corporation All Rights Reserved
Module -III-

LINE	CODE	TEXT	LINE	CODE	TEXT	LINE	CODE	TEXT
2		HERMITE INT	1		1.0 CURVE FORT HERMITE Spline Generator			
6								ENTRY POINTS
7		XDEF			INIT HERMITE,HERMITE,FINAL,HERMITE			STATIC SUBROUTINE
8		XDEF			SCALE,SET_BETA,CARMA_ALPHA,MAKE_NODES			VARIABLES
9		XDEF			SCALE,PREV,CUR,NEXT,FIRST,SECOND			
10		XDEF			PLOT_CHECK			
11	0000007F	LINEAR_INFLIED:	SET	127				
12	00000001	ISLINEAR:	SET	1				
13	00000002	CURVE:	SET	2				
14	00000030	CURP_INFLIED:	SET	40*2				
15	00000001	ISCUSP:	SET	1				
16	00000002	CORTIGIOUS:	SET	2				
17	0000005A	HDEC_40:	SET	40*2				
18	00000004	HDEC_90:	SET	90*2				
19	00000016	HDEC_180:	SET	180*2				
20	00004000	R_8_2:	SET	04000				
21	00007FFF	R_8_4:	SET	07FFF				
22	00000003	T_TOOFAR:	SET	3				
23	00000006	T_INCREMENT:	SET	3*SHORT				
24	0000000A	T_LENGTH:	SET	5*SHORT				
25	00000000	X1_LIMIT:	SET	1024*SHORT				
26	00000066	GAINSV_SCALE:	SET	1638				
27	00000000	GAINR_SCALE:	SET	0				GAIN SCALE FACTOR
28	00000008	GAINV_SCALE:	SET	0				
29	000000FF	R_8SV_SCALE:	SET	11<<8-1><<0				R/S SCALE
29	0000000D	R_8SR_SCALE:	SET	13				
29	00001FFF	SINRSV_SCALE:	SET	11<<13-1><<0				SINE/COSINE SCALE
30	00000000	SINRSH_SCALE:	SET	0				
30	00007FFF	TINRSH_SCALE:	SET	15				
31	00000000	TINRSV_SCALE:	SET	11<<15-1><<0				TIME FUNCTION SCALE
31	0000000F	TINRSH_SCALE:	SET	0				
31	00007FFF	TINRSH_SCALE:	SET	15				
31	00007FFF	TINRSH_SCALE:	SET	11<<15-1><<0				

42

Copyright (c) 1983 Allied Corporation All Rights Reserved
Module -III-

63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97																																
																																SCALE	SECTION 99																																	
																																SCALE:	EQU	MOVE.L D0/D1, -(SP)																																
																																48E7C000	MOVE.W DRU_Y(A1), D1																																	
																																32290004	MOVE.W DRU_X(A1), D0																																	
																																30290006	MOVE.W BLANT(A5), D7																																	
																																3E210004	BFG.S SCAL_1																																	
																																6712	JNR	SINE																																
																																4E00000000	MULS D1, D7																																	
																																CFC1	ASL.L #16-SINE06_SCALE, D7																																	
																																E387	SWAP D7																																	
																																4047	ADD.W D7, D0																																	
																																33470006	MOVE.W D7, DRU_X(A1)																																	
																																9261000C	ROT_Y(A5), D1																																	
																																9061000E	SUB.W ROT_X(A5), D0																																	
																																3F210006	MOVE.W ROTATE(A5), D7																																	
																																6752	BFG.S SCAL_6																																	
																																0C4700B4	CHP1.W #HDEC_90, D7																																	
																																6606	BNE.S SCAL_2																																	
																																C141	EXG D0, D1																																	
																																444A	NEG.W D0																																	
																																6046	BRA.S SCAL_6																																	
																																0C47FF4C	#-HDEC_90, D7																																	
																																6606	BNE.S SCAL_3																																	
																																C141	EXG D0, D1																																	
																																4441	NEG.W D1																																	
																																603A	BRA.S SCAL_6																																	
																																0C470168	#HDEC_100, D7																																	
																																6706	BFG.S SCAL_4																																	
																																0C47FE58	#-HDEC_100, D7																																	
																																6606	BNE.S SCAL_5																																	
																																4440	NEG.W D0																																	
																																4441	NEG.W D1																																	
																																6028	BRA.S SCAL_6																																	

Copyright (c) 1983 Allied Corporation All Rights Reserved
Module -III-

LOC	ADDRESS	INSTR	OPERAND	COMMENT
99	9:000072	BLANT	4E00000000000000	
100	9:00007B	3F07		
101	9:00007A	3F2D0000		
102	9:00007E	4E00000000000000		
103	9:000004	3C00		
104	9:000006	C0C7		
105	9:000008	CFC1		
106	9:00000A	C3D7		
107	9:00000C	D286		
108	9:00000E	E3B1		
109	9:000000	4B41		
110	9:000092	C1DF		
111	9:000094	94B7		
112	9:000096	E3B9		
113	9:000098	4B49		
114	9:00009A	D06D0012		
115	9:00009E	C1E1000A		
116	9:0000A2	F0B9		
117	9:0000A4	D06D0016		
118	9:0000A8	33400002		
119	9:0000AC	D26D0010		
120	9:0000B0	C3ED0008		
121	9:0000B4	F0B1		
122	9:0000B6	D26D0014		
123	9:0000BA	32B1		
124	9:0000BC	4C0F0003		
125	9:0000C0	4E75		

LOC	ADDRESS	INSTR	OPERAND	COMMENT
126	9:000072	BLANT	4E00000000000000	
127	9:00007B	3F07		
128	9:00007A	3F2D0000		
129	9:00007E	4E00000000000000		
130	9:000004	3C00		
131	9:000006	C0C7		
132	9:000008	CFC1		
133	9:00000A	C3D7		
134	9:00000C	D286		
135	9:00000E	E3B1		
136	9:000000	4B41		
137	9:000092	C1DF		
138	9:000094	94B7		
139	9:000096	E3B9		
140	9:000098	4B49		
141	9:00009A	D06D0012		
142	9:00009E	C1E1000A		
143	9:0000A2	F0B9		
144	9:0000A4	D06D0016		
145	9:0000A8	33400002		
146	9:0000AC	D26D0010		
147	9:0000B0	C3ED0008		
148	9:0000B4	F0B1		
149	9:0000B6	D26D0014		
150	9:0000BA	32B1		
151	9:0000BC	4C0F0003		
152	9:0000C0	4E75		

Copyright (c) 1983 Allied Corporation All Rights Reserved
Module -III-

H E R M I T E		H E R M I T E		* CODE SECTION
135	910000C2	135	SECTION 09	
136	910000C2	136	HERMITE: EQU	
137	910000C2	137	MOVE.L A1/A2, -(SP)	
138	910000C2	138	LEA.L CUR.A2	
139	910000C6	139	TST.W KNOT_CMT(A5)	
140	910000C6	140	BNE.S HRMT_1	
141	910000C6	141	MOVE.L D7, DRU_Y+CUR-CUR(A2)	
142	910000D0	142	MOVE.W D6, CURP+PREV-CUR(A2)	
143	910000D2	143	LEA.L (A2), A1	
144	910000D6	144	BSR SCALE	
145	910000DA	145	LEA.L PREV-CUR(A2), A1	
146	910000DC	146	BSR SET_BETA	
147	910000E0	147	BRA.S HRMT_0	
148	910000E4	148	MOVE.L D7, DRU_Y+NEXT-CUR(A2)	
149	910000E8	149	MOVE.W D6, CURP+CUR-CUR(A2)	
150	910000EA	150	LEA.L NEXT-CUR(A2), A1	
151	910000EE	151	BSR SCALE	
152	910000F2	152	EXG A2, A1	
153	910000F6	153	BSR SET_BETA	
154	910000FA	154	MOVED.L #NEXT-CUR, D6	
155	910000FC	155	LEA.L (A1), A2	
156	91000100	156	LEA.L PREV-CUR(A2), A1	
157	91000102	157	BSR CANTA_ALPHA	
158	91000104	158	BNE.S HRMT_4	
159	91000108	159	CMPL.W #2-1, KNOT_CMT(A5)	
160	9100010C	160	BNE.S HRMT_8	
161	9100010E	161	BNE.S MAKE_NODES	
162	91000114	162		
163	91000116	163		

Copyright (c) 1983 Allied Corporation All Rights Reserved
Module - III -

```

H E R M I T E
165 9:00011A 43EAFFF0
166 9:00011E 22DA
166 9:000120 22DA
166 9:000122 22DA
166 9:000124 22DA
169 9:000126
170 9:000126 22DA
170 9:000128 22DA
170 9:00012A 22DA
170 9:00012C 22DA
171 9:00012E 4CDF0600
172 9:000132 526D002A
173 9:000136 4E75
174 9:000138
175 9:000138 45EA0010
176 9:00013C 6148
177 9:00013E 43EAFFF0
178 9:000142 60E2
179 9:000144 3F2A000A
180 9:000148 9E69000A
181 9:00014C 4FB90000000000
182 9:000152 5EE0002F
184 9:000156 3E12
185 9:000158 BF51
191 9:00015A 50C7
192 9:00015C 6606
193 9:00015E BF2D002F
194 9:000162 57C7
195 9:000164 1B47002E
196 9:000168 43EA0020
197 9:00016C 43EAFFF0
198 9:000170 22DA
198 9:000172 22DA
198 9:000174 22DA
198 9:000176 22DA
201 9:000178 22DA
201 9:00017A 22DA
201 9:00017C 22DA
201 9:00017E 22DA
201 9:000180 43EAFFF0
202 9:000184 6094

B R N T _ 3 : L E A . L P R E V - C U R ( A 2 ) , A 1
              M O V E . L ( A 2 ) + , ( A 1 ) +
              M O V E . L ( A 2 ) + , ( A 1 ) +
              M O V E . L ( A 2 ) + , ( A 1 ) +
              M O V E . L ( A 2 ) + , ( A 1 ) +
B R N T _ 3 : E Q U *
              M O V E . L ( A 2 ) + , ( A 1 ) +
              M O V E . L ( A 2 ) + , ( A 1 ) +
              M O V E . L ( A 2 ) + , ( A 1 ) +
              M O V E . L ( A 2 ) + , ( A 1 ) +
B R N T _ 3 : M O V E . L ( A 2 ) + , ( A 1 ) +
              M O V E . L ( 8 F ) + , A 1 / A 2
              A D D Q . W # 1 , K R O T _ C N T ( A 5 )
              R T S
B R N T _ 4 : E Q U *
              L E A . L N E X T - C U R ( A 2 ) , A 3
              D S R . 8 S E T _ B E T A
              L E A . L C U R - N E X T ( A 2 ) , A 1
              B R A B R N T _ 3
B R N T _ 5 : M O V E . W B E T A ( A 2 ) , D 7
              S U B . W B E T A ( A 1 ) , D 7
              J S R F O L D _ A N G L E
              S C T I N I T _ C O L O R ( A 5 )
              M O V E . W ( A 2 ) , D 7
              C M P . W ( A 1 ) , D 7
              S L T D 7
              B R E . 6 B R N T _ 6
              C M P . B I N I T _ C O L O R ( A 5 ) , D 7
              S E Q D 7
B R N T _ 6 : M O V E . B D 7 , C O L O R ( A 5 )
              L E A . L F I R S T - C U R ( A 2 ) , A 1
              L E A . L P R E V - C U R ( A 2 ) , A 2
              M O V E . L ( A 2 ) + , ( A 1 ) +
              M O V E . L ( A 2 ) + , ( A 1 ) +
              M O V E . L ( A 2 ) + , ( A 1 ) +
              M O V E . L ( A 2 ) + , ( A 1 ) +
              M O V E . L ( A 2 ) + , ( A 1 ) +
              M O V E . L ( A 2 ) + , ( A 1 ) +
              M O V E . L ( A 2 ) + , ( A 1 ) +
              M O V E . L ( A 2 ) + , ( A 1 ) +
              L E A . L - ( K R O T _ L E N G T H ) ( A 2 ) , A 3
              B R A B R N T _ 3

```

Y T	B E T A	+	SET_BETA EQU	SECTION 09	* CODE SECTION
205	9:000106				
206	9:000106				
207	9:000106				
208	9:000106		3E2A0004	MOVE.W DRU_Y(A2),D7	
209	9:00010A		9E690004	SUB.W DRU_Y(A1),D7	
210	9:00010E		4047	SWAP D7	
211	9:000190		3E2A0006	MOVE.W DRU_X(A2),D7	
212	9:000194		9E690006	SUB.W DRU_X(A1),D7	
213	9:000198		2F07	MOVE.L D7, -(BP)	
214	9:00019A		4EB900000000	JSR ARCTAN	
215	9:0001A0		DF6D0006	ADD.W ROTATE(A5),D7	
216	9:0001A4		3347000A	MOVE.W D7,BETA(A1)	
217	9:0001A8		4A29000F	TST.B LINEAR(A1)	
218	9:0001AC		6622	BNE.B SBETA_4	
219	9:0001AE		3E1F	MOVE.W (BP)+,D7	
220	9:0001B0		6A02	BPL.B ++4	
220	9:0001B2	+	4447	NEG.W D7	
221	9:0001B4	+	0C47007F	CMPL.W *LINEAR_IMPLIED,D7	
222	9:0001B8		620E	BHI.B SBETA_3	
223	9:0001BA		3E1F	MOVE.W (BP)+,D7	
224	9:0001BC		6A02	BPL.B ++4	
224	9:0001BE	+	4447	NEG.W D7	
225	9:0001C0	+	0C47007F	CMPL.W *LINEAR_IMPLIED,D7	
226	9:0001C4		6204	BHI.B SBETA_3	
227	9:0001C6		4E75	RTS	
228	9:0001C8			SBETA_3 EQU	
229	9:0001CB	+	548F	ADDQ.L *SHORT,0F	
230	9:0001CA		5229000F	SBETA_3 ADDQ.B	
231	9:0001CE		4E75	RTS	
232	9:0001D0			SBETA_4 EQU	
233	9:0001D0	+	588F	ADDQ.L *LONG,0F	
234	9:0001D2		4E75	RTS	

48

total - 111 -

А М М А А Л Р Н А

© 1983 Allied Corporation All Rights Reserved

A	L	P	H	A	CHAL_4:	TST.B	CUSP(A2)
0000272	4A2A000E				BRE.B	BRE.B	CHAL_6
0000276	6622				MOVE.W	MOVE.W	BETA(A2),D7
0000278	3E2A000A				SUB.W	SUB.W	BETA(A1),D7
000027C	9E6A000A				JSR	JSR	FOLD_ANGLE
0000280	4EB000000000				TST.W	TST.W	D7
0000286	4A47				BPL.B	BPL.B	S++
000028B	6A02				NEG.W	NEG.W	D7
000028A	4447				CMPI.W	CMPI.W	CUSP_INFLIED,B7
000028C	0C470050				BLS.B	BLS.B	CHAL_5
0000290	6304				SUBQ.B	SUBQ.B	CONTIGIOUS-16CUSP,CUSP(A2)
0000292	532A000E				AUDQ.B	AUDQ.B	CONTIGIOUS,CUSP(A2)
0000296	542A000E				TST.W	TST.W	SLANT(A8)
000029A	4A6D0004				BRE.B	BRE.B	CHAL_6A
000029E	660A				MOVE.W	MOVE.W	CAIR_Y(A8),D7
00002A0	3E2A000B				CMPI.W	CMPI.W	CAIR_X(A8),D7
00002A4	BE6D000A				BPL.B	BPL.B	CHAL_8
00002AB	673E				CMPI.W	CMPI.W	2-1,KNOT_CNT(A8)
00002AA	0C6D0001002A				BRE.B	BRE.B	CHAL_7
00002B0	661B				MOVE.W	MOVE.W	(A2),D7
00002B2	3E12				SUB.W	SUB.W	(A1),D7
00002B4	9E51				SWAP	SWAP	D7
00002B6	4B47				MOVE.W	MOVE.W	HIS_X(A2),D7
00002B8	3E2A0002				SUB.W	SUB.W	HIS_X(A1),D7
00002BC	9E6A0002				JSR	JSR	ARCTAN
00002C0	4EB000000000				MOVE.W	MOVE.W	D7,BETA(A1)
00002C6	3347000A				TST.L	TST.L	D4
00002CA	4AB6				BPL.B	BPL.B	CHAL_8
00002CC	671A				MOVE.W	MOVE.W	HIS_Y(A2,D4.L),D7
00002CE	3E326000				SUB.W	SUB.W	(A2),D7
00002D2	9E52				SWAP	SWAP	D7
00002D4	4B47				MOVE.W	MOVE.W	HIS_X(A2,D4.L),D7
00002D6	3E326002				SUB.W	SUB.W	HIS_X(A2),D7
00002DA	9E6A0002				JSR	JSR	ARCTAN
00002DE	4EB000000000				MOVE.W	MOVE.W	D7,BETA(A2)
00002E4	3347000A						

© 1983 Allied Corporation All Rights Reserved

A L P H A	CHAL_8	CHP1_8	CONTIGIOUS, CUSP(A2)
02FB 0C2A0002000E		BNE_8	CHAL_13
02FE 6656		MOVE.W	BETA(A2), D7
02F0 3F2A000A		ADD.W	BETA(A1), D7
02F4 0F69000A		ASR.W	#1, D7
02FH F247		MOVE.W	D7, ALPHA(A2)
02FA 3547000B		MOVE.W	BETA(A2), D7
02FE 3F2A000A		SUB.W	BETA(A1), D7
0302 9F69000A		BPL_8	#4
0306 6A02		REG.W	D7
030B 4447		CHP1_8	HDEG_180, D7
030A 0C47016B		RLS_8	CHAL_10
030E 6312		MOVE.W	ALPHA(A2), D7
0310 3F2A000B		RM1_8	CHAL_9
0314 6B04		SUB.W	HDEG_180#2, D7
0316 044702D0		ADD.W	HDEG_180, D7
031A 0647016B		MOVE.W	D7, ALPHA(A2)
031E 3547000B		MOVE.W	ALPHA(A2), GAMMA(A1)
0322 336A000B000C		MOVE.W	LINEAR(A1), D7
032B 7F01		CHP_8	CHAL_11
032A 0F29000F		BNE_8	BETA(A1), ALPHA(A2)
032E 6606		MOVE.W	LINEAR(A2), D7
0330 3569000A000B		CHP_8	CHAL_12
0336 0F2A000F		BNE_8	BETA(A2), GAMMA(A1)
033A 6606		MOVE.W	TRUE, D7
033C 336A000A000C		RTS	
0342 7F01		MOVE.W	BETA(A1), GAMMA(A1)
0344 4E73		MOVE.W	BETA(A2), ALPHA(A2)
0346 3369000A000C		ERA	CHAL_12
034C 356A000A000B			
0352 60EE			

TAKE	NODES
405	
406	
407	
408	00000000
409	00000000
409	FFFFFFFF
410	00000002
410	FFFFFFFF
411	FFFFFFFF
412	00000006
412	FFFFFFFF
413	
414	
415	
416	

```

*
*
*LOCALS
MAKE_NODES
OFFSET
D8.W 1
SET --
D8.W 1
SET --
D8.W 1
SET --
D8.W 0
SET --
NTRPLS
ARCTAN,SINE,
TIME1,TBL,

```

```

DOBLE
DOBLE
DOBLE
DOBLE
      NTRPL6
      ARCTAN, SIN, COSINE, POLD_ANGLE,
      TIME_1_TBL, TIME_2_TBL, XI_TBL

```

Copyright (c) 1983 Allied Corporation All Rights Reserved
Module -III-

1 A K E	N O D E S	+	SECTION 09	* CODE SECTION
410	9:000354		UNLK A6	
419	9:000354	4E5E	MOVE.W (SP)+,D0/D1/D2/D3/D4/D5	
420	9:000356	4C9F003F	MOVE.W (A2),D6	
421	9:00035A		MOVE.W M18_X(A2),D7	
423	9:00035A	3C12	JSR M18_PL8	
432	9:00035C	3F2A0002	RTS	
434	9:000360	4F8900000000		
435	9:000366	4E73		
436				
437	9:000368		MOVE.W (A1),PLOT_Y(A8)	
439	9:000368	2D510020	CMPI.B #18,LINEAR,LINEAR(A1)	
444	9:00036C	0C290001000F	BEQ MKND_1	
445	9:000372	67E6	MOVE.W BETA(A1),D7	
446	9:000374	3F29000A	CMPI.W ALPHA(A1),D7	
447	9:000378	BF690008	BNE.B MKND_3	
448	9:00037C	6606	CMPI.W CANTA(A1),D7	
449	9:00037E	BE69000C	BEQ MKND_1	
430	9:000382	67D6		

A	K	E	N	O	D	E	R
452	9:000304	40A7FC00					
453	9:000308	4F56FFFA					
454	9:00030C	9E6A00B					
455	9:000309	4EB900000000					
456	9:000306	3007					
457	9:00030B	3F29000C					
458	9:00030C	9E69000A					
459	9:000309	4EB900000000					
460	9:000306	3207					
461	9:00030B	6602					
462	9:00030A	3200					
463	9:00030C	4A40					
464	9:00030E	6602					
465	9:00030B	3001					
466	9:000302	3E06					
467	9:000304	DE41					
468	9:000306	F247					
469	9:00030B	6A02					
469	9:00030A	4447					
470	9:00030C	0C47003A					
471	9:000300	6306					
472	9:000302	4447					
473	9:000304	064700B4					
474	9:00030B	41F900000000					
475	9:00030E	DE47					
476	9:000309	3E307000					
477	9:00030D	3407					
478	9:000306	0C470666					
479	9:00030A	640B					
480	9:00030C	320C4000					
481	9:000300	3001					
482	9:000302	6042					
483	9:000304	7C00					
484	9:000306	3C02					
485	9:00030B	E306					
486	9:00030A	3E01					
487	9:00030C	4EB900000000					
488	9:000302	6A02					
488	9:000304	4407					
489	9:000306	BFB6					
490	9:00030B	6506					
491	9:00030A	3E3C7FFF					
492	9:00030E	6006					

Copyright (c) 1983 Allied Corporation All Rights Reserved
Module -III-

HAKE	NO	DES				
494	9:000400	4847				
495	9:000402	F607				
496	9:000404	F6C2				
497	9:000406	3207				
498	9:000408	3F00				
499	9:00040A	4F09000000000000				
500	9:000410	6A02				
500	9:000412	4407				
501	9:000414	BE06				
502	9:000416	6506				
503	9:000418	3E307FFF				
504	9:00041C	6006				
505	9:00041E	4047				
506	9:000420	F607				
507	9:000422	F6C2				
508	9:000424	3007				
509	9:000426					
514	9:000426	3C2A0002				
516	9:00042A	9C6D0022				
517	9:00042E	3D46FFFE				
518	9:000432	6A02				
518	9:000434	4446				
520	9:000436	3E12				
523	9:000438	9E6D0020				
526	9:00043C	3D47FFFC				
527	9:000440	6A02				
527	9:000442	4447				
528	9:000444	BE46				
529	9:000446	620C				
530	9:000448	3E29000A				
531	9:00044C	4F09000000000000				
532	9:000452	600C				
533	9:000454	3C07				
534	9:000456	3E29000A				
535	9:00045A	4E09000000000000				
536	9:000460					
537	9:000460	6A02				
537	9:000462	4407				
538	9:000464	4046				
539	9:000466	4246				
540	9:000468	F206				
541	9:00046A	H0C7				
542	9:00046C	3406				
543	9:00046E	CC01				
544	9:000470	E306				
544	9:000472	4046				
545	9:000474	3206				
546	9:000476	3E02				
547	9:000478	CEC9				
548	9:00047A	F307				
548	9:00047C	4047				
549	9:00047E	3007				

MCMD	OP	SWAP	D7	#16-#808_SCALE,D7
MCMD_6:	ASR.L	D7	#16-#808_SCALE,D7	
	DIVU	D2,D7		
MCMD_7:	MOVE.W	D7,D1		
	MOVE.W	D0,D7		
	JSR	SIRE		
	RPL.S	#+4		
	NEG.L	D7		
	CHP.L	D6,D7		
	HCS.S	MCMD_8		
	MOVE.W	#R8-4,D7		
	BRA.S	MCMD_9		
MCMD_8:	SWAP	D7		
	ASR.L	#16-#808_SCALE,D7		
	DIVU	D2,D7		
MCMD_9:	MOVE.W	D7,D0		
MCMD_10:	EXU	*		
	MOVE.W	H18_X(A2),D6		
	SUB.W	PLOT_X(A5),D6		
	MOVE.W	D6,X2_X1(A6)		
	RPL.S	#+4		
	NEG.W	D6		
	MOVE.W	(A2),D7		
	SUB.W	PLOT_Y(A5),D7		
	MOVE.W	D7,Y2_Y1(A6)		
	RPL.S	#+4		
	NEG.W	D7		
	CHP.W	D6,D7		
	RHI.S	MCMD_11		
	MOVE.W	BETA(A1),D7		
	JSR	COSINE		
	BRA.S	MCMD_12		
MCMD_11:	MOVE.W	D7,D6		
	MOVE.W	BETA(A1),D7		
	JSR	SIRE		
MCMD_12:	EXU	*		
	RPL.S	#+4		
	NEG.L	D7		
	SWAP	D6		
	CLR.W	D6		
	ASR.L	#16-#808_SCALE,D6		
	DIVU	D7,D6		
	MOVE.W	D6,D2		
	MULU	D1,D6		
	ASL.L	#16-#808_SCALE+2,D6		
	SWAP	D6		
	MOVE.W	D6,D1		
	MOVE.W	D2,D7		
	MULU	D0,D7		
	ASL.L	#16-#808_SCALE+2,D7		
	SWAP	D7		
	MOVE.W	D7,D0		

Copyright (c) 1983 Allied Corporation All Rights Reserved
Module -III-

MAK E	N O D E S						
551	9:000404	4E900000000000					
552	9:000404	CFC1					
553	9:00040C	E787					
554	9:00040E	4047					
555	9:000490	3607					
556	9:000492	3F290000					
557	9:000496	4E900000000000					
558	9:00049C	CFC1					
559	9:00049E	E787					
560	9:0004A0	4047					
561	9:0004A2	3207					
562	9:0004A4	3F290000					
563	9:0004A6	4E900000000000					
564	9:0004AF	CFC0					
565	9:0004B0	E787					
566	9:0004B2	4047					
567	9:0004B4	3007					
568	9:0004B6	3F290000					
569	9:0004B8	4E900000000000					
570	9:0004C0	CFC0					
571	9:0004C2	E787					
572	9:0004C4	4047					
573	9:0004C6	3007					
574	9:0004C8	2F3C0000000000					
575	9:0004CE	0FC2					
576	9:0004D0	5247					
577	9:0004D2	0247FFFE					
578	9:0004D6	6602					
579	9:0004D8	7F02					
580	9:0004DA	3D47FFFA					
581	9:0004DE	4245					

JSR	COSINE
MULS	D1, D7
ASL.L	#16-(SIN208_SCALE-2), D7
SWAP	D7
MOVE.W	D7, D3
MOVE.W	ALPHA(A1), D7
JSR	SINE
MULS	D1, D7
ASL.L	#16-(SIN208_SCALE-2), D7
SWAP	D7
MOVE.W	D7, D1
MOVE.W	CANNA(A1), D7
JSR	COSINE
MULS	D6, D7
ASL.L	#16-(SIN208_SCALE-2), D7
SWAP	D7
MOVE.W	D7, D4
MOVE.W	CANNA(A1), D7
JSR	SINE
MULS	D6, D7
ASL.L	#16-(SIN208_SCALE-2), D7
SWAP	D7
MOVE.W	D7, D0
MOVE.L	#T_LENGTH, D7
DIVU	D2, D7
ADDQ.W	#SHORT-1, D7
ANDI.W	#-SHORT, D7
BNE.S	#4
MOVEQ.L	#SHORT, D7
MOVE.W	D7, T_STEP(A6)
CLA.W	D5

N	A	K	E	N	O	D	E	B	MCND_19	ADD.W	T_STEP(A6),D8
579	9	0	0	0	4	F	0	DA6FFFFA	MCND_20	CHP1.W	ST_LENGTH,D8
580	9	0	0	0	4	F	4	0C450000	MCND_99	MCND_99	
581	9	0	0	0	4	F	8	6C04F66A	MOVE.W	ST_LENGTH-SHORT,D7	
582	9	0	0	0	4	F	C	3E3067FE	SUB.W	D8,D7	
583	9	0	0	0	4	F	0	9E45	LEA.L	TIME_2_TBL,A0	
584	9	0	0	0	4	F	2	41F000000000	MOVE.W	0(A0,D7.W),D7	
585	9	0	0	0	4	F	8	3E307000	MULS	D0,D7	
586	9	0	0	0	4	F	C	CFC0	MOVE.W	0(A0,D8.W),D6	
587	9	0	0	0	4	F	E	3C305000	MULS	D1,D6	
588	9	0	0	0	5	0	2	CDC1	SUB.L	D7,D6	
589	9	0	0	0	5	0	4	9CH7	LEA.L	TIME_1_TBL,A0	
590	9	0	0	0	5	0	6	41F000000000	MOVE.W	Y2_Y1(A6),D7	
591	9	0	0	0	5	0	C	3F2FFFFC	MULS	0(A0,D8.W),D7	
592	9	0	0	0	5	1	0	CFD05000	ADD.L	D7,D6	
593	9	0	0	0	5	1	0	DC97	ASL.L	16-TIME08_SCALE,D6	
594	9	0	0	0	5	1	6	E306	SWAP	D6	
595	9	0	0	0	5	1	8	4046	ADD.W	(A1),D6	
596	9	0	0	0	5	1	A	DC51	MOVE.W	D6,D2	
601	9	0	0	0	5	1	A	3406	SUB.W	PLOT_Y(A5),D6	
602	9	0	0	0	5	1	E	9C6D0020	BNE.S	MCND_21	
603	9	0	0	0	5	2	2	6600	ADD.W	ST_INCREMENT,T_STEP(A6)	
604	9	0	0	0	5	2	4	066F000AFFFA	BRA	MCND_19	
605	9	0	0	0	5	2	A	60B4	EQU	*	
606	9	0	0	0	5	2	C	6A02	BPL.S	*+4	
607	9	0	0	0	5	2	C	4446	NEG.W	D6	
607	9	0	0	0	5	2	E	0C460003	ST_TOO_FAR,D6		
608	9	0	0	0	5	3	0	6C460003	MCND_22		
609	9	0	0	0	5	3	4	6300	BL8.S	ST_DECREMENT,T_STEP(A6)	
610	9	0	0	0	5	3	6	5DC0FFFFA	SUB.W	MCND_22	
611	9	0	0	0	5	3	A	6E06	BC7.S	SHORT,T_STEP(A6)	
612	9	0	0	0	5	3	C	3D7C0002FFFFA	MOVE.W		

Copyright (c) 1983 Allied Corporation All Rights Reserved
Module -III-

```

H A K E   M O D E S
614 9:000542 3F3C07FE
615 9:000546 9F43
616 9:000548 41F90000000000
617 9:00054E 3F347000
618 9:000552 CFC4
619 9:000554 3C305000
620 9:000558 CMC3
621 9:00055A 9C07
622 9:00055C 41F90000000000
623 9:000562 3F24FFFE
624 9:000566 CFF05000
625 9:00056A DF36
626 9:00056C F307
627 9:00056E 4047
631 9:000570 DF690002
633 9:000574 3C02
634 9:000576 4FB300000000
635 9:00057C 6004FF62
636
637

PCRD_23 MOVE.W D6,D7
SUB.W D6,D7
LEA.L TIME_2_TBL,A0
MOVE.W @ (A0,D7.W),D7
MULS D4,D7
MOVE.W @ (A0,D6.W),D6
MULS D3,D6
SUB.L D7,D6
LEA.L TIME_1_TBL,A0
MOVE.W @ (A0,D6.W),D7
MULS D6,D7
ADD.L D6,D7
ASL.L #16-TIME00_SCALE,D7
SWAP D7
MULX(A1),D7
ADD.W D2,D6
MOVE.W D6,D6
JSR RTNPLS
BRA PCRD_19
END
    
```

pyright (c) 1983 Allied Corporation All Rights Reserved
data -11-

```

* d e s c r i p t i o n      T i m e      T a b l e
* F u n c t i o n s
MFTABL 137      T i m e      F u n c t i o n s
6
7 C:000000      x d e f      t i m e _ 1 _ t b l , t i m e _ 2 _ t b l
8 C:000000      s e c t i o n 0 C      * C o d e S e c t i o n
9 C:000000      t i m e _ 1 _ t b l : e q u . x      * S t a t i s t i c
10 C:000010      d o . w      0 . 0 . 0 . 1 . 1 . 2 . 3 . 5
11 C:000020      d o . w      6 . 8 . 5 . 1 . 1 . 3 . 1 6 . 1 8 . 2 1
12 C:000030      d o . w      2 4 . 2 7 . 3 6 . 3 3 . 3 7 . 4 1 . 4 5 . 4 9
13 C:000040      d o . w      5 3 . 5 8 . 6 2 . 6 7 . 7 2 . 7 7 . 8 3 . 8 8
14 C:000050      d o . w      9 4 . 1 0 0 . 1 0 6 . 1 1 2 . 1 1 9 . 1 2 5 . 1 3 2 . 1 3 9
15 C:000060      d o . w      1 4 6 . 1 5 3 . 1 6 1 . 1 6 8 . 1 7 6 . 1 8 4 . 1 9 2 . 2 0 1
16 C:000070      d o . w      2 0 9 . 2 1 8 . 2 2 7 . 2 3 6 . 2 4 5 . 2 5 4 . 2 6 4 . 2 7 3
17 C:000080      d o . w      2 8 3 . 2 9 3 . 3 0 3 . 3 1 4 . 3 2 4 . 3 3 5 . 3 4 6 . 3 5 7
18 C:000090      d o . w      3 6 8 . 3 7 9 . 3 9 1 . 4 0 2 . 4 1 4 . 4 2 6 . 4 3 8 . 4 5 1
19 C:0000A0      d o . w      4 6 3 . 4 7 6 . 4 8 9 . 5 0 2 . 5 1 5 . 5 2 8 . 5 4 1 . 5 5 6
20 C:0000B0      d o . w      5 6 9 . 5 8 3 . 5 9 7 . 6 1 1 . 6 2 5 . 6 4 0 . 6 5 5 . 6 6 9
21 C:0000C0      d o . w      6 8 4 . 7 0 0 . 7 1 5 . 7 3 0 . 7 4 6 . 7 6 2 . 7 7 8 . 7 9 4
22 C:0000D0      d o . w      8 1 0 . 8 2 6 . 8 4 3 . 8 6 0 . 8 7 6 . 8 9 3 . 9 1 1 . 9 2 8
23 C:0000E0      d o . w      9 4 5 . 9 6 3 . 9 8 1 . 9 9 9 . 1 0 1 7 . 1 0 3 5 . 1 0 5 3 . 1 0 7 2
24 C:0000F0      d o . w      1 0 9 0 . 1 1 0 9 . 1 1 2 8 . 1 1 4 7 . 1 1 6 6 . 1 1 8 6 . 1 2 0 5 . 1 2 2 5
25 C:000100      d o . w      1 2 4 5 . 1 2 6 4 . 1 2 8 5 . 1 3 0 5 . 1 3 2 5 . 1 3 4 6 . 1 3 6 6 . 1 3 8 7
26 C:000110      d o . w      1 4 0 8 . 1 4 2 9 . 1 4 5 0 . 1 4 7 2 . 1 4 9 3 . 1 5 1 3 . 1 5 3 7 . 1 5 5 8
27 C:000120      d o . w      1 5 8 0 . 1 6 0 3 . 1 6 2 5 . 1 6 4 7 . 1 6 7 0 . 1 6 9 3 . 1 7 1 6 . 1 7 3 9
28 C:000130      d o . w      1 7 6 2 . 1 7 8 5 . 1 8 0 8 . 1 8 3 2 . 1 8 5 6 . 1 8 7 9 . 1 9 0 3 . 1 9 2 7
29 C:000140      d o . w      1 9 5 2 . 1 9 7 6 . 2 0 0 0 . 2 0 2 5 . 2 0 5 0 . 2 0 7 5 . 2 1 0 0 . 2 1 2 5
30 C:000150      d o . w      2 1 5 0 . 2 1 7 5 . 2 2 0 1 . 2 2 2 7 . 2 2 5 2 . 2 2 7 8 . 2 3 0 4 . 2 3 3 0
31 C:000160      d o . w      2 3 5 7 . 2 3 8 3 . 2 4 1 0 . 2 4 3 6 . 2 4 6 3 . 2 4 9 0 . 2 5 1 7 . 2 5 4 4
32 C:000170      d o . w      2 5 7 1 . 2 5 9 9 . 2 6 2 6 . 2 6 5 4 . 2 6 8 2 . 2 7 0 9 . 2 7 3 7 . 2 7 6 6
33 C:000180      d o . w      2 7 9 4 . 2 8 2 2 . 2 8 5 1 . 2 8 7 9 . 2 9 0 8 . 2 9 3 7 . 2 9 6 6 . 2 9 9 5
34 C:000190      d o . w      3 0 2 4 . 3 0 5 3 . 3 0 8 3 . 3 1 1 2 . 3 1 4 2 . 3 1 7 2 . 3 2 0 2 . 3 2 3 2
35 C:0001A0      d o . w      3 2 6 2 . 3 2 9 2 . 3 3 2 2 . 3 3 5 3 . 3 3 8 3 . 3 4 1 4 . 3 4 4 5 . 3 4 7 6
36 C:0001B0      d o . w      3 5 0 7 . 3 5 3 8 . 3 5 6 9 . 3 6 0 0 . 3 6 3 2 . 3 6 6 4 . 3 6 9 5 . 3 7 2 7
37 C:0001C0      d o . w      3 7 5 9 . 3 7 9 1 . 3 8 2 3 . 3 8 5 5 . 3 8 8 8 . 3 9 2 0 . 3 9 5 3 . 3 9 8 5
38 C:0001D0      d o . w      4 0 1 8 . 4 0 5 1 . 4 0 8 4 . 4 1 1 7 . 4 1 5 0 . 4 1 8 3 . 4 2 1 7 . 4 2 5 0
39 C:0001E0      d o . w      4 2 8 4 . 4 3 1 8 . 4 3 5 1 . 4 3 8 5 . 4 4 1 9 . 4 4 5 3 . 4 4 8 8 . 4 5 2 2
40 C:0001F0      d o . w      4 5 5 6 . 4 5 9 1 . 4 6 2 5 . 4 6 6 0 . 4 6 9 5 . 4 7 3 0 . 4 7 6 5 . 4 8 0 0
41 C:000200      d o . w      4 8 3 5 . 4 8 7 0 . 4 9 0 5 . 4 9 4 1 . 4 9 7 7 . 5 0 1 2 . 5 0 4 8 . 5 0 8 4
42 C:000210      d o . w      5 1 2 0 . 5 1 5 6 . 5 1 9 2 . 5 2 2 8 . 5 2 6 5 . 5 3 0 1 . 5 3 3 8 . 5 3 7 4
43 C:000220      d o . w      5 4 1 1 . 5 4 4 8 . 5 4 8 5 . 5 5 2 2 . 5 5 5 9 . 5 5 9 6 . 5 6 3 3 . 5 6 7 0
44 C:000230      d o . w      5 7 0 8 . 5 7 4 5 . 5 7 8 3 . 5 8 2 1 . 5 8 5 8 . 5 8 9 6 . 5 9 3 4 . 5 9 7 2
45 C:000240      d o . w      6 0 1 0 . 6 0 4 8 . 6 0 8 7 . 6 1 2 5 . 6 1 6 3 . 6 2 0 2 . 6 2 4 1 . 6 2 7 9
46 C:000250      d o . w      6 3 1 8 . 6 3 5 7 . 6 3 9 6 . 6 4 3 5 . 6 4 7 4 . 6 5 1 3 . 6 5 5 2 . 6 5 9 2
47 C:000260      d o . w      6 6 3 1 . 6 6 7 1 . 6 7 1 0 . 6 7 5 0 . 6 7 9 0 . 6 8 2 9 . 6 8 6 9 . 6 9 0 9
48 C:000270      d o . w      6 9 4 9 . 6 9 8 9 . 7 0 3 0 . 7 0 7 0 . 7 1 1 0 . 7 1 5 1 . 7 1 9 1 . 7 2 3 2
49 C:000280      d o . w      7 2 7 2 . 7 3 1 3 . 7 3 5 4 . 7 3 9 5 . 7 4 3 6 . 7 4 7 7 . 7 5 1 8 . 7 5 5 9
50 C:000290      d o . w      7 6 0 0 . 7 6 4 1 . 7 6 8 3 . 7 7 2 4 . 7 7 6 6 . 7 8 0 7 . 7 8 4 9 . 7 8 9 0
51 C:0002A0      d o . w      7 9 3 2 . 7 9 7 4 . 8 0 1 6 . 8 0 5 8 . 8 1 0 0 . 8 1 4 2 . 8 1 8 4 . 8 2 2 6
52 C:0002B0      d o . w      8 2 6 9 . 8 3 1 1 . 8 3 5 4 . 8 3 9 6 . 8 4 3 9 . 8 4 8 1 . 8 5 2 4 . 8 5 6 7
53 C:0002C0      d o . w      8 6 0 9 . 8 6 5 2 . 8 6 9 5 . 8 7 3 8 . 8 7 8 1 . 8 8 2 4 . 8 8 6 7 . 8 9 1 1
54 C:0002D0      d o . w      8 9 5 4 . 8 9 9 7 . 9 0 4 1 . 9 0 8 4 . 9 1 2 8 . 9 1 7 1 . 9 2 1 5 . 9 2 5 9
55 C:0002E0      d o . w      9 3 0 2 . 9 3 4 6 . 9 3 9 0 . 9 4 3 4 . 9 4 7 8 . 9 5 2 2 . 9 5 6 6 . 9 6 1 0
      d o . w      9 6 8 4 . 9 6 9 8 . 9 7 4 3 . 9 7 8 7 . 9 8 3 1 . 9 8 7 6 . 9 9 2 0 . 9 9 6 5

```

Copyright (c) 1983 Allied Corporation All Rights Reserved

domily o Time Table

56	C:0002F0	27A12746277327A0	do.w	10010,10054,10099,10144,10189,10233,10278,10323
57	C:000300	28B02AD28DA2907	do.w	10368,10413,10458,10503,10548,10594,10639,10684
58	C:000310	29F92A172A42A72	do.w	10729,10775,10820,10866,10911,10957,10992,11048
59	C:000320	20562DR320H1204F	do.w	11094,11139,11185,11231,11277,11323,11369,11415
60	C:000330	20C52CF320J1204F	do.w	11461,11507,11553,11599,11645,11691,11737,11784
61	C:000340	2F3A2EF642E932EC1	do.w	11830,11876,11923,11969,12016,12062,12108,12155
62	C:000350	2FA22FDR30673035	do.w	12202,12248,12295,12341,12388,12435,12482,12528
63	C:000360	311F314E317D31AC	do.w	12675,12722,12769,12816,12863,12910,12957,13004
64	C:000370	329732C632F53324	do.w	12951,12998,13045,13092,13139,13186,13234,13281
65	C:000380	341043F346F349F	do.w	13328,13375,13423,13470,13517,13565,13612,13659
66	C:000390	350B35BA35EA3619	do.w	13707,13754,13802,13849,13897,13944,13992,14039
67	C:000400	3707373637663796	do.w	14087,14134,14182,14230,14277,14325,14373,14420
68	C:000410	380430B430E33913	do.w	14468,14516,14563,14611,14659,14707,14754,14802
69	C:000420	3A923A323A623A91	do.w	14809,14856,14904,14951,14999,15046,15093,15140
70	C:000430	3001300130E13011	do.w	15233,15281,15329,15377,15424,15472,15520,15568
71	C:000440	3004303030E030F0	do.w	15616,15664,15712,15760,15808,15856,15904,15952
72	C:000450	3F04303030E030F0	do.w	16000,16048,16096,16144,16192,16240,16288,16336
73	C:000460	4000403040E04090	do.w	16384,16432,16480,16528,16576,16624,16672,16720
74	C:000470	4100410041F04210	do.w	16768,16816,16864,16912,16960,17008,17056,17104
75	C:000480	4300430043E04390	do.w	17152,17200,17248,17296,17344,17391,17439,17487
76	C:000490	447F44AF44DF450F	do.w	17536,17583,17631,17679,17727,17775,17823,17870
77	C:000500	45FE45E455E460D	do.w	17918,17966,18014,18061,18109,18157,18205,18252
78	C:000510	477C47AC47DD480B	do.w	18300,18348,18395,18443,18491,18538,18586,18634
79	C:000520	48F947294958490B	do.w	18681,18729,18776,18824,18871,18919,18966,19014
80	C:000530	4A754AA54AD44B03	do.w	19061,19109,19156,19203,19251,19298,19345,19393
81	C:000540	4F004F004F004F00	do.w	19440,19487,19534,19582,19629,19676,19723,19770
82	C:000550	4D604D904DC74DF6	do.w	19817,19864,19911,19958,20005,20052,20099,20146
83	C:000560	4FE14F104F3E4F6D	do.w	20193,20240,20286,20333,20380,20427,20473,20520
84	C:000570	505650B550B450F2	do.w	20566,20613,20660,20708,20755,20802,20849,20896
85	C:000580	51CA51F852275258	do.w	20938,20984,21031,21077,21123,21169,21216,21261
86	C:000590	533B5369539753C8	do.w	21307,21353,21399,21445,21491,21537,21583,21629
87	C:000600	54A54A0854065433	do.w	21674,21720,21766,21811,21857,21902,21948,21993
88	C:000610	561756445671569E	do.w	22039,22084,22129,22174,22220,22265,22310,22355
89	C:000620	57B057AD57DA5807	do.w	22400,22446,22490,22535,22580,22624,22669,22714
90	C:000630	58E659135940596C	do.w	22758,22803,22848,22892,22937,22981,23025,23070
91	C:000640	5A45A765AA25ACE	do.w	23114,23158,23202,23246,23290,23334,23378,23422
92	C:000650	5B0A5B031B05D30B8	do.w	23466,23509,23553,23597,23640,23684,23727,23771
93	C:000660	5E5F5E595E485DF	do.w	23814,23857,23901,23944,23987,24030,24073,24116
94	C:000670	5F5F5F5F5F5F5F5F	do.w	24189,24231,24274,24317,24360,24403,24446,24489
95	C:000680	6104612F615761H1	do.w	24599,24642,24684,24726,24768,24810,24852,24894
96	C:000690	6250627962A262CB	do.w	24961,25003,25045,25087,25129,25171,25213,25255
97	C:000700	639B63C063E96411	do.w	25168,25209,25250,25291,25332,25373,25414,25456
98	C:000710	64DL650365206533	do.w	25496,25536,25577,25617,25658,25698,25738,25779
99	C:000720	67526779679F67C6	do.w	25819,25859,25899,25939,25978,26018,26058,26097
100	C:000730	67526779679F67C6	do.w	26137,26176,26216,26255,26294,26333,26372,26411
101	C:000740	68B668AC68D268F8	do.w	26459,26499,26538,26577,26616,26655,26694,26733
102	C:000750	69BA69DA69FF6A24	do.w	26758,26796,26834,26872,26910,26948,26986,27024
103	C:000760	6AD060266026604B	do.w	27069,27108,27146,27184,27222,27260,27298,27336
104	C:000770	6AD060266026604B	do.w	27367,27406,27444,27482,27520,27558,27596,27634
105	C:000780	6C0060266026604B	do.w	27648,27686,27724,27762,27800,27838,27876,27914
106	C:000790	6D10604060360B86	do.w	27933,27971,28009,28047,28085,28123,28161,28199
107	C:000800	6F316F306F78F9F8	do.w	28212,28250,28288,28326,28364,28402,28440,28478
108	C:000810	6F446F666F878FA9	do.w	28484,28521,28559,28597,28635,28673,28711,28749

c) 1983 Allied Corporation All Rights Reserved

u n
 o t i c e
 T a b l e

0640	704F706F700F70B0	do.w	28750, 28783, 28810, 28848, 28880, 28913, 28945, 28977
0650	7151717171917110	do.w	29009, 29041, 29073, 29104, 29136, 29168, 29199, 29230
0660	7240726C720B72AA	do.w	29261, 29292, 29323, 29354, 29385, 29416, 29446, 29476
0670	73427360737E739C	do.w	29506, 29536, 29566, 29596, 29626, 29656, 29685, 29715
0680	7430744D746A74B7	do.w	29744, 29773, 29802, 29831, 29860, 29889, 29917, 29946
0690	75167532754F756B	do.w	29974, 30002, 30031, 30059, 30086, 30114, 30142, 30169
06A0	75F57610762B7646	do.w	30197, 30224, 30251, 30278, 30305, 30332, 30358, 30385
06B0	76C076E67700771A	do.w	30411, 30438, 30464, 30490, 30516, 30541, 30567, 30593
06C0	779A77B377CC77E5	do.w	30618, 30643, 30668, 30693, 30718, 30743, 30768, 30792
06D0	7860787979178A9	do.w	30816, 30841, 30865, 30889, 30912, 30936, 30960, 30983
06E0	791E7930794C7963	do.w	31006, 31029, 31052, 31075, 31098, 31121, 31143, 31165
06F0	79D479FA79FF7A15	do.w	31188, 31210, 31231, 31253, 31275, 31296, 31318, 31339
0700	7AB07A957AA7ABE	do.w	31360, 31381, 31402, 31422, 31443, 31463, 31483, 31504
0710	7B237B377B4B7B5E	do.w	31523, 31543, 31563, 31582, 31602, 31621, 31640, 31659
0720	7B077B007BE37BF5	do.w	31678, 31696, 31715, 31733, 31751, 31769, 31787, 31805
0730	7C4F7C607C717C83	do.w	31823, 31840, 31857, 31875, 31892, 31908, 31925, 31942
0740	7D677D637D717D06	do.w	31958, 31974, 31990, 32006, 32022, 32038, 32053, 32068
0750	7D547D637D717D00	do.w	32084, 32099, 32113, 32128, 32143, 32157, 32171, 32185
0760	7D077D057DF37DF0	do.w	32199, 32213, 32227, 32240, 32253, 32266, 32279, 32292
0770	7F317F3D7E4A7E56	do.w	32305, 32317, 32330, 32342, 32354, 32366, 32377, 32389
0780	7F907F9B7FA67EB1	do.w	32400, 32411, 32422, 32433, 32444, 32454, 32465, 32476
0790	7EE37EE77EF87F02	do.w	32483, 32495, 32504, 32514, 32523, 32532, 32541, 32550
07A0	7F207F377F407F48	do.w	32559, 32567, 32576, 32584, 32592, 32600, 32607, 32615
07B0	7F6E7F757F7C7F83	do.w	32622, 32629, 32636, 32643, 32649, 32656, 32662, 32668
07C0	7FA27FAB7FAD7FB3	do.w	32674, 32680, 32685, 32691, 32696, 32701, 32706, 32710
07D0	7FCB7FCF7FD37FD7	do.w	32715, 32719, 32723, 32727, 32731, 32735, 32738, 32741
07E0	7FE87FE87FEE7FE0	do.w	32744, 32747, 32750, 32752, 32755, 32757, 32759, 32760
07F0	7FFA7FFF7FFD7FFE	do.w	32762, 32763, 32765, 32766, 32767, 32767, 32767, 32767

62

1983 Allied Corporation All Rights Reserved

Y
T
I
M
E
T
A
B
L
E

70	0F7A0F860F920F9Z	do.w 3962,3974,3986,3998,4010,4021,4033,4045
80	0FD10FE30FF0FFA	do.w 4056,4067,4079,4090,4101,4112,4123,4133
90	1030103010451050	do.w 4144,4155,4165,4176,4186,4196,4207,4217
00	10831080D109610A0	do.w 4227,4237,4246,4256,4266,4276,4286,4294
10	10D10D10910F210FB	do.w 4304,4313,4322,4331,4340,4349,4358,4366
20	1117112011281130	do.w 4375,4384,4392,4400,4409,4417,4425,4433
30	1159116111691170	do.w 4441,4449,4457,4464,4472,4480,4487,4495
40	1196119D11A411AB	do.w 4502,4509,4516,4523,4530,4537,4544,4551
50	11CD11D411DB11E1	do.w 4557,4564,4571,4577,4583,4590,4596,4602
60	12001206120C1212	do.w 4608,4614,4620,4626,4631,4637,4643,4648
70	122F1233123B123D	do.w 4654,4659,4664,4669,4674,4679,4684,4689
80	1256125B12601264	do.w 4694,4699,4704,4708,4713,4717,4721,4726
90	127A127E12821286	do.w 4730,4734,4738,4742,4746,4750,4754,4757
00	1290129D12A012A3	do.w 4761,4765,4768,4771,4775,4778,4781,4784
10	12B112B612B912BC	do.w 4787,4790,4793,4796,4799,4802,4804,4807
20	12C112C612CE12D0	do.w 4809,4812,4814,4816,4819,4821,4823,4825
30	12D112D12D12D12D	do.w 4827,4829,4831,4832,4834,4836,4837,4839
40	12E112E912FB12FC	do.w 4840,4841,4843,4844,4845,4846,4847,4848
50	12F112F212F212F3	do.w 4849,4850,4850,4851,4852,4852,4853,4853
60	12F612F612F612F6	do.w 4854,4854,4854,4854,4854,4855,4855,4854
70	12F612F612F612F6	do.w 4854,4854,4854,4854,4854,4855,4855,4854
80	12F312F212F212F1	do.w 4854,4854,4854,4854,4854,4855,4855,4852
90	12F312F212F212F1	do.w 4854,4854,4854,4854,4854,4855,4855,4852
00	12F312F212F212F1	do.w 4854,4854,4854,4854,4854,4855,4855,4852
10	12F112DF12DF12DC	do.w 4854,4854,4854,4854,4854,4855,4855,4852
20	12D212D012CE12DC	do.w 4854,4854,4854,4854,4854,4855,4855,4852
30	12C112D112D112D1	do.w 4854,4854,4854,4854,4854,4855,4855,4852
40	12A112A712A412A1	do.w 4854,4854,4854,4854,4854,4855,4855,4852
50	1291120E120A1207	do.w 4854,4854,4854,4854,4854,4855,4855,4852
60	12751271126D1269	do.w 4854,4854,4854,4854,4854,4855,4855,4852
70	12551251124D1248	do.w 4854,4854,4854,4854,4854,4855,4855,4852
80	1235122E12291224	do.w 4854,4854,4854,4854,4854,4855,4855,4852
90	120C1207120211FD	do.w 4854,4854,4854,4854,4854,4855,4855,4852
00	11B111B211AD11A7	do.w 4854,4854,4854,4854,4854,4855,4855,4852
10	11A111A4117E117B	do.w 4854,4854,4854,4854,4854,4855,4855,4852
20	11591152114C1146	do.w 4854,4854,4854,4854,4854,4855,4855,4852
30	1125111E111B1111	do.w 4854,4854,4854,4854,4854,4855,4855,4852
40	10FF10FB10E110DA	do.w 4854,4854,4854,4854,4854,4855,4855,4852
50	10B710AF10A810A1	do.w 4854,4854,4854,4854,4854,4855,4855,4852
60	107C1074106D1065	do.w 4854,4854,4854,4854,4854,4855,4855,4852
70	103F1037102F102B	do.w 4854,4854,4854,4854,4854,4855,4855,4852
80	1000FF0F0F0F0F0F	do.w 4854,4854,4854,4854,4854,4855,4855,4852
90	0FDF0F0F0F0F0F0F	do.w 4854,4854,4854,4854,4854,4855,4855,4852
00	0F7C0F740F6B0F63	do.w 4854,4854,4854,4854,4854,4855,4855,4852
10	0F370F2F0F260F1D	do.w 4854,4854,4854,4854,4854,4855,4855,4852
20	0FF10FE00E0F0ED6	do.w 4854,4854,4854,4854,4854,4855,4855,4852
30	0FA90FA00E970E0E	do.w 4854,4854,4854,4854,4854,4855,4855,4852
40	0E5F0E560E400E43	do.w 4854,4854,4854,4854,4854,4855,4855,4852
50	0E110E000E0100FB	do.w 4854,4854,4854,4854,4854,4855,4855,4852
60	01C100D00E0D05DA0B	do.w 4854,4854,4854,4854,4854,4855,4855,4852
70	01710D710D670D5D	do.w 4854,4854,4854,4854,4854,4855,4855,4852
80	012C01220D180D0E	do.w 4854,4854,4854,4854,4854,4855,4855,4852
90	0C010C0D20C0C0C0E	do.w 4854,4854,4854,4854,4854,4855,4855,4852

Early Times 2

1

55

Copyright (c) 1983 Allied Corporation All Rights Reserved
Module - IV-

1	W E A N - I N T E R P O L A T I O N /	EXTENDED PRECISION	1.0	CURVE FONT LINEAR INTERPOLATOR	
2		IDNT			
3		NTRPLB			
4		XDEF		NTRPLB	
5		XREF		PLOT_INSERT	
6		XREF		VEKTOR, GRADE	
7		MACRO		INSERT DI	
8		IFNC		^1, ^.	
9		MOVE.W		^1, DI	
10		ENDC			
11		ADD.W		DI, DI	
12		TST.B		COLOR(A6)	
13		BEQ.B		#+4	
14		ADDQ.W		#1, DI	
15		JBR		PLOT_INSERT	
16		ENDM			
17					
18					

Copyright (c) 1983 Allied Corporation All Rights Reserved
Module -IV-

LINE	ADDRESS	INSTR	OPERANDS	COMMENT
20	91000000	4RA1C000		
21	91000004	3A07		
22	91000004	3A06		
23	91000006	4A3B7001		
24	91000008	673C		
25	9100000C	48E77FFE		
26	91000012	41F900000000		
27	91000018	3F210020		
28	9100001C	4447		
29	9100001E	DF6D0024		
30	91000022	30C7 11		
31	91000024	3F2D0022		
32	91000028	DE6D0026		
33	9100002C	30C7		
34	9100002E	3F04		
35	91000030	4447		
36	91000032	DE6D0024		
37	91000036	30C7		
38	91000038	3F03		
39	9100003A	DE6D0026		
40	9100003E	30B7		
41	91000040	4ED900000000		
42	91000044	4C0F7FFF		
43	91000046	302D0020		
44	9100004E	9849		
45	91000050	6646		
46	91000052	BA6D0023		
47	91000056	673A		
48	91000058	6EC7		
49	9100005A	BE2D002F		
50	9100005E	66C7		
51	91000060	BE2D002E		
52	91000064	672B		
53	91000066	1007		
54	91000068	322D0023		
55	9100006C	D241		
56	9100006E	4A2D002E		
57	91000072	6702		
58	91000074	D241		
59	91000076	4ED900000000		
60	9100007A	660C		
61	91000080	4C0F003C		
62	91000084	1047		
63	91000088	3F1F		
64	91000092	4C0F0033		
65	91000096	4E75		

Copyright (c) 1983 Allied Corporation All Rights Reserved
 today -IV-

LINE	DATA	EXTENDED	PRECISION
57	91000000 0F17	RTNP_4	SGT D7
58	91000000 0F17		CHP.B COLOR(AS),D7
59	91000000 0F17		RTNP_5
60	91000000 0F17		MOVE.W OLD_INTERCEPT(AS),D1
61	91000000 0F17		ADD.W D1,D1
62	91000000 0F17		TST.B COLOR(AS)
63	91000000 0F17		BEQ.B #4
64	91000000 0F17		ADDQ.W #1,D1
65	91000000 0F17		JHR PLDT_INSERT
66	91000000 0F17	RTNP_6	MOVEQ.L #1,D7
67	91000000 0F17		MOVE.W D4,D6
68	91000000 0F17		SLT COLOR(AS)
69	91000000 0F17		BCE.B RTNP_6
70	91000000 0F17		NEG.W D6
71	91000000 0F17		MOVEQ.L #1,D7
72	91000000 0F17	RTNP_6	CHP.W #1,D6
73	91000000 0F17		RTNP_7
74	91000000 0F17		MOVEQ.L D2/D3, -(SP)
75	91000000 0F17		MOVE.W D7, -(SP)
76	91000000 0F17		MOVE.W D0,D3
77	91000000 0F17		SUB.W OLD_INTERCEPT(AS), D3
78	91000000 0F17		EXT.L D2
79	91000000 0F17		MOVE.W D2,D1
80	91000000 0F17		RPL.B #4
81	91000000 0F17		NEG.V D1
82	91000000 0F17		MOVEQ.L #4,D7
83	91000000 0F17	RTNP_7	MOVEQ.L #13+1, D3
84	91000000 0F17		SUBQ.W #1,D3
85	91000000 0F17		BCE.B RTNP_8
86	91000000 0F17		ADD.W D7,D7
87	91000000 0F17		CHP.W D1,D7
88	91000000 0F17		BLB RTNP_7
89	91000000 0F17	RTNP_8	CHP.V #2,D6
90	91000000 0F17		BCE.B RTNP_11
91	91000000 0F17		CHP.W #4,D6
92	91000000 0F17		RTNP_11
93	91000000 0F17		MOVEQ.L #2,D7
94	91000000 0F17	RTNP_9	ADDQ.W #1,D3
95	91000000 0F17		CHP.V #10,D3
96	91000000 0F17		BCE.B RTNP_10
97	91000000 0F17		ADD.W D7,D7
98	91000000 0F17		CHP.W D6,D7
99	91000000 0F17		BLB RTNP_9
100	91000000 0F17	RTNP_10	ASL.L D3,D3
101	91000000 0F17		DIVB D6,D3
102	91000000 0F17		EXT.L D2
103	91000000 0F17		BRA.B RTNP_12

Copyright (c) 1983 Allied Corporation All Rights Reserved
Module -IV-

```

L I N E A R - I N T E R P O L A T O R / EXTENDED PRECISION
109 9:00010E E7A2 NTRP_11 ASL.L D3,D2
110 9:000110 E246 ASR.W #1,D6
111 9:000112 D646 ADD.W D6,D3
112 9:000114 3C2D0023 NTRP_12 MOVE.W OLD_INTERCEPT(AB),D6
113 9:000118 4BC6 EXT.L D6
114 9:00011A E7A6 ASL.L D3,D6
115 9:00011C 9B07 NTRP_13 SUB.W (SP),D4
116 9:00011E 6700FF62 BEQ NTRP_11
117 9:000122 D057 ADD.W (SP),D0
118 9:000124 DC82 ADD.L D2,D6
119 9:000126 2E06 MOVE.L D6,D7
120 9:000128 E6A7 ASR.L D3,D7
121 9:00012A 3207 MOVE.W D7,D1
121 9:00012C D241 ADD.W D1,D1
121 9:00012E 4A2D002E TST.B COLOR(AB)
121 9:000132 6702 BEQ.B #+4
121 9:000134 5241 ADDQ.W #1,D1
121 9:000136 4EB900000000 JBR PLOT_INTERPT
122 9:00013C 60DE BRA NTRP_13
123
124 END

```

Copyright (c) 1983 Allied Corporation All Rights Reserved
Module -V-

PLAQUEUE INSERT, OUTPUT, CLEAR, & CHECK

2	PLAQUE	IDMT	1.0 CURVE FONT MASTER SORTING SYSTEM
6	1	INCLUDE RCHPAR	
1	2	STRUKTUR DES PASCAL-PARAMETERBLOCKS	
2	3		
4	4	OFFSET	
5	5	RCHPAR EQU	
6	6	CHARJOB DS.W	
7	7		
8	8	JOBPARCEL EQU	
9	9	AUFLEX DS.W	
10	10	AUFLEX DS.W	
11	11	ZOOMP DS.W	
12	12		
13	13	ZEITAB EQU	
14	14	C2 DS.W	
15	15	FONADR DS.L	
16	16	MCNADR DS.L	
17	17	FRORNR DS.W	
18	18		
19	19		
20	20		
21	21		
22	22	DATENTYPEN FUER MEMO-CONTROL-BLOCK	
23	23	OFFSET	
24	24		
25	25	MCNLEN EQU	
26	26	V90C DS.B	
27	27	EQUCHK DS.B	
28	28	CFKLEN DS.W	
29	29	FONLEN DS.W	
30	30	SC DS.V	
31	31	CE DS.V	
32	32		
33	33	HIER MUESEN DANN SLANT_WIN, ROT_WIN REIN	
34	34	OPCL DS.W	
35	35	CHAR_OFF DS.W	
36	36	RASE_OFF DS.W	
37	37	SCANCT DS.L	
38	38	IPADR DS.L	
39	39	SEC_LACH DS.W	
40	40	HER_LACH DS.L	
41	41		
42	42		
43	43	ERROR NUMBERS	
44	44	BUTOVER SET	30
45	45	FONLEN SET	17
46	46	PARLEN SET	76

AUFGABEN: 0-SYSTEM-INIT, 1-JOB-INIT
 2-ZEICHEN-INIT
 JOB-PARAMETER
 ZEICHEN-PARAMETER
 POINTER AUF MCS
 SYSTEM-ERNUMMER
 FUER SCAN
 FONTLOC/FONN-KENNUNG
 FONTNR/LAUFEDE FONN-NR
 SCHRIFTCRAD
 CONDENSED-EXPAND
 ANSTAND IN MI'S REL-YRULL-ZEICHENKARTE
 ZEICHENKARTE->SCHRIFTLINIE
 -> -UNTERKANTE
 POINTER AUF MEMO-SECHENT
 MEMOSECHENT-LAENGE IN VORTEN

PLAQUEQUEL INSERT, OUTPUT, CLEAN, ● CHECK

70

Copyright (c) 1983 Allied Corporation All Rights Reserved
Module -V-

DEFINITIONS

```

42 *
43 *
44 *
45 *
46 *
47 *
48 *
49 *
50 *
51 *
52 *
53 *
54 *
55 *
56 *
57 *
58 *
59 *
60 *
61 *
62 *
63 *
64 *
65 *
66 *

```

MACROS TO HANDLE SECRET TABLES

```

THIS_SECRET MACRO
  MOVE.W D0,D7
  LSR.W #POSV_SECRET,D7
  INDEX.W LONG,D7
  LEA.L POSL_SECRET,A0
  LEA.L @ (A0,D7.W),A2
  MOVE.L (A2),D7
  ENDM

```

THIS_LINE

```

THIS_LINE MACRO
  MOVE.W D0,D7
  ANDI.W #POSL_LINE,D7
  INDEX.W LONG,D7
  IFX POSL_LINE#FFFFFFF00
  LEA.L POSL_LINE(A2,D7.W),A0
  MOVE.L (A0),%I
  NEXT
  ENDC
  LEA.L POSL_LINE(A2),A0
  LEA.L @ (A0,D7.W),A0
  MOVE.L (A0),%I
  ENDM

```

THIS_LINE_NEED_STORE

Copyright (c) 1983 Allied Corporation All Rights Reserved
 file -v-

A C R O B

	THIS_COUNT	MACRO	THIS_COUNT REGISTER(OPTIONAL)
68	MOVE.W	D0,D7	
69	ANDI.W	#F08H,L1RE,D7	
70	INDEX.W	BYTE,D7	
71	IFDA	POSTER_COUNT(0FFFFFFF)	
72	LEA.L	POSTER_COUNT(A2,D7.W),A0	
73	ENDC		
74	IFNE	POSTER_COUNT(0FFFFFFF)	
75	LEA.L	POSTER_COUNT(A2),A0	
76	LEA.L	0(A0,D7.W),A0	
77	ENDC		
78	IFNC	'\1','\1	
79	MOVE.B	(A0),\1	
80	ENDC		
81	ENDM		
82			
83			
84	INTO_MACRO	MACRO	INTO_MACRO VALUE
85	ADDQ.L	#1,D5	
86	SUBQ.L	#SHORT,D6	
87	IFC	'\0','\0'	
88	BCE.B	#+4	
89	ENDC		
90	IFNC	'\0','\0'	
91	BCE.B	#+6	
92	ENDC		
93	BSR.L	MORE_MACRO	
94	MOVE.W	\1,(A4)+	
95	ENDM		

Copyright (c) 1983 Allied Corporation All Rights Reserved
 file -v.

```

C R O B
97
1
2
3 *****
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50

INCLUDE STORMAC
* MACRO FOR MEMORY ALLOCATION
*
ONLYAREA: SET TRUE * IF ONLY 1 AREA

XREF AALOC
XREF AFREE

GETMAIN: MACRO
  MAINSIZE,% \1,<%% \3>
  JSR AALOC
  ENDM

  * SET A BLOCK FROM THE POOL
  FREEMAIN: MACRO
    IFC '% \2',%A0
    ENDC
    * ADDRESS TO FREE IN LONGWORD A0
    IFNG '% \2',%A0
    IFNG '% \3',%D7
    MOVE.L D7,A0
    ENDC
    IFNG '% \3',%D7
    LEA.L (% \3),A0
    ENDC
    ENDC
    MAINSIZE,% \1,<%% \3>
    JSR AFREE
    ENDM
    * RETURN BLOCK TO POOL

* .L= LONG REFERENCE
* .S= SHORT REFERENCE
* .N= SHORT REFERENCE
* .B= BY VALUE
MAINSIZE:
  MACRO '% \0',%B
  IFC '% \0',%D7
  EXT.L D7
  NEXT
  ENDC
  IFC '% \1',%
  NEXT
  ENDC
  * MEMORY REQUEST LENGTH GIVEN IN LONGWORD D7
  IFC '% \0',%
  MOVE.L % \1,D7
  NEXT
  ENDC
  IFC '% \0',%L
  MOVE.L % \1,D7
  NEXT
  ENDC
  MOVEQ.L % \0,D7
  MOVE.W % \1,D7
  ENDM
  
```

Copyright (c) 1983 Allied Corporation

PL	OT	CHECK	+	DEF	SECTION	PL	OT	CHECK
100	100	91000000		FEA	(A2)	100	100	91000000
101	101	4052		MOVE.W	D0,D7	101	101	91000000
102	102	3F00		LSR.W	POST_SCHT.D7	102	102	91000000
103	103	E04F		LSL.W	#2,D7	103	103	91000000
104	104	E04F		LEA.L	POST_SCHT.A0	104	104	91000000
105	105	E04F		LEA.L	(A0,D7.W),A2	105	105	91000000
106	106	45F07000		MOVE.L	(A2),D7	106	106	91000000
107	107	2E12		BDQ.S	PCBK_1	107	107	91000000
108	108	2447		MOVE.L	D7,A2	108	108	91000000
109	109	3F00		MOVE.W	D0,D7	109	109	91000000
110	110	024700FF		ANDI.W	POST_LINE.D7	110	110	91000000
111	111	41F27000		LEA.L	POST_COUNT(A2,D7.W),A0	111	111	91000000
112	112	00100000		BTST	#0,(A0)	112	112	91000000
113	113	6702		BDQ.S	PCBK_1	113	113	91000000
114	114	6104		BSR.S	PLT_INCREMENT	114	114	91000000
115	115	245F		MOVE.L	(BP)+,A2	115	115	91000000
116	116	4E75		PCBK_1	RTS	116	116	91000000

© 1983 Allied Corporation All Rights Reserved

```

M N E R T
0006 2447
0008 3E00
000A 024700FF
000E 41F27000
0002 1410
0004 4050
0006 3E00
0008 024700FF
000C E54F
000E 41EA0100
00A2 41F07000
00A6 2650
00AB 3602
00AA 02430007
00AE 670F
00B0 D643
00B2 41F33004
00B6 00000001
00BA 671A
00BC 6020
00DE 244B
00C0 7E10
00C2 4ED000000000
00C8 CF00
00CA 2607
00CC 240B
00CE 0430000E
00D2 41F00010
00D6 3F01
00DB 5247
00DA 3147FFFC
00DE 3101
00E0 205F
00E2 5302
00E4 1002
00E6 4C91000C
00EA 4C0F0C00
00EE 4E70

PINS_6 MOVE.L D7,A2
MOVE.W D0,D7
ANDI.W #PUSHLINE,B7
* D7 ALREADY INDEX TO CLEAR
LEA.L POSBL_COUNT(A3,B7.W),A0
MOVE.B (A0),D2
PEA (A0)
MOVE.W D0,D7
ANDI.W #PUSHLINE,B7
LEA.L #2,D7
LEA.L POSBL_COUNT(A2),A0
LEA.L 0(A0,D7.W),A0
MOVE.L (A0),A0
MOVE.W D2,D3
ANDI.W #POLSE_ENTRIES,D0
DEQ.B PINS_7
ADD.W D3,D3
LEA.L POLSE_VALUES(A3,D0.W),A0
BTST #1,D3
BFX.B PINS_8
BRA.B PINS_9

PINS_7 MOVE.L A0,A2
MOVE.L #POLSE_LENGTH,D7
JSR AALOC
EXG D7,A3
MOVE.L D7,(A3)
MOVE.L A3,(A2)
MOVE.W #1<<POLSE_ENTRIES+POLSE_ENTRIES,D0
LEA.L POLSE_ENTRIES+SHORT+POLSE_VALUES(A3),A0
MOVE.W D1,D7
ADDQ.W #1,D7
MOVE.W D7,-2+SHORT(A0)
MOVE.W D1,-(A0)
MOVE.L (BP)+,A0
SUBQ.B #1,D2
MOVE.B D2,(A0)
MOVE.W (BP)+,D2/D0
MOVE.L (BP)+,A3/A2
RTS

** ALLOCATE NEW LINE
* SET A BLOCK FROM THE POOL

```

LOC	CODE	ADDRESS	OPERATION	COMMENT
100	9:0000F0	40E7067A	MOVEN.L	D5/D6/A6/A2/A3/A4/A1, -(SP)
101	9:0000F0	40E7067A	MOVEN.W	D2/D3/D1/D0/D4, -(SP)
102	9:0000F4	23CF00000400	TST.B	LOOP_NOT_CLOSED(A5)
103	9:0000F0	4A2D002D	BNE	PDON_30
104	9:0000F0	660001FA	CLR.L	D5
105	9:0000F0	42B5	SUB.L	A6,A6
106	9:0000F0	9DCE	MOVE.L	PCRPAN,A4
107	9:0000F0	207700000000	MOVE.L	MCBADR(A4),A4
108	9:0000F0	2B6C000E	CLR.L	D6
109	9:0000F0	4206	MOVE.W	SEGLAEN(A4),D6
110	9:0000F0	302C0018	SUBQ.L	#1,D6
111	9:0000F0	53B6	ADD.L	D6,D6
112	9:0000F0	DCB6	MOVE.L	D6,MPROL_LENGTH
113	9:0000F0	23C000000404	MOVE.L	IPADR(A4),A4
114	9:0000F0	2B6C0014	MOVE.W	MAX_RASTER(A5),D0
115	9:0000F0	302D001C	MOVE.W	D0,D4
116	9:0000F0	3000	ADDQ.W	#1,D0
117	9:0000F0	5249	SUBQ.W	#1,D0
118	9:0000F0	5349	CMPL.W	MIN_RASTER(A5),D0
119	9:0000F0	B06D001B	BLT.B	PDON_2
120	9:0000F0	6D1A	MOVE.W	D0,D7
121	9:0000F0	3E04	LSR.W	#POSV_SHIFT,D7
122	9:0000F0	F04F	LSL.W	#2,D7
123	9:0000F0	F34F	LEA.L	POSSL_INCREMENT,A0
124	9:0000F0	41F000000000	LEA.L	#(A0,D7.W),A2
125	9:0000F0	45F07000	MOVE.L	(A2),D7
126	9:0000F0	2E12	BNE.B	PDON_5
127	9:0000F0	666C	ANDI.W	#POSSL_SHIFT,D0
128	9:0000F0	0249FF00	BRA	PDON_1
129	9:0000F0	60DE		

220	9:000152	5486	ADDQ.L	SHORT,D6	D4,MIN_RASTER(A5)
221	222	223	224	225	226
227	9:000152	5283	ADDQ.L	1,D6	
228	9:000154	5386	SUBQ.L	SHORT,D6	
229	9:000156	6004	BCE.B	26	
230	9:000158		BSR.L	MORE_MEMO	
231	9:00015A	61000168	MOVE.W	END_MEMO,(A4)+	
232	9:00015E	38FCDFE	BSR	RELEASE_QUEUE	
233	9:000162	610001E4			
234	9:000166	3D440018	MOVE.W	D4,MIN_RASTER(A5)	
235	9:00016A	E0ED001A	ASR.W	MIN_INTERCEPT(A5)	
236	9:00016E	6404	BCC.B	PDON_3	
237	9:000170	5361001E	SUBQ.W	1,MAX_INTERCEPT(A5)	
238	9:000174	E0E1001E	ASR.W	MAX_INTERCEPT(A5)	
239	9:000178	207900000000	MOVE.L	PCFAR,A0	
240	9:00017E	2143001A	MOVE.L	PCBADR(A0),A0	
241	9:000182	3F210024	MOVE.L	D5,PCFAR(A0)	
242	9:000186	9E61001C	SUB.W	LR,PCFAR(A0),D7	
243	9:00018A	3147000C	MOVE.W	MAX_RASTER(A5),D7	
244	9:00018E	3E21001C	MOVE.W	MAX_RASTER(A5),D7	
245	9:000192	3147000E	MOVE.W	D7,BASE_OFF(A0)	
246	9:000196	3C2D0018	MOVE.W	MIN_RASTER(A5),D6	
247	9:00019A	40C7	EXT.L	D7	
248	9:00019E	40C6	EXT.L	D6	
249	9:0001A0	9E36	SUB.L	D6,D7	
250	9:0001A2	21470010	MOVE.L	D7,SCARCT(A0)	
251	9:0001A4	2E7000000000	MOVE.L	PDON_SP,SP	
252	9:0001AB	4C9F001F	MOVE.W	(SP)+,D2/D3/D1/D0/D4	
253	9:0001AE	4C9F3E60	MOVE.W	(SP)+,D5/D6/A6/A2/A3/A4/A1	
254	9:0001B2	4E75	MOVE.W	(SP)+,D5/D6/A6/A2/A3/A4/A1	
255	9:0001B6	4E75	RTS		

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
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
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
84

5

10

15

20

25

30

35

40

45

50

55

Copyright (c) 1983 Allied Corporation All Rights Reserved
Module --V--

P L O T D O N E
317 9:00023B 0342
318 9:00023A 6008

8000.V #1.D2
BRA.8 P00N.L11

PC	LOC	DATA	INSTR	COMMENT
320	9:00020C	4A42	LD	
321	9:00021E	6602	BRN.B	PDOWN_10
322	9:000240	3447	MOVE.W	D7,A2
323	9:000242	5242	ADDQ.W	#1,D2
324	9:000244		*	
325	9:000244	3E1F	MOVE.W	(SP)+,D7
326	9:000246	9F41	SUB.W	D1,D7
327	9:000248	E247	ASR.W	#1,D7
328	9:00024A	6504	PCSR.B	PDOWN_12
329	9:00024C	5242	ADDQ.W	#1,D2
330	9:00024E	603A	BRA.B	PDOWN_15
331	9:000250	5342	SUBQ.W	#1,D2
332	9:000252	6636	BRN.B	PDOWN_18
333	9:000254	BE4A	CMP.W	A2,D7
334	9:000256	6F32	BLE.B	PDOWN_15
335	9:000258	3407	MOVE.W	D7,D2
336	9:00025A	D044	CMP.W	D4,D0
337	9:00025C	6714	BEQ.B	PDOWN_14
338	9:00025E	5344	SUBQ.W	#1,D4
339	9:000260	D044	CMP.W	D4,D0
340	9:000262	670E	BEQ.B	PDOWN_14
341	9:000264	5283	ADDQ.L	#1,D5
342	9:000266	5086	SUBQ.L	#SHORT,D6
343	9:000268	6C02	BCR.B	#+4
344	9:00026A	6180	BSR.B	MORE_HZRD
345	9:00026C	30FCFF	MOVE.W	#BLANK_RASTER,(A4)+
346	9:000270	60EC	BRA	PDOWN_15

Copyright (c) 1983 Allied Corporation All Rights Reserved
Module -V-

LOC	DATA	CODE	COMMENT
356	9:000272	ADDQ.L	#1,D3
357	9:000272	SUBQ.L	#SHORT,D6
357	9:000274	BCE.B	#4
357	9:000276	BSR.B	MORE_MEMO
357	9:000278	MOVE.W	A2,(A4)+
357	9:00027A	ADDQ.L	#1,D3
358	9:00027C	SUBQ.L	#SHORT,D6
358	9:00027E	BCE.B	#4
358	9:000280	BSR.B	MORE_MEMO
358	9:000282	MOVE.W	D2,(A4)+
359	9:000284	MOVE.L	A4,A6
360	9:000288	CLR.W	D2
361	9:00028A	SUBQ.W	#2*SHORT,D3
362	9:00028C	BNE	PDOR_B
363	9:00028E	MOVE.L	A6,D7
364	9:000290	BEQ.B	PDOR_18
365	9:000292	LFA	-SHORT(A6),A0
367	9:000296	MOVE.W	#E000,D7
368	9:00029A	AND.W	(A0),D7
369	9:00029C	CMPL.W	#END_MEMO0E000,D7
370	9:0002A0	BEQ.B	PDOR_16
371	9:0002A2	BRI.B	PDOR_17
373	9:0002A4	ORI.W	#END_RASTER,(A0)
374	9:0002A8	BRA	PDOR_1
376	9:0002AC	ADDQ.L	#1,D3
377	9:0002AE	SUBQ.L	#SHORT,D6
377	9:0002B0	BCE.B	#4
377	9:0002B2	BSR.B	MORE_MEMO
377	9:0002B4	MOVE.W	#BLANK_RASTER,(A4)+
378	9:0002B8	MOVE.L	A4,A6
379	9:0002BA	BRA	PDOR_16
381	9:0002BC	SUBQ.W	#1,D4
382	9:0002BE	MOVE.W	D4,MAX_RASTER(A8)
383	9:0002C2	BRA	PDOR_16

Copyright (c) 1983 Allied Corporation All Rights Reserved
Module -V-

NO RE MEMO

```

305 910002C4 38FCDFFE
306 910002C8 2C4C
307 910002CA 8283
308 910002CC 40E7FFFF
309 910002D0 598F
310 910002D2 4857
311 910002D4 4CF9280000000000
312 910002D6 4EB9000000000000
313 910002D8 4BF9280000000000
314 910002DA 285F
315 910002DC 4CDF6FFF
316 910002DE 7CFC
317 910002E0 DCB90000000000
318 910002E2 2E9C
319 910002E4 6746
320 910002E6 4E75
321 910002E8 48E7FFFF
322 910002EA 598F
323 910002EC 40700011
324 910002EE 406F0004
325 910002F0 4EB90000000000
326 910002F2 2EB3
327 910002F4 20790000000000
328 910002F6 317C00110012
329 910002F8 2F20000E
330 910002FA 203F
331 910002FC 2F07
332 910002FE 4CF9280000000000
333 91000300 4EB90000000000
334 91000302 4BF9280000000000
335 91000304 4CDF7FFF
336 91000306 6148
337 91000308 6000FE43

```

NO RE MEMO

```

305 MORE MEMO
306 MOVE.W END_MEMO,(A4)+
307 ADDQ.L A4,A6
308 ADDQ.L #1,D6
309 MOVE.L D6-D7/A0-A3/A5-A6,-(SP)
310 SUBQ.L #LONG,SP
311 PEA (SP)
312 MOVE.L PASCAL_REGISTER,PASCAL_REGISTER
313 JSR NEXT
314 MOVE.L PASCAL_REGISTER,PASCAL_REGISTER
315 MOVE.L (SP)+,A4
316 MOVE.L (SP)+,D6-D7/A0-A3/A5-A6
317 MOVE.L #SHORT,D6
318 ADD.L MEMO_LENGTH,D6
319 MOVE.L A4,D7
320 BQJ.B PDOR_B1
321 RTS
322 PDOR_B0
323 MOVE.L D6-D7/A0-A6,-(SP)
324 SUBQ.L #LONG,SP
325 PEA FONTEAR
326 PEA LONG(SP)
327 JSR MEMA
328 MOVE.L A0,(SP)
329 MOVE.L PCHPAR,A0
330 MOVE.W FONTEAR,ENRUMRR(A0)
331 MOVE.L MCBADR(A0),D7
332 MOVE.L (SP)+,A0
333 MOVE.L D7,-(SP)
334 MOVE.L PASCAL_REGISTER,PASCAL_REGISTER
335 JSR KILL
336 MOVE.L PASCAL_REGISTER,PASCAL_REGISTER
337 MOVE.L (SP)+,D6-D7/A0-A6
338 BSR.B PLOT_CLEAR
339 BNA PDOR_B4

```


[illegible]

2 2
22 RETURN BLOCK TO POOL.

86

5
10
15
20
25
30
35
40
45
50
55

© (c) 1983 Allied Corporation All Rights Reserved

C A R - S E G M E N T

000450
000450 4050
000452 41F000000000
000450 0E3C0100
000450 4290
00045E 0947
000460 66FA
000462 205F
000464 4E75

XDZF CLEAR_SEGMENT
CLEAR_SEGMENT EQU *
PEA (A0)
LEA.L POSS_SEGMENT,A0
MOVE.W #1<POSS_SEGMENT,D7
CLR.L (A0)+
SUBQ.W #LONG,D7
BNE 2-4
MOVE.L (SP)+,A0
RTS
END

[illegible]

Copyright (c) 1983 Allied Corporation All Rights Reserved
Module -VI-

A A I I O O

```

26 9:00000000
27 9:00000000
28 9:00000000
29 9:00000004
30 9:00000006
31 9:00000008
32 9:0000000A
33 9:0000000C
34 9:0000000E
35 9:00000010
36 9:00000016
37 9:0000001A
38 9:0000001C
39 9:00000020
40 9:00000022
41 9:00000024
42 9:00000026
43 9:00000028
44 9:0000002A
45 9:0000002C
46 9:0000002E
47 9:00000030
48 9:00000032
49 9:00000038
50 9:0000003A
51 9:0000003C
52 9:00000040
53 9:00000044
54 9:00000046
55 9:0000004A
56 9:0000004C
57 9:0000004E
58 9:00000050
59 9:00000054
60 9:00000058
61 9:0000005C
62 9:00000060
63 9:00000064
64 9:00000068
65 9:0000006C
66 9:0000006E
67 9:00000070
68 9:00000074
69 9:00000078
70 9:0000007C
71 9:0000007E
72 9:00000080
73 9:00000084
74 9:00000088
75 9:0000008C
76 9:0000008E
77 9:00000090
78 9:00000094
79 9:00000098
80 9:0000009C
81 9:0000009E
82 9:000000A0
83 9:000000A4
84 9:000000A8
85 9:000000AC
86 9:000000B0
87 9:000000B4
88 9:000000B8
89 9:000000BC
90 9:000000C0
91 9:000000C4
92 9:000000C8
93 9:000000CC
94 9:000000D0
95 9:000000D4
96 9:000000D8
97 9:000000DC
98 9:000000E0
99 9:000000E4
100 9:000000E8
101 9:000000EC
102 9:000000F0
103 9:000000F4
104 9:000000F8
105 9:000000FC
106 9:00000100
107 9:00000104
108 9:00000108
109 9:0000010C
110 9:00000110
111 9:00000114
112 9:00000118
113 9:0000011C
114 9:00000120
115 9:00000124
116 9:00000128
117 9:0000012C
118 9:00000130
119 9:00000134
120 9:00000138
121 9:0000013C
122 9:00000140
123 9:00000144
124 9:00000148
125 9:0000014C
126 9:00000150
127 9:00000154
128 9:00000158
129 9:0000015C
130 9:00000160
131 9:00000164
132 9:00000168
133 9:0000016C
134 9:00000170
135 9:00000174
136 9:00000178
137 9:0000017C
138 9:00000180
139 9:00000184
140 9:00000188
141 9:0000018C
142 9:00000190
143 9:00000194
144 9:00000198
145 9:0000019C
146 9:000001A0
147 9:000001A4
148 9:000001A8
149 9:000001AC
150 9:000001B0
151 9:000001B4
152 9:000001B8
153 9:000001BC
154 9:000001C0
155 9:000001C4
156 9:000001C8
157 9:000001CC
158 9:000001D0
159 9:000001D4
160 9:000001D8
161 9:000001DC
162 9:000001E0
163 9:000001E4
164 9:000001E8
165 9:000001EC
166 9:000001F0
167 9:000001F4
168 9:000001F8
169 9:000001FC
170 9:00000200
171 9:00000204
172 9:00000208
173 9:0000020C
174 9:00000210
175 9:00000214
176 9:00000218
177 9:0000021C
178 9:00000220
179 9:00000224
180 9:00000228
181 9:0000022C
182 9:00000230
183 9:00000234
184 9:00000238
185 9:0000023C
186 9:00000240
187 9:00000244
188 9:00000248
189 9:0000024C
190 9:00000250
191 9:00000254
192 9:00000258
193 9:0000025C
194 9:00000260
195 9:00000264
196 9:00000268
197 9:0000026C
198 9:00000270
199 9:00000274
200 9:00000278
201 9:0000027C
202 9:00000280
203 9:00000284
204 9:00000288
205 9:0000028C
206 9:00000290
207 9:00000294
208 9:00000298
209 9:0000029C
210 9:000002A0
211 9:000002A4
212 9:000002A8
213 9:000002AC
214 9:000002B0
215 9:000002B4
216 9:000002B8
217 9:000002BC
218 9:000002C0
219 9:000002C4
220 9:000002C8
221 9:000002CC
222 9:000002D0
223 9:000002D4
224 9:000002D8
225 9:000002DC
226 9:000002E0
227 9:000002E4
228 9:000002E8
229 9:000002EC
230 9:000002F0
231 9:000002F4
232 9:000002F8
233 9:000002FC
234 9:00000300
235 9:00000304
236 9:00000308
237 9:0000030C
238 9:00000310
239 9:00000314
240 9:00000318
241 9:0000031C
242 9:00000320
243 9:00000324
244 9:00000328
245 9:0000032C
246 9:00000330
247 9:00000334
248 9:00000338
249 9:0000033C
250 9:00000340
251 9:00000344
252 9:00000348
253 9:0000034C
254 9:00000350
255 9:00000354
256 9:00000358
257 9:0000035C
258 9:00000360
259 9:00000364
260 9:00000368
261 9:0000036C
262 9:00000370
263 9:00000374
264 9:00000378
265 9:0000037C
266 9:00000380
267 9:00000384
268 9:00000388
269 9:0000038C
270 9:00000390
271 9:00000394
272 9:00000398
273 9:0000039C
274 9:000003A0
275 9:000003A4
276 9:000003A8
277 9:000003AC
278 9:000003B0
279 9:000003B4
280 9:000003B8
281 9:000003BC
282 9:000003C0
283 9:000003C4
284 9:000003C8
285 9:000003CC
286 9:000003D0
287 9:000003D4
288 9:000003D8
289 9:000003DC
290 9:000003E0
291 9:000003E4
292 9:000003E8
293 9:000003EC
294 9:000003F0
295 9:000003F4
296 9:000003F8
297 9:000003FC
298 9:00000400
299 9:00000404
300 9:00000408
301 9:0000040C
302 9:00000410
303 9:00000414
304 9:00000418
305 9:0000041C
306 9:00000420
307 9:00000424
308 9:00000428
309 9:0000042C
310 9:00000430
311 9:00000434
312 9:00000438
313 9:0000043C
314 9:00000440
315 9:00000444
316 9:00000448
317 9:0000044C
318 9:00000450
319 9:00000454
320 9:00000458
321 9:0000045C
322 9:00000460
323 9:00000464
324 9:00000468
325 9:0000046C
326 9:00000470
327 9:00000474
328 9:00000478
329 9:0000047C
330 9:00000480
331 9:00000484
332 9:00000488
333 9:0000048C
334 9:00000490
335 9:00000494
336 9:00000498
337 9:0000049C
338 9:000004A0
339 9:000004A4
340 9:000004A8
341 9:000004AC
342 9:000004B0
343 9:000004B4
344 9:000004B8
345 9:000004BC
346 9:000004C0
347 9:000004C4
348 9:000004C8
349 9:000004CC
350 9:000004D0
351 9:000004D4
352 9:000004D8
353 9:000004DC
354 9:000004E0
355 9:000004E4
356 9:000004E8
357 9:000004EC
358 9:000004F0
359 9:000004F4
360 9:000004F8
361 9:000004FC
362 9:00000500
363 9:00000504
364 9:00000508
365 9:0000050C
366 9:00000510
367 9:00000514
368 9:00000518
369 9:0000051C
370 9:00000520
371 9:00000524
372 9:00000528
373 9:0000052C
374 9:00000530
375 9:00000534
376 9:00000538
377 9:0000053C
378 9:00000540
379 9:00000544
380 9:00000548
381 9:0000054C
382 9:00000550
383 9:00000554
384 9:00000558
385 9:0000055C
386 9:00000560
387 9:00000564
388 9:00000568
389 9:0000056C
390 9:00000570
391 9:00000574
392 9:00000578
393 9:0000057C
394 9:00000580
395 9:00000584
396 9:00000588
397 9:0000058C
398 9:00000590
399 9:00000594
400 9:00000598
401 9:0000059C
402 9:000005A0
403 9:000005A4
404 9:000005A8
405 9:000005AC
406 9:000005B0
407 9:000005B4
408 9:000005B8
409 9:000005BC
410 9:000005C0
411 9:000005C4
412 9:000005C8
413 9:000005CC
414 9:000005D0
415 9:000005D4
416 9:000005D8
417 9:000005DC
418 9:000005E0
419 9:000005E4
420 9:000005E8
421 9:000005EC
422 9:000005F0
423 9:000005F4
424 9:000005F8
425 9:000005FC
426 9:00000600
427 9:00000604
428 9:00000608
429 9:0000060C
430 9:00000610
431 9:00000614
432 9:00000618
433 9:0000061C
434 9:00000620
435 9:00000624
436 9:00000628
437 9:0000062C
438 9:00000630
439 9:00000634
440 9:00000638
441 9:0000063C
442 9:00000640
443 9:00000644
444 9:00000648
445 9:0000064C
446 9:00000650
447 9:00000654
448 9:00000658
449 9:0000065C
450 9:00000660
451 9:00000664
452 9:00000668
453 9:0000066C
454 9:00000670
455 9:00000674
456 9:00000678
457 9:0000067C
458 9:00000680
459 9:00000684
460 9:00000688
461 9:0000068C
462 9:00000690
463 9:00000694
464 9:00000698
465 9:0000069C
466 9:000006A0
467 9:000006A4
468 9:000006A8
469 9:000006AC
470 9:000006B0
471 9:000006B4
472 9:000006B8
473 9:000006BC
474 9:000006C0
475 9:000006C4
476 9:000006C8
477 9:000006CC
478 9:000006D0
479 9:000006D4
480 9:000006D8
481 9:000006DC
482 9:000006E0
483 9:000006E4
484 9:000006E8
485 9:000006EC
486 9:000006F0
487 9:000006F4
488 9:000006F8
489 9:000006FC
490 9:00000700
491 9:00000704
492 9:00000708
493 9:0000070C
494 9:00000710
495 9:00000714
496 9:00000718
497 9:0000071C
498 9:00000720
499 9:00000724
500 9:00000728
501 9:0000072C
502 9:00000730
503 9:00000734
504 9:00000738
505 9:0000073C
506 9:00000740
507 9:00000744
508 9:00000748
509 9:0000074C
510 9:00000750
511 9:00000754
512 9:00000758
513 9:0000075C
514 9:00000760
515 9:00000764
516 9:00000768
517 9:0000076C
518 9:00000770
519 9:00000774
520 9:00000778
521 9:0000077C
522 9:00000780
523 9:00000784
524 9:00000788
525 9:0000078C
526 9:00000790
527 9:00000794
528 9:00000798
529 9:0000079C
530 9:000007A0
531 9:000007A4
532 9:000007A8
533 9:000007AC
534 9:000007B0
535 9:000007B4
536 9:000007B8
537 9:000007BC
538 9:000007C0
539 9:000007C4
540 9:000007C8
541 9:000007CC
542 9:000007D0
543 9:000007D4
544 9:000007D8
545 9:000007DC
546 9:000007E0
547 9:000007E4
548 9:000007E8
549 9:000007EC
550 9:000007F0
551 9:000007F4
552 9:000007F8
553 9:000007FC
554 9:00000800
555 9:00000804
556 9:00000808
557 9:0000080C
558 9:00000810
559 9:00000814
560 9:00000818
561 9:0000081C
562 9:00000820
563 9:00000824
564 9:00000828
565 9:0000082C
566 9:00000830
567 9:00000834
568 9:00000838
569 9:0000083C
570 9:00000840
571 9:00000844
572 9:00000848
573 9:0000084C
574 9:00000850
575 9:00000854
576 9:00000858
577 9:0000085C
578 9:00000860
579 9:00000864
580 9:00000868
581 9:0000086C
582 9:00000870
583 9:00000874
584 9:00000878
585 9:0000087C
586 9:00000880
587 9:00000884
588 9:00000888
589 9:0000088C
590 9:00000890
591 9:00000894
592 9:00000898
593 9:0000089C
594 9:000008A0
595 9:000008A4
596 9:000008A8
597 9:000008AC
598 9:000008B0
599 9:000008B4
600 9:000008B8
601 9:000008BC
602 9:000008C0
603 9:000008C4
604 9:000008C8
605 9:000008CC
606 9:000008D0
607 9:000008D4
608 9:000008D8
609 9:000008DC
610 9:000008E0
611 9:000008E4
612 9:000008E8
613 9:000008EC
614 9:000008F0
615 9:000008F4
616 9:000008F8
617 9:000008FC
618 9:00000900
619 9:00000904
620 9:00000908
621 9:0000090C
622 9:00000910
623 9:00000914
624 9:00000918
625 9:0000091C
626 9:00000920
627 9:00000924
628 9:00000928
629 9:0000092C
630 9:00000930
631 9:00000934
632 9:00000938
633 9:0000093C
634 9:00000940
635 9:00000944
636 9:00000948
637 9:0000094C
638 9:00000950
639 9:00000954
640 9:00000958
641 9:0000095C
642 9:00000960
643 9:00000964
644 9:00000968
645 9:0000096C
646 9:00000970
647 9:00000974
648 9:00000978
649 9:0000097C
650 9:00000980
651 9:00000984
652 9:00000988
653 9:0000098C
654 9:00000990
655 9:00000994
656 9:00000998
657 9:0000099C
658 9:000009A0
659 9:000009A4
660 9:000009A8
661 9:000009AC
662 9:000009B0
663 9:000009B4
664 9:000009B8
665 9:000009BC
666 9:000009C0
667 9:000009C4
668 9:000009C8
669 9:000009CC
670 9:000009D0
671 9:000009D4
672 9:000009D8
673 9:000009DC
674 9:000009E0
675 9:000009E4
676 9:000009E8
677 9:000009EC
678 9:000009F0
679 9:000009F4
680 9:000009F8
681 9:000009FC
682 9:00000A00
683 9:00000A04
684 9:00000A08
685 9:00000A0C
686 9:00000A10
687 9:00000A14
688 9:00000A18
689 9:00000A1C
690 9:00000A20
691 9:00000A24
692 9:00000A28
693 9:00000A2C
694 9:00000A30
695 9:00000A34
696 9:00000A38
697 9:00000A3C
698 9:00000A40
699 9:00000A44
700 9:00000A48
701 9:00000A4C
702 9:00000A50
703 9:00000A54
704 9:00000A58
705 9:00000A5C
706 9:00000A60
707 9:00000A64
708 9:00000A68
709 9:00000A6C
710 9:00000A70
711 9:00000A74
712 9:00000A78
713 9:00000A7C
714 9:00000A80
715 9:00000A84
716 9:00000A88
717 9:00000A8C
718 9:00000A90
719 9:00000A94
720 9:00000A98
721 9:00000A9C
722 9:00000AA0
723 9:00000AA4
724 9:00000AA8
725 9:00000AAC
726 9:00000AB0
727 9:00000AB4
728 9:00000AB8
729 9:00000ABC
730 9:00000AC0
731 9:00000AC4
732 9:00000AC8
733 9:00000ACC
734 9:00000AD0
735 9:00000AD4
736 9:00000AD8
737 9:00000ADC
738 9:00000AE0
739 9:00000AE4
740 9:00000AE8
741 9:00000AEC
742 9:00000AF0
743 9:00000AF4
744 9:00000AF8
745 9:00000AFC
746 9:00000B00
747 9:00000B04
748 9:00000B08
749 9:00000B0C
750 9:00000B10
751 9:00000B14
752 9:00000B18
753 9:00000B1C
754 9:00000B20
755 9:00000B24
756 9:00000B28
757 9:00000B2C
758 9:00000B30
759 9:00000B34
760 9:00000B38
761 9:00000B3C
762 9:00000B40
763 9:00000B44
764 9:00000B48
765 9:00000B4C
766 9:00000B50
767 9:00000B54
768 9:00000B58
769 9:00000B5C
770 9:00000B60
771 9:00000B64
772 9:00000B68
773 9:00000B6C
774 9:00000B70
775 9:00000B74
776 9:00000B78
777 9:00000B7C
778 9:00000B80
779 9:00000B84
780 9:00000B88
781 9:00000B8C
782 9:00000B90
783 9:00000B94
784 9:00000B98
785 9:00000B9C
786 9:00000BA0
787 9:00000BA4
788 9:00000BA8
789 9:00000BAC
790 9:00000BB0
791 9:00000BB4
792 9:00000BB8
793 9:00000BBC
794 9:00000BC0
795 9:00000BC4
796 9:00000BC8
797 9:00000BCC
798 9:00000BD0
799 9:00000BD4
800 9:00000BD8
801 9:00000BDC
802 9:00000BE0
803 9:00000BE4
804 9:00000BE8
805 9:00000BEC
806 9:00000BF0
807 9:00000BF4
808 9:00000BF8
809 9:00000BFC
810 9:00000C00
811 9:00000C04
812 9:00000C08
813 9:00000C0C
814 9:00000C10
815 9:00000C14
816 9:00000C18
817 9:00000C1C
818 9:00000C20
819 9:00000C24
820 9:00000C28
821 9:00000C2C
822 9:00000C30
823 9:00000C34
824 9:00000C38
825 9:00000C3C
826 9:00000C40
827 9:00000C44
828 9:00000C48
829 9:00000C4C
830 9:00000C50
831 9:00000C54
832 9:00000C58
833 9:00000C5C
834 9:00000C60
835 9:00000C64
836 9:00000C68
837 9:00000C6C
838 9:00000C70
839 9:00000C74
840 9:00000C78
841 9:00000C7C
842 9:00000C80
843 9:00000C84
844 9:00000C88
845 9:00000C8C
846 9:00000C90
847 9:00000C94
848 9:00000C98
849 9:00000C9C
850 9:00000CA0
851 9:00000CA4
852 9:00000CA
```

Copyright (c) 1983 Allied Corporation All Rights Reserved
Module -VI-

A	A	L	L	O	C	06	9:00004C	43F00FF0		ALCA_7:	LEA.L	HUNK-FREESE_LENGTH,A1	
						67	9:000050	B0B9			CMF.L	A1,D0	
						68	9:000052	6504			BCR.B	ALCR.B	* LOWZA
						69	9:000054	2240			MOVE.L	D0,A1	
						70	9:000056	6002			BRA.B	ALCR_9	
						71	9:000058	50B9		ALCA_8:	ADDQ.L	*FREESE_LENGTH,A1	
						72	9:00005A	2F09		ALCA_9:	MOVE.L	A1,D7	
						73	9:00005C	0607000003FF			ADD.L	*1023,D7	
						74	9:000062	E0BF			LSR.L	*8,D7	
						75	9:000064	E4BF			LSR.L	*10-8,D7	
						76	9:000066	4BF7FF7E			MOVEN.L	D0-D7/A1-A6,-(BP)	
						77	9:00006A	59BF			SUBQ.L	*LONG,SP	
						78	9:00006C	4057			PEA.L	(SP)	
						79	9:00006E	2F07			MOVEN.L	D7,-(BP)	* MAKE ROOM FOR POINTER TO NEW
						80	9:000070	4CF92B0000000000			MOVEN.L	PASCAL_REGISTERS,PASCAL_RCLST	* GET NEW MEMORY
						81	9:00007B	4EB900000000			JSR	FETCH_MEM	*
						82	9:00007E	4BF92B0000000000			MOVEN.L	PASCAL_RCLST,PASCAL_REGISTERS	
						83	9:000086	205F			MOVEN.L	(BP)+,A0	
						84	9:000088	4CD77EFF			MOVEN.L	(BP)+,D0-D7/A1-A6	
						85	9:00008C	2F09			MOVE.L	A1,D7	* LENGTH OF THIS HOLE
						86	9:00008E	51ED002C			SP	USED_1_SEGMENT(A0)	
						87	9:000092	610C			DSR.B	AFRE_9	
						88	9:000094	6000FF7A			BRA	ALCA_1	

Copyright (c) 1983 Allied Corporation All Rights Reserved
Module -VI-

131 9:0000E4	6204	APRE_6:	BRI_8	APRE_7	
132 9:0000E6	2F0A		MOVE.L	A2, D7	
133 9:0000E8	6623	BRE_8	APRE_20		* TRAP ON TROUBLE
134 9:0000EA	208A	APRE_7:	MOVE.L	A2, (A0)	* INSERT THIS HOLE
135 9:0000FC	2F09	APRE_8:	MOVE.L	A1, D7	
136 9:0000FE	6710		BFI_8	APRE_10	
137 9:0000F0	B1C9		CHP.L	A1, A0	
138 9:0000F2	6314		BIS_8	APRE_10	
139 9:0000F4	DEA00004		ADD.L	FREE9L_SIZE(A1), D7	
140 9:0000F8	B1C7		CHP.L	D7, A0	
141 9:0000FA	6608		BRE_8	APRE_9	* NO COLLAPSE BACKWARD
142 9:0000FC	2294		MOVE.L	(A0), (A1)	
143 9:0000FE	D1A00004		ADD.L	D0, FREE9L_SIZE(A1)	
144 9:000102	600B		BRA_8	APRE_11	
145 9:000104	630C	APRE_9:	BIS_8	APRE_20	
146 9:000106	2203		MOVE.L	A0, (A1)	* TRAP ON TROUBLE
147 9:000108	21400004	APRE_10	MOVE.L	D0, FREE9L_SIZE(A0)	* PREVIOUS LINK TO NEW
148 9:00010C	4CDF0601	APRE_11	MOVE.L	(BP)+, D0/A1/A2	
149 9:000110	4E75		RTS		
150					
151 9:000112	207900000000	APRE_20	PCBP, A0		
152 9:000118	317C004B0012		MOVE.W	*PARERR, ERRORNN(A0)	
153 9:00011E	59BF		SUBQ.L	*LONG, BP	
154 9:000120	4073004B		PEA	PARERR	
155 9:000124	406F0004		PEA	LONG(BP)	
156 9:000128	4E9000000000		JSR	MEVA	
157 9:00012E	207900000000		MOVE.L	PCBP, A0	
158 9:000134	2F20000E		MOVE.L	PCBADR(A0), -(BP)	
159 9:000138	4CF9200000000000		MOVE.L	PASCAL_REGISTERS, PSCAL_RGLST	
160 9:000140	4E9000000000		JSR	*KILL	
161 9:000146	40F9200000000000		MOVE.L	PSCAL_RGLST, PASCAL_REGISTERS	
162 9:00014E	4EF90000000000		JMP	CRV_ERROR	

Copyright (c) 1983 Allied Corporation All Rights Reserved
Module -V1-

```

I N I T I A L I Z A T I O N
164 9:000154 4A2D002C
165 9:000158 6702
166 9:00015A 605E
167 9:00015C 40F70300
168 9:000160 207908000000
169 9:000166 2F18
170 9:000168 6740
171 9:00016A 23C790000000
172 9:000170 0E00000000
173 9:000176 67E0
174 9:000178 40F7F7FE
175 9:00017C 2F07
176 9:00017E 4057
177 9:000180 2F10
178 9:000182 0207000003FF
179 9:000188 6600
180 9:00018A 2F10
181 9:00018C 0607000003FF
182 9:000192 F00F
183 9:000194 F40F
184 9:000196 2F07
185 9:000198 4CF9200000000000
186 9:0001A0 4F0900000000
187 9:0001A6 40F9200000000000
188 9:0001AE 000F
189 9:0001B0 4C0F7777
190 9:0001D4 60AA
191 9:0001B6 4C0F01C0
192
193 9:0001BA 4030
194 9:0001BC 2F3900000000
195 9:0001C2 6710
196 9:0001C4 2047
197 9:0001C6 23C000000000
198 9:0001CC 4200
199 9:0001CE 20B900000000
200 9:0001D4 00F0002C
201 9:0001D0 205F
202 9:0001DA 4E75
203 9:0001DC 40F77777
204 9:0001E0 407900000000
205 9:0001E6 40700014
206 9:0001EA 4CF9200000000000
207 9:0001F2 4F0900000000
208 9:0001F8 40F9200000000000
209 9:000200 4C0F7777
210 9:000204 23FC000000000000
211 9:00020E 60AC
212
213

```

```

STORE_FINAL TEST.B USED_1_SEGMENT(A0)
REQ.B A+4
BRA.B STORE_INIT
MOVE.L D6/D7/A0,-(SP)
STORE_1 MOVE.L PLOT_AREA,A0
MOVE.L (A0)+,D7
REQ.B STORE_2
MOVE.L D7,PLOT_AREA
CMP.L FIRST_SEGMENT,D7
REQ.B STORE_1
MOVE.L D6/D7/A0,-(SP)
MOVE.L D7,-(SP)
PEA (SP)
MOVE.L (A0),D7
AND.L #1023,D7
BNE AFRE_20
MOVE.L (A0),D7
ADD.L #1023,D7
LSR.L #8,D7
LSR.L #10-8,D7
MOVE.L D7,-(SP)
MOVE.L PASCAL_REGISTERS,PASCAL_REGIST
JSR DISP_MEM
MOVE.L PASCAL_REGIST,PASCAL_REGISTERS
ADDQ.L #LONG,SP
MOVE.L (SP)+,D6-D7/A0-A6
BRA STORE_1
STORE_2 MOVE.L (SP)+,D6/D7/A0
STORE_INIT PEA (A0)
STORE_1 MOVE.L FIRST_SEGMENT,D7
REQ.B STORE_2
MOVE.L D7,A0
MOVE.L A0,PLOT_AREA
CLR.L (A0)+
MOVE.L FIRST_LENGTH,(A0)
STORE_1 USED_1_SEGMENT(A0)
MOVE.L (SP)+,A0
RTS
STORE_2 MOVE.L D6-D7/A0-A6,-(SP)
PEA.L FIRST_SEGMENT
PEA 20
MOVE.L PASCAL_REGISTERS,PASCAL_REGIST
JSR FETCH_MEM
MOVE.L PASCAL_REGIST,PASCAL_REGISTERS
MOVE.L (SP)+,D6-D7/A0-A6
MOVE.L #20*1024,FIRST_LENGTH
BRA STORE_1
END

```

* LENGTH OF PARTITION
* PARTITION WAS NOT CLOSTED

5

10

15

20

25

30

35

40

45

50

55

Copyright (c) 1983 Allied Corporation All Rights Reserved
Module -VII-

T R I G F U N C T I O N S

1	2	6	7	8	9	10	11	12
			+ * ANGULAR CONSTANTS					
			HDEC_90: SET 90*2					
			HDEC_89P9: SET 89*2+1					
			HDEC_90: SET 90*2					
			HDEC_180: SET HDEC_90*2					
			HDEC_360: SET HDEC_180*2					

T R I O I D W T 1 . 0 C U R V E P O N T A N G T A R C E N T / S I N E / C O S I N E F U N C T I O N S

+

* ANGULAR CONSTANTS

HDEC_90: SET 90*2
HDEC_89P9: SET 89*2+1
HDEC_90: SET 90*2
HDEC_180: SET HDEC_90*2
HDEC_360: SET HDEC_180*2

8 000000A0
9 000000B3
10 000000B4
11 00000160
12 000002D0

Copyright (c) 1983 Allied Corporation All Rights Reserved
Module -VII-

ARCTANGENT

```

14 * ARCTAN
15 *
16 *CALL FRAME
17 * D7(16:31)= DELTA Y
18 * D7(00:16)= DELTA X
19 *
20 *RETURN FRAME
21 * D7(00:31)= ARCTANGENT ANGLE IN 1/2 DEGREES
22 *
23 *REGISTERS DESTROYED - NONE
24 *REGISTERS USED - DELTA_Y(D0)/D7/A0
25 *
26 *FUNCTION
27 * GIVEN DELTA_Y AND DELTA_X A ANGLE IN 1/2 DEGREES IS RETURNED BY TAKING
28 * THE SLOPE (IE ABS(DELTA_Y)/ABS(DELTA_X)) AND LOOKING IT UP INTO A TABLE.
29 * THE TABLES ARE BROKEN INTO 3 PARTS.
30 * 1) ARCTANGENT(SLOPE) 0<SLOPE<2 BY 1/1280
31 * 2) ARCTANGENT(SLOPE) 2<SLOPE<6 BY 1/640
32 * 3) TANGENT(THETA) 80<THETA<=89 BY 1/20000
33 * FINALLY THE SIGNS OF DELTA_X AND DELTA_Y ARE USED TO SET THE QUADRANT
34 * OF THE RETURN VALUE
35 *
36 *REGISTERS
37 *DELTA_Y SET D0
38 *DELTA_Y.W SET D0.W
39
40
41
42 * ARCTAN * ARCTANGENT FUNCTION
43 * XREF ATN_0_2,ATN_2_6,TAN_0_0 * FROM TABLES
44 * SECTION 09 * CODE SECTION
45
46 *
47 *
48 *
49 *
50 *
51 *
52 *
53 *
54 *
55 *
56 *
57 *
58 *
59 *
60 *

```

Copyright (c) 1983 Allied Corporation All Rights Reserved
 July -VII-

62	9:00002B	6A*2	ARCTN_3	POU	*	ARCTN_3	POU	*	
63	9:00002B	4400		RPL_8	++4		RPL_8	++4	
64	9:00002A	40C7		NFG.L	D0		NFG.L	D0	
65	9:00002C	6006		EXT.L	D7		EXT.L	D7	
66	9:000030	3F3C00B4		BNE.B	ARCTN_4		BNE.B	ARCTN_4	
67	9:000034	60DA		MOVE.W	#HDEC_90.D7		MOVE.W	#HDEC_90.D7	
68	9:000036			BRA	ARCTN_1		BRA	ARCTN_1	
69	9:00003B	6A02		POU	*		POU	*	
70	9:00003B	4407		RPL_8	++4		RPL_8	++4	
71	9:00003A	2047		NFG.L	D7		NFG.L	D7	
72	9:00003C	D1C8		MOVE.L	D7.A0		MOVE.L	D7.A0	
73	9:00003E	B1C0		ADD.L	A0.A0		ADD.L	A0.A0	
74	9:000040	6312		CHP.L	D0.A0		CHP.L	D0.A0	
75	9:000042	EF00		RIS.B	ARCTN_6		RIS.B	ARCTN_6	
76	9:000044	41F900000000		ASL.L	#7.D0		ASL.L	#7.D0	
77	9:00004A	80C7		LEA.L	ATN_0_2.A0		LEA.L	ATN_0_2.A0	
78	9:00004C	7E80		ARCTN_8	DIVU		ARCTN_8	DIVU	
79	9:00004E	1F300000		* D0 ALREADY INDEX TO CLEAR					
80	9:000052	60BC		MOVE.L	#0.D7		MOVE.L	#0.D7	
81	9:000054	D1C7		BRA	ARCTN_1		BRA	ARCTN_1	
82	9:00005B	B1C0		ADD.L	D7.A0		ADD.L	D7.A0	
83	9:00005A	630A		CHP.L	D0.A0		CHP.L	D0.A0	
84	9:00005C	ED00		RIS.B	ARCTN_7		RIS.B	ARCTN_7	
85	9:00005E	41F9FFFFF7F00		ASL.L	#6.D0		ASL.L	#6.D0	
86	9:000064	60F4		LEA.L	ATN_2_6-120#CLEAR.A0		LEA.L	ATN_2_6-120#CLEAR.A0	
87	9:000066	2047		BRA	ARCTN_8		BRA	ARCTN_8	
88	9:00006A	EF07		MOVE.L	D7.A0		MOVE.L	D7.A0	
89	9:00006A	BE09		ASL.L	#16-9.D7		ASL.L	#16-9.D7	
90	9:00006C	6316		CHP.L	D0.D7		CHP.L	D0.D7	
91	9:00006E	2E08		RIS.B	ARCTN7A		RIS.B	ARCTN7A	
92	9:000070	F100		MOVE.L	A0.D7		MOVE.L	A0.D7	
93	9:000072	F309		ASL.L	#8.D0		ASL.L	#8.D0	
94	9:000074	00C7		ASL.L	#9-8.D0		ASL.L	#9-8.D0	
95	9:000076	690C		DIVU	D7.D0		DIVU	D7.D0	
96	9:00007B	41F90000000000		BVS.B	ARCTN7A		BVS.B	ARCTN7A	
97	9:00007E	B0600024		LEA.L	TAN_R0_09.A0		LEA.L	TAN_R0_09.A0	
98	9:000082	6506		CHP.W	18#SHORT(A0),D0		CHP.W	18#SHORT(A0),D0	
99	9:000084	7E12		BVS.B	ARCTN_8		BVS.B	ARCTN_8	
100	9:00008A	3F3C00B3		BRA	ARCTN_1		BRA	ARCTN_1	
101	9:00008C	60707000		MOVE.L	#0.A0.D7.L),D0		MOVE.L	#0.A0.D7.L),D0	
102	9:00008E	7EFE		CHP.W	#0.A0.D7.L),D0		CHP.W	#0.A0.D7.L),D0	
103	9:000090	6404		RVC.L	#SHORT.D7		RVC.L	#SHORT.D7	
104	9:000092	0407		MOVE.L	#SHORT.D7		MOVE.L	#SHORT.D7	
105	9:000094	B0707000		CHP.L	#0.A0.D7.L),D0		CHP.L	#0.A0.D7.L),D0	
106	9:000096	62F0		BHI.B	ARCTN_9		BHI.B	ARCTN_9	
107	9:00009A	F247		ASR.W	#1.D7		ASR.W	#1.D7	
108	9:00009C	064700A0		ADD.W	#HDEC_00.D7		ADD.W	#HDEC_00.D7	
109	9:00009E	6000FF6C		BRA	ARCTN_1		BRA	ARCTN_1	
110	9:0000A2								
111	9:0000A2								

* SAVE X
 * X*2
 * Y < 30X

* OFFSET BY BYTES

* SET QUADRANT
 * X*3
 * X*6
 * Y < 60X
 * Y*64

* LOWER

* HIGH OR SAME

* UNDO SHORT INDEX

abt (c) 1983 Allied Corporation All Rights Reserved
-VII-

E C O N T E N T S

```

* S I N E / C O S I N E
* CALL FRAMZ
* D7(0:10) = +/- THETA IN 1/8 DEGREES
* RETURN FRAMZ
* D7(0:31) = +/- SINE(THETA)/COSINE(THETA) * 2**18
* REGISTERS DESTROYED - NONE
* REGISTERS USED - D7/A0
* FUNCTION
* IF COMPENSATE FOR COSINE IE. COSINE(THETA) == SINE(THETA+90)
* THE VALUE OF ABS(THETA) MOD 360 IS REDUCED WITHIN RANGE OF (0..90)
* THIS NUMBER IS USED TO LOOKUP INTO A SINE TABLE
* THEN COMPENSATE FOR NEGATIVE THETA IE. SINE(-THETA) == -SINE(THETA)
XDEF SINE, COSINE * SINE / COSINE FUNCTIONS
XREF SINE_TBL * FROM TABLES

+
COSINE: SECTION 09 * CODE SECTION
EQU *
ADDI.W #DEG_90, D7
SINE: EQU *
PEA.L (A0) * SAVE A0
LEA.L SINE_TBL, A0 * SET D7(31) TO SIGN OF THETA
EXT.L D7
BPL.S ++4
NEG.W D7 * HIGH OR SAME
SINE_1: SUBI.W #DEG_360, D7
BCC.B SINE_1
ADDI.W #DEG_360, D7
CMPI.W #DEG_180, D7
BLS.B SINE_2
SUBI.W #DEG_180, D7
RCHG.L #31, D7
SINE_2: CMPI.W #DEG_90, D7
BLS.B SINE_3
NEG.W D7
SINE_3: ADDI.W #DEG_180, D7
EQU *
ADD.W D7, D7
MOVE.W 0(A0, D7, W), D7
MOVE.L (SP), A0
TST.L D7
BPL.S ++4
NEG.W D7
EXT.L D7
RTS

```

```

9:0000A6 064700B4
9:0000A6 4B50
9:0000AA 41F300000000
9:0000AC 4BC7
9:0000B2 6A02
9:0000B6 4447
9:0000B8 044702D0
9:0000BC 64FA
9:0000BE 064702D0
9:0000C2 0C47016B
9:0000C6 630B
9:0000CB 0447016B
9:0000CC 0B47001F
9:0000D0 0C4700B4
9:0000D4 6306
9:0000D6 4447
9:0000D8 0647016B
9:0000DC
9:0000E0 DE47
9:0000E2 3E307000
9:0000E4 203F
9:0000F4 4AB7
9:0000F6 6A02
9:0000F8 4447
9:0000FA 4BC7
9:0000EC 4E7B

```

ght (c) 1983 Allied Corporation All Rights Reserved
-V11-

D A N G L E

```
* F O L D - A N G L E
*
*CALL FRAME
* D7= DELTA_ANGLE = ANGLE2-ANGLE1    ANGLE2.ANGL1=(-180..+180)
*
*RETURN FRAME
* D7= DELTA_ANGLE SUCH THAT DELTA_ANGLE=(-180..+180)
*
*REGISTERS DESTROYED - NONE
*REGISTERS USED - D7
*
*FUNCTION
* IF DELTA_ANGLE > 180 THEN -360,RETURN
* IF DELTA_ANGLE < -180 THEN +360,RETURN
```

```
XDEF FOLD_ANGLE
SECTION 09 * CODE SECTION
FOLD_ANGLE: EQU *
    CHPI.W #HDEC_180,D7
    BLE.S FANG_1
    SUBI.W #HDEC_360,D7
    BRA.S FANG_2
    CHPI.W #HDEC_180,D7
    MCL.S FANG_2
    ADDI.W #HDEC_360,D7
    EXT.L D7
    RTS
END
```

```
9:0000FE
9:0000FE
9:0000FE 0C470168
9:0000F2 6F06
9:0000F4 044702D0
9:0000F8 600A
9:0000FA 0C47FE98
9:0000FE 6C05
9:000100 064702D0
9:000104 40C7
9:000106 4E75
```

5

10

15

20

25

30

35

40

45

50

55

it (a) 1983 Allied Corporation All Rights Reserved
VII-

```

only Trlg Table
segment Table
    86 Arotament Table
    111 segment Table
    136 Xl Table

:00000000 + section 00 x Code Section

xdef sim02,sim2_6,sim026,tan00_09,simtan.tbl
xdef simtbl,tan.tbl,xl.tbl

```

R e a d o n l y T r i g T a b l e s
A c t a n g e n t T a b l e s

10	C:000000	0BA40C480CEFF0DCB	tan_00_B9: equ *	* tangent of 00.89 by 1/2deg
11	C:000000	18121AEA1EB42339	dc.w	2900,3144,3327,3531,3762,4025,4326,4675,5085,5572
12	C:000014	0001020304040506	dc.w	6162,6090,7012,9017,10659,13031,16758,23465,-26424
13	C:000026	0703090A0B0C0D	atn_0_2: equ *	* arctangent of 0.2 by 1/128
14	C:000026	0F0F101112131414	dc.b	0,1,2,3,4,5,6
15	C:00002E	15161718191A1B	dc.b	7,8,9,10,11,12,13
16	C:000036	1C1D1E1F202122	dc.b	14,15,16,17,18,19,20,20
17	C:00003E	23242526272828	dc.b	21,22,23,24,25,26,26,27
18	C:000046	292A2B2C2D2E2F	dc.b	28,29,30,31,31,32,33,34
19	C:00004E	2F303132333434	dc.b	35,36,36,37,38,39,40,40
20	C:000056	30313233343535	dc.b	41,42,43,43,44,45,46,47
21	C:00005E	31323334353636	dc.b	47,48,49,49,50,51,52,52
22	C:000066	32333435363737	dc.b	53,54,55,55,56,57,57,58
23	C:00006F	33343536373838	dc.b	59,59,60,61,61,62,63,63
24	C:000076	34353637383939	dc.b	64,65,65,66,67,67,68,68
25	C:00007E	35363738394040	dc.b	69,70,70,71,71,72,73,73
26	C:000086	36373839404141	dc.b	74,74,75,75,76,77,77,78
27	C:00008E	37383940414242	dc.b	78,79,79,80,80,81,81,82
28	C:000096	38394041424343	dc.b	82,83,83,84,84,85,85,86
29	C:00009E	39404142434444	dc.b	86,87,87,88,88,89,89,90
30	C:0000A6	40414243444545	dc.b	90,90,91,91,92,92,93,93
31	C:0000AE	41424344454646	dc.b	93,94,94,95,95,96,96,96
32	C:0000B6	42434445464747	dc.b	97,97,98,98,99,99,99
33	C:0000BE	43444546474848	dc.b	100,100,101,101,101,102,102,102
34	C:0000C6	44454647484949	dc.b	103,103,103,104,104,104,105,105
35	C:0000CE	45464748495050	dc.b	105,106,106,106,107,107,107,108
36	C:0000D6	46474849505151	dc.b	108,108,109,109,109,109,110,110
37	C:0000DE	47484950515252	dc.b	110,111,111,111,112,112,112,112
38	C:0000E6	48495051525353	dc.b	113,113,113,113,114,114,114,115
39	C:0000EE	49505152535454	dc.b	115,115,115,116,116,116,116,117
40	C:0000FE	50515253545555	dc.b	117,117,117,118,118,118,118,118
41	C:0000FE	51525354555656	dc.b	119,119,119,119,120,120,120,120
42	C:000106	52535455565757	dc.b	121,121,121,121,122,122,122,122
43	C:00010E	53545556575858	dc.b	122,122,123,123,123,123,123,124
44	C:000116	54555657585959	dc.b	124,124,124,124,125,125,125,125
45	C:00011E	55565758596060	dc.b	125,126,126,126,126,126,127,127
47	C:000026	71727374757676	atn_026: set atn_0_2	* atn_0_2 is followed by atn_2_6
49	C:000026	73747576777777	atn_tan_tbl: set atn_0_2	* tan_00_90 is -19 shortwords of tan_0_2
53	C:000126	75767778797979	atn_2_6: equ *	* arctangent 2.6 by 1/64
54	C:00012E	76777879808080	dc.b	127,127,-128,-128,-127,-127,-127
55	C:000136	77787980818181	dc.b	-126,-126,-126,-125,-125,-125,-124
56	C:00013E	78798081828282	dc.b	-124,-124,-123,-123,-122,-122,-122
57	C:000146	79808182838383	dc.b	-122,-121,-121,-121,-121,-120,-120
58	C:00014E	80818283848484	dc.b	-120,-119,-119,-119,-118,-118,-118
59	C:000156	81828384858585	dc.b	-118,-117,-117,-117,-117,-116,-116
60	C:00015E	82838485868686	dc.b	-116,-116,-116,-115,-115,-115,-115
61	C:000166	83848586878787	dc.b	-114,-114,-114,-114,-113,-113,-113
62	C:00016E	84858687888888	dc.b	-113,-113,-113,-112,-112,-112,-112
63	C:000176	85868788898989	dc.b	-111,-111,-111,-111,-111,-111,-110
64	C:00017E	86878889909090	dc.b	-110,-110,-110,-110,-109,-109,-109
65	C:000186	87888990919191	dc.b	-109,-109,-109,-109,-108,-108,-108
66	C:00018E	88899091929292	dc.b	-108,-108,-107,-107,-107,-107,-106

Copyright (c) 1983 Allied Corporation All Rights Reserved
Module -VII-

Local Memory Trig Table

67	C:000196	96969696979797	do.b	-106,-106,-106,-106,-106,-106,-106,-106,-106,-106
68	C:000197	97979797979797	do.b	-105,-105,-105,-105,-105,-105,-105,-105,-105,-105
69	C:000198	98989898989898	do.b	-104,-104,-104,-104,-104,-104,-104,-104,-104,-104
70	C:000199	99999999999999	do.b	-103,-103,-103,-103,-103,-103,-103,-103,-103,-103
71	C:00019A	9A9A9A9A9A9A9A	do.b	-102,-102,-102,-102,-102,-102,-102,-102,-102,-102
72	C:00019B	9B9B9B9B9B9B9B	do.b	-101,-101,-101,-101,-101,-101,-101,-101,-101,-101
73	C:00019C	9C9C9C9C9C9C9C	do.b	-100,-100,-100,-100,-100,-100,-100,-100,-100,-100
74	C:00019D	9D9D9D9D9D9D9D	do.b	-99,-99,-99,-99,-99,-99,-99,-99,-99,-99
75	C:00019E	9E9E9E9E9E9E9E	do.b	-98,-98,-98,-98,-98,-98,-98,-98,-98,-98
76	C:00019F	9F9F9F9F9F9F9F	do.b	-97,-97,-97,-97,-97,-97,-97,-97,-97,-97
77	C:000196	96969696969696	do.b	-96,-96,-96,-96,-96,-96,-96,-96,-96,-96
78	C:000197	97979797979797	do.b	-95,-95,-95,-95,-95,-95,-95,-95,-95,-95
79	C:000198	98989898989898	do.b	-94,-94,-94,-94,-94,-94,-94,-94,-94,-94
80	C:000199	99999999999999	do.b	-93,-93,-93,-93,-93,-93,-93,-93,-93,-93
81	C:00019A	9A9A9A9A9A9A9A	do.b	-92,-92,-92,-92,-92,-92,-92,-92,-92,-92
82	C:00019B	9B9B9B9B9B9B9B	do.b	-91,-91,-91,-91,-91,-91,-91,-91,-91,-91
83	C:00019C	9C9C9C9C9C9C9C	do.b	-90,-90,-90,-90,-90,-90,-90,-90,-90,-90
84	C:00019D	9D9D9D9D9D9D9D	do.b	-89,-89,-89,-89,-89,-89,-89,-89,-89,-89
85	C:00019E	9E9E9E9E9E9E9E	do.b	-88,-88,-88,-88,-88,-88,-88,-88,-88,-88
86	C:00019F	9F9F9F9F9F9F9F	do.b	-87,-87,-87,-87,-87,-87,-87,-87,-87,-87
87	C:000226	00000000000000	do.w	-86,-86,-86,-86,-86,-86,-86,-86,-86,-86
88	C:000226	00000000000000	do.w	-85,-85,-85,-85,-85,-85,-85,-85,-85,-85
89	C:000236	0012001400160019	do.w	-84,-84,-84,-84,-84,-84,-84,-84,-84,-84
90	C:000246	002400260029002B	do.w	-83,-83,-83,-83,-83,-83,-83,-83,-83,-83
91	C:000256	0036003900420045	do.w	-82,-82,-82,-82,-82,-82,-82,-82,-82,-82
92	C:000266	0048005100540057	do.w	-81,-81,-81,-81,-81,-81,-81,-81,-81,-81
93	C:000276	0060006300660069	do.w	-80,-80,-80,-80,-80,-80,-80,-80,-80,-80
94	C:000286	0072007500780081	do.w	-79,-79,-79,-79,-79,-79,-79,-79,-79,-79
95	C:000296	0084008700900093	do.w	-78,-78,-78,-78,-78,-78,-78,-78,-78,-78
96	C:0002A6	0096009901020105	do.w	-77,-77,-77,-77,-77,-77,-77,-77,-77,-77
97	C:0002B6	00A800AB00AD00B0	do.w	-76,-76,-76,-76,-76,-76,-76,-76,-76,-76
98	C:0002C6	00BA00BD00C000C4	do.w	-75,-75,-75,-75,-75,-75,-75,-75,-75,-75
99	C:0002D6	00CB00CE00D000D4	do.w	-74,-74,-74,-74,-74,-74,-74,-74,-74,-74
100	C:0002E6	00DD00E000E400E8	do.w	-73,-73,-73,-73,-73,-73,-73,-73,-73,-73
101	C:0002F6	00EF00F200F600FA	do.w	-72,-72,-72,-72,-72,-72,-72,-72,-72,-72
102	C:000306	00F000F400F800FC	do.w	-71,-71,-71,-71,-71,-71,-71,-71,-71,-71
103	C:000316	00F200F600FA00FE	do.w	-70,-70,-70,-70,-70,-70,-70,-70,-70,-70
104	C:000326	00F400F800FE0100	do.w	-69,-69,-69,-69,-69,-69,-69,-69,-69,-69
105	C:000336	00F600FA00FE0104	do.w	-68,-68,-68,-68,-68,-68,-68,-68,-68,-68
106	C:000346	00F800FE01000108	do.w	-67,-67,-67,-67,-67,-67,-67,-67,-67,-67
107	C:000356	00FA00FC0104010C	do.w	-66,-66,-66,-66,-66,-66,-66,-66,-66,-66
108	C:000366	00FC00FE01080110	do.w	-65,-65,-65,-65,-65,-65,-65,-65,-65,-65
109	C:000376	00FE00FA010C0114	do.w	-64,-64,-64,-64,-64,-64,-64,-64,-64,-64
110	C:000386	0100000000000000	do.w	-63,-63,-63,-63,-63,-63,-63,-63,-63,-63

tan.tbl	eqn #	* tangent of 0..90 degrees by 1/32 binary(15,8)
do.w	0.2,4.7,9.11,13,16	
do.w	18.20,22,25,27,29,31,34	
do.w	36.38,41,43,45,47,50,52	
do.w	54.57,59,61,64,66,69,71	
do.w	73.76,78,81,83,86,88,91	
do.w	93.96,98,101,103,106,109,111	
do.w	114,117,119,122,125,128,130,133	
do.w	136,139,142,145,148,151,154,157	
do.w	156,163,166,169,173,176,179,183	
do.w	186,189,193,196,200,204,207,211	
do.w	215,219,223,226,231,235,239,243	
do.w	247,252,256,261,265,270,275,279	
do.w	284,289,294,300,305,311,316,322	
do.w	328,334,340,346,352,359,366,372	
do.w	380,387,394,402,410,418,426,435	
do.w	443,452,462,471,481,492,502,513	
do.w	525,537,549,562,578,589,603,618	
do.w	634,650,667,685,703,723,743,765	
do.w	788,812,837,864,893,923,955,990	
do.w	1027,1066,1109,1155,1204,1258,1317,1381	
do.w	1452,1530,1616,1713,1822,1945,2085,2247	
do.w	2435,2659,2926,3253,3661,4186,4855,5653	
do.w	7331,9776,14666,29335,32767	

Copyright (c) 1983 Allied Corporation All Rights Reserved
Module -VII-

Table

Line	Table	Equ	* x sine of 0.90 degree by 1/2 Binary(15,18)
112	C:000390	0000011E023C035A	0.286,572,858,1144,1429,1710,2000
113	C:000390	00E0A0B0R280C45	2286,2871,2856,3141,3425,3709,3993,4277
114	C:0003A0	08E0A0B0R280C45	4560,4843,5126,5408,5690,5971,6252,6533
115	C:0003B0	11D012E014061520	6813,7092,7371,7650,7927,8204,8481,8757
116	C:0003C0	1A9D1B041C0B1DE2	9032,9307,9580,9854,10126,10397,10668,10938
117	C:0003D0	2340245B256C267E	11207,11476,11743,12010,12278,12540,12803,13066
118	C:0003E0	2D072CD42DDF2EFA	13328,13589,13848,14107,14365,14621,14876,15131
119	C:0003F0	341035153618371B	15384,15636,15886,16136,16384,16631,16877,17121
120	C:000400	3C183D143E0F3F08	17364,17606,17847,18086,18324,18560,18795,19028
121	C:000410	43D444C645B746A6	19261,19491,19720,19948,20174,20399,20622,20843
122	C:000420	4R3D4C234D084DEC	21063,21281,21498,21713,21926,22138,22348,22556
123	C:000430	5247532153FA54D1	22763,22967,23170,23372,23571,23769,23965,24169
124	C:000440	58FB59B75A28B4C	24351,24542,24730,24917,25102,25285,25466,25645
125	C:000450	5F1F5FDE09A6155	25822,25997,26170,26341,26510,26677,26842,27005
126	C:000460	64DF658D63A66E5	27166,27328,27482,27636,27789,27939,28088,28234
127	C:000470	6A1F6ABD6B5A6BF4	28378,28529,28660,28797,28932,29066,29197,29325
128	C:000480	6EDA6F686FF4707D	29452,29576,29698,29818,29935,30050,30163,30274
129	C:000490	730C73887402747A	30382,30488,30592,30693,30792,30888,30983,31075
130	C:0004A0	76A57718770077E5	31164,31251,31336,31419,31499,31576,31651,31724
131	C:0004B0	79D17A137A687AB8	31795,31863,31928,31991,32052,32110,32166,32219
132	C:0004C0	7C37C777CB77CF7	32270,32319,32365,32408,32449,32488,32524,32557
133	C:0004D0	7E0E7E37FE6D7E98	32588,32617,32643,32667,32688,32707,32723,32737
134	C:0004E0	7F4C7F697FB37F9B	32748,32787,32763,32767,32767
135	C:0004F0	7FEC7F697FB37F9F	
136	C:0004FA		
137	C:0004FA	0000011E023C035A	0.286,572,858,1143,1429,1714,1999
138	C:0004FA	08E0A0B0R280C45	2283,2567,2850,3133,3416,3698,3979,4259
139	C:00050A	11B012D113E614FB	4538,4817,5094,5371,5647,5921,6195,6467
140	C:00051A	1A521B601C6D1D7B	6738,7008,7277,7544,7810,8074,8336,8598
141	C:00052A	22993398249B259A	8807,9115,9371,9626,9878,10129,10378,10624
142	C:00053A	2A752B682C592D48	10869,11112,11353,11592,11828,12063,12295,12524
143	C:00054A	31D432B13390346C	12752,12977,13200,13420,13638,13853,14066,14276
144	C:00055A	3B9339603A2A3AF2	14483,14688,14890,15090,15286,15480,15671,15860
145	C:00056A	3EAD3F64401740C5	16045,16228,16407,16584,16757,16928,17095,17260
146	C:00057A	440D44AC454745DF	17421,17609,17738,17887,18036,18181,18324,18463
147	C:00058A	48A7492C49AD4A2B	18599,18732,18861,18987,19110,19230,19346,19459
148	C:00059A	4C704CDA4D41	19568,19674,19777
149	C:0005AA		
150	C:0005B0		
151	C:0005B0		
152	C:0005B0		
total-table: set *			
end			

Claims

1. A method of creating a data base on a storage medium in a computing device including the steps of:
 - (i) generating a series of signals representing nodes on a locus of a curve partially defined by a set of

related knots, encoded as data on an encodable medium, with said knots defining end points of respective curve segments of said curve locus and with said knots being in a successive order in relation to said locus, and

(ii) encoding said node signals as data stored on said storage medium for use when representing said curve segments in a separate additional curve generating process responsive to the shape of said curve locus, as represented by said encoded node signals;

said method of creating said data base further comprising the steps of:

(a) defining the locations and the successive order of said knots on said curve locus and encoding as data on said storage medium, signals indicative of said knots,

(b) for a first knot (Z_a) representing a first end point of a first curve segment (21), deriving a first angle (a) measured with respect to an arbitrary reference line and indicative of the average of two interknot angles (B_{a-1} and B_a), wherein (Z_{a-1}) is the knot immediately preceding the first knot (Z_a), the inter-knot angle (B_{a-1}) is the angle formed between the reference line and a straight line connecting the said two knots (Z_a and Z_{a-1}), and wherein (Z_{a+1}) is the knot immediately succeeding knot (Z_a) in said successive order and interknot angle B_a is the angle formed between the reference line and a straight line connecting said knots (Z_a and Z_{a+1}), and encoding, as data on said storage medium, signals indicative of said first angle (a),

(c) at a second of said knots (Z_{a+1}), representing a second end point of said first curve segment (21), establishing a second angle (B_{a+1}) for said first curve segment, and encoding as data in said computing device, signals indicative of said second angle,

(d) establishing a compiler in a memory of a computing device, which compiler generates locus points for compiling data as a function of:

- i. a predetermined cubic parametric polynomial function of a parameter "t" incremented in steps;
- ii. said knots; and

- iii. said angles at said end points of said first curve segments;

(e) establishing a limited range "R" of values for said parameter "t" in the compiler,

(f) in said memory of said computing device, applying said signals indicative of said locations of said first and second knots of said first curve segment, to said compiler,

(g) in said memory of said computing device, applying said signals indicative of said first and second angles of said first curve segment to said compiler,

(h) generating a signal indicative of a distinct selected value of said parameter "t" within said range "R" by incremental signal generation, and applying said signal to said compiler to derive a signal indicative of a respective node location on said first curve segment according to said predetermined cubic parametric polynomial function,

(i) repeating step (h) by applying signals indicative of additional distinct selected values of said parameter "t", within said range "R", to derive a plurality of signals indicative of respective node locations on said locus of said first curve segment for respective distinct selected values of said parameter "t", and

(j) encoding said signals derived in steps (h) and (i), in said data base in said computing device memory for later sorting to represent said first curve segment for later generation of a visual representation of said curve by using said data signals directly to control a display process to visually display the curve.

2. A method as in claim 1 further comprising a step k) wherein steps b) through j) are repeated for at least a second curve segment within the locus of said curve.

3. A method as in claim 1 wherein said compiler is in the Hermite form.

4. A method as in claim 1 wherein the reference line is a straight, horizontal line in a normalized coordinate system formed on a master grid and within which said angles are measured.

5. A method as in claim 4 wherein said compiler of step d) is in the Hermite form.

6. The method as in claim 1 wherein said step (a) further comprises defining said knot locations with respect to an encoding grid having first and second coordinates and said range of values "R" is defined independently of said first and second coordinates.

7. A method as in claim 1 wherein said incremental signal generation in step (h) further comprises generating a signal corresponding to an incremental value of said parameter "t" and generating a multiple of said incremental value as said signal indicative of said distinct selected value.

8. The method as in claim 7, wherein said step (a) of defining said knot locations further comprises defining said knot locations with respect to an encoding grid having first and second coordinates and said step of generating a signal corresponding to an incremental value of parameter "t" further comprises generating said incremental value signal corresponding to a multiple of the reciprocal of the

distance between said first and second knots.

9. A method as in claim 1, wherein in step (a) said successive order of knots defines an outline loop and said encoding signals further comprises encoding said knot signals in a data order indicative of said knot order and said data order defines a data loop indicative of said outline loop.

10. The method as in claim 9, wherein step (j) further comprises encoding said node signals in said data loop, in an order indicative of the order of said nodes along the curve segments generated between knots in the knot order whereby the encoded node signals for a respective curve segment are positioned in the data loop between the encoded knot signals corresponding to the endpoints of that respective curve segment.

11. A method as in claim 1, wherein said second angle at said second knot (Z_{a+1}) is ϕ and said first angle at said first knot (Z_a) is θ and wherein said compiler generates node location signals ($Z(t)$) between knots (Z_a) and (Z_{a+1}) according to the equation:

$$Z(t) = Z_a + (3t^2 - 2t^3)(Z_{a+1} - Z_a) + rt(1-t)^2S_1 - st^2(1-t)S_2$$

$$S_1 = e^{j\theta} (Z_{a+1} - Z_a)$$

$$S_2 = e^{-j\phi} (Z_{a+1} - Z_a)$$

$$0 \leq t < 1$$

where r and s are positive real numbers.

12. The method as in claim 11, wherein said angle θ is an entrance angle for said first curve segment, at said first knot (Z_a), and said angle ϕ is an exit angle for said first curve segment at said second knot (Z_{a+1}).

13. A method as in claim 1, wherein said step (a) further comprises encoding the coordinate positions of said knots with regard to a normalized encoding grid thereby defining the outline loop at a normalized size, deriving a conversion factor relating the normalized encoding grid to a display coordinate grid, and applying said conversion factor to the coordinate positions for each knot thereby defining the outline loop with respect to the display coordinate grid.

14. A method as in claim 11 wherein:

$$r = \frac{2 \cdot \sin \phi}{(1 + |\cos \Psi|) \sin \Psi}$$

$$s = \frac{2 \cdot \sin \theta}{(1 + |\cos \Psi|) \sin \Psi}$$

$$\Psi = (\theta + \phi) / 2$$

15. A method as in claim 14, wherein said compiler is in the Hermite form and said display process of step (j) comprises the operation of a raster scan photo typesetter to generate font characters.

16. A method as in claim 1 or 13 further comprising accessing said signals encoded in said data base and using a microprocessor to control imaging means responsive to said accessed signals to reproduce said first curve segment.

17. A method as in claim 2 further comprising accessing said signals encoded in said data base and using a microprocessor to control imaging means responsive to said accessed signals to reproduce said first curve segment and at least a second curve segment.

19. A method as in claim 13 or 18 further comprising accessing said signals encoded in said data base and using a microprocessor to control imaging means responsive to said accessed signals to reproduce said curve.

20. The method as in claim 16 or 19, wherein said imaging means is a raster display and said accessing of said data base signals is conducted in a time relation to generation of raster lines by said imaging means to reproduce said first curve segment at a selected location on an imaging surface.

21. A method as in claim 15, further comprising the step of applying said display knot coordinates to said compiler and wherein the signals derived from said compiler, in steps (h) and (i), are indicative of node locations on said curve segment with respect to said display coordinate grid.

22. A system for generating images comprising a data base having data stored thereon, said data having been encoded into said data base in accordance with the method of claim 1, processing means for

processing said data, connected to imaging means for generating a curve image corresponding to said data, said imaging means being controlled by said processing means.

Revendications

5

1. Procédé pour créer une base de données sur un support d'enregistrement dans un dispositif de calcul, comprenant les étapes consistant à :

10

(i) produire une série de signaux représentant des noeuds sur un lieu d'une courbe partiellement définie par un ensemble de noeuds associés, codés en tant que données sur un support codable, lesdits noeuds définissant des points d'extrémité de segments respectifs dudit lieu de la courbe, lesdits noeuds étant dans un ordre séquentiel en rapport avec ledit lieu, et

15

(ii) coder lesdits signaux des noeuds en tant que données mémorisées sur ledit support d'enregistrement pour leur utilisation lors de la représentation desdits segments de courbe dans un processus additionnel séparé de production de la courbe, en réponse à la forme dudit lieu de la courbe, tel que représentée par lesdits signaux codés des noeuds;

ledit procédé de création de ladite base de données comprenant en outre les étapes consistant à :

20

(a) définir les emplacements et l'ordre séquentiel desdits noeuds sur ledit lieu de la courbe et coder, en tant que données sur ledit support d'enregistrement, des signaux indicatifs desdits noeuds,

20

(b) pour un premier noeud (Z_a), représenter un premier point d'extrémité d'un premier segment de courbe (21) devant être produit, obtenir un premier angle (α) mesuré par rapport à une ligne de référence quelconque et indicatif d'une moyenne de deux angles entre noeuds (B_{a-1} et B_a), (Z_{a-1}) étant le noeud précédant immédiatement le noeud (Z_a) dans ledit ordre séquentiel et l'angle entre noeuds (B_{a-1}) étant l'angle formé entre la ligne de référence et une ligne droite raccordant lesdits deux noeuds (Z_a et Z_{a-1}), alors que (Z_{a+1}) est le noeud succédant directement au noeud (Z_a) dans ledit ordre séquentiel et l'angle entre noeuds (B_a) est l'angle formé entre la ligne de référence et une

25

ligne droite raccordant lesdits noeuds (Z_a et Z_{a+1}), et coder, en tant que données dans ledit support d'enregistrement, des signaux indicatifs dudit premier angle (α),

30

(c) en un second desdits noeuds (Z_{a+1}), représenter un second point d'extrémité dudit premier segment de courbe (21), établir un second angle (B_{a+1}) pour ledit segment de courbe et coder, sous la forme de données, dans ladite mémoire, lesdits signaux du dispositif de calcul indicatifs dudit second angle,

(d) établir, dans une mémoire, un compilateur qui produit des points de lieu pour compiler des données en fonction de :

35

- i. une fonction polynomiale cubique paramétrique prédéterminée d'un paramètre "t", qui est incrémenté selon des échelons;
- ii. lesdits noeuds; et
- iii. lesdits angles au niveau desdits points d'extrémité desdits premiers segments de courbe;

(e) établir une gamme limitée (R) de valeurs pour le paramètre "t" dans le compilateur,

40

(f) dans ladite mémoire dudit dispositif de calcul, appliquer lesdits signaux indicatifs desdits emplacements desdits premier et second noeuds dudit premier segment de courbe audit compilateur,

(g) dans ladite mémoire dudit dispositif de calcul, appliquer lesdits signaux indicatifs desdits premier et second angles dudit premier segment de courbe audit compilateur,

45

(h) produire un signal indicatif d'une valeur sélectionnée distincte dudit paramètre "t" dans ladite gamme "R" au moyen d'une production incrémentale de signaux, et appliquer ledit signal audit compilateur pour obtenir un signal indicatif d'un emplacement respectif d'un noeud sur ledit premier segment de courbe conformément à ladite fonction polynomiale cubique paramétrique prédéterminée,

50

(i) répéter l'étape (h) en appliquant des signaux indicatifs de valeurs sélectionnées additionnelles distinctes dudit paramètre "t" dans ladite gamme "R" pour obtenir une pluralité de signaux indicatifs des emplacements respectifs de noeuds sur ledit lieu dudit premier segment de courbe pour des valeurs respectives du paramètre "t", et

55

(j) coder lesdits signaux obtenus lors des étapes (k) et (i), dans ladite base de données dans ladite mémoire du dispositif de calcul pour un échantillonnage ultérieur en vue de représenter ledit premier segment de courbe pour produire ultérieurement une représentation visuelle de ladite courbe moyennant l'utilisation desdits signaux codés de données pour commander un processus d'affichage afin d'afficher optiquement la courbe.

2. Procédé selon la revendication 1 comprenant en outre une étape k), lors de laquelle les étapes b) à j) sont répétées pour au moins un second segment de courbe sur le lieu de ladite courbe.
3. Procédé selon la revendication 1, selon lequel ledit compilateur se présente sous la forme de Hermite.
4. Procédé selon la revendication 1, selon lequel la ligne de référence est une ligne droite horizontale dans un système de coordonnées normalisées formé sur une grille maître et par rapport à laquelle lesdits angles sont mesurés.
5. Procédé selon la revendication 4, selon lequel ledit compilateur de l'étape d) possède la forme de Hermite.
6. Procédé selon la revendication 1, selon lequel ladite étape (a) comprend en outre la définition desdits emplacements de noeuds par rapport à une grille de codage possédant des première et seconde coordonnées, et ladite gamme de valeur "R" est définie indépendamment desdites première et seconde coordonnées.
7. Procédé selon la revendication 1, selon lequel ladite production incrémentale des signaux lors de l'étape (k) comprend en outre la production d'un signal correspondant à une valeur incrémentale dudit paramètre "t" et la production d'un multiple de ladite valeur incrémentale pour constituer ledit signal indicatif de ladite valeur distincte sélectionnée.
8. Procédé selon la revendication 6, selon lequel ladite étape (a) consistant à définir lesdits emplacements de noeuds comprend en outre la définition desdits emplacements de noeuds par rapport à une grille de codage possédant des première et seconde coordonnées, et ladite étape consistant à produire un signal correspondant à une valeur incrémentale du paramètre "t" comprend en outre la production dudit signal de valeur incrémentale correspondant à un multiple de l'inverse de la distance entre lesdits premier et second noeuds.
9. Procédé selon la revendication 1, selon lequel lors de l'étape (a), ledit ordre séquentiel des noeuds définit une boucle de contour et lesdits signaux de codage comprennent en outre un codage desdits signaux de noeuds dans un ordre des données indicatif dudit ordre des noeuds, et ledit ordre des données définit une boucle de données indicative de ladite boucle de contour.
10. Procédé selon la revendication 8, selon lequel l'étape (j) comprend le codage desdits signaux de noeuds dans ladite boucle de données, dans un ordre indicatif de l'ordre desdits noeuds le long du segment de courbe produit entre les noeuds dans l'ordre des noeuds, ce qui permet de positionner les signaux codés de noeuds pour un segment de courbe respectif dans la boucle des données entre les signaux codés de noeuds correspondant aux points d'extrémité de ce segment de courbe respectif.
11. Procédé selon la revendication 1, selon lequel ledit second angle au niveau dudit second noeud (Z_{a+1}) est ϕ et ledit premier angle au niveau dudit premier noeud (Z_a) est θ , et selon lequel ledit compilateur produit des signaux d'emplacement de noeuds ($Z(t)$) entre des noeuds (Z_a) et (Z_{a+1}) conformément aux équations :

$$Z(t) = Z_a + (3t^2 - 2t^3)(Z_{a+1} - Z_a) + rt(1-t)^2S_1 - st^2(1-t)S_2$$

$$S_1 = e^{j\theta} (Z_{a+1} - Z_a)$$

$$S_2 = e^{-j\phi} (Z_{a+1} - Z_a)$$

$$0 \leq t < 1$$

r et s étant des nombres réels positifs.
12. Procédé selon la revendication 10, selon lequel ledit angle θ est un angle d'entrée pour ledit premier segment de courbe, au niveau dudit premier noeud (Z_a), et ledit angle ϕ est un angle de sortie pour ledit premier segment de courbe au niveau dudit second noeud (Z_{a+1}).
13. Procédé selon la revendication 1, selon lequel ladite étape (c) comprend en outre le codage des positions de coordonnées desdits noeuds par rapport à une grille normalisée de codage, ce qui permet

de définir la boucle de contour à une taille normalisée, obtenir un facteur de conversion rapportant la grille normalisée de codage à une grille de coordonnées d'affichage, et appliquer ledit facteur de conversion aux positions de coordonnées pour chaque noeud, ce qui définit la boucle de contour par rapport à la grille de coordonnées d'affichage.

5

14. Procédé selon la revendication 11, selon lequel :

10

$$\begin{aligned} r &= \left| \frac{2 \cdot \sin \phi}{(1 + |\cos \Psi|) \sin \Psi} \right| \\ s &= \left| \frac{2 \cdot \sin \theta}{(1 + |\cos \Psi|) \sin \Psi} \right| \\ \Psi &= (\theta + \phi) / 2 \end{aligned}$$

15

15. Procédé selon la revendication 14, selon lequel ledit compilateur possède la forme de Hermite et ledit processus d'affichage de l'étape (j) inclut l'opération d'une photocomposeuse à balayage de trame pour produire des caractères d'une police.

20

16. Procédé selon la revendication 1 ou 13, comprenant en outre l'accès auxdits signaux codés dans ladite base de données et l'utilisation d'un microprocesseur pour commander des moyens de formation d'images, répondant auxdits signaux, auxquels l'accès est réalisé, pour reproduire ledit premier segment de courbe.

25

17. Procédé selon la revendication 2, comprenant en outre l'accès auxdits signaux codés dans ladite base de données, l'utilisation d'un microprocesseur pour commander des moyens de formation d'images, répondant auxdits signaux, auquel l'accès a été réalisé, pour reproduire ledit premier segment de courbe et au moins un second segment de courbe.

30

18. Procédé selon la revendication 3, selon lequel lesdites étapes b) à j) sont répétées un nombre suffisant de fois pour l'ensemble du lieu de la courbe à reproduire.

35

19. Procédé selon la revendication 13 ou 18, comprenant en outre l'accès auxdits signaux codés dans ladite base de données et l'utilisation d'un microprocesseur pour commander les moyens de formation d'images, répondant auxdits signaux, auxquels l'accès a été réalisé, pour reproduire ladite courbe.

40

20. Procédé selon la revendication 16 ou 19, selon lequel lesdits moyens de formation d'images sont un dispositif d'affichage de trames et ledit accès auxdits signaux de base de données est exécuté dans une relation temporelle par rapport à la production des lignes de trames par lesdits moyens de formation d'images pour reproduire ledit premier segment de courbe en un emplacement sélectionné sur une surface de formation d'images.

45

21. Procédé selon la revendication 15, comprenant en outre l'étape consistant à appliquer lesdites coordonnées de noeuds d'affichage audit compilateur, et selon lequel les signaux délivrés par ledit compilateur, lors des étapes (h) et (i), sont indicatifs d'emplacements de noeuds sur ledit segment de courbe par rapport à ladite grille de coordonnées d'affichage.

50

22. Système pour produire des images comprenant une base de données dans laquelle sont mémorisées des données, lesdites données ayant été introduites de façon codée dans ladite base de données conformément au procédé selon la revendication 1, des moyens de traitement pour traiter lesdites données, raccordés auxdits moyens de formation d'images pour produire l'image d'une courbe correspondant auxdites données, lesdits moyens de formation d'images étant commandés par lesdits moyens de traitement.

55

Patentansprüche

1. Verfahren zur Erzeugung einer Datenbank auf einem Speichermedium in einem Rechner, mit den Schritten:

(i) Erzeugen einer Reihe von Signalen, die Knotenpunkte auf einem Ort der Kurve darstellen, die teilweise durch einen Satz zueinander in Beziehung stehender Knoten definiert ist, die als Daten auf einem kodierbaren Medium kodiert sind, wobei die Knoten Endpunkte entsprechender Kurvensegmente des Kurvenorts definieren und wobei die Knoten in einer fortlaufenden Reihenfolge bezüglich des Orts vorliegen, und

(ii) Kodieren der Knotenpunktsignale als in dem Speichermedium gespeicherte Daten zur Verwendung bei der Darstellung der Kurve in einem gesonderten Kurvendarstellungsverfahren entsprechend der Form des Kurvenorts, wie er durch die kodierten Knotenpunktsignale dargestellt ist,

wobei das Verfahren zur Erzeugung der Datenbank weiterhin die Schritte umfaßt:

(a) Bestimmen der Orte und der fortlaufenden Reihenfolge der Knoten auf dem Kurvenort und Kodieren von Signalen, welche die Knoten angeben, als Daten auf dem Speichermedium,

(b) für einen ersten Knoten (Z_a), der einen ersten Endpunkt eines ersten Kurvensegments (21) darstellt, Ableiten eines ersten Winkels (α), der bezüglich einer willkürlichen Bezugslinie gemessen ist und einen Durchschnitt zweier Zwischenknoten-Winkel (B_{a-1} und B_a) angibt, wobei (Z_{a-1}) der Knoten ist, der dem ersten Knoten unmittelbar vorangeht, der Zwischenknoten-Winkel (B_{a-1}) der zwischen der Bezugslinie und einer geraden Linie gebildete Winkel ist, der die beiden Knoten (Z_a und Z_{a-1}) verbindet, und wobei (Z_{a+1}) der Knoten ist, der dem Knoten (Z_a) in der fortlaufenden Reihenfolge unmittelbar folgt, und der Zwischenknoten-Winkel B_a der Winkel ist, der zwischen der Bezugslinie und einer geraden Linie gebildet ist, welche die Knoten (Z_a und Z_{a+1}) verbindet, sowie Kodieren als Daten auf dem Speichermedium, wobei die Signale den ersten Winkel (α) angeben,

(c) bei einem zweiten Knoten (Z_{a+1}), der einen zweiten Endpunkt des ersten Kurvenabschnitts (21) darstellt, Ansetzen eines zweiten Winkels (B_{a+1}) für das erste Kurvensegment und Kodieren von Signalen, welche den zweiten Winkel (B_{a+1}) angeben, als Daten in dem Rechner,

(d) Bilden eines Kompilierers in einem Speicher eines Rechners, wobei der Kompilierer Ortspunkte zum Kompilieren von Daten erzeugt als Funktion von:

i. einer vorherbestimmten kubischen parametrischen Polynominalfunktion eines Parameters " t ", der in Schritten inkrementiert ist;

ii. den Knoten; und

iii. den Winkeln an den Endpunkten der ersten Kurvensegmente;

(e) Aufstellen eines begrenzten Wertebereichs " R " für Parameter " t " in dem Kompilierer,

(f) in dem Speicher des Rechners Einsetzen der Signale, welche die Positionen der ersten und zweiten Knoten des ersten Kurvensegments angeben, in den Kompilierer,

(g) in dem Speicher des Rechners Einsetzen der Signale, welche die ersten und zweiten Winkel des ersten Kurvensegments darstellen, in den Kompilierer,

(h) Generieren eines Signals, welches einen bestimmten ausgewählten Wert des Parameters " t " innerhalb des Bereichs " R " angibt, durch inkrementale Signalerzeugung, und Eingeben des Signals in den Kompilierer, um ein Signal abzuleiten, welches eine entsprechende Knotenposition auf dem ersten Kurvensegment entsprechend der vorbestimmten kubischen parametrischen Polynominalfunktion angibt,

(i) Wiederholen von Schritt (h) durch Eingeben von Signalen, welche zusätzliche bestimmte ausgewählte Werte des Parameters " t " innerhalb des Bereichs " R " angeben, um eine Vielzahl von Signalen abzuleiten, welche entsprechende Knotenorte auf dem Ort des ersten Kurvensegments für entsprechende bestimmte ausgewählte Werte des Parameters " t " angeben,

(j) Kodieren der in den Schritten (h) und (i) abgeleiteten Signale in der Datenbank in dem Rechner-Speicher zum späteren Sortieren, um das erste Kurvensegment zum anschließenden Generieren einer sichtbaren Darstellung der Kurve darzustellen, indem die Datensignale zur direkten Steuerung eines Darstellungsvorgangs zur sichtbaren Darstellung der Kurve verwendet werden.

2. Verfahren nach Anspruch 1,

dadurch gekennzeichnet,

daß weiterhin die Schritte (b) bis (j) für wenigstens ein zweites Kurvensegment innerhalb des Orts der Kurve wiederholt werden.

3. Verfahren nach Anspruch 1,

dadurch gekennzeichnet,

daß der Kompilierer in Hermite-Form vorliegt.

4. Verfahren nach Anspruch 1,
dadurch gekennzeichnet,

5 daß die Bezugslinie eine gerade, horizontale Linie in einem normalisierten Koordinatensystem ist, welches auf einer Gittervorlage (Schriftzeichenvorlage) ausgebildet ist und innerhalb dessen die Winkel gemessen werden.

5. Verfahren nach Anspruch 4,
dadurch gekennzeichnet,

10 daß der Kompilierer des Schrittes (d) in Hermite-Form vorliegt.

6. Verfahren nach Anspruch 1,
dadurch gekennzeichnet,

15 daß Schritt (a) weiterhin eine Bestimmung der Knotenstellen hinsichtlich eines Kodier-Gitternetzes mit ersten und zweiten Koordinaten umfaßt und daß der Wertebereich "R" unabhängig von den ersten und zweiten Koordinaten bestimmt wird.

7. Verfahren nach Anspruch 1,
dadurch gekennzeichnet,

20 daß das inkrementale Erzeugen der Signale nach Schritt (h) weiterhin das Erzeugen eines Signals entsprechend einem inkrementalen Wert des Parameters "t" und das Erzeugen einer Vielzahl des inkrementalen Werts als Signal, welches den bestimmten ausgewählten Wert angibt, umfaßt.

8. Verfahren nach Anspruch 7,
dadurch gekennzeichnet,

25 daß Schritt (a), mit dem die Orte der Knoten bestimmt werden, weiterhin das Bestimmen der Orte der Knoten bezüglich eines Kodiernetzes mit ersten und zweiten Koordinaten umfaßt, und daß der Schritt des Erzeugens eines Signals entsprechend einem inkrementalen Wert des Parameters "t" weiterhin das Erzeugen des Signals des inkrementalen Werts umfaßt, der einem Vielfachen des Kehrwerts der Entfernung zwischen den ersten und zweiten Knoten entspricht.

9. Verfahren nach Anspruch 1,
dadurch gekennzeichnet,

35 daß nach Schritt (a) die Abfolge der Knoten eine Umrißlinienschleife bestimmt und die Signalkodierung weiterhin eine Kodierung der Knotensignale in einer Folge der Daten einschließt, welche die Folge der Knoten angibt, und daß die Folge der Daten eine Datenschleife definiert, welche die Umrißlinienschleife angibt.

10. Verfahren nach Anspruch 9,
dadurch gekennzeichnet,

40 daß Schritt (j) weiterhin das Kodieren der Knotensignale in der Datenschleife in einer Folge umfaßt, welche die Folge der Knoten entlang den Kurvensegmenten angibt, welche zwischen Knoten in der Knotenfolge gebildet werden, wodurch die kodierten Knotensignale für ein entsprechendes Kurvensegment in der Datenschleife zwischen den kodierten Knotensignalen entsprechend den Endpunkten des entsprechenden Kurvensegments angeordnet werden.

11. Verfahren nach Anspruch 1,
dadurch gekennzeichnet,

50 daß für den zweiten Winkel bei dem zweiten Knoten (Z_{a+1}) gleich ϕ und den ersten Winkel bei dem ersten Knoten (Z_a) gleich θ der Kompilierer Signale der Knotenorte ($Z(t)$) zwischen Knoten (Z_a) und (Z_{a+1}) entsprechend den Gleichungen generiert:

$$Z(t) = Z_a + (3t^2 - 2t^3) (Z_{a+1} - Z_a) + rt(1-t)^2 S_1 - st^2(1-t) S_2$$

$$S_1 = e^{j\theta} (Z_{a+1} - Z_a)$$

$$S_2 = e^{-j\phi} (Z_{a+1} - Z_a)$$

$$0 \leq t \leq 1$$

worin r und s positive reelle Zahlen sind.

12. Verfahren nach Anspruch 11,
dadurch gekennzeichnet,

5 daß der Winkel θ ein Eintrittswinkel für das erste Kurvensegment bei dem ersten Knoten (Z_a) ist und der Winkel $\emptyset$ ein Austrittswinkel für das erste Kurvensegment bei dem zweiten Knoten (Z_{a+1}) ist.

13. Verfahren nach Anspruch 1,
dadurch gekennzeichnet,

10 daß Schritt (a) weiterhin ein Kodieren der Koordinatenpositionen der Knoten bezüglich eines normalisierten Kodier-Gitternetzes umfaßt, wodurch die Umrißlinienschleife mit einer normalisierten Größe bestimmt wird, Ableiten eines Umrechnungsfaktors bezüglich des normalisierten Kodier-Gitternetzes auf ein Darstellungskoodinatengitternetz umfaßt und Anwenden des Umrechnungsfaktors auf Koordinatenpositionen für jeden Knoten umfaßt, wodurch die Umrißschleife bezüglich dem Darstellungskoodinatengitternetz bestimmt wird.

14. Verfahren nach Anspruch 11,
gekennzeichnet durch

20

$$\begin{aligned} r &= \left| \frac{2 \cdot \sin \phi}{(1 + |\cos \psi|) \sin \psi} \right| \\ s &= \left| \frac{2 \cdot \sin \theta}{(1 + |\cos \psi|) \sin \psi} \right| \\ \psi &= (\theta + \phi) / 2 \end{aligned}$$

25

30

15. Verfahren nach Anspruch 14,
dadurch gekennzeichnet,

35 daß der Compiler in Hermite-Form vorliegt und daß der Darstellungsvorgang nach Schritt (j) den Betrieb eines Rasterabastphotosetzgeräts zum Erzeugen der Schriftzeichensatz-Schriftzeichen umfaßt.

16. Verfahren nach Anspruch 1 oder 13,
dadurch gekennzeichnet,

40 daß auf die in der Datenbank kodierten Signale zugegriffen wird und ein Mikroprozessor zum Steuern von Abbildungsmitteln verwendet wird, die auf die zugegriffenen Signale zum Reproduzieren des ersten Kurvensegments ansprechen.

17. Verfahren nach Anspruch 2,
dadurch gekennzeichnet,

45 daß weiterhin auf die in der Datenbank kodierten Signale zugegriffen wird und ein Mikroprozessor zum Steuern von Abbildungsmitteln verwendet wird, die auf die zugegriffenen Signale zum Reproduzieren des ersten Kurvensegments und wenigstens eines zweiten Kurvensegments ansprechen.

18. Verfahren nach Anspruch 2,
dadurch gekennzeichnet,

50 daß die Schritte (b) bis (j) genügend oft wiederholt werden, um den gesamten Kurvenort zu reproduzieren.

19. Verfahren nach Anspruch 13 oder 14,
dadurch gekennzeichnet,

55 daß auf die in der Datenbank kodierten Signale zugegriffen wird und ein Mikroprozessor zum Steuern von Abbildungsmitteln verwendet wird, der auf die zugriffenen Signale zum Reproduzieren der Kurve anspricht.

20. Verfahren nach Anspruch 16 oder 19,

dadurch gekennzeichnet,

daß die Abbildungsmittel eine Rasterdarstellungseinrichtung sind und der Zugriff der Datenbank-Signale in zeitlicher Beziehung zu der Erzeugung von Rasterlinien durch die Abbildungsmittel erfolgt, um das erste Kurvensegment an einer ausgesuchten Stelle auf einer Abbildungsoberfläche zu reproduzieren.

21. Verfahren nach Anspruch 15,

dadurch gekennzeichnet,

daß in einem weiteren Schritt die Darstellungs-Knotenkoordinaten in den Kompilierer eingegeben werden, wobei die von dem Kompilierer in den Schritten (h) und (i) abgeleiteten Signale Knotenstellen auf dem Kurvensegment bezüglich des Darstellungskoordinatengitternetzes angeben.

22. Einrichtung zum Erzeugen von Bildern mit einer Datenbank, welche in ihr gespeicherte Daten aufweist,

welche in der Datenbank entsprechend dem Verfahren nach Anspruch 1 kodiert sind, mit Datenverarbeitungsmitteln für die Daten, welche an die Abbildungsmittel zum Erzeugen eines Kurvenbilds entsprechend den Daten angeschlossen sind, wobei die Abbildungsmittel durch die Verarbeitungsmittel gesteuert werden.

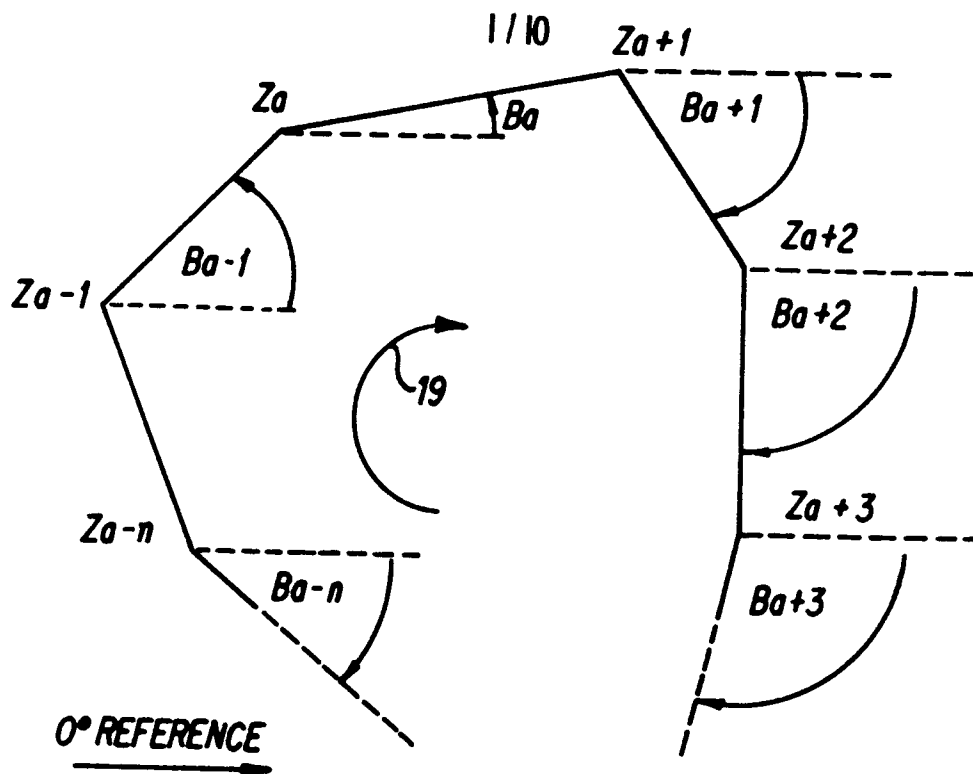


FIG. 1a

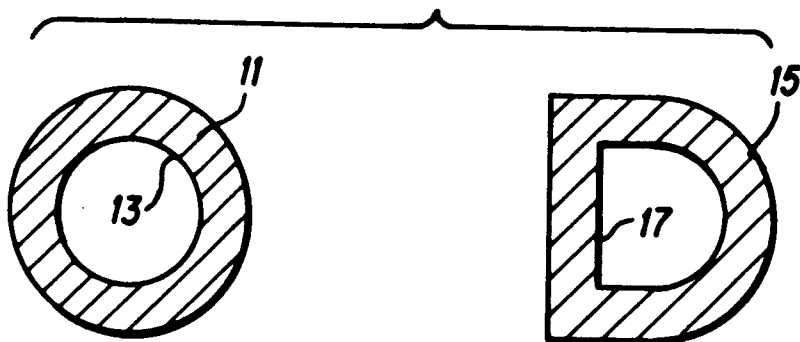


FIG. 1b

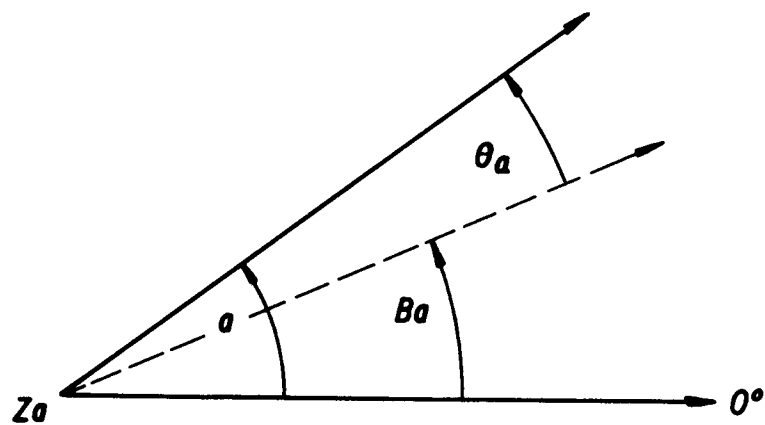


FIG. 1d

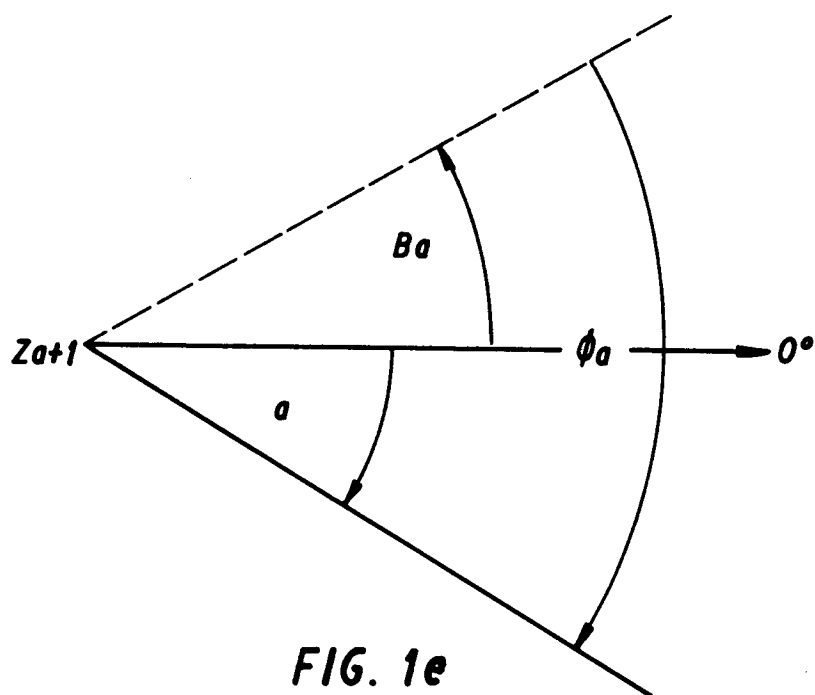


FIG. 1e

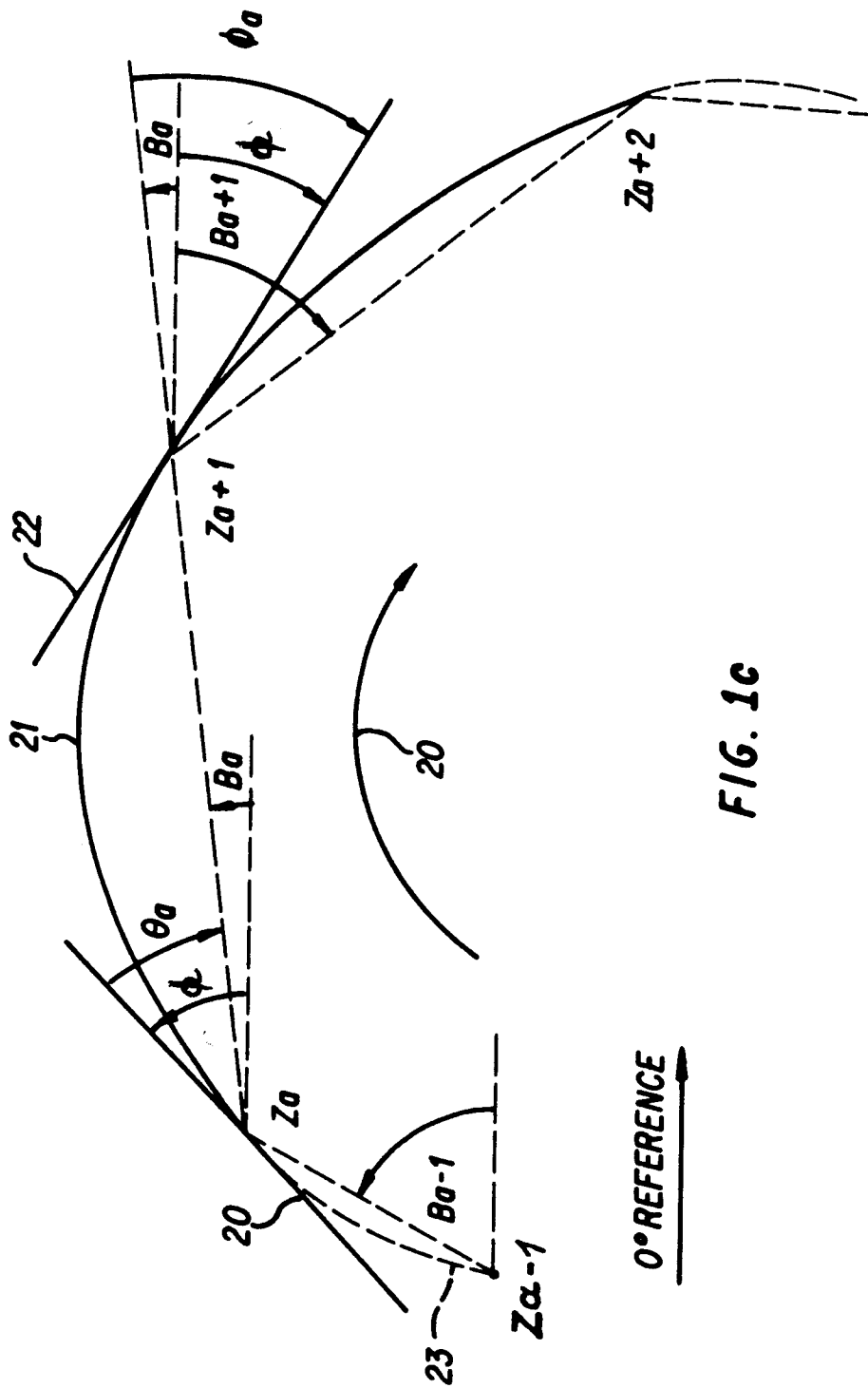


FIG. 1c

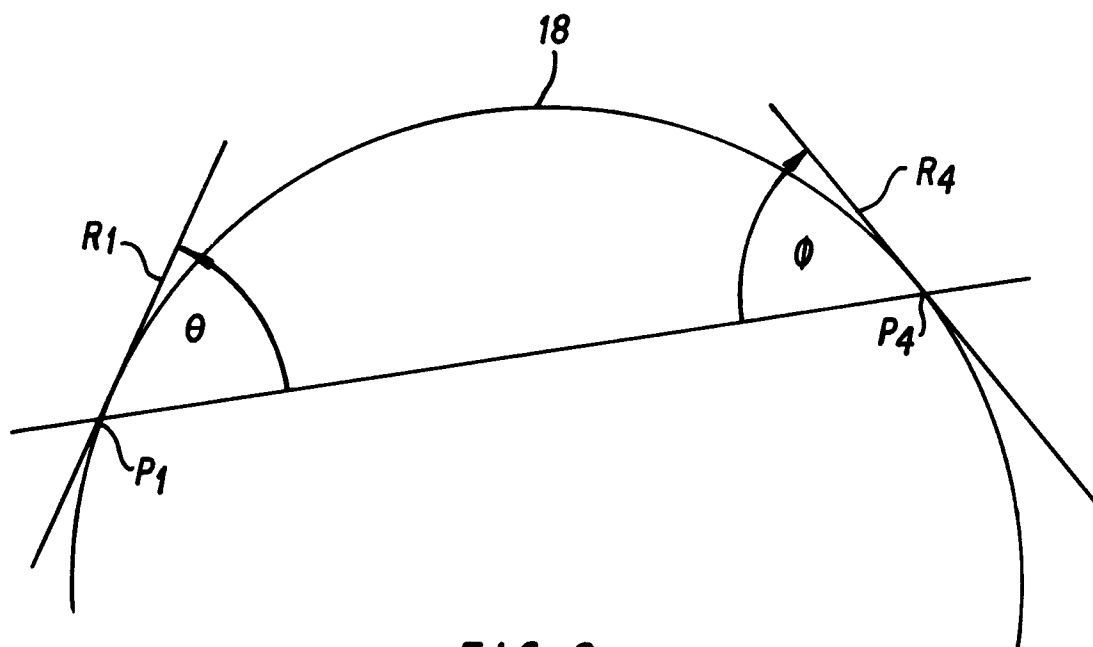


FIG. 2

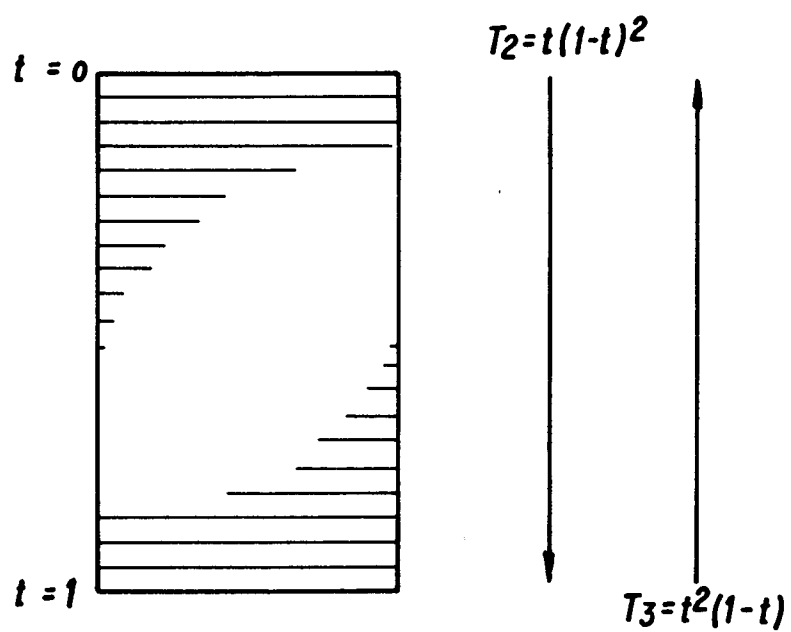


FIG. 4

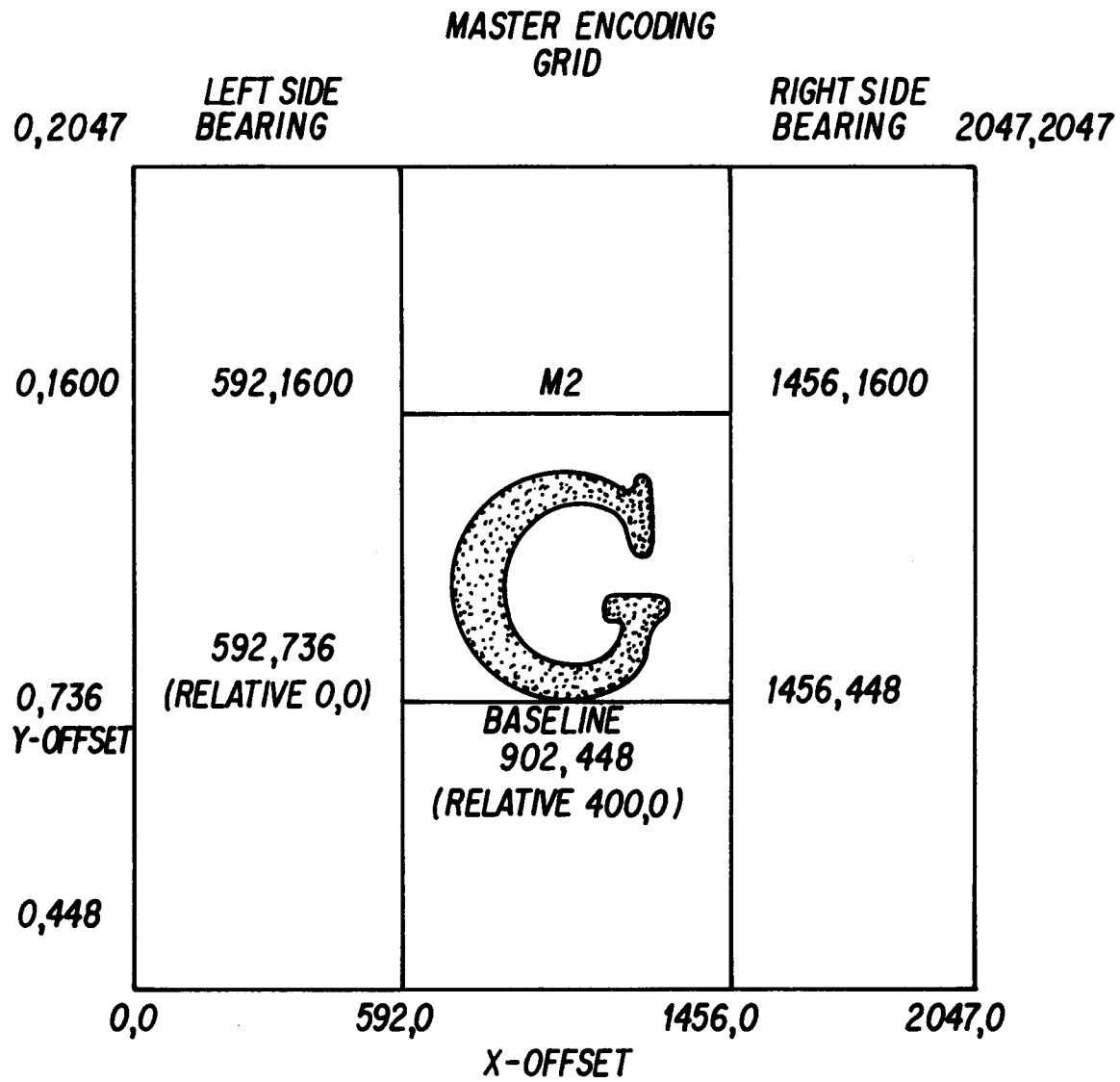
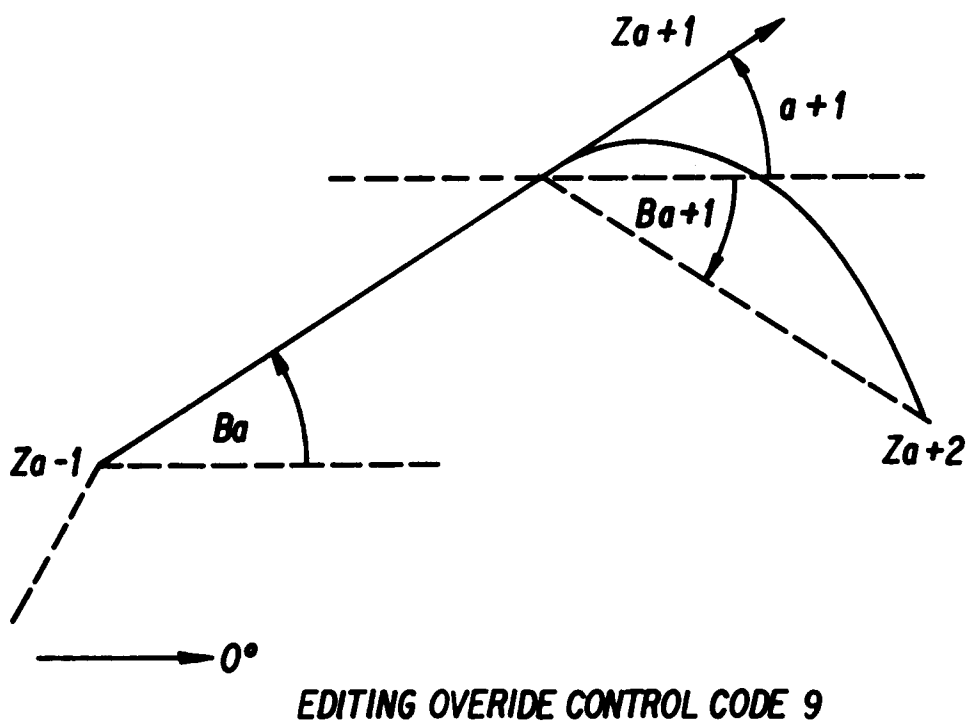
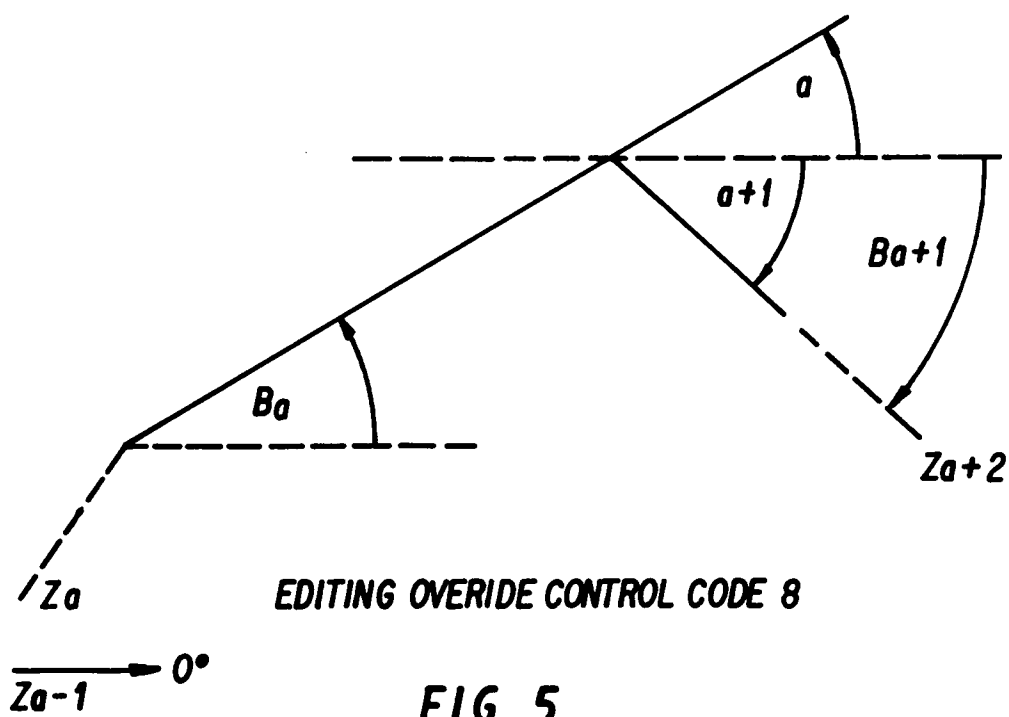


FIG. 3



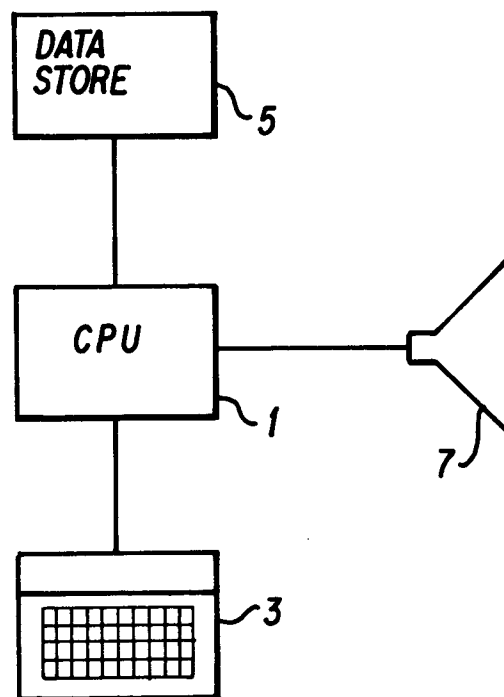
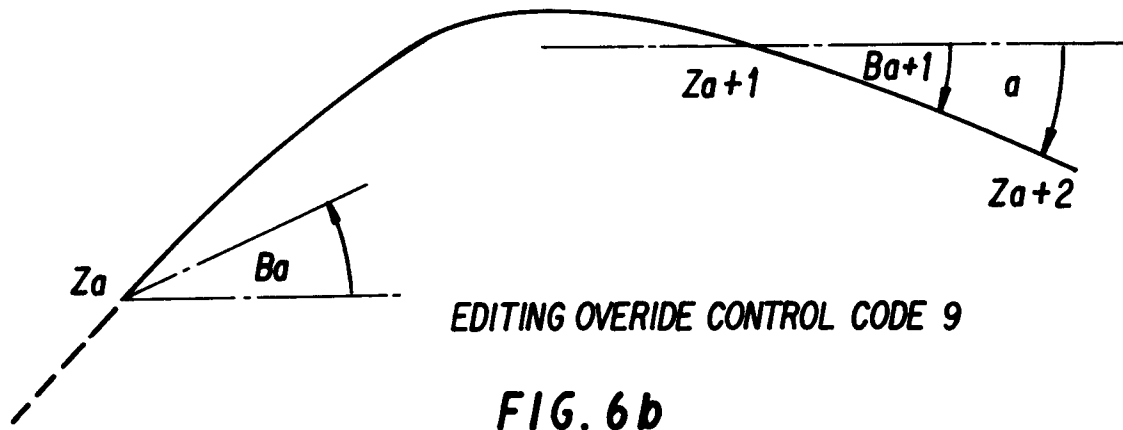


FIG. 10

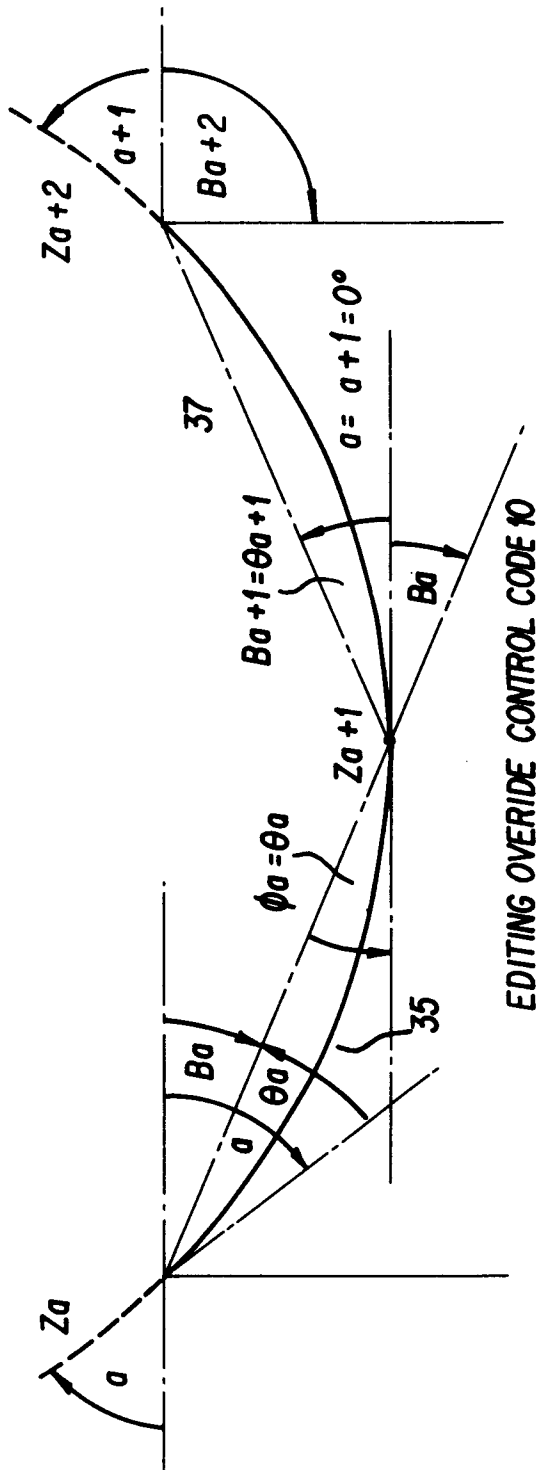


FIG. 7a

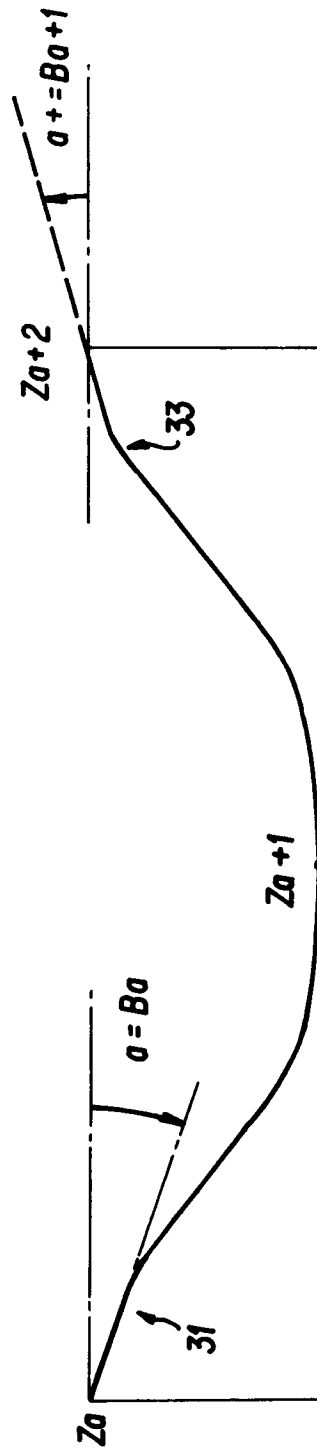


FIG. 7b

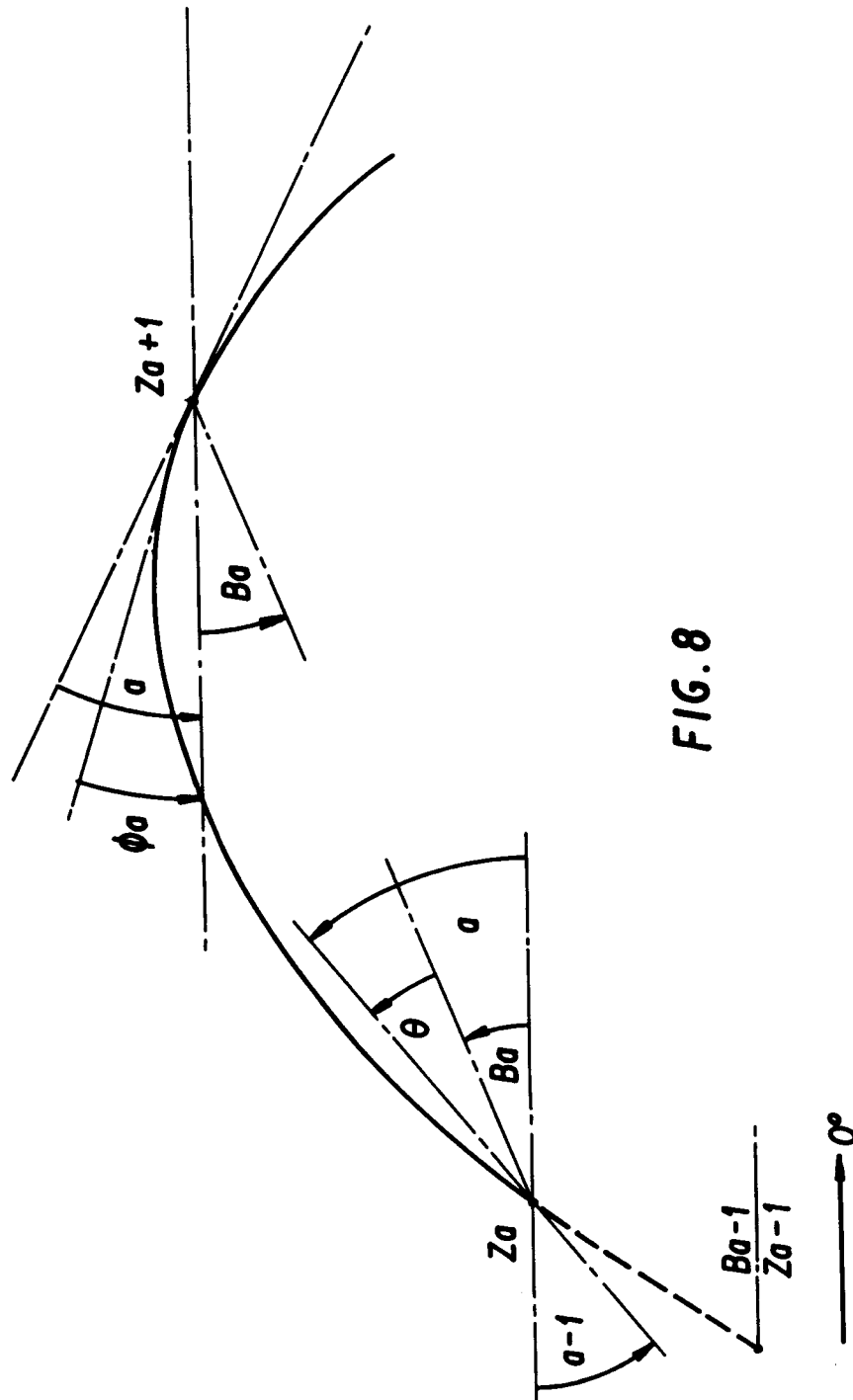


FIG. 8

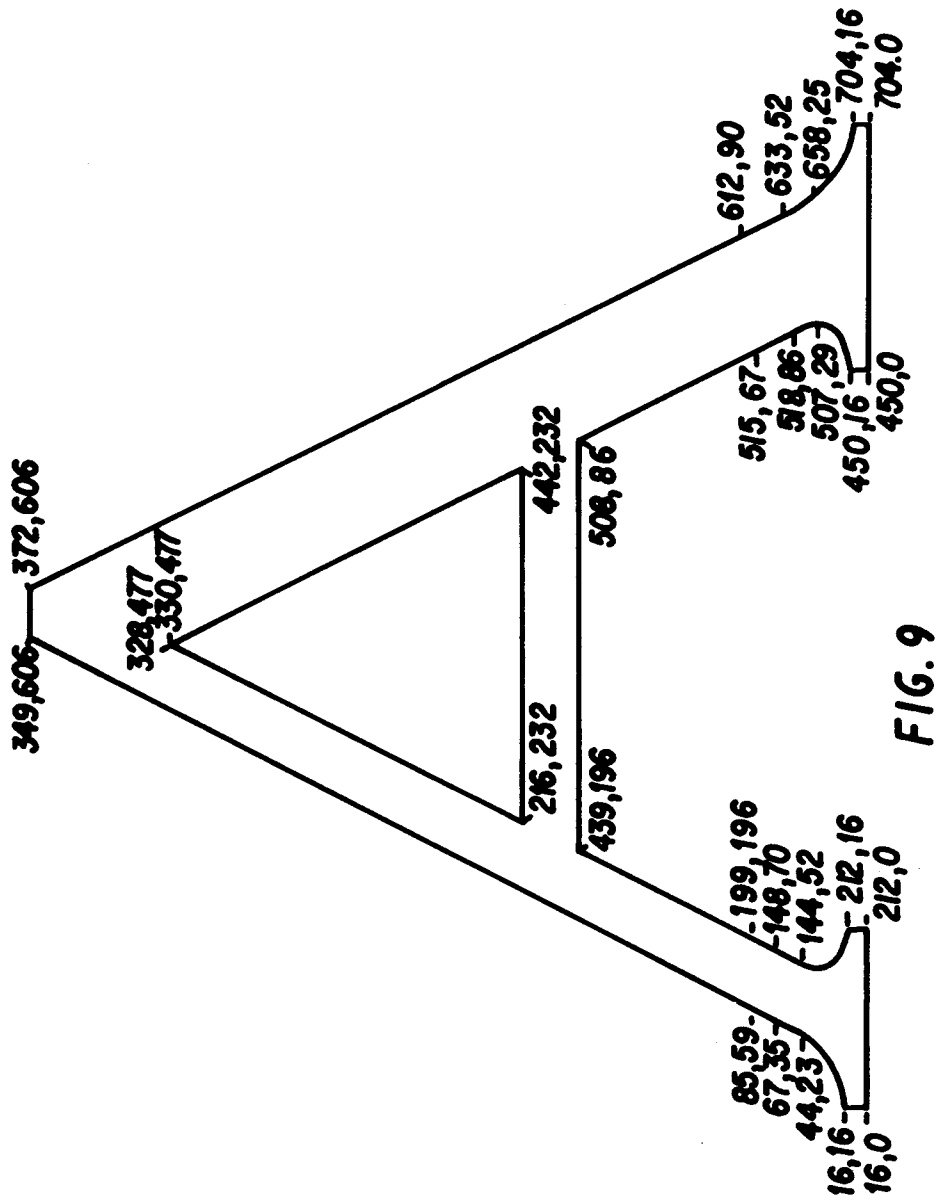


FIG. 9