



INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

<p>(51) International Patent Classification ⁵ : G09G 1/10, 1/14, G06F 15/72</p>	A1	<p>(11) International Publication Number: WO 90/05355 (43) International Publication Date: 17 May 1990 (17.05.90)</p>
<p>(21) International Application Number: PCT/US89/04911 (22) International Filing Date: 1 November 1989 (01.11.89) (30) Priority data: 266,048 1 November 1988 (01.11.88) US (71) Applicant: HONEYWELL INC. [US/US]; Honeywell Plaza, Minneapolis, MN 55408 (US). (72) Inventors: ANDERSON, Bruce, M. ; 2402 West Kathleen Road, Phoenix, AZ 85023 (US). BRODERICK, Brian, E. ; RR2, Box 105, New Palestine, IN 46163 (US). (74) Agent: BLINN, Clyde, C.; Honeywell Inc., MN12-8251, Honeywell Plaza, Minneapolis, MN 55408 (US).</p>		<p>(81) Designated States: AT (European patent), BE (European patent), CH (European patent), DE (European patent), FR (European patent), GB (European patent), IT (European patent), JP, LU (European patent), NL (European patent), SE (European patent). Published <i>With international search report. Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.</i></p>
<p>(54) Title: DIGITAL VECTOR GENERATOR APPARATUS FOR PROVIDING MATHEMATICALLY PRECISE VECTORS AND SYMMETRICAL PATTERNS</p> <p>(57) Abstract</p> <p>The vector generator employs a digital accumulator for each vector component. The accuracy of generated vectors is optimized by presetting the fractional portions of the accumulators to a value of either 1/2 or to 1/2 less 1 LSB. Accumulation of errors over concatenated vectors is eliminated by employing high enough fractional resolution relative to the maximum vector length so that error accumulation on any single vector is limited to the fractional portions of the accumulators, and by setting the fractional portions of the accumulators, on the last clock cycle of each vector, to a value of 1/2 or to 1/2 less 1 LSB. Also, by setting/presetting the fractions to only 1/2 less 1 LSB, and by enabling the carry input to a given accumulator on only the first clock cycle of vectors that have certain orientations, then symmetry is forced or favored in complex symbology that is generated by concatenated vectors.</p>		

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AT	Austria	FI	Finland	ML	Mali
AU	Australia	FR	France	MR	Mauritania
BB	Barbados	GA	Gabon	MW	Malawi
BE	Belgium	GB	United Kingdom	NL	Netherlands
BF	Burkina Fasso	HU	Hungary	NO	Norway
BG	Bulgaria	IT	Italy	RO	Romania
BJ	Benin	JP	Japan	SD	Sudan
BR	Brazil	KP	Democratic People's Republic of Korea	SE	Sweden
CF	Central African Republic	KR	Republic of Korea	SN	Senegal
CG	Congo	LJ	Liechtenstein	SU	Soviet Union
CH	Switzerland	LK	Sri Lanka	TD	Chad
CM	Cameroon	LJ	Liechtenstein	TG	Togo
DE	Germany, Federal Republic of	LJ	Liechtenstein	US	United States of America
DK	Denmark	LJ	Liechtenstein		
ES	Spain	MC	Monaco		
		MG	Madagascar		

DIGITAL VECTOR GENERATOR APPARATUS FOR PROVIDING
MATHEMATICALLY PRECISE VECTORS AND SYMMETRICAL PATTERNS

1

BACKGROUND OF THE INVENTION

1. Field of the Invention

The invention relates to digital vector generators, specifically as applied to electronic displays.

5

2. Description of the Prior Art

Digital vector generators are utilized in the art in various applications, including electronic displays. For example, they are used for: raster scanned or caligraphically generated cathode ray tube displays; X-Y plotters; numerically controlled machines; and robotics. Known vector generation procedures utilize either open-loop or closed-loop curve generation techniques. Open-loop techniques are often advantageous because various vector instruction formats may be incorporated which share the same basic hardware. For example, with open-loop techniques, both polar coordinate and rectangular coordinate vector formats can be effectively utilized. The execution of polar coordinate vectors can greatly simplify the rotation of symbology. Additionally, the open-loop algorithm permits achieving constant velocity vector generation, independent of vector angle. Constant velocity vector generators can provide for the generation of dotted and dashed vector patterns that do not change as a function of vector angle.

20

For the rectangular coordinate system, two-dimensional open-loop vector generation is generally implemented utilizing two accumulators for the X and Y coordinate axes, respectively, of the vector generation system. To generate a vector, the accumulators are loaded with the X and Y coordinates of the initial point of the vector. The vector is then generated by adding the incremental values of

25

-2-

1 DX/N and DY/N to the respective accumulators during each of N clock cycles,
where DX is the X component of the vector and DY is the Y component
thereof. Generally, each accumulator has an integer portion and a
fractional portion, where the integer portions provide the vector generator
5 outputs that select the picture elements (pixels) to be illuminated for
displaying the vector. This technique of vector generation is denoted as
open-loop integration. Although the subject invention will be described for
the application of two-dimensional vector generation, open-loop vector
generation is easily extended to three-dimensional space by the addition of
10 a third accumulator. It should be appreciated that the improvements of the
invention are also applicable to three-dimensional vector generation.

Open-loop integration generally utilizes either the tangent
algorithm or the sine/cosine algorithm. In the tangent algorithm, the
accumulator corresponding to the axis of the largest vector component is
15 advanced by one integer unit on every clock cycle and the other accumulator
is advanced by the tangent of the vector angle. In the sine/cosine
algorithm, the X and Y coordinates are advanced on each clock cycle by the
cosine and sine, respectively, of the vector angle. An example of a tangent
algorithm vector generator that may be configured in accordance with the
20 present invention is disclosed in U.S. Patent 4,115,863; issued September
19, 1978; entitled "Digital Stroke Display with Vector, Circle and Character
Generation Capability". An example of a sine/cosine vector generator that
may be configured in accordance with the present invention is disclosed in
United States Patent 4,481,605; issued November 6, 1984; entitled "Display
25 Vector Generator Utilizing Sine/Cosine Accumulation". Said patents
4,115,863 and 4,481,605 are incorporated herein by reference.

SUBSTITUTE SHEET

1 With vector generation utilizing open-loop integration, truncation
errors in the incremental values that are added to the X and Y accumulators
can result in an accumulation of errors as vectors are generated. Utilizing
open-loop integration results in a compromise to the accuracy of the
5 displayed symbology causing a degradation in display quality. Such
degradation in accuracy results in the illumination of pixels that are not
located on the mathematical locus of the desired vector, resulting in, for
example, line ends that do not meet and open curves that should be closed.
The truncation errors that result in these display anomalies are inherent in
10 digital vector generation apparatus utilizing open-loop integration.

Truncation errors can be eliminated by utilizing a closed-loop
vector algorithm which is inherently self-correcting and in which the
maximum error is inherently bounded. In closed-loop vector generation, the
error does not increase with vector length. The closed-loop algorithms are,
15 however, incompatible with vector instructions that utilize the polar
coordinate system to simplify the rotation of symbology. The inherent
advantage of polar vectors for the rotation of symbology can outweigh the
error problems due to truncation and coordinate transformation.
Additionally, as discussed above, constant velocity vector generation cannot
20 readily be achieved utilizing a closed-loop algorithm.

SUMMARY OF THE INVENTION

The present invention permits vector generation with open-loop
algorithms that is free from the anomalies that normally result from
truncation errors. This is accomplished by presetting the accumulator
25 fractional portions to a value of one-half, or to one-half less 1 LSB, prior
to the generation of each vector, and by limiting the maximum vector length
relative to the accumulator fractional resolution.

-4-

1 The presetting of the accumulator fractional portions provides
mathematically perfect vectors in the absence of truncation errors. When
truncation errors are present, the fractional presetting and the limiting of
the maximum vector length restricts the error at the vector end point to the
5 fractional bits of the accumulator. Presetting of the fractional
accumulator bits prior to the generation of each vector eliminates the
buildup of truncation errors over consecutive vectors. Also, a further
limitation of the maximum vector length ensures the generation of
10 mathematically perfect vectors utilizing the tangent algorithm.

10 The present invention also forces the generation of symmetrical
patterns when vectors are concatenated to generate complex symbology.
Symmetry is forced by, in effect, controlling the fractional accumulator
presetting to either one-half or to one-half less 1 LSB. This is achieved
by always presetting to one-half less 1 LSB and then conditionally enabling
15 the carry input to the accumulators, on the initial clock cycle of a vector,
as a function of the octant in which the vector angle resides.

BRIEF DESCRIPTION OF THE DRAWINGS

20 Figure 1 is a schematic block diagram of a vector processor
utilizing open-loop integration implemented in accordance with the present
invention.

 Figure 1A is a chart illustrating the formats of the various
instructions that can be executed by the hardware of Figure 1 of the present
invention.

25 Figure 2A is an illustration of the generation of a vector in
accordance with the prior art.

 Figure 2B is an illustration of the generation of a vector
utilizing the present invention.

SUBSTITUTE SHEET

-5-

1 Figure 3A is an illustration of the generation of the letter "v"
in accordance with the present invention with the accumulators initialized
to one-half.

5 Figure 3B is an illustration of the generation of the letter "v"
in accordance with the present invention with the accumulators initialized
to one-half minus 1 LSB.

 Figure 3C is an illustration of the generation of the letter "v"
in accordance with the invention utilizing a symmetrical algorithm.

10 Figure 4 is a diagram of the cartesian coordinate octant
definitions utilized in the symmetrical algorithms of the present invention.

 Figure 5A is an illustration of the generation, in accordance with
the invention, of an octagon utilizing concatenated vectors without using a
symmetrical algorithm of the present invention.

15 Figure 5B is an illustration of the generation, in accordance with
the invention, of an octagon utilizing concatenated vectors using a
symmetrical algorithm of the present invention.

 Figure 6 is an annotated octagonal chart useful in explaining the
operation of the symmetrical algorithms of the present invention.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

20 The present invention modifies the basic technique of open-loop
vector generation so as to eliminate the errors that normally result
therefrom. A two-dimensional vector in the rectangular coordinate system
can be generated with two digital accumulators, one for the X axis and one
for the Y axis. A vector is generated by adding an incremental value DX/N
25 to the X accumulator and an incremental value DY/N to the Y accumulator on
each of N clock cycles. Since the fractional portions of the accumulators
are of a finite size, the DX/N and DY/N values that are added thereto can
include truncation errors. Errors can therefore increase

SUBSTITUTE SHEET

-6-

1 in the X and Y accumulators over a number of clock cycles until the integer
values thereof, which are the vector generator outputs, are in error. Since
the probability of an integer error increases with the number of clock
cycles, errors will be prevalent when vectors are concatenated to generate
5 extensive complex symbology. The present invention, however, eliminates
error accumulation over the generation of multiple vectors by assuring that
no integer error can occur at the end point of any single vector, and by
resetting the fractional portion of each accumulator on the last clock cycle
of every vector in order to discard any error that has accumulated. The
10 present invention also generates mathematically perfect vectors using the
tangent algorithm, provided that the fractional resolution of the
accumulators is sufficiently high relative to the length of the generated
vector. A mathematically perfect vector has no integer error at any point
thereof. The invention also provides additional modifications to the open-
15 loop vector generator that can force the generation of symmetrical patterns
when vectors are concatenated to generate complex symbology.

In the most general approach to a hardware vector generator, using
the technique of open-loop integration, an accumulator device is used for
each vector axis. Thus, two accumulators are required for two-dimensional
20 vector generation, and three are required for three-dimensional vector
generation. Input data buffer registers, one for each accumulator, can be
used to hold the incremental values that are added to each respective
accumulator on every clock cycle during the generation of a given vector.
In order to control the number of clock cycles that are used in the
25 generation of a given vector, a counter circuit is also employed.

The hardware circuitry comprising: the set of accumulators; the
input data buffer registers; the length counter; and all required control
logic, together constitute a vector execution circuit. A low-cost vector

SUBSTITUTE SHEET

-7-

1 generator system might incorporate a general-purpose microcomputer along
with such a vector execution circuit. The microcomputer can execute output
instructions in order to transfer the required data parameters to the vector
execution circuit and to initiate each vector generation operation. This
5 approach to vector generation is, however, rather limited in its
performance. For high throughput vector generation, a special purpose
processor is often employed.

Figure 1 is a block diagram of a special purpose processor that is
designed to execute vector instructions in accordance with the invention.
10 It is designed for execution of the instruction types that are illustrated
by Figure 1A. The vector processor of Figure 1 is designed for two-
dimensional vectors. Accordingly, it incorporates two accumulators. The X
accumulator 30 and the Y accumulator 10 are identical in design. The X
accumulator incorporates a register 31 whose integer and fractional inputs
15 are derived, respectively, from separate integer 32 and fractional 33
multiplexers. Also included in the X accumulator is an adder 34 whose
inputs consist of the X register output 47 and an incremental value that is
derived from the output 35 of an X multiplexer 54 that is external to the X
accumulator. The range of values for the incremental input is generally of
20 less magnitude than the range of values of the X register output. However,
the value 35 from the X multiplexer 54 can be sign extended so that the
incremental input to the adder has the same number of bits as does the X
register.

Although the X register output 47 incorporates both integer and
25 fractional bits, only the integer portion thereof is used as the accumulator
output 48. Similarly, for the Y accumulator 10, only the integer portion 28
of the Y register output 27 is used as the accumulator output. For the
application of a display symbol generator for a raster scanned display, the

SUBSTITUTE SHEET

-8-

1 X 48 and Y 28 outputs are normally used to address a two-dimensional display
image memory for the purpose of writing vectors into said memory. For a
caligraphic CRT display, the X and Y outputs are each converted to an analog
signal. These analog signals then control the position of the electron beam
5 of the CRT for directly "writing" vectors onto the CRT display screen.

All the registers and counters of Figure 1 can be loaded or
advanced, as required, on the same edge of a single clock signal, thereby
allowing the use of a single phase clock. During an instruction fetch
cycle, the address of the instruction to be executed is held in the program
10 counter 61. The output of the program counter 87 is gated through the
multiplexer 63 to the address input 82 of the program memory 64. At the end
of the instruction fetch cycle, the transition of the single phase clock
signal will increment the program counter so that during the next clock
cycle the program counter will hold the program memory address of the next
15 sequentially higher location in the memory. Said memory location may
contain the next instruction to be executed or may contain the second word
of the instruction that was just fetched.

Note that each of the instruction formats of Figure 1A
incorporates an operation code field that is always located in the more
20 significant bits of a single-word instruction 120 or in the more significant
bits of the first word of a two-word instruction 101. At the end of an
instruction fetch cycle, the instruction word (or the first word of a two-
word instruction) is available on the data output 81 of the program memory
64. The transition of the clock signal at the end of the instruction fetch
25 cycle loads the entire instruction word into the D register 59. Also, the
operation code bits of the instruction word 88 are loaded into the OP code
register 52; and the lower bits of the instruction word 89, that are used
for a length count in the TAN vector instruction 104 and in the POLAR VECTOR
instruction 113, are loaded into the L counter 53.

SUBSTITUTE SHEET

- 9 -

1 The output 95 of the OP code register 52 is used by
the control logic 51 during each instruction execution sequence
in order to derive the control signals that are required for
the proper execution of the instruction. The control signals
5 required of the two accumulators are shown in Figure 1.
however, in the interest of simplicity, the control signals to
the other blocks of Figure 1 are not depicted. The control
signals to the X accumulator consist of: A carry input 36 to
10 the least significant bit of the adder 34; A select input 37 to
the fractional multiplexer 33 to control whether the fractional
portion 42 of the adder output 40 or the fixed binary value 43
of .0111...1 is gated to the multiplexer output 45; A select
input 38 the integer multiplexer 32 to control whether the
15 integer portion 41 of the adder output 40 or the accumulator
input 35 is gated to the multiplexer output 44; and a gated
clock signal 39 for conditionally loading the register 31 of
the X accumulator at the end of a given clock cycle. An
alternative to the use of a gated clock signal 39 would be to
20 employ a type of register that makes use of the primary clock
signal, which is active on every cycle, along with an enable
signal that determines the cycles on which the register is to
be loaded.

The reference numerals 11-28 denote components in
25 the Y accumulator 10 identical to those described with respect
to the X accumulator 30.

The preferred embodiment of the invention uses the
binary two's complement number system. The accumulators of
Figure 1 incorporate features that are specifically designed to
30 accommodate the invention. One such feature is the conditional
carry input to the two's complement adder of each accumulator.
Another is the use of the constant value .0111...1 for the
initial value of the fractional portion of the accumulator.
Still another is the use of separate multiplexers for
35 independently selecting the fractional and integer inputs to be
loaded into the accumulator register.

SUBSTITUTE SHEET

-10-

1 On the first cycle of either a jump instruction or a subroutine
jump instruction, the lower bits of the D register output 84, that
correspond to the address field bits of the jump instruction 121 or of the
subroutine jump instruction 123, are gated through the multiplexer 62 to the
5 input 86 of the program counter 61 and loaded into the program counter at
the end of the first clock cycle of the respective instruction. The second
clock cycle of these instructions is an instruction fetch cycle. It loads
the next instruction that is to be executed, which is located at the program
memory address previously loaded into the program counter. Also, for the
10 subroutine jump instruction, the output of the program counter 87 is loaded
into the A register 60 at the end of the first clock cycle. This saves, in
register A, the address of the instruction that follows the subroutine jump
instruction. On the first cycle of a return instruction, the A register
output 85 is gated through the multiplexer 62 to the program counter input
15 86 and is loaded into the program counter at the clock transition occurring
at the end of said first cycle. The second cycle of a return instruction is
an instruction fetch cycle. The return instruction is therefore followed by
the execution of the instruction that immediately follows the most recently
executed subroutine jump instruction. It should be appreciated that if
20 multiple levels of subroutines are required (i.e., nested subroutines), then
the A register 60 can be replaced with a last-in-first-out register stack.

 The load control instruction requires only a single clock cycle
for execution. In addition to being an instruction fetch cycle, this single
clock cycle also loads the V register 65 with the lower bits of the output
25 84 from the D register. These bits correspond to the miscellaneous field
119 and the video field 118 of the load control instruction word. The
output of the V register 99 can be used to control the video intensity

SUBSTITUTE SHEET

-11-

1 and/or the color of displayed vectors. Some of the V register output bits (e.g., those corresponding to the miscellaneous bits 119) might also be used to enable/disable a video modulation for the generation of vectors having a selectable dotted or dashed pattern.

5 On the first clock cycle of a load bias instruction, the second word of the instruction, which contains the rotation angle 112, is loaded into the D register from the output 81 of the program memory. The program counter 61 is also incremented in order to address the next instruction. The second clock cycle is an instruction fetch cycle, and it also loads the
10 bias register 57 with the output 84 of the D register. The bias register is used with polar vector instructions to cause a rotation of the generated vectors.

The load XY instruction is used to load an initial X value 100 and an initial Y value 103 into the X and Y accumulators, respectively. On the
15 first clock cycle of the instruction execution, the output 84 of the D register is gated through the addend logic 58 to the input 92 of the X multiplexer 54. It is also gated through the X multiplexer 54 to the input 35 of the integer multiplexer 32, gated through the integer multiplexer to the integer portion 44 of the input 46 to the register 31 of the X
20 accumulator, and loaded into the integer portion of said register 31 at the end of the cycle. Also, the value 0.111...1 is gated through the fractional multiplexer 33 to the fractional portion 45 of the input 46 to the register 31 of the X accumulator, and it is loaded into the fractional portion of said register at the end of the cycle.

25 Also on the first clock cycle of the load XY instruction, the second word of the instruction, which includes the initial value of Y 103, is transferred from the program memory to the D register. The program counter is also incremented. The second clock cycle is an instruction fetch

SUBSTITUTE SHEET

1 cycle. Also on the second clock cycle, the D register is gated through the
addend logic 58, the Y multiplexer 50, and the Y integer multiplexer 12, and
is loaded into the integer portion of the register 11 of the Y accumulator.
The value 0.111...1 is also gated through the Y fractional multiplexer 13
5 and loaded into the fractional portion of the register 11 of the Y
accumulator.

On the first clock cycle of either the tan vector or the polar
vector instruction, the program counter 61 is incremented and the second
word of the respective instruction is transferred from the program memory to
10 the D register. Note that the length parameter for the tan vector
instruction 104, or for the polar vector instruction 113, which specifies
the number of accumulation cycles to be used in generating the given vector,
would have been loaded into the L counter on the clock cycle that was used
to fetch the respective instruction.

15 After the first clock cycle of the instruction execution sequence,
the incremental values that are to be added to the two accumulators are
generated. The correct incremental value to be added to the X accumulator
is gated through the X multiplexer 54 to the X accumulator input 35.
Similarly, the correct incremental value to be added to the Y accumulator is
20 gated through the Y multiplexer 50 to the Y accumulator input 15. Because
of the time required for these incremental values to be generated and to
become stable at the respective accumulator inputs, the first accumulation
clock cycle will occur at the third clock cycle of the vector instruction
execution. For a single step vector, which is determined by the value in
25 the L counter 53, this third clock cycle would also be the last clock cycle
of the instruction execution, and would therefore be an instruction fetch
cycle. For vectors with multiple steps, the accumulation clock cycles are
repeated, with the L counter decremented at each said clock cycle, until the
value in the L counter indicates the occurrence of the last accumulation
30 clock cycle, which is also an instruction fetch cycle.

-13-

1 For the tan vector instruction, the incremental addend value to at least one of the accumulators will be either +1 or -1.

The octant code 107 for the vector, which is contained in the D register 59 and which is an input 84 to the control logic 51, determines whether the X

5 multiplexer 54 or the Y multiplexer 50, shall gate the fixed value of either +1 or -1 to its respective output. The octant code also determines which of these two values is to be gated. The other of these two multiplexers will gate the output 92 from the addend logic 58. The slope value 108 consists of only the fractional portion of the two's complement incremental value

10 that is the correct input to the appropriate accumulator. The integer bits for this input are appended to the fractional bits by the addend logic 58, and they are a function of the octant code. For most cases, the integer bits will be all ones or all zeroes, according to the sign of the incremental value as is determined by the octant code. However, the value

15 of +1 or -1 is required for vectors that are oriented at an angle of exactly 45° from either the X or the Y axis. Note that the octants are defined so that a given octant includes only one of the three values: 0; +1; or -1, as possible slope values. For any of these values, and for only these values, all of the bits of the fractional slope 108 are equal to zero. Therefore,

20 the correct integer bits can be appended to the fractional slope value by the addend logic as a function of the octant code and a function of whether or not the fractional slope bits are all equal to zero.

The polar vector instruction is designed to allow the simple rotation of symbology by automatically rotating all polar vectors by a

25 rotation angle 112 that is held in the bias register 57 as a result of the execution of a load bias instruction. The angle 117 of a polar vector is loaded into the D register 59 on the first clock cycle of the instruction

SUBSTITUTE SHEET

-14-

1 execution, and it is then added to the output 90 of the bias register 57 in
the adder 56. The output 91 of the adder 56, which is the correct angle for
the rotated vector, is the input to a SIN/COS generator 55. The SIN/COS
generator can be implemented, for example, with read-only-memories, using
5 the table-look-up technique. For the polar vector instruction, the COS
output 94 of the SIN/COS generator is used for the addend input to the X
accumulator and it is therefore gated through the X multiplexer 54. The SIN
output 93 is used for the addend to the Y accumulator, and it is gated
through the Y multiplexer 50. The first accumulation clock cycle for the
10 polar vector instruction occurs on the third clock cycle of the instruction
execution, as was the case for the Tan instruction. The last accumulation
clock cycle, as determined by the value of the L counter, is also an
instruction fetch cycle.

By loading the fractional value of .0111...1 into the fractional
15 portions of the accumulator registers 11, 31 on the load XY instruction, the
invention improves the accuracy of vectors generated with both the tan
vector instruction and the polar vector instruction. The accuracy of
vectors generated with the tan vector instruction are further improved by
loading the value of .011...1 into the fractional portions of the
20 accumulator registers on the last accumulation clock cycle of each vector,
which also loads the integer portion of the register with the integer
portion of the output from the accumulator adder 14, 34. Because the number
of fractional accumulator bits is made large enough relative to the maximum
number of accumulation clock cycles for any vector, the effect of the above
25 technique is to eliminate all errors at the end points of vectors that are
drawn with the tan vector instruction. Also, when the fractional resolution
is sufficient, errors at any point along a vector are eliminated. The

SUBSTITUTE SHEET

- 15 -

1 invention also forces, or favors, the generation of symmetrical
2 symbology using concatenated tan vector instructions, by
3 selectively enabling the carry inputs 16, 36 to the accumulator
4 adders 14, 34 on only the first accumulation clock cycle of
5 some vectors, whereby said enabling is, for each accumulator, a
6 function of the octant 107 of the vector.

7 The program memory 64 of the vector processor is
8 designed to be time-shared by the vector processor and by a
9 general purpose processor (not shown). The memory address 83
10 from general purpose processor can be gated through the
11 multiplexer 63 to the address input 82 of the program memory.
12 Data 80 from the general processor can be written to the
13 program memory so that the general processor can store a
14 program in the memory for subsequent execution by the vector
15 processor.

16 It should be appreciated that various techniques
17 could be used to improve the performance of the vector
18 processor shown in Figure 1. For example, two independent
19 program memories might be used so that memory access conflicts
20 between the vector processor, and the general processor could
21 be eliminated. Also, the vector execution hardware (i.e., the
22 accumulators, L counter and required control logic) might be
23 designed independently from the instruction fetch and format
24 hardware. With buffer registers, or first-in-first-out memory,
25 between the hardware of these two functions, the vector
26 execution operation could occur simultaneously with the
27 fetching and formatting of new instructions, thereby improving
28 over-all throughput. The instruction set of the vector
29 processor, Figure 1A, might also be expanded. For example, it
30 could include a SIN/COS type of vector instruction. General
31 purpose instruction types (e.g., Add, Subtract, Load, Store)
32 could also be included in the instruction set.

SUBSTITUTE SHEET

-16-

1 As discussed above, only the integer portions of the X and Y
accumulators are used as outputs. For the application of a raster CRT
display system, the X and Y outputs are utilized to address a two-
dimensional display-image memory. The vector generator stores a complete
5 image in the memory by writing the appropriate sequence of vectors. The
memory is then read out in synchronization with the display raster scan in
order to provide the video information to refresh the display. In real-time
applications, two image memories may be utilized so that the vector
generator can update the display image in one memory while the second memory
10 is utilized to provide the display video information. The functions of the
memories are interchanged when the vector generator completes an updated
version of the display image. For many other applications, such as for a
caligraphic CRT display or a numerically controlled machine, analog X and Y
signals are required. In these applications, the integer portions of the X
15 and Y accumulators are applied as inputs to digital-to-analog converters.

A nominal magnitude of one integer unit is often used for the
incremental step size on each of the N clock cycles utilized to generate a
vector. This requires a minimum range from -1 to +1, inclusive, for the
DX/N and DY/N inputs to the accumulators of Figure 1. A general purpose
20 design, however, may require a significantly larger range for these inputs.
For example, one design may be utilized in several applications that have
different resolution requirements. A low resolution application does not
require the entire integer portions of the X and Y accumulators as outputs,
and may utilize only the more significant bits of these integers. The less
25 significant integer bits (i.e., those not being used for the X and Y
outputs) are loaded with the value zero when the X and Y accumulators are
initialized. The DX/N and DY/N inputs would then be increased in magnitude

SUBSTITUTE SHEET

- 17 -

1 for such low-resolution applications in order to achieve an
adequate incremental step size at each clock cycle of a vector.
High-resolution applications, in particular those utilizing
analog X and Y output signals, can also make use of a larger
5 than normal step size in order to increase the speed at which
vectors are drawn.

For the tangent vector algorithm and the sine/cosine
vector algorithm, two different approaches are normally
utilized for determining the number of clock cycles, N, to be
10 used in generating a vector. For the tangent algorithm, the
value of N is normally provided by the absolute magnitude of DX
or the absolute magnitude of DY, whichever is the larger. For
example, if $|DX| \geq |DY|$, then N is given by: $N = |DX|$; and the
incremental values added to the X and Y accumulators are,
15 respectively, $DX/|DX|$ (i.e., either +1 or -1) and $DY/|DX|$. The
value added to the Y accumulator in this case is equal in
magnitude to the tangent of the acute angle defined by the
intersection of the desired vector with the X axis, hence the
designation of tangent algorithm. Scaled versions of the
20 tangent algorithm, for example, where N is given by twice the
larger of $|DX|$ or $|DY|$ or by approximately 1/2 the larger of DX
or DY, will not be described. It is appreciated, however,
that the invention is applicable to such variations of the
tangent vector algorithm.

25 For the unscaled tangent algorithm, at least one of
the accumulators is advanced by exactly one integer unit on
each clock cycle of a vector. For a 45° vector, both
accumulators are advanced by one integer unit on each clock
cycle. With the tangent algorithm, the magnitude of the
30 incremental step size at each clock cycle of a vector is a
function of the vector angle. The incremental step size varies
from a magnitude of 1, for a horizontal or vertical vector, to
a magnitude of the square root of 2, for a 45° vector. The
magnitude of the velocity of the generated vector varies with
35 vector angle in the same manner as does the step size.

SUBSTITUTE SHEET

-18-

1 The tangent algorithm is inherently more accurate than the
sine/cosine algorithm, and it generally results in more aesthetically
pleasing vectors. However, because of the variation in step size that
occurs with the tangent algorithm, the sine/cosine algorithm is more
5 appropriate for some applications. For example, in some display
applications, dotted or dashed vectors are generated by modulating the video
signal on and off with a pattern that is controlled by the accumulator clock
signal. With the tangent algorithm, the distance over which the dot/dash
pattern is repeated, is a function of vector angle. This is due to the
10 variation in step size with vector angle. This problem does not occur with
the sine/cosine algorithm, since it has a relatively constant step size.

For the unscaled sine/cosine algorithm, the number of clock cycles
used in generating a vector is nominally given by: $[DX^2 + DY^2]^{1/2}$. This
results in a step size of one integer unit regardless of the vector angle.
15 It is appreciated, however, that the number of clock cycles must be an
integer, and the value N is actually derived by rounding off the value of
 $[DX^2 + DY^2]^{1/2}$ to the nearest integer.

The present invention improves the accuracy of vectors generated
with either the tangent or the sine/cosine algorithms. However, when used
20 with the inherently more accurate tangent algorithm, the invention can
generate vectors of the highest possible mathematical accuracy. Although
the invention is primarily described herein as applied to the tangent
algorithm, it is also applicable to the sine/cosine algorithm.

Prior to discussing further details of the invention, a commonly
25 employed variation in the hardware implementation of the tangent algorithm
will now be described. Instead of utilizing two accumulators, as
illustrated in Figure 1, the tangent vector algorithm may be implemented

SUBSTITUTE SHEET

-19-

1 with two up/down counters (i.e., one for the X output and one for the Y
 output) and one accumulator that does not include an integer portion, but
 only the circuitry required to detect overflow in the adder from the
 fractional accumulator into the integer portion. As described above with
 5 respect to the tangent algorithm, at least one of the outputs (i.e., either
 X or Y) advances by exactly one integer unit on each clock cycle of the
 generated vector. This can be implemented by advancing the corresponding X
 or Y counter of this alternative implementation, either up or down, on each
 clock cycle. The other counter is advanced only on clock cycles in which
 10 the fractional accumulator overflows into the integer units, which would be
 on every cycle in the case of a 45° vector. The values of the X and Y
 outputs with this alternate implementation are identical to those of the
 embodiment of Figure 1. In applying the invention to this alternative
 implementation, the fractional accumulator is treated, for any given vector,
 15 as the accumulator of Figure 1 that corresponds to the axis of the smallest
 vector component.

In order to contrast vector generation in accordance with the
 prior art and vector generation in accordance with the present invention,
 the following Table 1 illustrates the generation of a vector in accordance
 20 with the tangent algorithm with $DX = +3$ and $DY = -8$.

TABLE 1

CLOCK CYCLE	1A		1B	
	INITIAL VALUE = 0		INITIAL VALUE = 0.5	
	X	Y	X	Y
INITIAL	0	0	0.5	0.5
1	0.375	-1	0.875	-0.5
2	0.75	-2	1.25	-1.5
3	1.125	-3	1.625	-2.5
4	1.5	-4	2.0	-3.5
5	1.875	-5	2.375	-4.5
6	2.25	-6	2.75	-5.5
7	2.625	-7	3.125	-6.5
8	3.0	-8	3.5	-7.5

SUBSTITUTE SHEET

-20-

1 Table 1 delineates the initial values in the X and Y accumulators
of Figure 1 and the values after each of the eight clock cycles utilized to
generate the vector. Table 1A provides the X and Y values for the prior art
generation of the vector and Table 1B provides the X and Y values for
5 generating the vector with the fractional portions of the X and Y
accumulators initialized to the value of 1/2 in accordance with the
invention. Figure 2A illustrates the pixels illuminated for generating the
vector in accordance with the prior art as delineated in Table 1A. Figure
2B illustrates the vector generated in accordance with the invention as
10 delineated in Table 1B. It is appreciated that only the integer portions of
the values delineated in Table 1 are utilized for the outputs that determine
the illuminated pixels of the displayed vectors illustrated in Figures 2A
and 2B. Although the accumulator values in Table 1 are, for convenience,
provided in decimal, the vectors displayed in Figures 2A and 2B are
15 predicated on a two's complement binary hardware implementation.

The present invention modifies the prior art vector generator to
improve the accuracy of the resulting symbology. The first modification is
to initialize the fractional portion of the accumulators to a value of 1/2,
or to a value of 1/2 less the value of the LSB of the accumulator (i.e.,
20 where $1/2 = .100\dots0$ and $1/2 - 1 \text{ LSB} = .011\dots1$, in binary two's
complement). For the exemplified vector, this modification results in the
accumulator values delineated in Table 1B above and the corresponding vector
illustrated in Figure 2B.

The accuracy of the vector of Figure 2B, initialized to the value
25 of 1/2 in accordance with the invention, and the vector of Figure 2A,
initialized to zero in accordance with the prior art, may be compared by
lines 71 and 70 drawn between the respective vector end points. The lines

SUBSTITUTE SHEET

-21-

1 70 and 71 define the location of the actual vector the hardware is
attempting to generate. The line 71, compared to the line 70, illustrates
that the present invention has generated a more accurate vector than the
prior art. In the absence of truncation errors, the present invention
5 illuminates only pixels that are within the distance of 1/2 the pixel-to-
pixel spacing from the actual vector. This is the closest representation of
the actual vector that can be achieved with a given display resolution, and
it defines a "mathematically perfect vector". The prior art vector
generator, which clears the fractional portions of the accumulators when the
10 integer portions are loaded, can illuminate pixels that approach a distance
of a full pixel-to-pixel spacing from the actual vector.

Since the correct value of DX/N or DY/N can be an irrational
number, the actual value that is utilized by the vector generator can
incorporate a truncation error. These errors can build up over a number of
15 clock cycles until the integer value of an accumulator is in error. This
problem is most likely to occur when consecutive vectors, including blank
vectors, are used to generate extensive complex symbology. The result is a
degradation in the appearance of the symbology.

A further modification to the prior art vector generator can
20 however, prevent this problem. If the accumulator fractions are initialized
to either 1/2 or to 1/2 less 1 LSB; if the number of fractional bits in the
accumulator is K ; if the incremental value added to the accumulator is
rounded to the nearest LSB (i.e., to the K th fractional bit); and if the
number of clock cycles used to generate the vector is limited to a maximum
25 value of 2^K ; then the error at the vector end point will be restricted to
the fractional bits of the accumulator. Therefore, by re-initializing the
fractional bits of the accumulators on the last clock cycle of every
generated vector (i.e., by loading them with 1/2 or 1/2 less 1 LSB), the
build-up of truncation errors over consecutive vectors is eliminated.

SUBSTITUTE SHEET

- 1 The previously described modifications to the prior art vector generator guarantee that, for either the tangent or sine/cosine algorithm, every vector will terminate precisely at its correct end point. Mathematically perfect vectors are, however, not necessarily achieved.
- 5 Errors can still occur in the integer portions of the accumulators at intermediate points along a generated vector. For the tangent algorithm, these errors can also be eliminated if the fractional resolution of the accumulators is increased relative to the maximum vector length (i.e., the vector length as determined by the maximum number of clock cycles). For
- 10 example, if the accumulators have K fractional bits, then mathematically perfect vectors are generated for all vectors with the number of clock cycles less than or equal to $2^{K/2}$.

Even with mathematically perfect vectors, a problem arises in the generation of symmetrical symbology. When a given vector lies exactly

15 between two pixels, it is mathematically arbitrary which pixel is illuminated. When, however, vectors are concatenated to form symbols, this choice is no longer arbitrary if symmetrical symbology is to be generated. An example of this problem is illustrated by the letter "V" of Figure 3A. This symbol, which is clearly unsymmetrical, was generated with a vector

20 having $DX = +3$ and $DY = -8$ (as in the previous example), followed by a vector with $DX = +3$ and $DY = +8$. Table 1B delineates the accumulator values after each clock cycle of the first vector. After clock cycle 4, the integer values of X and Y are equal to $+2$ and -4 , respectively (i.e., as the accumulator values are expressed in the binary 2's complement number

25 system). At $Y = -4$, the actual vector is at $X = 1.5$. Therefore, it is mathematically arbitrary whether the pixel at $X = 1, Y = -4$ or the pixel at $X = 2, Y = -4$ is illuminated. Similarly, for the second vector it is arbitrary whether the pixel at $X = 4, Y = -4$ or the pixel at $X = 5, Y = -4$ is illuminated.

-23-

1 Figure 3A illustrates the symbol generated with the accumulator
fractions initialized to $1/2$. If the accumulator fractions are initialized
to $1/2$ less 1 LSB, then the opposite choices are effected with respect to
which of the mathematically arbitrary pixels are illuminated. The symbol
5 illustrated in Figure 3B is thus generated. If one of the vectors is
initialized to $1/2$, and the other initialized to $1/2$ less 1 LSB, the
resulting symbol will be symmetrical as illustrated in Figure 3C. The
mathematically arbitrary pixel in the first vector is denoted by reference
numerals 75 and 75' and the mathematically arbitrary pixel of the second
10 vector is denoted by reference numerals 76 and 76'. Thus, a further
advantage of the present invention is illustrated by the comparison of the
three examples of the generation of the letter "V" depicted in Figures 3A,
3B and 3C.

 The present invention generates symmetrical symbology by
15 initializing the accumulator fractions to the value of $1/2$ less 1 LSB, and
then enabling the accumulator carry input (lines 16, 36 of Figure 1) on the
first clock cycle of some vectors. Enabling the carry input on the first
clock cycle is equivalent to initializing the accumulator fraction to $1/2$,
instead of to $1/2$ less 1 LSB, and not enabling the carry input. The
20 decision on whether to enable the carry input on the first clock cycle of a
vector is predicated on the octant in which the vector to be generated
resides.

 Figure 4 illustrates the octant definitions utilized in generating
symmetrical symbology. The following Table 2 provides two algorithms for
25 the generation of symmetrical symbology utilizing concatenated vectors with
the tangent vector generation algorithm.

SUBSTITUTE SHEET

TABLE 2

		CARRY INPUT ON 1ST CLOCK								
ALGORITHM	AXIS	OCTANT #								
		0	1	2	3	4	5	6	7	
#1	X	*	1	1	*	*	0	0	*	
	Y	0	*	*	1	1	*	*	0	
#2	X	*	0	0	*	*	1	1	*	
	Y	1	*	*	0	0	*	*	1	

* = DON'T CARE

1 The algorithms of Table 2 are implemented by a portion of the control logic
51 of the vector processor of Figure 1. The X and Y carry input signals are
applied to the lines 36 and 16, respectively.

In the embodiments of the invention heretofore described, the
5 values DX/N and DY/N are provided by rounding to the LSB bit position of the
accumulators. The aspects of the invention regarding symmetrical symbology,
as discussed, did not consider the effect of truncation errors with respect
to the choice between two mathematically arbitrary pixels. The possibility
of truncation errors requires that, to guarantee the generation of
10 symmetrical symbology using concatenated vectors, a different truncation
algorithm must be utilized in providing the DX/N and DY/N signals. For K
fractional bits, symmetrical symbology, as well as mathematically perfect
vectors, can be generated with vectors having up to $2^{K/2}$ clock cycles.
However, when the number of clock cycles is an even integer, the DX/N or
15 DY/N value must be truncated for all vectors in octants with the carry input
to the corresponding accumulator disabled on the first clock cycle of the
vector. Also, when the number of clock cycles is an even integer, the DX/N
or DY/N value must be rounded up, when not represented exactly in K
fractional bits, for all vectors in octants with the carry input to the
20 corresponding accumulator enabled on the first clock cycle of the vector.
For vectors with an odd number of clock cycles, the DX/N and DY/N values are
always rounded to the nearest LSB.

1 Alternatively, for K fractional bits, symmetrical symbology is
generated if the maximum number of clock cycles in a vector is limited to
2^{(K-1)/2}, and if the DX/N and DY/N values for all vectors (i.e., regardless
of whether the number of clock cycles is an even or odd integer) are
5 determined in the same manner as described above for the vectors having an
even number of clock cycles.

 It is appreciated that the software system that establishes the
value for the slope 108 of the tan vector instruction word of Figure 1A
accommodates the above-described limitations so that symmetrical symbology
10 is generated in the presence of truncation errors.

 Referring again to Figures 3A-3C, the advantage of symmetrical
symbology generated in accordance with the present invention is exemplified
by a comparison of Figure 3C, which was generated by algorithm number 1 of
Table 2 above, with Figures 3A and 3B which are non-symmetrical. Referring
15 to Figures 5A and 5B, the advantages of the symmetrical symbology aspect of
the present invention are further exemplified. Figure 5A illustrates the
generation of an octagon in a counterclockwise direction with
mathematically perfect concatenated vectors. The accumulator fractions are
initialized to one-half in accordance with the invention. The octagon of
20 Figure 5B is generated in the counterclockwise direction utilizing
concatenated vectors with algorithm #1 of Table 2. The advantages of the
present invention regarding the generation of symmetrical symbology are
appreciated by comparing the appearance of the octagon of Figure 5B with
that of Figure 5A.

25 As previously described, the present invention can utilize the
sine/cosine vector generation algorithm to eliminate error accumulation when
generating concatenated vectors. Unlike the tangent algorithm, however, the
sine/cosine algorithm cannot generate mathematically

SUBSTITUTE SHEET

1 perfect vectors. Nevertheless, the sine/cosine algorithm can be utilized to generate symmetrical symbology with concatenated vectors. The following Table 3 defines four symmetry algorithms for sine/cosine vector generation.

TABLE 3
CARRY INPUT ON 1ST CLOCK

ALGORITHM	AXIS	OCTANT #							
		0	1	2	3	4	5	6	7
#1	X	1	1	1	1	0	0	0	0
	Y	0	0	1	1	1	1	0	0
#2	X	0	0	0	0	1	1	1	1
	Y	1	1	0	0	0	0	1	1
#3	X	0	1	1	0	1	0	0	1
	Y	0	1	0	1	1	0	1	0
#4	X	1	0	0	1	0	1	1	0
	Y	1	0	1	0	0	1	0	1

The octant definition with respect to Table 3 is illustrated in Figure 4.

5 It is noted that the algorithms utilized to generate symmetrical symbology using the sine/cosine technique are similar to those for the tangent algorithm, except that the accumulator carry inputs on the first clock cycle are never "Don't Cares", as is the situation in half of the table entries of Table 2 above.

10 Referring to Figure 6, an annotated octagon is illustrated for use in the development of the symmetry algorithms. The octagon is drawn with concatenated vectors in the counterclockwise direction and the numbers inside the octagon denote the octant of each such vector. The notations outside the octagon provide the X and Y directions from each vector toward the outside of the octagon. Specifically, the notations outside the octagon
 15 comprise two notations associated with each vector, one of which is either X+ or X-, and the other of which is either Y+ or Y-. For each vector, the notations indicate which X direction (i.e., X+ or X-) and which Y direction, is on the side of the vector that lies outside of the octagon. Enabling the
 20 carry input on the first clock cycle will bias an accumulator, and the associated component of the

-27-

1 generated vector, toward the positive direction (i.e., toward
the X+ or Y+ direction). Similarly, by disabling the carry
input on the first clock cycle, the accumulator is biased
toward the negative direction (i.e., toward X- or Y-).

5 Symmetrical symbology is generated if the carry inputs to the
accumulators are selected in accordance with the bias that is
shown in Figure 6 to always select between mathematically
arbitrary pixels such that the pixel towards the outside of the
octagon will be illuminated. Assuming that the octagon is
10 drawn in the counterclockwise direction, this selection results
in algorithm #1 of Table 2, for vectors generated with the
tangent algorithm, and algorithm #1 of Table 3, for vectors
generated with the sine/cosine algorithm. Symmetry will also
result if the carry inputs are selected so that the pixels
15 toward the inside of the octagon are illuminated. This results
in algorithm #2 in both Tables 2 and 3. It is noted that
algorithm #1 illuminates the pixels that are toward the outside
of the octagon only when the octagon is drawn in the
counterclockwise direction and illuminates the pixels that are
20 toward the inside when the octagon is drawn in the clockwise
direction. Similarly, algorithm #2 illuminates the inside
pixels when drawing the octagon in the counterclockwise
direction, and the outside pixels when drawing in the clockwise
direction.

25 For the sine/cosine technique, two additional
algorithms are provided for the generation of symmetrical
symbology. Algorithm #3 of Table 3 results from the selection
of the outside pixels for the axis that corresponds to the
shortest component of each vector, and the selection of the
30 inside pixels for the axis that corresponds to the longest
component of each vector, when the octagon is generated in the
counterclockwise direction. Similarly, algorithm #4 results
from the selection of the inside pixels for the axis that
corresponds to the shortest component of each

SUBSTITUTE SHEET

-28-

1 vector, and the selection of the outside pixels for the axis that
corresponds to the longest component of each vector, when the octagon is
generated in the counterclockwise direction. In the manner described above
with respect to Table 2, Table 3 is implemented by a portion of the control
5 logic 51.

As described above, the present invention eliminates truncation
errors at the vector end point. The following is an error analysis for the
correct vector end point and provides a derivation of the maximum error at a
vector end point.

10 Assume a vector is drawn from point X_1, Y_1 , to X_2, Y_2 . Then, $DX =$
 $X_2 - X_1$ and $DY = Y_2 - Y_1$, where DX and DY are integers. If a total of N
clock cycles are used to generate the vector, the incremental values that
should be added to the X and Y accumulators at each clock cycle are given by
 DX/N and DY/N , respectively. Consider only the error in the Y accumulator.
15 The analysis is also valid for the X accumulator because of symmetry.

The actual value added to the Y accumulator on every clock cycle
is given by $DY/N + e$, where the error term, e , is given by:

1) $e = \text{Actual Value} - \text{Correct Value} = [DY/N + e] - DY/N$. The error term
results from the necessity of representing DY/N with a finite number of
20 fractional bits. If DY/N is an irrational number, then its exact
representation in 2's complement binary requires an infinite number of
fractional bits. If the accumulator has K fractional bits, the weight of
the Least Significant Bit (LSB) of the accumulator is equal to 2^{-K} . In the
exact 2's complement representation of DY/N , all the bit positions to the
25 right of the K^{TH} fractional bit contribute some value, E , to the total
amount of DY/N (where E is greater than, or equal to, zero; and is less than
 2^{-K}). If the value added to the accumulator is derived by dropping (i.e.,

SUBSTITUTE SHEET

-29-

1 truncating) the value E from the exact representation of DY/N , e will be given by:

$$2) \quad e = -E \text{ (for truncation)}$$

If, for the case when $E \geq 1/2$ LSB, the value added to the accumulator is rounded up (i.e., by adding +1 to the LSB and truncating the value E), then

$$3) \quad e = \text{LSB} - E \text{ (for rounding up)}.$$

The integer portion of the Y accumulator is initially loaded with the value Y_1 , and the fractional portion of the accumulator is initialized to the value of $1/2$ or to the value $1/2$ less 1 LSB. If the initial fractional value of Y is designated as F , then the value of the Y accumulator at the

$$4) \quad Y = Y_1 + F + [(DY/N) + e]N = Y_1 + DY + F + e.N,$$

where Y_1 and DY are integers and F is the initial value of the fraction. If no error is allowed in the integer value of the Y accumulator, then:

$$5) \quad 0 \leq F + e.N < 1$$

Using worst case values of F (i.e., $1/2$ for the positive limit and $1/2$ less 1 LSB for the negative limit), and solving for e , gives:

$$6) \quad -(1/2N) + (\text{LSB}/N) \leq e < 1/2N$$

The number of fractional bits in the accumulators must be large enough so that the error term in the incremental value added to the accumulator is within the range defined by inequality 6); where N is the number of clock cycles used to generate a vector, and LSB is the value of the least significant bit of the accumulator. This will ensure that the integer value of the accumulator will not be in error at the vector end point. The fractional portions of the accumulators can then be initialized (i.e., loaded with $1/2$ or with $1/2 - 1$ LSB) on the last clock cycle of every vector. This will prevent error build up on consecutive vectors.

SUBSTITUTE SHEET

-30-

1 If K bits are available to define the value N, then the maximum
 number of clock cycles is equal to 2^K . (Note that zero length vectors can
 be disallowed. The value of N can then be defined as one greater than the
 number defined within the K bits). It will now be shown that with $N \leq 2^K$, a
 5 total of K fractional bits are sufficient to meet the requirements of 6).

The worst case value of N, relative to inequality 6), is given by
 2^{K-1} (note that e will be 0 for $N = 2^K$). Substituting into 6), with $LSB =$
 2^{-K} , gives:

$$10 \quad 7) \quad - \frac{1}{2^{K+1} - 2} + \frac{2^{-K}}{2^{K-1}} \leq e < \frac{1}{2^{K+1} - 2}$$

The actual value used for DY/N can include an error value, e, due to the
 limitation of K fractional bits. If any number between -1 and +1 were
 possible for the correct value of DY/N , then by rounding to the closest LSB
 (i.e., nearest 2^{-K}), the actual range of e would be given by:

$$15 \quad 8) \quad - \frac{1}{2^{K+1}} < e \leq \frac{1}{2^{K+1}}$$

If the actual range of e (given by inequality 8) lies within the required
 range (given by inequality 7), then K fractional bits would indeed be
 sufficient to prevent integer errors at the end point. This is clearly the
 case for the positive limits of the inequalities. It is not, however, the
 20 case for the negative limits.

If inequality 6) is solved for $N = 2^{K-2}$, then the result is given by:

$$9) \quad - \frac{1}{2^{K+1} - 4} + \frac{2^{-K}}{2^K - 2} \leq e < \frac{1}{2^{K+1} - 4}$$

It can be demonstrated that the negative limit of 8) is equal to
 25 the negative limit requirement of 9). Therefore, K fractional bits are
 sufficient to prevent integer errors with vectors having $2^K - 2$, or fewer,
 clock cycles.

SUBSTITUTE SHEET

-31-

1 Inequality 8) was derived under the assumption that DY/N could be any value
 between -1 and $+1$. Because this is not precisely true, the negative limit
 of inequality 8) is less than the value that can actually occur. The limit
 assumed a truncated value, E , that is arbitrarily close to (but still less
 5 than) the value of $1/2$ LSB (where $1/2$ LSB = 2^{-K-1}). A more exact value for
 the negative limit of e will now be determined.

If the correct value to be added to the Y accumulator,
 DY/N , is divided by the weight of the LSB (i.e.; by 2^{-K}), this has the
 effect of moving the binary point to the right by K bit positions.
 10 Therefore, the fractional portion of the result of this division will
 consist of all the bits to be truncated in deriving the value that will be
 used to approximate DY/N (i.e., in deriving the value $(DY/N) + e$). If the
 worst case fractional result of this division (i.e., the fraction that is
 closest to, but still less than, $1/2$) is multiplied by the weight of the
 15 LSB, this will give the magnitude of the worst case negative limit for e .

The previously prescribed division can be expressed as:

$$10) \quad \frac{(DY)}{(N)} \cdot \frac{1}{LSB} = Q + \frac{R}{N}$$

where Q is an integer and R is a positive integer that is less than N . The
 largest value of R/N that is less than $1/2$ occurs when N is the largest
 20 possible odd integer. It is given by:

$$11) \quad \text{Fraction} = \frac{N/2 - 1/2}{N} = \frac{N-1}{2N}$$

The magnitude of the negative error limit can now be determined by
 multiplying this fraction by the LSB (i.e., where $LSB = 2^{-K}$ and $N = 2^{K-1}$).

The result is given by:

$$25) \quad \text{Neg. limit} = - \frac{2^K - 1 - 1}{2(2^{K-1})} (2^{-K}) = - \frac{1}{2^{K+1} - 2} + \frac{2^{-K+1}}{2^{K+1} - 2}$$

Substituting this value for the negative limit of inequality 8) gives:

SUBSTITUTE SHEET

-32-

$$1 \quad 13) \quad - \frac{1}{2^{K+1} - 2} + \frac{2^{-K+1}}{2^{K+1} - 2} \leq e \leq \frac{1}{2^{K+1}}$$

Inequality 13) gives the actual limits in the error term of the value that is added to the accumulator (i.e., for K fractional bits and 2^K maximum clock cycles). These limits lie within the required limits that are given by inequality 7). (In fact, the negative limits of inequalities 7) and 13) are equal). Therefore, no error can occur in the integer bits of the accumulators at the vector end point when:

- 1) The accumulator fractions are initialized to either $1/2$ or $1/2 - 1$ LSB.
- 2) K fractional bits are available for the accumulators; 3) the value added to an accumulator is rounded to the nearest LSB (i.e., to the K^{TH} fractional bit); and 4) a maximum of 2^K clock cycles is used to generate the vector.

Increasing the fractional resolution of the accumulators beyond that required for the elimination of vector end-point errors, can, with the tangent algorithm, result in the generation of mathematically perfect vectors. The following error analysis derives the resolution requirements for mathematically perfect vectors. The requirements for combining symmetrical symbology with mathematically perfect vectors are also derived.

Assume a vector is drawn from point X_1, Y_1 to point X_2, Y_2 .

Let $X_2 - X_1 = N$, and $Y_2 - Y_1 = M$. Both N and M are integers. Consider the tangent algorithm with $|N|$ clock cycles (i.e., $|N| \geq |M|$).

CASE #1: initial value of accumulator fraction = $1/2$. The "correct" value for the Y accumulator after clock cycle "i" (i.e., with no truncation errors) is given by:

$$14) \quad Y_C(i) = Y_1 + \frac{1}{2} + \frac{M}{|N|} \cdot i$$

But the approximate value that is added to the Y accumulator is given by:

$(M/|N|) + e$, where the error term, e, results from truncation (and possibly

SUBSTITUTE SHEET

-33-

1 from rounding). Therefore, the "actual" value of Y after clock cycle i is:

$$15) \quad Y_A(i) = Y_1 + \frac{1}{2} + \frac{M}{|N|} \cdot i + e \cdot i$$

Note that e is given by:

$$16) \quad e = \text{Actual slope value} - \text{Correct slope value} = \\ 5 \quad [(M/|N|) + e] - M/|N|$$

Therefore, if the value E (as defined above) is dropped (or truncated) from the correct value to form the actual value used, then:

$$17) \quad e = -E \quad (\text{for truncation})$$

If the actual value is derived by rounding up (i.e., by adding +1 to the LSB and then truncating the value E) (i.e., when $E \geq 1/2$ LSB), then:

$$18) \quad e = \text{LSB} - E \quad (\text{for rounding up})$$

The term, $(M/|N|) \cdot i$, in equations 14) and 15) can be expressed as:

$$19) \quad \frac{M}{|N|} \cdot i = Q + \frac{R}{|N|},$$

where Q and R are both integers, and $|R/N| < 1$.

15 Substituting into equations 14) and 15) gives:

$$20) \quad Y_C(i) = Y_1 + \frac{1}{2} + Q + \frac{R}{|N|}$$

$$21) \quad Y_A(i) = Y_1 + \frac{1}{2} + Q + \frac{R}{|N|} + e \cdot i$$

The highest susceptibility to an error in the integer value of the Y accumulator occurs when $R/|N|$ is very close to (but not equal to) $+1/2$ or $-1/2$. (When $R/|N| = \pm 1/2$, either of two integers is a "mathematically" equivalent approximation to the Y value. Therefore, this case is relevant to generating symmetrical symbols (e.g., $N = 6$, $M = 1$ or 5), but is not a worst case for generating "mathematically" correct vectors). The $R/|N|$ term will be closest to $\pm 1/2$ when N is an odd integer, and when N is large. For these cases,

$$22) \quad R = \pm \frac{N}{2} \pm \frac{1}{2}$$

SUBSTITUTE SHEET

1 where N is an odd integer.

Substituting into 20) and 21) gives:

23)
$$Y_C(i) = Y_1 + \frac{1}{2} + Q \pm \frac{1}{2} \pm \frac{1}{2N}$$

24)
$$Y_A(i) = Y_1 + \frac{1}{2} + Q \pm \frac{1}{2} \pm \frac{1}{2N} + e_i$$

5 where the sign of the term $\pm 1/2$ will be the same as the sign of the value Q, and N is a large odd integer.

If $Y_A(i)$ is to have the same integer value as $Y_C(i)$ then, from equations 23) and 24),

25)
$$-\left| \frac{1}{2N} \right| \leq e_i < \left| \frac{1}{2N} \right|, \text{ and}$$

10 26)
$$-\left| \frac{1}{2Ni} \right| \leq e < \left| \frac{1}{2Ni} \right|$$

The most stringent requirement on e occurs when N is the largest magnitude odd integer, and when i is given by:

27)
$$i = |N| - 1$$

(Note that when $i = |N|$, the vector end point is attained and the error sensitivity is very low).

15 Now, if L bits are available to specify the vector length (i.e., the number of clock cycles), and if zero length vectors are not allowed, then the maximum length is given by 2^L , and the maximum odd value is:

28)
$$N = 2^L - 1$$

20 Substituting 27) and 28) into equation 26) gives:

29)
$$-\frac{1}{2^{2L+1} - 2^{L+2} - 2^{L+1} + 4} \leq e < \frac{1}{2^{2L+1} - 2^{L+2} - 2^{L+1} + 4}$$

It would therefore be sufficient (but not quite necessary) to require that:

30)
$$-\frac{1}{2^{2L+1}} < e < \frac{1}{2^{2L+1}}$$

25 in order to prevent errors in the Y integer for Case #1 with the fraction

-35-

1 initially equal to 1/2. This requirement, which is more
 stringent than 29), is the result of rounding the value of M/N
 to the nearest value that can be represented with 2L fractional
 bits. Therefore, "mathematically" perfect vectors will result
 5 with 2L fractional bits, rounded slope values, and fractions
 initialized to 1/2. It can be demonstrated that 2L-1
 fractional bits would be insufficient for this case (e.g., with
 N = 55, i = 54, M = 27).

CASE #2: Initial value of accumulator is equal to 1/2 less
 10 the LSB of the accumulator fraction.

This case is of interest primarily for the special
 case of forcing symmetrical symbology for concatenated vectors.
 For vectors that are exactly between two pixels, the system
 must control which of the pixels is illuminated if symmetrical
 15 symbols are to result.

As a first step, it will be determined if
 "mathematically" perfect vectors will result for the same
 criteria as for Case #1 (i.e., with L-bits for vector length,
 2L fractional accumulator bits, and rounded values of M/N).
 20 The "arbitrary" case of a vector being exactly between two
 pixels will not be considered.

The equations for the "Correct" value of Y and the
 "Actual" value of Y are given by:

$$31) \quad Y_C(i) = Y_1 + 1/2 - \text{LSB} + \frac{M}{|N|} \cdot i, \text{ and}$$

$$25 \quad 32) \quad Y_A(i) = Y_1 + \frac{1}{2} - \text{LSB} + \frac{M}{|N|} \cdot i + e_i, \text{ where}$$

$$33) \quad \text{LSB} = \frac{1}{2^{2L}}$$

Recalling from Case #1 that:

$$19) \quad \frac{M}{|N|} \cdot i = Q + \frac{R}{|N|}, \text{ gives:}$$

SUBSTITUTE SHEET

1 34) $Y_C(i) = Y_1 + \frac{1}{2} - \text{LSB} + Q + \frac{R}{|N|}$, and

35) $Y_A(i) = Y_1 + \frac{1}{2} - \text{LSB} + Q + \frac{R}{|N|} + ei$

The greatest susceptibility to an error in the integer value of $Y_A(i)$ occurs when $R/|N|$ is close to (but not equal to) $+1/2$ or $-1/2$. (Note that when

5 $R/|N| = \pm 1/2$, then the vector will be exactly between two pixels and it is mathematically arbitrary which pixel is illuminated. This case is not of immediate interest). The $R/|N|$ term will be closest to $\pm 1/2$ when

22) $R = \pm \frac{N}{2} \pm \frac{1}{2}$,

where N is a large odd integer.

10 Substituting into 34) and 35) gives:

36) $Y_C(i) = Y_1 + \frac{1}{2} - \text{LSB} + Q \pm \frac{1}{2} \pm \frac{1}{2N}$

37) $Y_A(i) = Y_1 + \frac{1}{2} - \text{LSB} + Q \pm \frac{1}{2} \pm \frac{1}{2N} + ei$

From equations 36) and 37)

38) $-\frac{1}{2N} + \text{LSB} \leq ei < \frac{1}{2N} + \text{LSB}$,

15 if no error is to occur in the integer value of Y_A .

Solving for e , and substituting 33), 28) and 27) for LSB , N , and i , respectively, gives:

39) $-\frac{1}{(2^{L+1}-2)(2^L-2)} + \frac{1}{2^{2L}(2^L-2)} \leq e < \frac{1}{(2^{L+1}-2)(2^L-2)} + \frac{1}{2^{2L}(2^L-2)}$

20 The above expression defines the acceptable range of e for the generation of mathematically perfect vectors. The actual range of e for $2L$ fractional bits with rounding of M/N was given by:

30) $-\frac{1}{2^{2L+1}} < e < \frac{1}{2^{2L+1}}$

Since the acceptable range of e in 39) has a smaller magnitude for the negative limit of the range than for the positive limit, it will only be

SUBSTITUTE SHEET

1 necessary to compare the negative limits of 30) and 39). If the magnitude
of the acceptable negative limit of 39) is larger than (or equal to) the
magnitude of the actual negative limit of 30), then 2L fractional bits are
sufficient for mathematically perfect vectors. This is shown to be true as
5 follows:

$$\frac{1}{(2^{2L+1}-2)(2^L-2)} - \frac{1}{2^{2L}(2^L-2)} \stackrel{>?}{>} \frac{1}{2^{2L+1}}$$

$$\frac{2^{2L+1}-2}{(2^{2L+1}-2)(2^L-2)} \stackrel{>?}{>} \frac{(2^{L+1}-2)(2^L-2)}{(2^{L+1}-2)(2^L-2)2^{2L+1}}$$

$$2^{2L+1}-2^{L+2}+4 \stackrel{>?}{\text{QED}} 2^{2L+1}-2^{L+2}-2^{L+1}+4$$

10 It has now been demonstrated that "mathematically" perfect
vectors will result when: the fractional portions of the accumulators are
initialized to either 1/2 or to 1/2 less 1 LSB; the slope value is rounded
to the nearest LSB; and 2L bits are available for the slope and accumulator
fractions (i.e., where L bits define the vector length in terms of the
number of accumulator clock cycles). When the desired vector lies exactly
15 between two pixels, it is mathematically arbitrary which pixel is
illuminated. However, if symmetrical symbology is desired when vectors are
concatenated to form symbols, then it is no longer arbitrary which of these
mathematically equivalent pixels is illuminated.

20 If the problem of truncation errors is neglected, then
symmetrical symbols can be generated with chained vectors by using an
algorithm that initializes the accumulator fractions to either 1/2 or to 1/2
less 1 LSB, depending on the direction (i.e., the octant) of the vector to
be drawn. This can also be accomplished by always initializing the
fractions to 1/2 less 1 LSB, and then forcing a carry input to the

25

-38-

1 accumulator/adder on the first clock cycle of vectors that are in selected
 octants. Then for octants with the carry input enabled (i.e., on the first
 clock cycle only), the larger, or more positive, of two mathematically
 equivalent integers (pixel locations) will be selected. For octants with
 5 the carry input disabled, the lesser, or more negative, of two mathematically
 equivalent integers is selected. This approach generates symmetrical
 symbology, using chained vectors, when the slope value is exact. When,
 however, the slope cannot be represented exactly by 2L fractional bits, then
 a different approach to truncation is required (i.e., other than rounding to
 10 the nearest LSB).

CASE #3: Symmetrical vectors for octants with carry input enabled.

This case is equivalent to initializing the accumulator fraction
 with 1/2 and not enabling the carry input (i.e., as in Case #1). The
 equations for the correct and the actual values for the Y accumulator were
 15 given by:

$$20) \quad Y_C(i) = Y_1 + \frac{1}{2} + Q + \frac{R}{|N|}$$

$$21) \quad Y_A(i) = Y_1 + \frac{1}{2} + Q + \frac{R}{|N|} + e_i$$

For the mathematically arbitrary case: $R/|N| = \pm 1/2$. The carry input was
 enabled on the first clock cycle in order to force the larger of the
 20 mathematically equivalent integers for the Y accumulator. Therefore,
 considering only the case where $R/|N| = \pm 1/2$, if no error is to be allowed
 in the integer value of $Y_A(i)$, then:

$$40) \quad 0 \leq e_i < 1$$

Of course, inequality 30) must still be met (i.e., in addition to 40), to
 25 meet the requirements for mathematically correct vectors. Inequality 40)
 requires that, for symmetrical symbology, a simple truncation can no longer
 be used on the slope value of vectors that are in octants that have the

SUBSTITUTE SHEET

-39-

1 carry input enabled on the first clock cycle. If the slope
value cannot be exactly represented, then it must always be
rounded up to the next LSB. Using this method to derive the
slope value, and meeting the requirements of both inequalities
5 30) and 40), would require a minimum of $2L + 1$ fractional bits
for the accumulators and the slope value. The slope error
would then be given by:

$$41) \quad 0 \leq e < \frac{1}{2^{2L+1}}$$

By separating Case #3 into two distinct "subcases",
10 and by using different algorithms for determining the slope
values for these subcases, it is possible to have
mathematically correct vectors and symmetrical symbology with
only $2L$ fractional bits (i.e., instead of $2L+1$ bits).

CASE #3A: Carry input enabled with N , an odd integer

15 It is noted that a vector can lie exactly between
two pixels only when the term R/N of equations 20) and 21) is
given by $\pm 1/2$. But, since R and N are integers, this is only
possible when N is an even integer. Therefore, when N is odd,
the error term can be given by:

$$20 \quad 30) \quad - \frac{1}{2^{2L+1}} < e \leq \frac{1}{2^{2L+1}}$$

as in Case #1. This requires $2L$ bits for the slope value with
rounding to the nearest LSB.

CASE #3B: Carry input enabled with N , an even integer

The analysis of Case #1 examined the worse case
25 error susceptibility, which occurs when N is an odd integer.
Therefore, this must now be considered for the case with N an
even integer. Equations 20) and 21) show that the highest
susceptibility to an error in the integer value of Y_A occurs
when $R/|N|$ is very close to (but not equal to) $+1/2$ or $-1/2$.
30 (The case for $R/|N| = \pm 1/2$ pertains only to symmetrical
symbols and results in the inequality of 40)).

The worst case error susceptibility for the case
with N an even integer occurs when

SUBSTITUTE SHEET

1 42)
$$R = \frac{+N}{2} \pm 1$$
 ⁻⁴⁰⁻

Substituting into equations 20) and 21) gives:

43)
$$Y_C(i) = Y_1 + \frac{1}{2} + Q \pm \frac{1}{2} \pm \frac{1}{N}$$

44)
$$Y_A(i) = Y_1 + \frac{1}{2} + Q \pm \frac{1}{2} \pm \frac{1}{N} + e_i$$

5 Therefore, to prevent errors in the integer value of $Y_A(i)$ requires that:

45)
$$-\frac{1}{|N|} \leq e \cdot i < \frac{1}{|N|}$$

The most stringent requirement on e occurs for large values of N and i , and specifically for

10 46)
$$N = 2^L - 2, \text{ and}$$

47)
$$i = N - 1,$$

where L bits are available to specify the vector length. (Recall that N is an even integer, and note that $N = 2^L$ will not result in truncation errors for the slope value).

15 Substituting 46) and 47) into 45) and solving for e gives:

48)
$$-\frac{1}{2^{2L} - (3)2^{L+1} + 6} \leq e < \frac{1}{2^{2L} - (3)2^{L+1} + 6}$$

It would therefore be sufficient to require that:

49)
$$-\frac{1}{2^{2L}} < e \leq \frac{1}{2^{2L}}$$

20 which could be realized with $2L-1$ fractional bits with rounding of the slope value to the nearest LSB. If, however, symmetrical symbology is required, then both 48) and 49) must be met. This gives:

50)
$$0 \leq e \leq \frac{1}{2^{2L}}$$

25 This requirement must be met when N is an even integer and the carry is enabled (on the first clock cycle), in order to force symmetrical symbology. The requirement is met with $2L$ fractional bits for the slope. When the slope value is not exact, it must always be rounded up to the next LSB value.

SUBSTITUTE SHEET

-41-

1 To complete the analysis for symmetrical symbology, the case for octants with the carry input disabled is now examined.

CASE #4: Symmetrical symbology for vectors in octants with the carry input disabled.

5 This case is similar to Case #2 and the equations for Y_C and Y_A were given by:

$$34) \quad Y_C(i) = Y_1 + \frac{1}{2} - \text{LSB} + Q + \frac{R}{|N|}$$

$$35) \quad Y_A(i) = Y_1 + \frac{1}{2} - \text{LSB} + Q + \frac{R}{|N|} + e_i$$

For the mathematically arbitrary case: $R/|N| = \pm 1/2$. If no error is allowed in the integer value of Y_A for this case, then:

$$51) \quad -1 + \text{LSB} \leq e_i \leq 0$$

(Note that the actual upper limit of " $e_i < 1 \text{ LSB}$ " has been simplified to " $e_i \leq 0$ ". These limits are equivalent for all practical purposes due to the slope resolution of 1 LSB).

15 Combining the inequality of 51), required for symmetry, with the inequality of 30), required for mathematically correct vectors, gives:

$$52) \quad -\frac{1}{2^{2L+1}} < e_i \leq 0$$

This range of slope error can be obtained by always truncating, or dropping, all fractional bits of lower significance than the LSB; and by using a minimum of $2L + 1$ fractional bits. However, only $2L$ fractional bits are required if this case is divided into two subcases, as was Case #3; and different algorithms are used for determining the slope values of each subcase.

CASE 4A: Symmetrical symbology; carry input disabled; and N is odd integer.

25 As noted in Case #3A, the term $R/|N|$ cannot equal $\pm 1/2$ when N is odd. Therefore, this case is equivalent to Case #2. The error limits are given by 30), $2L$ fractional bits are required, and the slope value must be rounded to the nearest LSB.

SUBSTITUTE SHEET

-42-

1 CASE 4B: Symmetrical symbology; carry input disabled; and N is an even integer.

As with Case #3B, the worst case error susceptibility, for N an even integer, occurs when:

5 42)
$$R = \pm \frac{N}{2} \pm 1$$

Substituting 42) into equations 34) and 35) for Y_C and Y_A gives

53)
$$Y_C(i) = Y_1 + \frac{1}{2} - \text{LSB} + Q \pm \frac{1}{2} \pm \frac{1}{N}$$

54)
$$Y_A(i) = Y_1 + \frac{1}{2} - \text{LSB} + Q \pm \frac{1}{2} \pm \frac{1}{N} + ei$$

If no error is allowed in the $Y_A(i)$ integer, then:

10 55)
$$-\frac{1}{N} + \text{LSB} \leq ei < \frac{1}{N} + \text{LSB}$$

Substituting for LSB, N, and i in equation 55) (from equations 33), 46) and 47)) gives:

56)
$$-\frac{1}{(2^{L-2})(2^{L-3})} + \frac{1}{2^{2L}(2^{L-3})} \leq e < \frac{1}{(2^{L-2})(2^{L-3})} + \frac{1}{2^{2L}(2^{L-3})}$$

It can be shown that more stringent limits are expressed by:

15 57)
$$-\frac{1}{2^{2L}} < e < \frac{1}{2^{2L}}$$

which requires only $2L-1$ fractional bits with rounding. Combining

41) and 57), however, gives:

58)
$$-\frac{1}{2^{2L}} < e \leq 0$$

20 This can be met with $2L$ fractional bits and truncation of the slope value (i.e., never rounding up).

The following is concluded from the foregoing Error Analysis for Perfect Vectors and Symmetrical Symbology.

1) For "Mathematically" Perfect Vectors (Tangent Algorithm)

Fractional portions of accumulators should be initialized to $1/2$,

-43-

1 or to $1/2$ less 1 LSB, when accumulators are loaded (i.e., initial position instruction).

Fractional portions of accumulators should also be loaded with $1/2$, or $1/2$ less 1 LSB, on the last clock cycle of every vector. (This
5 initializes fractions for the execution of consecutive vectors by clearing any error accumulation).

If the initialization value of $1/2$ less 1 LSB is used, then the adder carry input can be enabled on the first clock cycle of a vector. This has the same effect as changing the initialization value to $1/2$.

10 The slope value to be added to one of the accumulators should be rounded off to the nearest LSB.

If K bits are available for the slope and for the accumulator fractions, then vector lengths of up to $2^{K/2}$ clock cycles can be generated with mathematical perfection. (For an Accumulator with an 11-bit fraction,
15 this allows mathematically perfect vectors of up to 45 steps).

Vectors that are longer than $2^{K/2}$ clock cycles, and therefore might not achieve mathematical perfection, can also be drawn. However, no vector longer than 2^K clock cycles should be drawn (where K is the number of fractional bits in the accumulator and in the rounded slope value). This
20 restriction is necessary in order to guarantee that no errors will occur in the integer portions of the accumulators at vector end points.

It is noted that "Perfect Vectors" illuminate only pixels that lie within half of the pixel-to-pixel spacing from the actual desired vector.

2. For Symmetrical Symbology

25 Symmetrical symbology from chained vectors can be generated by controlling which of two pixels is illuminated when both are equally distant from the desired vector. This is achieved by always initializing the

SUBSTITUTE SHEET

-44-

1 accumulator fractions to $1/2$ less 1 LSB and by enabling the carry input to an accumulator adder, or to both accumulator adders, on the first clock cycle of the vectors in certain octants.

5 Although symmetrical symbology can be achieved with either the SIN/COS or the tan vector generator algorithm, only the tan algorithm also results in the exclusive generation of mathematically perfect vectors. The basic resolution requirement for the accumulator fractions is, in fact, the same
10 for the generation of symmetrical symbology as for the generation of mathematically perfect vectors with the tan algorithm. However, a different algorithm must be used to derive the values that are added to the accumulators. Two such algorithms are available.

15 For accumulators having K fractional bits, symmetry can be realized for all vectors that are generated with a maximum of $2^{K/2}$ clock cycles if: 1) the values added to the accumulators are derived by rounding to the K th fractional bit for all vectors having an odd number of clock cycles; and 2)
20 for vectors having an even number of clock cycles, the value added to an accumulator having the carry input disabled on all clock cycles is truncated to K fractional bits; and 3) also for vectors having an even number of clock cycles, the value added to an accumulator having the carry input enabled on only the
25 first clock cycle of the vector is rounded up (i.e., by adding 1 to the K th fractional bit position) when not represented exactly within the K fractional bits. Note that the carry inputs to the accumulators on the first clock cycle of each vector are determined by the octant of the generated vector
30 according to one of the symmetry algorithms of Table 2 or Table 3.

 Again for accumulators having K fractional bits, symmetry can also be realized for all vectors that are generated with a maximum of $2^{(K-1)/2}$ clock cycles if: 1) the
35 value to be added to an accumulator having the

SUBSTITUTE SHEET

-45-

1 carry input disabled on all clock cycles is truncated to K fractional bits;
and if 2) the value added to an accumulator having the carry input enabled
on only the first clock cycle is rounded up when not represented exactly
within the available K fractional bits.

5 Neither of these algorithms, for deriving the values to be added to
the accumulators, would normally be used for vectors having more than the
maximum number of clock cycles for symmetry. The second algorithm can
result in errors in the integer bits of the accumulators, at vector end
points, for vectors having more than 2^{K-1} clock cycles.

10 While the invention has been described in its preferred
embodiments, it is to be understood that the words which have been used are
words of description rather than limitation and that changes may be made
within the purview of the appended claims without departing from the true
scope and spirit of the invention in its broader aspects.

SUBSTITUTE SHEET

-46-

CLAIMS

1 1. Vector generator apparatus for generating a vector having an
2 initial point and an end point comprising:

3 first accumulator means having a first integer portion and a first
4 fractional portion, said first integer portion providing a signal
5 representative of a first component of said vector,

6 second accumulator means having a second integer portion and a
7 second fractional portion, said second integer portion providing a signal
8 representative of a second component of said vector, and

9 first and second presetting means for presetting said first and
10 second fractional portions, respectively, to a non-zero value prior to
11 generating said vector so that no error exists in said first and second
12 integer portions at said end point of said vector.

1 2. The apparatus of Claim 1 wherein:

2 said first presetting means comprises means for presetting said
3 first fractional portion to a value selected from the group consisting of
4 the value of one-half and the value of one-half less the value of the least
5 significant bit of said first accumulator means, and

6 said second presetting means comprises means for presetting said
7 second fractional portion to a value selected from the group consisting of
8 the value of one-half and the value of one-half less the value of the least
9 significant bit of said second accumulator means.

SUBSTITUTE SHEET

-47-

1 3. The apparatus of Claim 1 wherein each said first and second
2 accumulator means has a carry input and said first and second presetting
3 means comprises:

4 means for presetting said first fractional portion to a value of
5 one-half less the value of the least significant bit of said first
6 accumulator means,

7 means for presetting said second fractional portion to a value of
8 one-half less the value of the least significant bit of said second
9 accumulator means,

10 first carry enabling means for selectively enabling said carry
11 input of said first accumulator means, and

12 second carry enabling means for selectively enabling said carry
13 input of said second accumulator means.

SUBSTITUTE SHEET

- 48 -

4. The apparatus of Claim 3 wherein:
 said vector is generated at an angle residing in one
of a plurality of angular sectors, and
 said first and second carry enabling means comprises
means for selectively enabling said carry inputs of said first
and second accumulator means in accordance with said angular
sector so that said vector generator apparatus generates
symmetrical symbology.

5. The apparatus of Claim 4 wherein said plurality of
angular sectors comprise eight octants.

6. The apparatus of Claim 2 further including:
 clock means having clock cycles for controlling
generating said vector,
 First incrementing means for providing a first
incremental value signal to said first accumulator means for
accumulation therein during each said clock cycle, and
 second incrementing means for providing a second
incremental value signal to said second accumulator means for
accumulation therein during each said clock cycle.

SUBSTITUTE SHEET

-49-

1 7. The apparatus of Claim 6 wherein:
2 said first and second fractional portions each comprise K bits,
3 said first and second incrementing means comprises means for
4 providing said first and second incremental value signals rounded to the
5 nearest least significant bit of said first and second accumulator means,
6 and
7 means for limiting said clock cycles utilized to generate said
8 vector to a maximum value selected from the group consisting of 2^K and $2^{K/2}$.

1 8. The apparatus of Claim 5 further including:
2 clock means having clock cycles for controlling generating said
3 vector,
4 first incrementing means for providing a first incremental value
5 signal to said first accumulator means for accumulation therein during each
6 said clock cycle, and
7 second incrementing means for providing a second incremental value
8 signal to said second accumulator means for accumulation therein during each
9 said clock cycle.

1 9. The apparatus of Claim 8 further including means for limiting said
2 clock cycles utilized to generate said vector to a maximum of $2^{K/2}$ clock
3 cycles and wherein said first and second fractional portions each comprise
4 K bits.

SUBSTITUTE SHEET

- 50 -

10. The apparatus of Claim 9 wherein
said number of clock cycles for generating said
vector comprises and even integer,
said first incrementing means comprises means for
providing said first incremental signal rounded up, when not
represented exactly in K fractional bits, whenever said first
carry enabling means is enabling said carry input of said first
accumulator means and for providing said first incremental
signal truncated whenever said first carry enabling means is
disabling said carry input of said first accumulator means, and
said second incrementing means comprises means for
providing said second incremental signal rounded up when not
represented exactly in K fractional bits, whenever said second
carry enabling means is enabling said carry input of said
second accumulator means and for providing said second
incremental signal truncated whenever said second carry
enabling means is disabling said carry input of said second
accumulator means.

SUBSTITUTE SHEET

1 11. The apparatus of Claim 9 wherein
2 said number of clock cycles for generating said vector comprises an
3 odd integer,
4 said first incrementing means comprises means for providing said
5 first incremental signal rounded to the nearest least significant bit of
6 said first accumulator means,
7 said second incrementing means comprises means for providing said
8 second incremental signal rounded to the nearest least significant bit of
9 said second accumulator means.

SUBSTITUTE SHEET

-52-

12. The apparatus of Claim 8 further including means for limiting said clock cycles utilized to generate said vector to a maximum of $2^{(K-1)/2}$ clock cycles and wherein

said first and second fractional portions each comprise K bits,

said first incrementing means comprises means for providing said first incremental signal rounded up, when not represented exactly in K fractional bits, whenever said first carry enabling means is enabling said carry input of said first accumulator means and for providing said first incremental signal truncated whenever said first carry enabling means is disabling said carry input of said first accumulator means, and

said second incrementing means comprises means for providing said second incremental signal rounded up, when not represented exactly in K fractional bits, whenever said second carry enabling means is enabling said carry input of said second accumulator means and for providing said second incremental signal truncated whenever said second carry enabling means is disabling said carry input of said second accumulator means.

SUBSTITUTE SHEET

1 13. The apparatus of Claim 5 wherein said first and second carry
2 enabling means comprises means for enabling said carry input of said first
3 accumulator means and said carry input of said second accumulator means in
4 accordance with an algorithm selected from the group consisting of algorithm
5 1 and algorithm 2 of Table 2.

1 14. The apparatus of Claim 5 wherein said first and second carry
2 enabling means comprises means for enabling said carry input of said first
3 accumulator means and said carry input of said second accumulator means in
4 accordance with an algorithm selected from the group consisting of
5 algorithm 1, algorithm 2, algorithm 3 and algorithm 4 of Table 3.

1 15. The apparatus of Claim 1 wherein said first and second presetting
2 means comprises means for presetting said first and second fractional
3 portions, respectively, to non-zero values prior to generating said vector
4 so as to bias initial accumulator values approximately midway between the
5 next larger integer value and the next smaller integer value.

1 16. The apparatus of Claim 15 further including first and second
2 integer presetting means for presetting said first and second integer
3 portions to the respective values that correspond to the initial point of a
4 vector that is to be generated.

-54-

1 17. Vector generator apparatus for generating a vector having an
2 initial point and an end point comprising: first accumulator means having a
3 first integer portion and a first fractional portion, said first integer
4 portion providing a signal representative of a first component of said
5 vector, second accumulator means having a second integer portion and a
6 second fractional portion, said second integer providing a signal
7 representative of a second component of said vector, first and second
8 integer presetting means for presetting said first and second integer
9 portions to the respective values that correspond to the initial point of a
10 vector that is to be generated, and first and second fractional presetting
11 means for presetting said first and second fractional portions,
12 respectively, to non-zero values prior to generating said vector so as to
13 bias initial accumulator values approximately midway between the next larger
14 integer value and the next smaller integer value.

1 18. The apparatus of Claim 17 wherein: said first fractional
2 presetting means comprises means for presetting said first fractional
3 portion to a value selected from the group consisting of the value of one-
4 half and the value of one-half less the value of the least significant bit
5 of said first accumulator means, and said second fractional presetting means
6 comprises means for presetting said second fractional portion to a value
7 selected from the group consisting of the value of one-half and the value of
8 one-half less the value of the least significant bit of said second
9 accumulator means.

SUBSTITUTE SHEET

-55-

1 19. The apparatus of Claim 17 or of Claim 18 wherein said first
2 and second accumulators incorporate, respectively, a first and second
3 accumulation means comprising: first addition means for setting the first
4 accumulator integer and fractional values to a value representative of the
5 sum of the present accumulator value and the value of an addend input to
6 said first accumulator; and second addition means for setting the second
7 accumulator integer and fractional values to a value representative of the
8 sum of the present accumulator value and the value of an addend input to
9 said second accumulator, and wherein each accumulator incorporates a set of
10 input control signal means for controlling the initial presetting means and
11 the accumulation means of each respective accumulator, and wherein each
12 vector generation operation is effected by a sequence of one or more
13 activation cycles of a clock signal means whereby each activation cycle in
14 said sequence results in a single accumulation operation by each accumulator
15 so that said generated vector consists of a sequence of integer accumulator
16 outputs whereby each set of accumulator outputs within said sequence
17 corresponds to the location of a point that is approximately on the desired
18 vector and whereby said sequence of outputs corresponds to a sequence of
19 points that lie approximately along the path of the desired vector.

SUBSTITUTE SHEET

-56-

1 20. The apparatus of Claim 19 wherein said first and second
2 accumulator means also incorporate an optional accumulation means
3 comprising: addition means for setting the integer portion of the first
4 accumulator value to the integer portion of a value representative of the
5 sum of the present accumulator value and the value of an addend input to
6 said accumulator, and the essentially concurrent means for setting the
7 fractional portion of the first accumulator value to a value selected from
8 the group consisting of the value of one-half and the value of one-half less
9 the value of the least significant bit of said first accumulator means; and
10 addition means for setting the integer portion of the second accumulator
11 value to the integer portion of a value representative of the sum of the
12 present accumulator value and the value of an addend input to said
13 accumulator, and the essentially concurrent means for setting the fractional
14 portion of the second accumulator value to a value selected from the group
15 consisting of the value of one-half and the value of one-half less the value
16 of the least significant bit of said second accumulator means.

SUBSTITUTE SHEET

-57-

21. A vector generator apparatus as in Claim 20 in which the accumulation, or the sequence of accumulations, that are required by each accumulator in the generation of a single vector are of the types described in Claim 19 or 20; and in which, for each generated vector, the last accumulation of each accumulator is always of the optional type described in Claim 20, whereas any other accumulations prior to said last accumulation are of the type described in Claim 19, and where furthermore: the number of fractional bits in each accumulator is large enough relative to the largest number of accumulations used in the generation of any single vector so that any build-up of accumulator errors that results from truncation or round-up errors in the addend inputs to the accumulators shall, at the end point of any vector, be limited to only the fractional portions of the sum values whose integer portions are used as inputs to the integer portions of the respective accumulators on the last accumulation cycle of any vector, so that the complete elimination of accumulator errors at the end points of all generated vectors is achieved, and furthermore, so that complex symbology can be generated by the concatenation of vectors and that said complex symbology generation is achieved without error accumulation and without a requirement for initializing the accumulators between vectors as a means of preventing error accumulation.

SUBSTITUTE SHEET

-58-

1 22. A vector generator apparatus as in Claim 20 that also
2 incorporates the means for generating vectors by either one of two types of
3 vector generator algorithms whereby one of these algorithm types is as
4 described in Claim 21 in which the last accumulation operation of each
5 accumulator for a given vector employs the optional accumulation technique
6 of Claim 20 and in which any other accumulations for a given vector employ
7 the accumulation technique of Claim 19, and whereby the second type of
8 vector generation algorithm employs accumulation operations that are
9 comprised exclusively of the accumulation technique that is described in
10 Claim 19.

SUBSTITUTE SHEET

-59-

1 23. The vector generator apparatus as in any of Claims 19 through
2 22 which also incorporates an optional addition means for each accumulator
3 means whereby each addition means also incorporates an input carry signal to
4 the least significant bit of the addition circuit means, and whereby the
5 output of said addition circuit when said carry input is disabled is given
6 by the sum of the present accumulator value and the value of an addend input
7 to said accumulator, where said sum is identical to that described for Claim
8 19 or 20, and whereby the output of said addition circuit for the optional
9 case of the carry input signal being enabled is given by the sum of: the
10 present accumulator value; the value of the addend input to the accumulator;
11 and the value of the carry input, where the carry input value is equal to
12 the value of the least significant fractional bit of the accumulator; and
13 where furthermore, the selection of the optional addition means described
14 herein, in which case the carry input signal is enabled, may be controlled
15 independently for each accumulator, and is independent from the selection of
16 an optional accumulation means as described in Claim 20, whereby the
17 fractional portion of an accumulator is set to either the value $1/2$ or to
18 the value $1/2$ less the value of the least significant bit of the accumulator
19 fraction, instead of set to the fractional portion of the output from the
20 addition circuit.

SUBSTITUTE SHEET

- 60 -

24. The vector generator apparatus of Claim 23 except that when the integer portion of an accumulator is preset to a value corresponding to the beginning point of a vector to be generated, as described in Claim 18, the fractional portion of said accumulator is preset to only the value of $1/2$ less the value of the least significant bit of the accumulator fraction; and except that when the optional accumulation technique described in Claim 20 is utilized to set the fractional portion of an accumulator on the last accumulation of a generated vector, then the fractional portion of said accumulator shall be set to only the value of $1/2$ less the least significant bit of the accumulator; and whereby each vector that is generated with said apparatus is generated at an angle that resides in one of a plurality of angular sectors; and whereby the input carry signal for each accumulator of said apparatus is enabled on only the first accumulation cycle of selected vectors, where the state of the carry input signal on the first accumulation cycle is, for each accumulator, a function of the angular sector of the vector to be generated, and whereby said functions are defined according to one of a number of algorithms that are designed to force or to favor the generation of a symmetrical symbology when multiple vectors are concatenated for the purpose of generating complex symbols.

SUBSTITUTE SHEET

- 61 -

1 25. The vector generator apparatus of Claim 17 or
Claim 18 that also incorporates a third accumulator apparatus
of the same type as the other two accumulators that are
described in the corresponding claims and whereby the integer
5 portion of said third accumulator provides a third output that,
along with the integer outputs from the other two accumulators,
provide for the generation of three-dimensional vectors.

 26. The apparatus of Claim 24 wherein said
plurality of angular sectors comprises eight octants, and
10 wherein the carry input to the accumulators on the first
accumulation clock cycle of a generated vector are determined
by the octant of the vector according to either; algorithm #1;
algorithm #2; algorithm #3; or algorithm #4, all of Table 3,
where the octants are defined as in Figure 4.

15 27. The apparatus of Claim 24 wherein said
plurality of angular sectors comprises eight octants, wherein
the addend input to the accumulator associated with the axis of
the longest vector component is given by either the value +1 or
the value -1, and wherein the carry inputs to the accumulators
20 on the first accumulation clock cycle of a generated vector are
determined by the octant of the vector according to either
algorithm #1 or algorithm #2 of Table 2, where the octants are
defined as in Figure 4.

SUBSTITUTE SHEET

-62-

1 28. The vector generator apparatus of Claim 17 or Claim 18
wherein the value of the addend input to the specific accumulator that
corresponds to the axis of the longest component of a vector that is to be
generated is restricted to a value of either +1 or -1, and whereby the
5 integer portion of the addition means for each accumulator is thereby
simplified to an increment or a decrement means, and whereby the fractional
accumulator and fractional addition means of the two accumulators in
each of Claims 17 through 22 is replaced by a single fractional accumulator
that incorporates addition means whereby the range of possible values of
10 addend inputs to said single fractional accumulator includes the values of
+1 and -1 and whereby said single fractional accumulator incorporates an
overflow detection means for designating a carry output condition or a
borrow output condition from the fractional portion of said addition means
and where furthermore: a given vector is generated by a sequence of one or
15 more accumulation clock cycles in which the integer accumulator that
corresponds to the axis of the longest vector component is either
incremented or decremented, according to the orientation of the given
vector, on each of said accumulation clock cycles, and in which, for a given
vector, the second integer accumulator is either incremented or decremented
20 on only those corresponding accumulation clock cycles in which the
fractional addition means designates, respectively, a carry output condition
or a borrow output condition, and whereby the single fractional accumulator
is operated, in regards to accumulation operations and setting and
presetting operations, in an identical manner as is prescribed in each of
25 Claims 17 through 22 for the two fractional accumulators of Claims 17
through 22.

SUBSTITUTE SHEET

-63-

1 29. A vector generator apparatus as in Claim 28 whereby the single
2 fractional accumulator is set, on the last accumulation clock cycle used in
3 the generation of any given vector, to the value of $1/2$ less the value of
4 the least significant bit of the fractional accumulator, and also set to
5 this same value whenever the integer accumulators are set to values that
6 correspond to the initial point of a vector that is to be generated, and
7 whereby the addition means of said fractional accumulator incorporates a
8 carry input signal such that when said carry input signal is disabled, the
9 output of said addition means is equal to the sum of the present value of
10 the fractional accumulator and the value of the addend input to the
11 accumulator, and when said carry input is enabled the output of the addition
12 means is equal to the sum of: the present value of the fractional
13 accumulator; the addend input; and the carry input, where the carry input is
14 equal to the value of the least significant bit of the fractional
15 accumulator, and where furthermore: each vector that is generated with said
16 apparatus is generated at an angle that resides in one of a plurality of
17 angular sectors, and where the input carry signal is enabled on only the
18 first accumulation cycle of selected vectors, where the state of the carry
19 input signal on the first accumulation is a function of the angular sector
20 of the vector to be generated, and whereby said function is defined
21 according to one of a number of algorithms that are designed to force or to
22 favor the generation of symmetrical symbology when multiple vectors are
23 concatenated for the purpose of generating complex symbols.

SUBSTITUTE SHEET

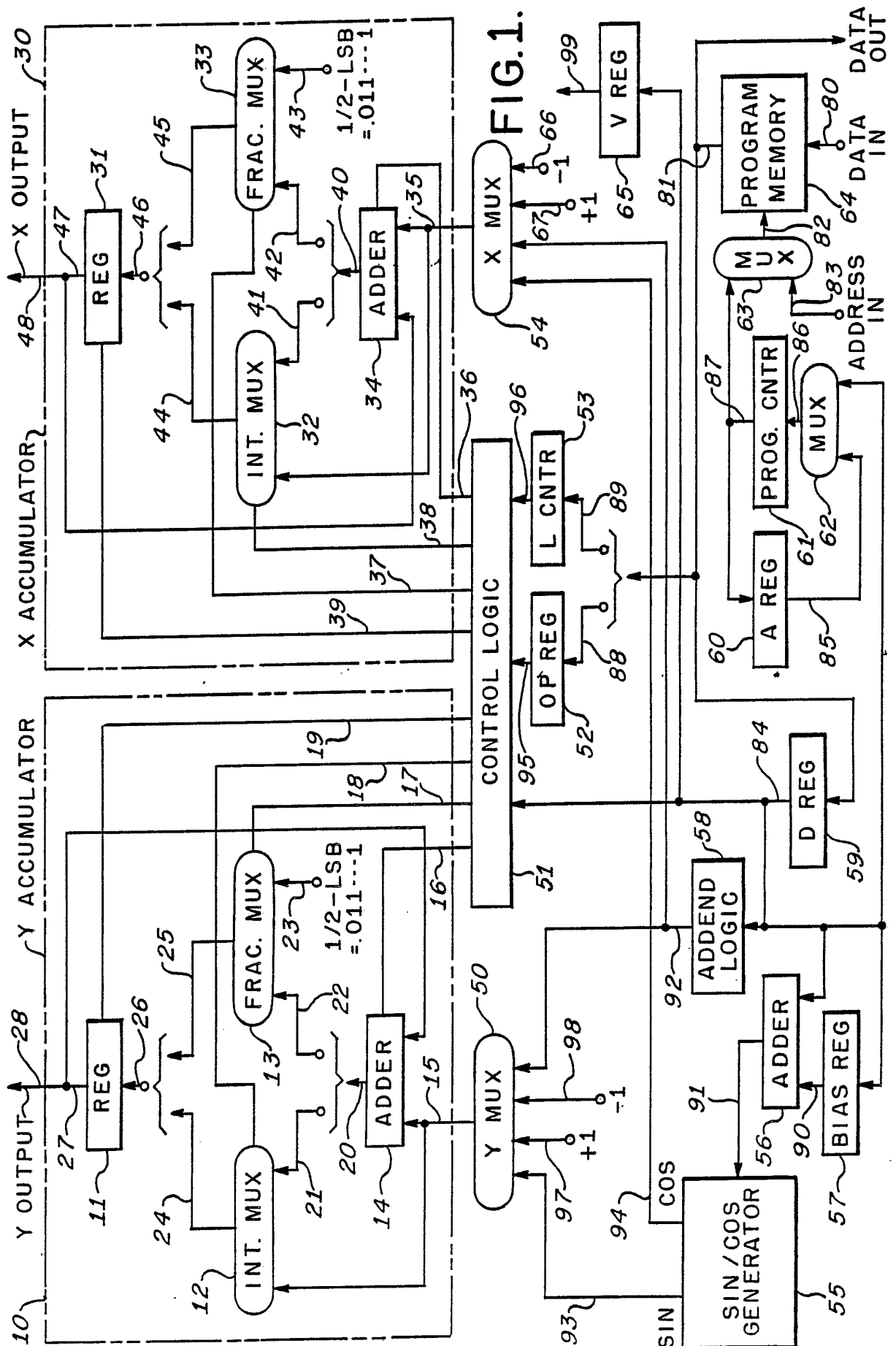


FIG. 1.

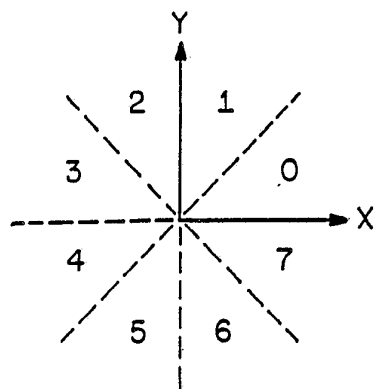
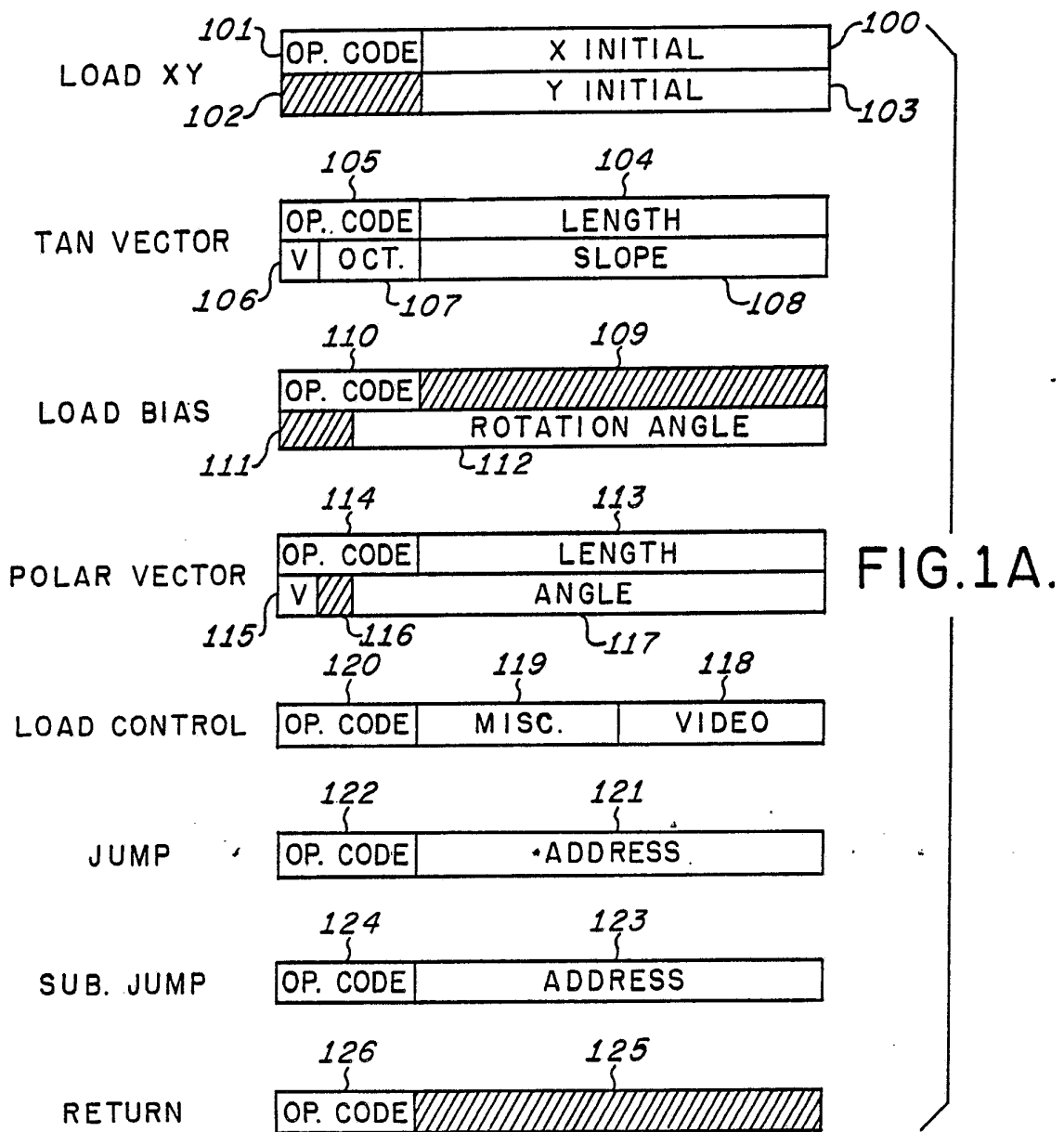


FIG. 4.

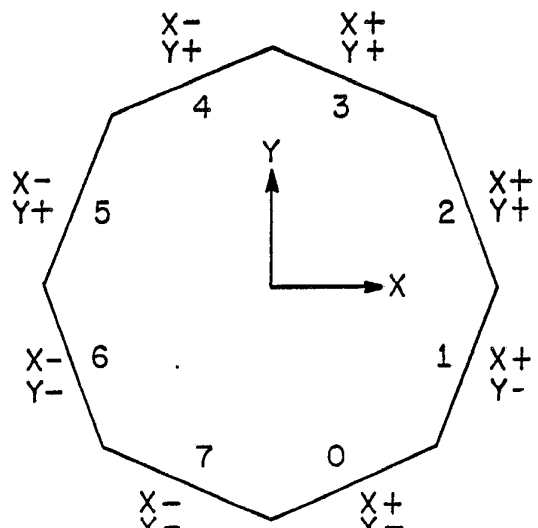


FIG. 6.

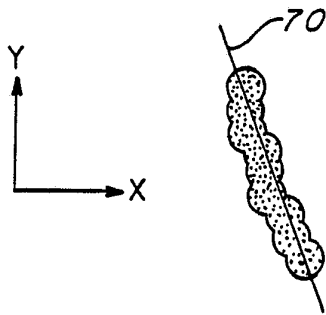


FIG. 2A.
PRIOR ART

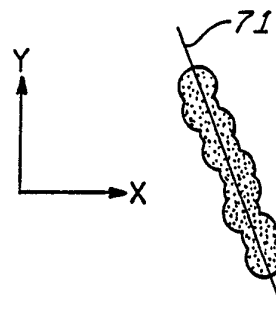


FIG. 2B.

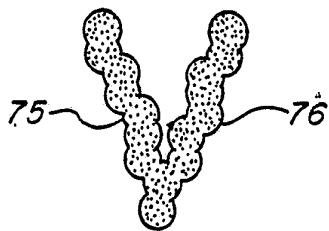


FIG. 3A.

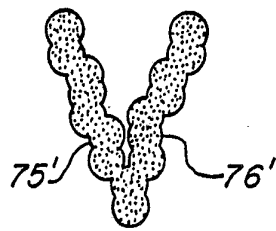


FIG. 3B.

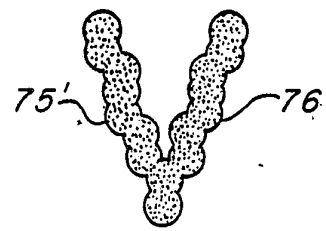


FIG. 3C.

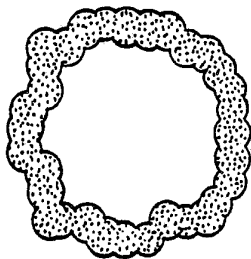


FIG. 5A.

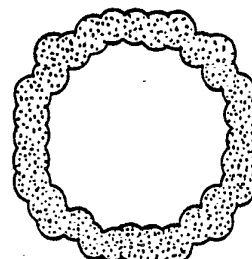


FIG. 5B.

INTERNATIONAL SEARCH REPORT

International Application No PCT/US 89/04911

I. CLASSIFICATION OF SUBJECT MATTER (if several classification symbols apply, indicate all) ⁶		
According to International Patent Classification (IPC) or to both National Classification and IPC		
IPC ⁵ : G 09 G 1/10, G 09 G 1/14, G 06 F 15/72		
II. FIELDS SEARCHED		
Minimum Documentation Searched ⁷		
Classification System ¹	Classification Symbols	
IPC ⁵	G 09 G, G 06 G	
Documentation Searched other than Minimum Documentation to the Extent that such Documents are Included in the Fields Searched ⁸		
III. DOCUMENTS CONSIDERED TO BE RELEVANT ⁹		
Category ¹⁰	Citation of Document, ¹¹ with indication, where appropriate, of the relevant passages ¹²	Relevant to Claim No. ¹³
A	US, A, 4481605 (CHASE) 6 November 1984, see column 3, line 64 - column 6, line 22 (cited in the application)	1,17
	--	
A	WO, A1, 82/02441 (WESTERN ELECTRIC CO. INC.) 22 July 1982, see page 8, line 22 - page 13, line 20	1,17
	--	
A	US, A, 4027148 (ROSENTHAL) 31 May 1977, see claim 1; column 4, line 31 - column 6, line 43	1,17

<p>¹⁰ Special categories of cited documents:</p> <p>"A" document defining the general state of the art which is not considered to be of particular relevance</p> <p>"E" earlier document but published on or after the international filing date</p> <p>"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)</p> <p>"O" document referring to an oral disclosure, use, exhibition or other means</p> <p>"P" document published prior to the international filing date but later than the priority date claimed</p> <p>"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention</p> <p>"X" document of particular relevance: the claimed invention cannot be considered novel or cannot be considered to involve an inventive step</p> <p>"Y" document of particular relevance: the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.</p> <p>"&" document member of the same patent family</p>		
IV. CERTIFICATION		
Date of the Actual Completion of the International Search	Date of Mailing of this International Search Report	
28th March 1990	23. 04. 90	
International Searching Authority	Signature of Authorized Officer	
EUROPEAN PATENT OFFICE	M. Peis	M. PEIS

**ANNEX TO THE INTERNATIONAL SEARCH REPORT
ON INTERNATIONAL PATENT APPLICATION NO.**

US 8904911
SA 32483

This annex lists the patent family members relating to the patent documents cited in the above-mentioned international search report. The members are as contained in the European Patent Office EDP file on 11/04/90. The European Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US-A- 4481605	06-11-84	EP-A, B 0088568	14-09-83
		JP-A- 58153995	13-09-83
WO-A- 8202441	22-07-82	US-A- 4365305	21-12-82
		CA-A- 1169983	26-06-84
		EP-A, B 0068013	05-01-83
		GB-A, B 2090709	14-07-82
US-A- 4027148	31-05-77	None	

EPO FORM P4179

For more details about this annex : see Official Journal of the European Patent Office, No. 12/82