X-521-67-271

NASA TM X- 55828

SOME VECTOR GENERATING TECHNIQUES FOR CATHODE RAY TUBE (CRT) DISPLAY SYSTEMS

BY

HERBERT R. DURBECK

JUNE 1967

NASA

GODDARD SPACE FLIGHT CENTER

GREENBELT, MARYLAND

X-521-67-271

SOME VECTOR GENERATING TECHNIQUES FOR CATHODE RAY TUBE (CRT) DISPLAY SYSTEMS

Herbert R. Durbeck

June 1967

GODDARD SPACE FLIGHT CENTER
Greenbelt, Maryland

CONTENTS

SOME VECTOR GENERATING TECHNIQUES FOR CATHODE RAY TUBE (CRT) DISPLAY SYSTEMS

INTRODUCTION

There exists at present a widespread use of Cathode Ray Tube (CRT) Systems for the display of computer generated data. Many of the older CRT display systems and some of the lower priced newer systems contain character generators for the production of alphanumeric characters, but have no vector generators for drawing graphic information. If at some time graphic display capability is required, a decision must be made as to whether the needed vector generator is to be purchased or developed in-house. There is a definite lack of information about vector generating techniques upon which to base this decision. This is also the case when buying a new CRT display system with an integral vector generator and trying to decide which system offers the most suitable graphic capability.

The purpose of this report is to provide a general discussion of vector generating techniques and to describe two straight-forward line drawing schemes for CRT display systems.

GENERAL DISCUSSION

This report will be confined to considering vector generating procedures for non-scan CRT displays, which constitute the majority of CRT display systems available today. In these systems electrical signals are supplied to horizontal (X) and vertical (Y) deflection circuits and the electron beam is moved to a position on the CRT screen, determined by the magnitude of these signals. Normally, digital X,Y information is generated by the computer to which the display system is attached, or by internal logic circuitry, and passed through D/A converters, amplifier and compensation circuitry to the deflection system of the CRT. The electron beam is moved to this position and a symbol, specified by an input code from the attached computer, is drawn by the character generator.

In order to produce a line segment on the CRT screen, the initial X, Y coordinates and the terminal X,Y coordinates must be presented to a vector generator by the attached computer. The electron beam is then moved from the initial to the terminal position in such a manner as to produce a straight line segment.

Alternately, the initial X,Y coordinates and the horizontal (Δx) and vertical (ΔY) increments can be presented to the vector generator with the same effect. The vector generator can produce line segments in two different ways. It can generate a series of points between the initial and final coordinates of the vector, the incremental point positioning method, or it can continuously provide the deflection signal required to move the electron beam smoothly in a straight line path from the initial to the final location, the stroke method.

INCREMENTAL POINT POSITIONING VS STROKE METHOD

The incremental point positioning method has the following advantages: complete compatibility with the original display system requiring no modification of the display system analog circuitry, accuracy and inherent stability, complete endpoint matching for connected vectors.

Furthermore, the intensity compensation that is required for some stroke type vector generators, when drawing vectors of different lengths, is not needed when using the incremental point positioning method, since all vectors, regardless of length, have equal brightness.

The incremental point positioning method has, however, the following disadvantages: line segments are a collection of points and not continuous, the time required for vector generation is high, there is a noticeable departure from a straight line for some of the simpler point generating techniques, there is little flexibility for adaptation to more complex curve generation (circular, parabolic, hyperbolic, etc.).

The quality of the vectors generated by the incremental point positioning method depends primarily on the point generating algorithm, the spot size of the electron beam, and the raster matrix used for point positioning. By the raster matrix we mean here the matrix of individually addressable discrete points, formed on the face of the CRT. Roughly, if the spot diameter is equal to or larger than the distance between adjacent raster points, good line continuity is achieved by a vector generator that plots points in near adjacent locations in generating a line. Of course, the more raster points that are addressable the better a straight line segment can be approximated, but on the other hand, the more points must be produced to construct a given vector. Most CRT display systems have a 1024 by 1024 raster matrix.

The principal reason for having a vector generator of the incremental point positioning type, rather than none at all, is to remove the necessity of storing

all the intermediate points of a given line segment in the display computer, or the refresh buffer memory. No significant writing speed advantage is achieved by this method, except when the display computer or buffer refresh memory have a very slow memory cycle time.

AN INCREMENTAL POINT POSITIONING METHOD FOR VECTOR GENERATION

In this section one scheme of vector generation, using the incremental point positioning method will be presented. For the purposes of this discussion let us assume that we are given a typical display system having the following characteristics:

Random Positioning time of the beam on the CRT face	15 μsec
Time Required to generate an adjacent point	1 μsec
CRT display area	10" x 10"
Raster matrix	1024 x 1024
Display refresh rate	30 times/second

In order to keep the vector drawing time within reasonable limits let us assume that the vector generator will produce up to 64 points per given increments ΔX and ΔY. This corresponds to 6 bits and one sign bit in the ΔX and ΔY registers, respectively, and a vector write time of 64 μsec plus positioning time. For this system up to approximately 333 inches of connected vectors can be generated per flicker free frame.*

The basic circuit requirements for this vector generator are:

(a) a seven bit ΔX register

(b) a seven bit ΔY register

(c) a 1 megahertz clock

---

*This approximation is obtained by noting that for the system specified, the distance between adjacent raster points is approximately 0.01 inches. Since points are generated at a rate of one per microsecond for one thirtieth of a second (then the display must be refreshed), a total of $\frac{.0333 \text{ sec}}{1 \times 10^{-6} \text{ sec}} = 3.33 \times 10^4$ points can be generated. This corresponds to 333 inches of line display.

(d) a six stage counter

(e) logic gates required to implement the point generating algorithm (this will be described in more detail in Appendix A)

(f) The ten bit X and Y registers (normally found in the display system) which specify the instantaneous location of the electron beam will be replaced by two ten stage up/down counters with parallel input. This will not affect the normal operation of the display system.

The functional diagram of the vector generator is given in Figure 1.

The display computer loads the X and Y up/down counters (by parallel input), and the ΔX and ΔY registers. The coordinates of the starting point of the

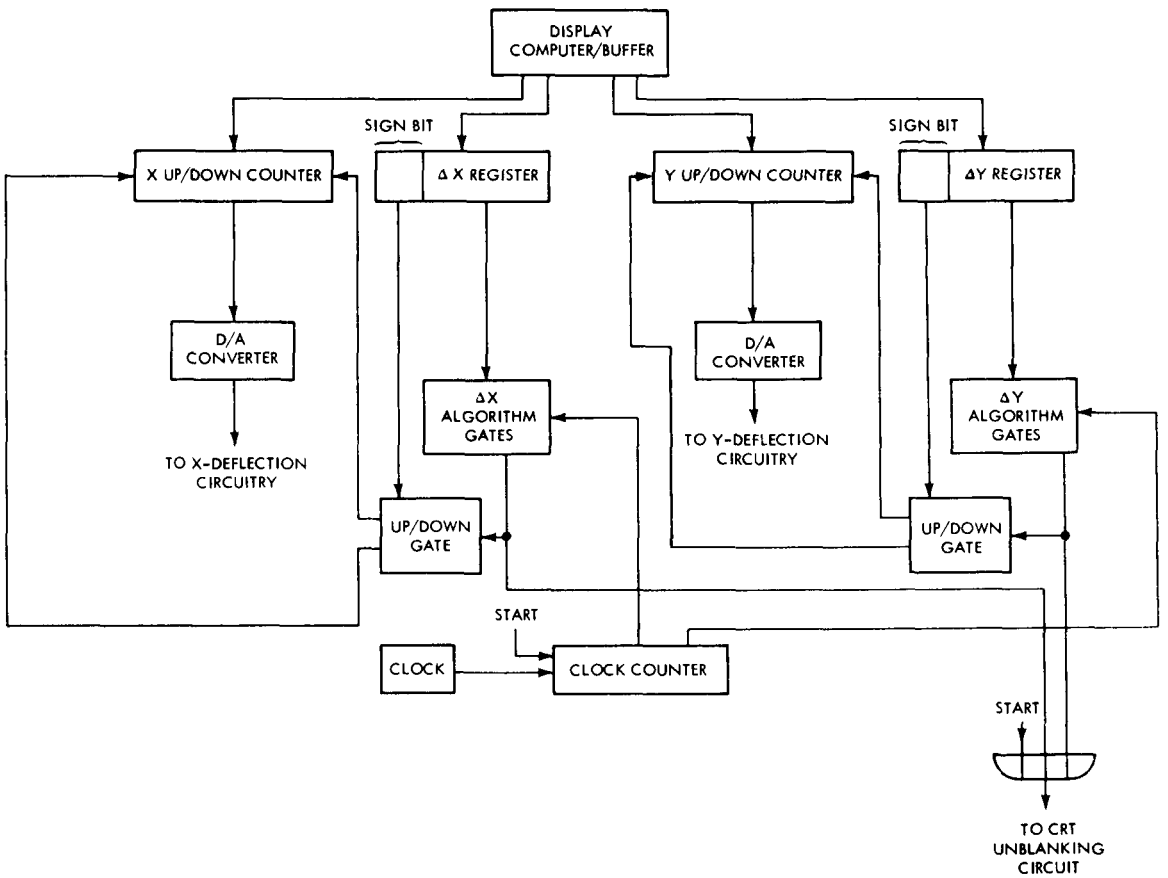

Figure 1.

vector are stored in the X and Y up/down counters, respectively, and the X and Y increments are stored in the ΔX and ΔY registers, respectively. The internal logic of the display system will generate a signal to start the clock counter sequence. The successive states of the clock counter are gated with the ΔX and ΔY registers through the ΔX and ΔY Algorithm Gates and the resultant pulses are either added to or subtracted from the numbers stored in the X and Y up/down counters, depending on the state of the sign bit in the ΔX and ΔY registers. Each time a pulse passes through either the ΔX or ΔY Algorithm Gates a signal is routed to the CRT unblanking circuitry and a point is generated on the face of the CRT. The START pulse is also passed on to the unblanking circuits and is used to produce a point at the starting position of the vector. By this procedure a set of points is generated which will closely approximate the desired vector. The comparison between the actual vector and the generated set of points will be made in Appendix A, where the Algorithm Gates are discussed. The gating for the START pulse, the STOP and RESET of the clock counter at the end of the 64th count is not shown, but implied in Figure 1.

At the end of the 64th count the clock counter is reset and clock pulses to the counter are inhibited. The display computer can now load the X and Y counters and ΔX and ΔY registers again and a new vector generating cycle can begin. For connected vectors only the ΔX and ΔY registers will be loaded by the computer, since the electron beam is already at the starting point of the vector.

The main limitation of this method is the time required to generate a given vector. Other considerations are the fact that for the system just described the time required to draw a short and long vector is constant (64 μsec + positioning time) and that the maximum vector length for one set of computer instructions is only one-sixteenth of the full diagonal distance on the CRT face. These last two limitations can be overcome by additional logic circuitry as will be shown in Appendix B. But it is unlikely that the time required to plot adjacent points can be reduced much below 1 μsec; so that little can be done to lessen the total vector generating time. When the requirement for endpoint accuracy, stability, and compatibility with the original display system are the over-riding considerations, a longer vector generating time might be tolerable (especially if only a small amount of graphic information has to be displayed at one time). The extreme accuracy and stability of this method for vector generation makes its use desirable in computer aided design systems.

GENERAL DISCUSSION OF STROKE TYPE VECTOR GENERATORS

The two major limitations of the incremental point positioning technique, excessive vector drawing time and lack of continuity of the generated vectors,

can be eliminated with the stroke method of line generation. Here a continuous signal is sent to the deflection circuitry of the CRT by the vector generator and the electron beam is moved smoothly from the initial to the terminal point.

There are basically two different types of deflection systems available for CRT's: electrostatic and electromagnetic.*

The characteristics of both types of deflection systems are summarized below:†

Characteristic	Electromagnetic	Electrostatic
Beam focus	Well focused throughout tube area	Beam defocused at tube edges
Internal Structure	Simple	Complex
Tube Length	Short	Long
Spot brightness	High	Moderate
Deflection Sensitivity	High	Low
Complexity of Amplifiers	High	Low
Deflection Speed	Low	High
Weight	Heavy	Light

Since electromagnetic deflection offers the advantages of a well defined beam over the entire tube face, high spot brightness, and high deflection sensitivity most present day display systems employ this mode of deflection.

For both types of systems, however, the shapes of the signals that must be produced by the vector generators are very similar. The difference between the two systems are found primarily in the different types of amplifiers that are required to drive the deflection circuits.

---

*There is also a hybrid electrostatic-electromagnetic system where the electromagnetic portion is used for large deflections and the electrostatic for small deflections. In this case the electrostatic deflection is utilized to generate alphanumeric symbols at locations specified by the large scale electromagnetic deflection. The electromagnetic deflection is used for vector generation.

†Charles W. Adams Associates, Inc., Computer Display Review, II. 11.0

For electrostatic deflection*

$$D = K_1 \frac{V_1}{V}$$

where, D = distance the beam is deflected from the center of the CRT face by the deflection voltage V_1.

V_1 = potential difference between deflection plates

V = accelerating voltage of the electron beam

K_1 = constant factor that is determined by the tube construction

For electromagnetic deflection†

$$D = K_2 \frac{H}{\sqrt{V}}$$

where D = distance the beam is deflected from the CRT face center by the magnetic field H

H = magnitude of the magnetic field at the center of the deflection coil

V = accelerating voltage of the electron beam

K_2 = constant factor that is determined by the tube construction

But since for a current carrying coil (containing no ferromagnet) the magnetic field in the interior is approximately proportional to this current.‡

$$H = K'' i$$

and,

$$D = \frac{K_3 i}{\sqrt{V}}$$

where i = current through the coil

K_3 = tube parameter

K'' = deflection coil parameter

*Theodore Soller, Merle A. Starr, George E. Valley, Jr., Editors, MIT Radiation Laboratory Series, Cathode Ray Tube Display, McGraw Hill Book Company, 1948, p. 64

†Ibid, p. 304

‡Ibid, p. 3

Thus it can be seen that in the electrostatic case the deflection of the electron beam is proportional to the potential difference between the deflection plates and in the electromagnetic case the deflection is proportional to the current in the deflection coil. The deflection speed of the beam is limited primarily by the capacitative time constant of the deflection plates in the electrostatic case and by the inductive time constant of the deflection coil in the electromagnetic case.

The function of the vector generator is to present linearly changing electrical signals to the X and Y deflection circuitry of the CRT, such that the amplitudes of the signals are proportional to ΔX and ΔY respectively.

There are two ways in which the vector generator can do this. It can produce deflection signals which will cause the electron beam to go from the initial to the final point in a constant time interval, regardless of vector length – constant time vector generator. Or it can produce deflection signals which will cause the electron beam to travel from the initial to the final point with a fixed velocity – constant velocity vector generator.

The constant velocity method is preferable for two reasons. First, since the beam travels with a uniform speed over the face of the CRT, all vectors, regardless of length, will have the same brightness. Second, since most graphic information consists of a large collection of short length vectors, or combinations at short and long vectors, a more efficient use of the display system is made. In contrast, for the constant time vector generating scheme it takes as long to draw short vectors as it does long vectors, and the vector drawing time must be chosen to be of such a magnitude so as to allow the generation of the longest vector in the fixed time interval. Furthermore, since both long and short vectors are drawn in the same time period, the short vectors will be generated more slowly across the CRT than the long vectors and thus appear brighter, unless intensity compensation circuitry is provided.

However, it appears that constant time vector generators of good quality are easier to implement than the corresponding constant velocity generators.

For constant velocity vector generation the following vector generating procedure is followed (after the coordinates of the starting point and ΔX and ΔY have been loaded into the appropriate registers of the display system by the display computer):

1. Compute the total vector length "L"

$$L = \sqrt{(\Delta X)^2 + (\Delta Y)^2}$$

2. Compute the time "T" required to draw the given vector

   $T = L/S$

   where S = constant vector speed

3. Compute the X and Y components of velocity "S_1" and "S_2"

   where $S_1 = \Delta X/T$

   $S_2 = \Delta Y/T$

4. Generate a sawtooth signal for the X-deflection circuitry whose slope is "S_1", for a time interval "T".

5. Simultaneously generate a sawtooth signal for the Y-deflection circuitry whose slope is "S_2" for a time interval "T".

6. At the same time unblank the CRT beam for a time interval "T".

Even though the required calculations can be performed by analog, digital, and/or approximational techniques, all of the above functional steps must be carried out by the constant velocity vector generator.

On the other hand for the constant time vector generator the electron beam unblanking time is usually fixed. In that time interval sawtooth waveforms, whose slopes depend on the values found in the ΔX and ΔY registers, are presented to the X and Y deflection circuitry of the CRT. Various methods are used to accomplish this, from the charging of delay lines to the use of operational amplifiers as high quality integrators. Perhaps the simplest, but not necessarily best, method is to convert the ΔX and ΔY values (by D/A converters) to two square wavepulses whose amplitudes are proportional to the magnitudes of the numbers stored in the ΔX and ΔY registers respectively. Then these pulses are presented to the X and Y deflection circuits and the triangular waveforms (whose forward slopes are used to deflect the CRT beam) are generated by the integration of the applied pulses through the inherent impedances of the deflection circuits. The main problem in this case is the lack of control over the integrating time constants, since these are determined by the structure of the deflection circuits. Also, unless the X and Y deflection circuit time constants are identical, amplitude compensation circuits must be provided after the D/A conversion of ΔX and ΔY. Normally, vectors generated by this method are excessively distorted.

A constant time method of vector generation that provides control of the integrating time constant and yet is fairly simple to implement will be discussed in a section below.

The requirement of intensity compensation for constant time vector generators generally involves the determination of the vector length, since the intensity modulation of the CRT beam is proportional to this length.

Mathematically,

$$I = kL$$

where, I = intensity modulation required

L = length of the generated vector

k = parameter that depends on the CRT characteristics

As before,

$$L = \sqrt{(\Delta X)^2 + (\Delta Y)^2}$$

For the purposes of intensity compensation the following crude assumption can be made,

$$L \approx \Delta X + \Delta Y$$

So that,

$$I \approx k\,(\Delta X + \Delta Y)$$

This approximation will not provide perfect intensity compensation, but will considerably facilitate implementation. One method for intensity compensation is then, clearly, to pass the magnitudes of ΔX and ΔY through D/A converters, sum the result, and use it to modulate the amplifier that controls the intensity of the CRT beam.

At present the majority of the commerically available vector generators is of the constant time stroke variety. However, this is only a slim majority and, as component prices drop and the demand for display system increases, the constant velocity vector generators are becoming more and more popular.

A STROKE TYPE VECTOR GENERATOR

In this section a fairly simple and easy to implement approach to constant time stroke type vector generation will be presented.

Basically, this method consists of producing sawtooth waveforms whose slopes depend on the magnitude of the numbers stored in the ΔX and ΔY registers by means of two digitally controlled sawtooth generators (DCSG's) (the design of these DCSG's will be discussed in Appendix C). The resultant sawtooth waveforms are either inverted or not, depending on the values of the sign bits in the ΔX and ΔY registers, and then passed on to the X and Y deflection circuitry of the CRT. A block diagram of this system is given in Figure 2.

The pulse generator will produce a timing pulse of fixed duration, starting at the beginning of the vector drawing sequence. In this time interval, the

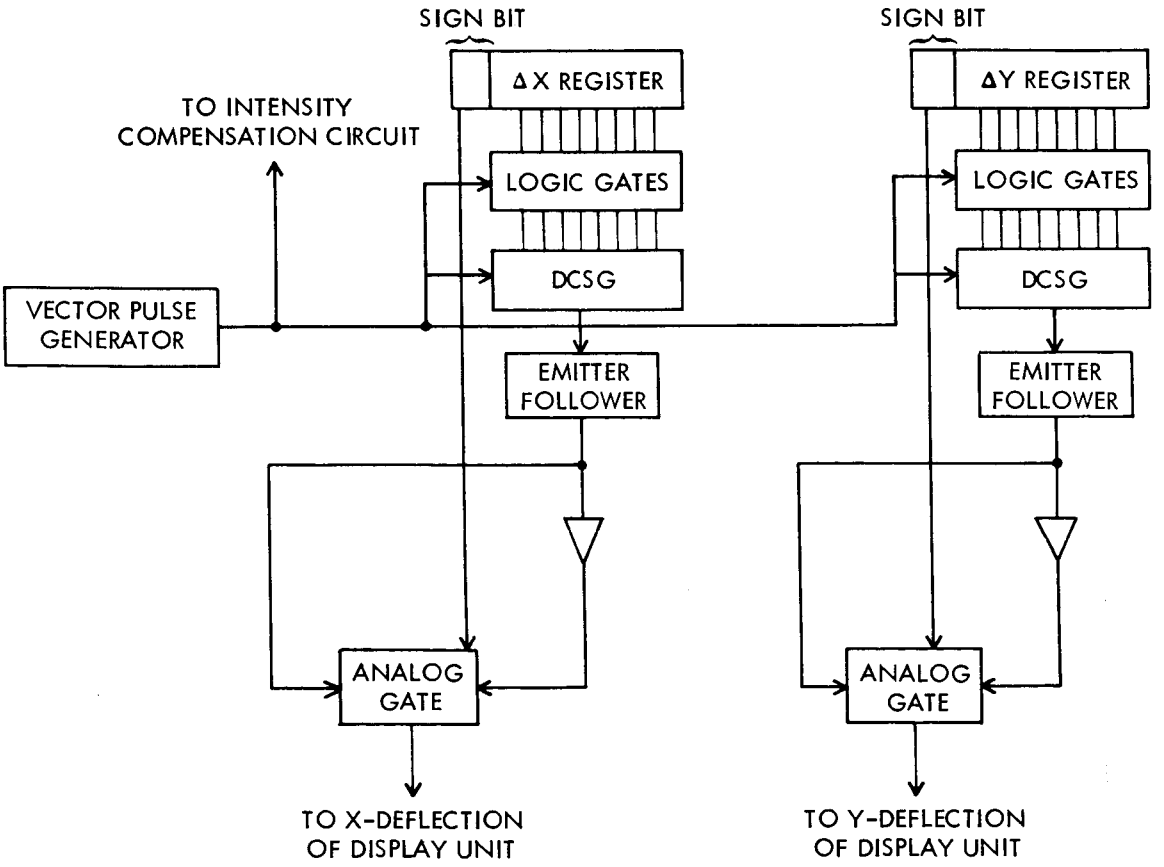

Figure 2.

variable slope ΔX and ΔY waveforms are produced and the CRT beam is unblanked. The time interval, which is essentially limited by the electrical inertia of the deflection circuitry, must be of sufficient length to allow the CRT beam to be deflected through the maximum vector increment.

Each analog gate in Figure 2 is really a combination of 2 analog gates, both controlled by the logical state of the respective sign bit, allowing either the sawtooth or its inverse to be transmitted to the CRT deflection circuitry.

The logic gates indicated in Figure 2 were implemented (see Appendix C) using Packard Bell Transistor Logic Cards. If digital logic cards, operating between different voltage levels are used, then the base biasing resistors of Figure C-1 will have to be adjusted accordingly.

The inverters shown in the functional diagram of the system (Figure 2) should be high quality DC amplifiers having as small a DC offset as possible. This can be achieved with a good grade operational amplifier, whose feedback loop is adjusted to provide a gain of unity.*

The analog gates required to switch either the sawtooth or its inverse into the deflection system of the CRT can be implemented using Diode Bridge, transistor, or FET switching methods given in Johnson** or Korn and Korn†. Good quality analog switches are also commercially available at reasonable prices††.

The intensity compensation required can be of the simple summation type mentioned at the end of the last section and is shown in Figure 3. As can be seen the modulation of the electron beam is only present in the vector drawing time interval as specified by the vector pulse generator.

This method offers the advantage of being able to display continuous vectors at high rates (being able to display a large amount of graphic information at one time). It can be readily applied in man/machine command and control consoles, where overall status is presented graphically and parameters requiring a high degree of accuracy are presented numerically, or in time shared remote consoles for the handling of mathematical problems, requiring the analysis of complex curves.

---

*Burr-Brown Research Corporation, Handbook of Operational Amplifier Applications, pp. 8-10

**T. E. Johnson, Analog Generator for Real Time Display of Curves, MIT Lincoln Laboratory, Technical Report 398, 28 July 1965

†Granine A. Korn and Theresa M. Korn, Electronic Analog and Hybrid Computers, McGraw Hill Book Company, 1964, pp. 219-233

††Philco, System 4-D Digital System Building Blocks Applications Manual, SU-02, SU-04, SU-06, SU-03 analog Switch units specification sheets

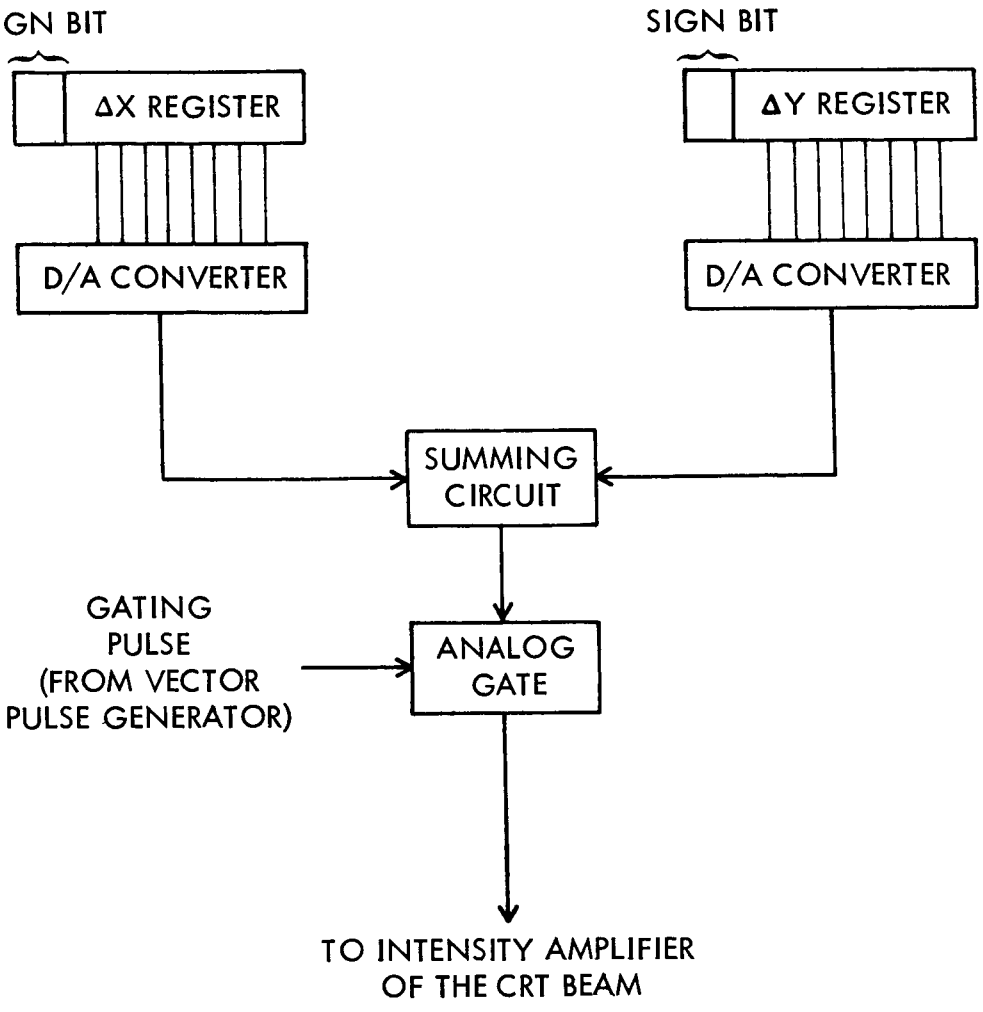

Figure 3.

CONCLUSION

It can be seen that vector generators, either of the incremental point positioning or stroke type, depending on the user's requirements can be readily constructed. The incremental point positioning type vector generator can be implemented using any standard commercial digital logic line, requiring a minimum amount of system development.

The stroke type vector generator, while probably needing less hardware requires more development, because of its hybrid nature, but has the potential of providing more effective utilization of the graphic capability of the CRT.

APPENDIX A

Algorithm Gates for the Seven Bit Incremental Point Positioning Vector Generator

As was mentioned, when the functional diagram of this vector generator (Figure 1) was discussed, the ΔX and ΔY Algorithm Gates basically extract point generating pulses from the clock counter states, depending on the values stored in the ΔX and ΔY registers.

Since the operation of the ΔX and ΔY Algorithm Gates is identical, only the ΔX Algorithm Gates will be described. The gating action will take place between the ΔX register and the clock counter, and the resultant point generating pulses are passed on to the X up/down counter.

Let A = X up/down counter

B = ΔX register

C = clock counter

Let the letters with subscripts indicate the condition of a bit stage of a given counter or register. Thus A_0 represents the first stage of the X up/down counter. The convention followed here is that the largest digit is stored in the lowest numbered stage. Thus A_1 stores a digit that is twice as large as an equivalent digit in A_2, and A_9 is least significant digit of the A up/down counter.

A Up/Down Counter	A_0	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9
B Register	B_0	B_1	B_2	B_3	B_4	B_5	B_6			
C Counter	C_0	C_1	C_2	C_3	C_4	C_5				

In the B register B_0 will represent the sign of ΔX. Let "0" represent a "+" sign and "1" represent a "-" sign, where "+" causes the X up/down counter "A" to be set to count up and the "-" sign causes the X up/down counter A to be set to count down.

Basically, the higher digits of the ΔX register are logically "anded" with the lower digits of the clock counter and the resultant pulses are either added

to or subtracted from the X up/down counter, depending on the value of the sign bit in ΔX. For example, if the ΔX register consists of all "1's" then every clock counter pulse will pass through the Algorithm Gates and produce a point on the CRT. If B_1 is "1" and the rest of the digits are all zero, then every other counter pulse is passed through the Algorithm gates and 32 points are produced.

If any of the more significant bits in the ΔX register are "0's" followed by "1's" then we say that there exists a "gap" at the location of the first less significant "1".

Example, 000110

Gap position

In this case all the counter bits up to the corresponding gap are "anded" with zeros and pulses are passed on to the X up/down counter, if an "and/or" condition exists for all consecutive "1's" (in the B register) following the gap.

In this way the correct total number of pulses are passed on to the X up/down counter (as will be illustrated by some examples in this appendix), producing the required vector increment ΔX.

Sign bit

For example if the ΔX register reads $\overbrace{0}\,0\,0\,0\,1\,1\,0$ then only the following clock counter states are passed through Algorithm Gates to produce point pulses:

C_0	C_1	C_2	C_3	C_4	C_5	Decimal Equivalent of the clock counter state
0	0	1	0	0	0	8
0	1	0	0	0	0	16
0	1	1	0	0	0	24
1	0	1	0	0	0	40
1	1	0	0	0	0	48
1	1	1	0	0	0	56

As can be seen, the 8th, 16th, etc. clock counts produce points. Of course, another point is produced at the starting location by the start pulse. Clearly, 6 incremental points are generated corresponding to $\Delta X = 6$.

A nice feature is the fact that the generated points are fairly evenly distributed over the clock counter cycle, corresponding to a nearly uniform change

in the X coordinate with time. Since the same procedure is followed by the Y portion of the vector generator, the requirement of a constant slope for the generated line is almost satisfied.

Let us express the logical gating required in terms of Boolean Algebra.

The most significant bit of the ΔX register is "anded" with the least significant bit of the clock counter, the next most significant bit of ΔX with the next least significant bit of the clock counter, etc., providing that no "gap" condition exists. If a "gap" is present then in order for a point pulse to be produced, all the less significant bit positions of the clock counter up to the "gap" must be zero, and all the following bits of the clock counter are "anded" with the bits of the B register, as before.

A point pulse "P_x" is produced when any one of the following conditions exists:

							Bit configuration of the ΔX register (X here indicates that a given bit is either "0" or "1")					
							B_1	B_2	B_3	B_4	B_5	B_6
B_1	C_5						1					
B_1	B_2	C_4					1	1				
$\overline{B}_1$	B_2	C_4	$\overline{C}_5$				0	1				
B_1	B_2	B_3	C_3				1	1	1			
$\overline{B}_1$	B_2	B_3	C_3	$\overline{C}_5$			0	1	1			
$\overline{B}_2$	B_3	C_3	$\overline{C}_4$	$\overline{C}_5$			X	0	1			
B_1	B_2	B_3	B_4	C_2			1	1	1	1		
$\overline{B}_1$	B_2	B_3	B_4	C_2	$\overline{C}_5$		0	1	1	1		
$\overline{B}_2$	B_3	B_4	C_2	$\overline{C}_4$	$\overline{C}_5$		X	0	1	1		
$\overline{B}_3$	B_4	C_2	$\overline{C}_3$	$\overline{C}_4$	$\overline{C}_5$		X	X	0	1		
B_1	B_2	B_3	B_4	B_5	C_1		1	1	1	1	1	
$\overline{B}_1$	B_2	B_3	B_4	B_5	C_1	$\overline{C}_5$	0	1	1	1	1	
$\overline{B}_2$	B_3	B_4	B_5	C_1	$\overline{C}_4$	$\overline{C}_5$	X	0	1	1	1	
$\overline{B}_3$	B_4	B_5	C_1	$\overline{C}_3$	$\overline{C}_4$	$\overline{C}_5$	X	X	0	1	1	
$\overline{B}_4$	B_5	C_1	$\overline{C}_2$	$\overline{C}_3$	$\overline{C}_4$	$\overline{C}_5$	X	X	X	0	1	

								B_1	B_2	B_3	B_4	B_5	B_6
B_1	B_2	B_3	B_4	B_5	B_6	C_0		1	1	1	1	1	1
$\overline{B}_1$	B_2	B_3	B_4	B_5	B_6	C_0	$\overline{C}_5$	0	1	1	1	1	1
$\overline{B}_2$	B_3	B_4	B_5	B_6	C_0	$\overline{C}_4$	$\overline{C}_5$	X	0	1	1	1	1
$\overline{B}_3$	B_4	B_5	B_6	C_0	$\overline{C}_3$	$\overline{C}_4$	$\overline{C}_5$	X	X	0	1	1	1
$\overline{B}_4$	B_5	B_6	C_0	$\overline{C}_2$	$\overline{C}_3$	$\overline{C}_4$	$\overline{C}_5$	X	X	X	0	1	1
$\overline{B}_5$	B_6	C_0	$\overline{C}_1$	$\overline{C}_2$	$\overline{C}_3$	$\overline{C}_4$	$\overline{C}_5$	X	X	X	X	0	1

$$
\begin{aligned}
P_x = {} & B_1\ C_5 + B_1\ B_2\ C_4 + \overline{B}_1\ B_2\ C_4\ \overline{C}_5 \\
& + B_1\ B_2\ B_3\ C_3 + \overline{B}_1\ B_2\ B_3\ C_3\ \overline{C}_5 + \overline{B}_2\ B_3\ C_3\ \overline{C}_4\ \overline{C}_5 \\
& + B_1\ B_2\ B_3\ B_4\ C_2 + \overline{B}_1\ B_2\ B_3\ B_4\ C_2\ \overline{C}_5 + \overline{B}_2\ B_3\ B_4\ C_2\ \overline{C}_4\ \overline{C}_5 \\
& + \overline{B}_3\ B_4\ C_2\ \overline{C}_3\ \overline{C}_4\ \overline{C}_5 + B_1\ B_2\ B_3\ B_4\ B_5\ C_1 \\
& + \overline{B}_1\ B_2\ B_3\ B_4\ B_5\ C_1\ \overline{C}_5 + \overline{B}_2\ B_3\ B_4\ B_5\ C_1\ \overline{C}_4\ \overline{C}_5 \\
& + \overline{B}_3\ B_4\ B_5\ C_1\ \overline{C}_3\ \overline{C}_4\ \overline{C}_5 + \overline{B}_4\ B_5\ C_1\ \overline{C}_2\ \overline{C}_3\ \overline{C}_4\ \overline{C}_5 \\
& + B_1\ B_2\ B_3\ B_4\ B_5\ B_6\ C_0 + \overline{B}_1\ B_2\ B_3\ B_4\ B_5\ B_6\ C_0\ \overline{C}_5 \\
& + \overline{B}_2\ B_3\ B_4\ B_5\ B_6\ C_0\ \overline{C}_4\ \overline{C}_5 + \overline{B}_3\ B_4\ B_5\ B_6\ C_0\ \overline{C}_3\ \overline{C}_4\ \overline{C}_5 \\
& + \overline{B}_4\ B_5\ B_6\ C_0\ \overline{C}_2\ \overline{C}_3\ \overline{C}_4\ \overline{C}_5 + \overline{B}_5\ B_6\ C_0\ \overline{C}_1\ \overline{C}_2\ \overline{C}_3\ \overline{C}_4\ \overline{C}_5
\end{aligned}
$$

This expression represents the action of the X Algorithm Gates. It can be reduced to:

$$
\begin{aligned}
P_x = {} & B_1\ C_5 + B_2\ (B_1 + \overline{B}_1\ \overline{C}_5)\ \Big(C_4 + B_3\ C_3 + B_3\ B_4\ (C_2 + B_5\ C_1 \\
& + B_5\ B_6\ C_0)\Big) + B_5\ \overline{C}_4\ \overline{C}_5\ (C_1 + B_6\ C_0)\ (\overline{B}_4\ \overline{C}_2\ \overline{C}_3 + \overline{B}_3\ B_4\ \overline{C}_3 + \overline{B}_2\ B_3\ B_4) \\
& + \overline{C}_4\ \overline{C}_5\ \Big(\overline{B}_2\ B_3\ (C_3 + B_4\ C_2) + \overline{C}_3\ \big(\overline{B}_3\ B_4\ C_2 + \overline{B}_5\ B_6\ C_0\ \overline{C}_1\ \overline{C}_2\big)\Big)
\end{aligned}
$$

Before actual implementation, this Boolean expression should be minimized by using a suitable computer minimization program.

It must be remembered that identical Algorithm Gates must be provided for the Y portion of the vector generator.

At this point it will be useful to consider some examples of vector generation by this method

	B_1	B_2	B_3	B_4	B_5	B_6
Let ΔX = 20 (Decimal) =	0	1	0	1	0	0
	B_1'	B_2'	B_3'	B_4'	B_5'	B_6'
ΔY = 14 (Decimal) =	0	0	1	1	1	0

where the primes indicate digit stages of the ΔY register.

The following clock pulses will be passed through:

ΔX Algorithm Gates

0	0	0	0	1	0	(2)
0	0	0	1	1	0	(6)
0	0	1	0	0	0	(8)
0	0	1	0	1	0	(10)
0	0	1	1	1	0	(14)
0	1	0	0	1	0	(18)
0	1	0	1	1	0	(22)
0	1	1	0	0	0	(24)
0	1	1	0	1	0	(26)
0	1	1	1	1	0	(30)
1	0	0	0	1	0	(34)
1	0	0	1	1	0	(38)
1	0	1	0	0	0	(40)
1	0	1	0	1	0	(42)
1	0	1	1	1	0	(46)
1	1	0	0	1	0	(50)
1	1	0	1	1	0	(54)
1	1	1	0	0	0	(56)
1	1	1	0	1	0	(58)
1	1	1	1	1	0	(62)

20 pulses pass through the

ΔX Algorithm Gates

ΔY Algorithm Gates

0	0	0	1	0	0	(4)
0	0	1	0	0	0	(8)
0	0	1	1	0	0	(12)
0	1	0	0	0	0	(16)
0	1	0	1	0	0	(20)
0	1	1	0	0	0	(24)
0	1	1	1	0	0	(28)
1	0	0	1	0	0	(36)
1	0	1	0	0	0	(40)
1	0	1	1	0	0	(44)
1	1	0	0	0	0	(48)
1	1	0	1	0	0	(52)
1	1	1	0	0	0	(56)
1	1	1	1	0	0	(60)

14 pulses pass through the

ΔY Algorithm Gates

Each one of these pulses produces a point on the CRT. The result is shown in Figure A-1.

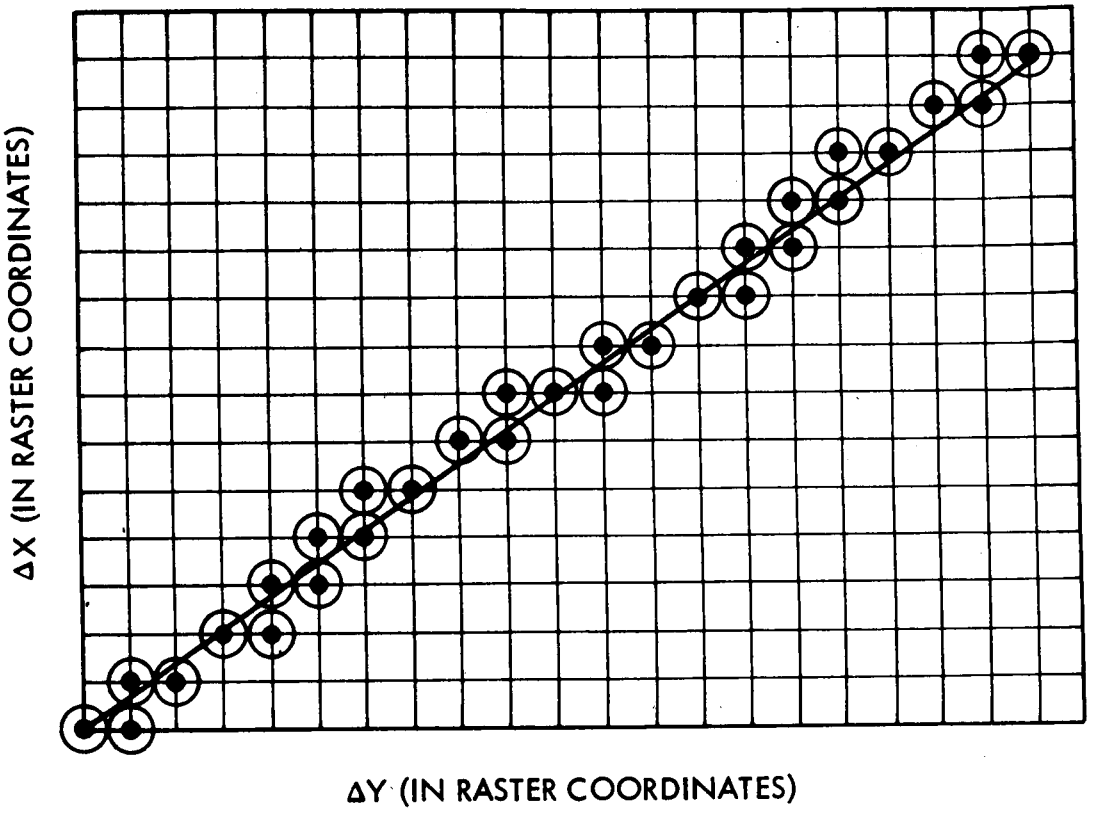

Figure A-1.

For a second example

	B_1	B_2	B_3	B_4	B_5	B_6
let ΔX = 18 (Decimal) =	0	1	0	0	1	0
	B'_1	B'_2	B'_3	B'_4	B'_5	B'_6
ΔY = 5 (Decimal) =	0	0	0	1	0	1

the following clock pulses will pass through

ΔX Algorithm Gates

0	0	0	0	1	0	(2)
0	0	0	1	1	0	(6)
0	0	1	0	1	0	(10)
0	0	1	1	1	0	(14)
0	1	0	0	0	0	(16)
0	1	0	0	1	0	(18)
0	1	0	1	1	0	(22)
0	1	1	0	1	0	(26)
0	1	1	1	1	0	(30)
1	0	0	0	1	0	(34)
1	0	0	1	1	0	(38)
1	0	1	0	1	0	(42)
1	0	1	1	1	0	(46)
1	1	0	0	0	0	(48)
1	1	0	0	1	0	(50)
1	1	0	1	1	0	(54)
1	1	1	0	1	0	(58)
1	1	1	1	1	0	(62)

18 pulses pass through the

ΔX Algorithm Gates

ΔY Algorithm Gates

0	0	1	0	0	0	(8)
0	1	1	0	0	0	(24)
1	0	0	0	0	0	(32)
1	0	1	0	0	0	(40)
1	1	1	0	0	0	(56)

5 pulses pass through the

ΔY Algorithm Gates

The resultant vector is shown in Figure A-2.

As can be seen from the two examples, the more nearly the two numbers stored in the ΔX and ΔY registers agree in magnitude, the better will be the approximation to a straight line, and vice versa. It should, however, be kept in mind that the two illustrations are highly magnified in size and that on the CRT screen the deviation from a straight line would be hardly noticeable. If we take the CRT specifications to be the ones given at the beginning of this discussion, the maximum length of our vector corresponding to 64 points would be $10\sqrt{2}/16$ inches $\approx$ 0.885 inches. The magnitudes of the vectors in our two illustrations would be correspondingly smaller.

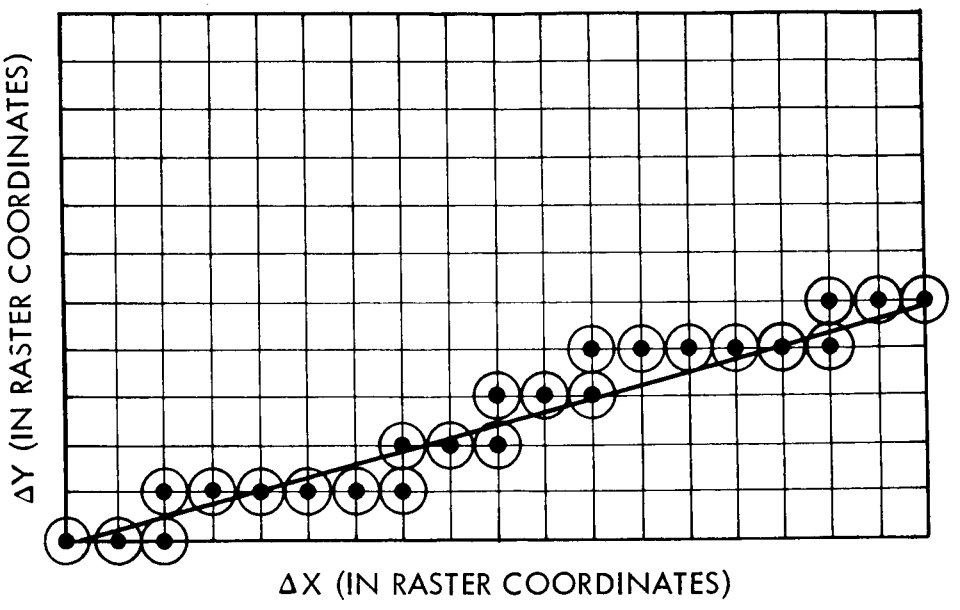

Figure A-2.

APPENDIX B

Adaptation of the Incremental Point Positioning Vector Generator to the Production of Larger Vector Increments

The vector generator discussed in Appendix A can produce vectors corresponding to 7 bit increments ΔX and ΔY (6 bit magnitude plus the sign bit). In order to extend this generator to 8 bit increments the method of arriving at the required logic gates in Appendix A must be re-examined and extended.

The ΔX register in this case will be the 8 bit B register with B_0 again representing the sign of ΔX.

The clock counter C will have seven stages:

B Register

B_0	B_1	B_2	B_3	B_4	B_5	B_6	B_7

Clock Counter

C_0	C_1	C_2	C_3	C_4	C_5	C_6

Again the 1 megahertz clock will drive the clock counter, in this case through 128 steps, beginning with the arrival of the "start" pulse.

As in Appendix A, a point pulse "P_x" will be passed on to the X Up/Down Counter when one of the following conditions exists:

	Bit configuration of ΔX register						
	B_1	B_2	B_3	B_4	B_5	B_6	B_7
$B_1\ C_6$	1						
$B_1\ B_2\ C_5$	1	1					
$\overline{B}_1\ B_2\ C_5\ \overline{C}_6$	0	1					
$B_1\ B_2\ B_3\ C_4$	1	1	1				
$\overline{B}_1\ B_2\ B_3\ C_4\ \overline{C}_6$	0	1	1				
$\overline{B}_2\ B_3\ C_4\ \overline{C}_5\ \overline{C}_6$	X	0	1				

									B_1	B	B_3	B_4	B_5	B_6	B_7
B_1	B_2	B_3	B_4	C_3					1	1	1	1			
$\overline{B}_1$	B_2	B_3	B_4	C_3	$\overline{C}_6$				0	1	1	1			
$\overline{B}_2$	B_3	B_4	C_3	$\overline{C}_5$	$\overline{C}_6$				X	0	1	1			
$\overline{B}_3$	B_4	C_3	$\overline{C}_4$	$\overline{C}_5$	$\overline{C}_6$				X	X	0	1			
B_1	B_2	B_3	B_4	B_5	C_2				1	1	1	1	1		
$\overline{B}_1$	B_2	B_3	B_4	B_5	C_2	$\overline{C}_6$			0	1	1	1	1		
$\overline{B}_2$	B_3	B_4	B_5	C_2	$\overline{C}_5$	$\overline{C}_6$			X	0	1	1	1		
$\overline{B}_3$	B_4	B_5	C_2	$\overline{C}_4$	$\overline{C}_5$	$\overline{C}_6$			X	X	0	1	1		
$\overline{B}_4$	B_5	C_2	$\overline{C}_3$	$\overline{C}_4$	$\overline{C}_5$	$\overline{C}_6$			X	X	X	0	1		
B_1	B_2	B_3	B_4	B_5	B_6	C_1			1	1	1	1	1	1	
$\overline{B}_1$	B_2	B_3	B_4	B_5	B_6	C_1	$\overline{C}_6$		0	1	1	1	1	1	
$\overline{B}_2$	B_3	B_4	B_5	B_6	C_1	$\overline{C}_5$	$\overline{C}_6$		X	0	1	1	1	1	
$\overline{B}_3$	B_4	B_5	B_6	C_1	$\overline{C}_4$	$\overline{C}_5$	$\overline{C}_6$		X	X	0	1	1	1	
$\overline{B}_4$	B_5	B_6	C_1	$\overline{C}_3$	$\overline{C}_4$	$\overline{C}_5$	$\overline{C}_6$		X	X	X	0	1	1	
$\overline{B}_5$	B_6	C_1	$\overline{C}_2$	$\overline{C}_3$	$\overline{C}_4$	$\overline{C}_5$	$\overline{C}_6$		X	X	X	X	0	1	
B_1	B_2	B_3	B_4	B_5	B_6	B_7	C_0		1	1	1	1	1	1	1
$\overline{B}_1$	B_2	B_3	B_4	B_5	B_6	B_7	C_0	$\overline{C}_6$	0	1	1	1	1	1	1
$\overline{B}_2$	B_3	B_4	B_5	B_6	B_7	C_0	$\overline{C}_5$	$\overline{C}_6$	X	0	1	1	1	1	1
$\overline{B}_3$	B_4	B_5	B_6	B_7	C_0	$\overline{C}_4$	$\overline{C}_5$	$\overline{C}_6$	X	X	0	1	1	1	1
$\overline{B}_4$	B_5	B_6	B_7	C_0	$\overline{C}_3$	$\overline{C}_4$	$\overline{C}_5$	$\overline{C}_6$	X	X	X	0	1	1	1
$\overline{B}_5$	B_6	B_7	C_0	$\overline{C}_2$	$\overline{C}_3$	$\overline{C}_4$	$\overline{C}_5$	$\overline{C}_6$	X	X	X	X	0	1	1
$\overline{B}_6$	B_7	C_0	$\overline{C}_1$	$\overline{C}_2$	$\overline{C}_3$	$\overline{C}_4$	$\overline{C}_5$	$\overline{C}_6$	X	X	X	X	X	0	1

A point pulse "P_x" is produced when any one of these conditions exists.

$$P_x = B_1\ C_6 + B_1\ B_2\ C_5 + \overline{B}_1\ B_2\ C_5\ \overline{C}_6 + B_1\ B_2\ B_3\ C_4 + \overline{B}_1\ B_2\ B_3\ C_4\ \overline{C}_5$$
$$+ \overline{B}_2\ B_3\ C_4\ \overline{C}_5\ \overline{C}_6 + B_1\ B_2\ B_3\ B_4\ C_3 + \overline{B}_1\ B_2\ B_3\ B_4\ C_3\ \overline{C}_6 + \overline{B}_2\ B_3\ B_4\ C_3\ \overline{C}_5\ \overline{C}_6$$
$$+ \overline{B}_3\ B_4\ C_3\ \overline{C}_4\ \overline{C}_5\ \overline{C}_6 + B_1\ B_2\ B_3\ B_4\ B_5\ C_2 + \overline{B}_1\ B_2\ B_3\ B_4\ B_5\ C_2\ \overline{C}_6$$
$$+ \overline{B}_2\ B_3\ B_4\ B_5\ C_2\ \overline{C}_5\ \overline{C}_6 + \overline{B}_3\ B_4\ B_5\ C_2\ \overline{C}_4\ \overline{C}_5\ \overline{C}_6 + \overline{B}_4\ B_5\ C_2\ \overline{C}_3\ \overline{C}_4\ \overline{C}_5\ \overline{C}_6$$
$$+ B_1\ B_2\ B_3\ B_4\ B_5\ B_6\ C_1 + \overline{B}_1\ B_2\ B_3\ B_4\ B_5\ B_6\ C_1\ \overline{C}_6 + \overline{B}_2\ B_3\ B_4\ B_5\ B_6\ C_1\ \overline{C}_5\ \overline{C}_6$$
$$+ \overline{B}_3\ B_4\ B_5\ B_6\ C_1\ \overline{C}_4\ \overline{C}_5\ \overline{C}_6 + \overline{B}_4\ B_5\ B_6\ C_1\ \overline{C}_3\ \overline{C}_4\ \overline{C}_5\ \overline{C}_6 + \overline{B}_5\ B_6\ C_1\ \overline{C}_2\ \overline{C}_3\ \overline{C}_4\ \overline{C}_5\ \overline{C}_6$$
$$+ B_1\ B_2\ B_3\ B\ B_5\ B_6\ B_7\ C_0 + \overline{B}_1\ B_2\ B_3\ B_4\ B_5\ B_6\ B_7\ C_0\ \overline{C}_6 + \overline{B}_2\ B_3\ B_4\ B_5\ B_6\ B_7$$
$$C_0\ \overline{C}_5\ \overline{C}_6 + \overline{B}_3\ B_4\ B_5\ B_6\ B_7\ C_0\ \overline{C}_4\ \overline{C}_5\ \overline{C}_6 + \overline{B}_4\ B_5\ B_6\ B_7\ C_0\ \overline{C}_3\ \overline{C}_4\ \overline{C}_5\ \overline{C}_6$$
$$+ \overline{B}_5\ B_6\ B_7\ C_0\ \overline{C}_2\ \overline{C}_3\ \overline{C}_4\ \overline{C}_5\ \overline{C}_6 + \overline{B}_6\ B_7\ C_0\ \overline{C}_1\ \overline{C}_2\ \overline{C}_3\ \overline{C}_4\ \overline{C}_5\ \overline{C}_6$$

As in Appendix A, this Boolean expression should be minimized by a computer minimization program before implementation. Again – identical Algorithm Gates must be provided for the Y portion of the vector generator. By comparing the Algorithm Gates given in Appendix A and Appendix B, it becomes easy to generalize this vector generator for any number of bit position of the incremental registers ΔX and ΔY. Thus any vector length can be generated by this method provided that sufficient logic circuitry is found in the system.

The advantage gained in being able to draw longer vectors with a single set of instructions in this extended version of the incremental point positioning vector generator appears to be offset by the fact that the clock counter now has to cycle through 128 counts for all vectors, or that the vector drawing time has been increased to 128 μsec., regardless of vector length.

One way to overcome this disadvantage is to have the clock counter cycle through a smaller number of counts for short vectors than for long vectors.

For example, if the most significant bits (not the sign bits) of the ΔX and ΔY registers are both zero, then the clock will drive stage C_5 (instead of C_6) of the clock counter. Thus the clock counter will only cycle through 64 counts, since its least significant stage is bypassed.

The following terms will produce counts through the Algorithm Gates in this configuration:

	Bit configuration of ΔX register						
	B_1	B_2	B_3	B_4	B_5	B_6	B_7
$\overline{B}_1\ B_2\ C_5\ \overline{C}_6$	0	1					
$\overline{B}_1\ B_2\ B_3\ C_4\ \overline{C}_6$	0	1	1				
$\overline{B}_2\ B_3\ C_4\ \overline{C}_5\ \overline{C}_6$	X	0	1				
$\overline{B}_1\ B_2\ B_3\ B_4\ C_3\ \overline{C}_6$	0	1	1	1			
$\overline{B}_2\ B_3\ B_4\ C_3\ \overline{C}_5\ \overline{C}_6$	X	0	1	1			
$\overline{B}_3\ B_4\ C_3\ \overline{C}_4\ \overline{C}_5\ \overline{C}_6$	X	X	0	1			
$\overline{B}_1\ B_2\ B_3\ B_4\ B_5\ C_2\ \overline{C}_6$	0	1	1	1	1		
$\overline{B}_2\ B_3\ B_4\ B_5\ C_2\ \overline{C}_5\ \overline{C}_6$	X	0	1	1	1		
$\overline{B}_3\ B_4\ B_5\ C_2\ \overline{C}_4\ \overline{C}_5\ \overline{C}_6$	X	X	0	1	1		
$\overline{B}_4\ B_5\ C_2\ \overline{C}_3\ \overline{C}_4\ \overline{C}_5\ \overline{C}_6$	X	X	X	0	1		
$\overline{B}_1\ B_2\ B_3\ B_4\ B_5\ B_6\ C_1\ \overline{C}_6$	0	1	1	1	1	1	
$\overline{B}_2\ B_3\ B_4\ B_5\ B_6\ C_1\ \overline{C}_5\ \overline{C}_6$	X	0	1	1	1	1	
$\overline{B}_3\ B_4\ B_5\ B_6\ C_1\ \overline{C}_4\ \overline{C}_5\ \overline{C}_6$	X	X	0	1	1	1	
$\overline{B}_4\ B_5\ B_6\ C_1\ \overline{C}_3\ \overline{C}_4\ \overline{C}_5\ \overline{C}_6$	X	X	X	0	1	1	
$\overline{B}_5\ B_6\ C_1\ \overline{C}_2\ \overline{C}_3\ \overline{C}_4\ \overline{C}_5\ \overline{C}_6$	X	X	X	X	0	1	
$\overline{B}_1\ B_2\ B_3\ B_4\ B_5\ B_6\ B_7\ C_0\ \overline{C}_6$	0	1	1	1	1	1	1
$\overline{B}_2\ B_3\ B_4\ B_5\ B_6\ B_7\ C_0\ \overline{C}_5\ \overline{C}_6$	X	0	1	1	1	1	1
$\overline{B}_3\ B_4\ B_5\ B_6\ B_7\ C_0\ \overline{C}_4\ \overline{C}_5\ \overline{C}_6$	X	X	0	1	1	1	1
$\overline{B}_4\ B_5\ B_6\ B_7\ C_0\ \overline{C}_3\ \overline{C}_4\ \overline{C}_5\ \overline{C}_6$	X	X	X	0	1	1	1
$\overline{B}_5\ B_6\ B_7\ C_0\ \overline{C}_2\ \overline{C}_3\ \overline{C}_4\ \overline{C}_5\ \overline{C}_6$	X	X	X	X	0	1	1
$\overline{B}_6\ B_7\ C_0\ \overline{C}_1\ \overline{C}_2\ \overline{C}_3\ \overline{C}_4\ \overline{C}_5\ \overline{C}_6$	X	X	X	X	X	0	1

Since B_1 and C_6 are both always zero in this case $\overline{B}_1$ and $\overline{C}_6$ will be logical "1" and can be ignored. If the terms of the B register are relabeled so that B_2 corresponds to B_1 of Appendix A, B_3 corresponds to B_2, and so forth, then we see that the pulse producing terms here correspond to the ones found for the 7 bit ΔX register of Appendix A.

In summary, if the most significant bits of ΔX and ΔY are both zero, then the clock pulses are routed to the next least significant stage of the clock counter. If the most significant bits of ΔX and ΔY are not both zero then the clock pulses are fed to the least significant stage of the clock counter, which then cycles through 128 counts.

This modification allows the system to draw short vectors in 64 μsec and longer vectors in 128 μsec.

There is no reason why this procedure should not be applied to all stages of the ΔX and ΔY registers. If the more significant bits of both registers are zero, then the clock pulses will bypass a number of least significant stages of the clock counter equal to the number of more significant bits that are zero in both ΔX and ΔY registers.

This will allow a variable vector drawing time, depending on the vector length, and thus a more efficient production of line segments.

APPENDIX C

Digitally Controlled Sawtooth Generator (DCSG)

The Digitally Controlled Sawtooth Generator (DCSG) can be used to produce a sawtooth voltage signal whose slope is controlled by the number stored in a given register. This approach is based on the fact that in an RC sawtooth generator the slope of the voltage output waveform depends on the time constant of the system. By digitally changing the value of the charging resistance one obtains a proportional change in the output waveform.

The circuit used is shown in Figure C-1, where the incoming pulse turns off the normally conducting transistor Q_1 and turns on transistors Q_2 and/or Q_3 ..... Q_9, depending on the condition of the flip-flops in the ΔX or ΔY register. At the end of the pulse all transistors Q_2 ..... Q_9 are cut off and Q_1 is turned on providing a discharge path for the capacitor C.

The DCSG was implemented using Packard Bell Transistor Logic Cards for the digital portion. Transistors Q_2 to Q_9 were 2N703's, and Q_1 was 2N995. These transistors were chosen mainly for the low value of their leakage currents. Base bias resistors R_9 and R_{10} were selected at 300 and 510 ohms respectively. R_{12} was 9.1 KΩ and R_{11} was 3.6 KΩ.

The charging resistors R_1, R_2 ..... R_8 should form a binary sequence corresponding to the binary weighting of the different bits in the ΔX or ΔY register. The actual values chosen for the breadboard model of this system depended on the availability of resistors in our laboratory (without having to form resistor combinations) and are shown in the following table together with the peak output voltages of the generated sawtooth.

Resistor	Magnitude	Peak Output Voltage
R_1	470 Ω	- 6.4 V
R_2	910 Ω	- 3.7 V
R_3	1.8 K Ω	- 1.85 V
R_4	3.6 K Ω	- 0.86 V
R_5	7.5 K Ω	- 0.42 V
R_6	15 K Ω	- 0.21 V
R_7	30 K Ω	- 0.092 V
R_8	62 K Ω	- 0.04 V

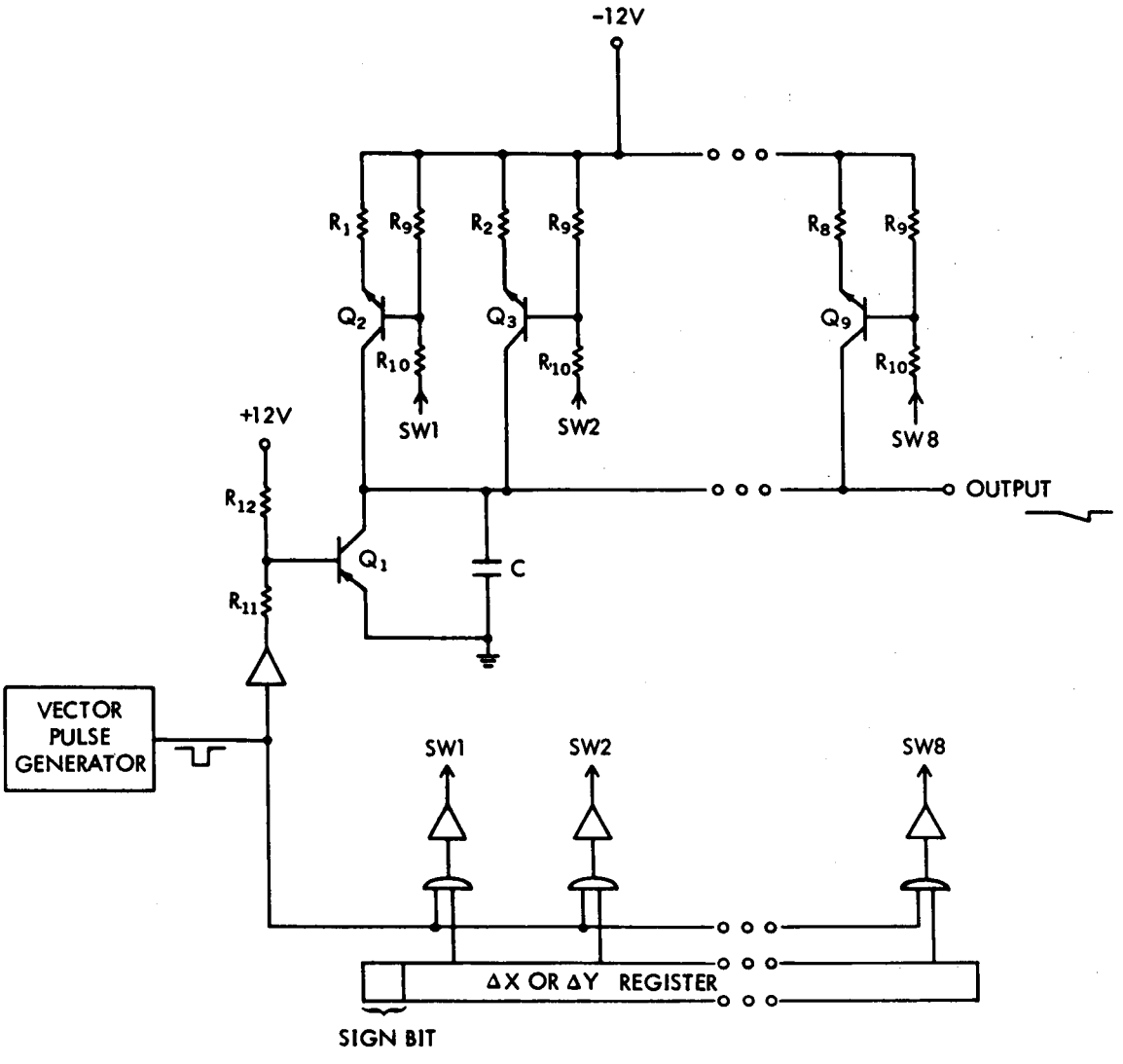

Figure C-1.

All of the above resistors were of standard (5%) quality and the voltage measurements were made on a Hewlett Packard 175 A oscilloscope.

The capacitor "C" was chosen so that the sawtooth pulse could reach the largest distortion free amplitude in the pulse interval. The pulse interval was somewhat arbitrarily placed at 11.6 microseconds and for this interval the value of the capacitor was 0.015 μf. Other time intervals could be selected, probably requiring a change in the value of "C". The inverters after the logical "AND" gates in Figure C-1 and the values of the biasing resistors R_9 and R_{10} were chosen to be compatible with the Packard Bell Logic Levels, where the logical "1" is -10 volts and logical zero is 0 volts.

Originally it was intended to construct a DCSG that would produce a variable slope sawtooth corresponding to 9 bits in the ΔX or ΔY register.

However, at an early experimental stage it was found that the unshielded circuit was subject to noise levels on the order of magnitude of the amplitude of the sawtooth produced by the least significant bit in the register. It is a matter of conjecture at this point if the DCSG as it stands could have been refined by providing a cleaner pulse source, mounting the system on a card with the accompanying shortening of wire leads, appropriate grounding, etc., to operate as a nine bit system since the DCSG never went beyond an early breadboard stage. However, it is felt that the DCSG could be made to perform reliably as an eight bit system. The linearity of the output waveform was very good over this wide range of slope variations. The sawtooth start and stop points were well defined.

A small amount of D.C. offset is present at the output due to the action of transistor Q_1, manifesting itself as a constant amplitude pulse of about 0.01 volts when zero is stored in the ΔX and ΔY register and the vector pulse is applied. This can be overcome by the addition of some logic gates, which would allow the sawtooth waveform or its inverse to be transmitted through the analog gate, only if a non-zero value is stored in the ΔX or ΔY register.

The method used to change the resistance of the charging circuit, and thus the slope of the output sawtooth, requires that a binary sequence of precision resistors R, 2R ..... 128R be placed in series with the switching transistors, as shown in Figure C-1. It is often difficult to obtain precision resistors over this wide range of resistance values. However, it is possible to substitute an R/2R binary ladder* for the charging resistors of Figure C-1. The modified DCSG is shown in Figure C-2, where the charging network now only requires two different high precision resistors R and 2R. Transistors Q_1 and Q_2 should be complementary. They are required at each node of the ladder network, since for proper operation either zero volts or the charging voltage (-12 volts) must be present at each node point. When a given stage of the ΔX or ΔY register is "0" then zero volts are switched in at this node; when a given stage is "1" then the charging voltage (-12 volts) is switched in at the corresponding node. Otherwise the operation of this system is identical to the one shown in Figure C-1.

*Alfred K. Susskind, editor, Notes on Analog Digital Conversion Techniques, Technology Press of MIT and John Wiley and Sons, Inc., 1960, pp. 5-32 – 5-35

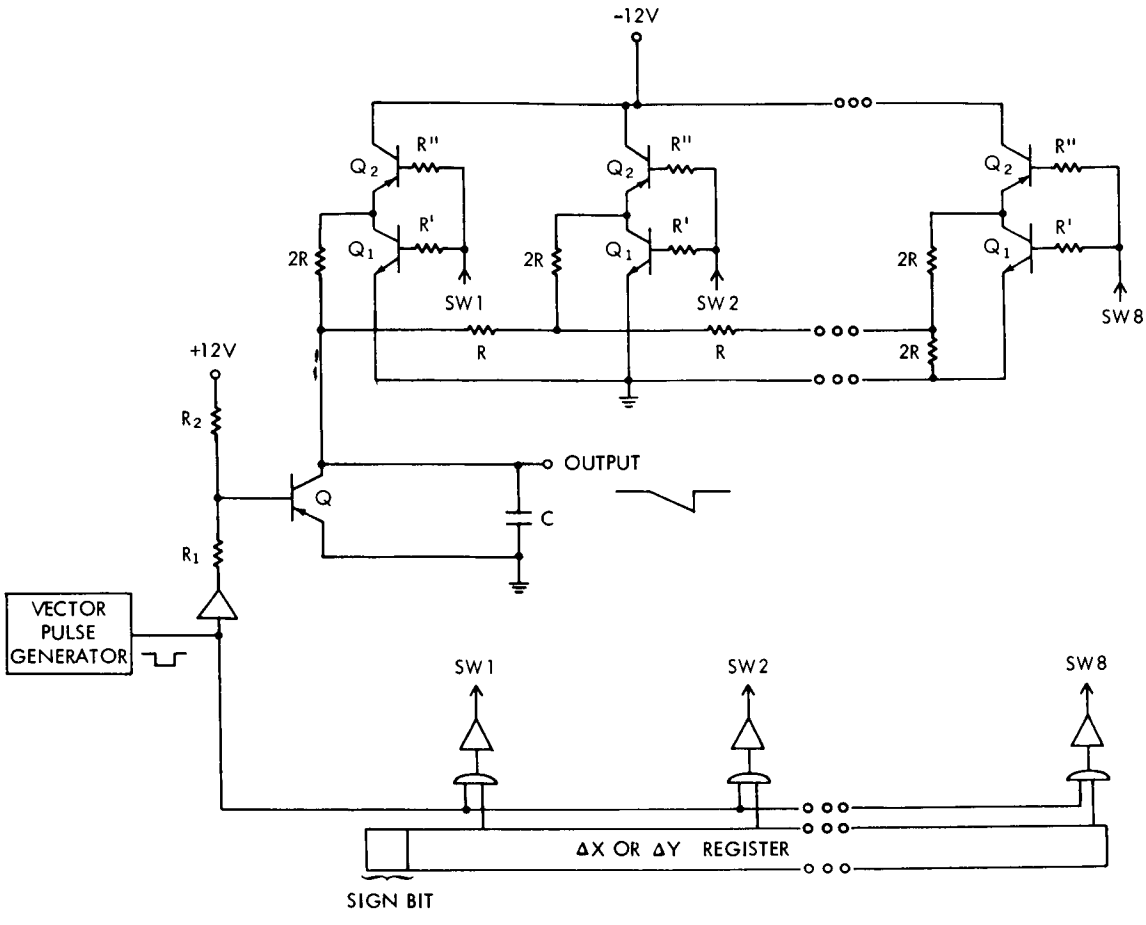
-12V
R"
Q2
R'
Q1
2R
SW1
SW2
SW8
R
+12V
R2
OUTPUT
Q
C
R1
VECTOR
PULSE
GENERATOR
ΔX OR ΔY REGISTER
SIGN BIT

Figure C-2.