ABSTRACT

A synthetic graphics generating system, using a C.R.T. scanned in a T.V. type raster. The video signal for each linescan is calculated digitally in real time, either during the linescan flyback periods, or during execution of the preceding linescan. The described system is used to generate a number of vectors, the calculations for which are performed on a time-shared basis by common circuitry. To prevent a stepped appearance, progressive increase and decrease in brightness is used, preferably using three discrete brightness levels.

9 Claims, 14 Drawing Figures
Fig. 8a

Write Into N Reg.  
Write Into Tan @ Reg.  
Add Reg. := A Reg.  
Reset Bistables G, H, I, K and Z  
Is Nav(Addr. Reg.) = 1?  
Address MH with Addr. Reg.  
Is Bistable G Set?  
Is Bistable J Set?  
P Reg. := Addr. Reg. Set Bistable H  
Is Nav = 0?  
L Reg. := Next Address Set Bistable G  
Set Bistable Z  
S Reg. := Addr. Reg. Set Bistable K  
RASTER-SCANNED DISPLAY DEVICES

This invention relates to raster-scanned display devices and, more specifically, is concerned with apparatus for generating a video signal for input to a raster-scanned display device.

One example of a raster-scanned display device is a cathode ray tube arranged to be scanned in a television-type raster. However, other forms of raster-scanned display device are known: for example, an array of light emitting diodes arranged to be scanned electronically, or a screen arranged to be scanned by a laser beam.

One object of the invention is to provide a novel form of apparatus for generating video signals for input to a raster-scanned display device.

According to a first aspect of the invention, there is provided apparatus for generating a video signal for input to a raster-scanned display device to cause the device to display at least one graphical symbol the apparatus comprising: an input store for holding, in digital form, input parameters of said at least one graphical symbol; digital data-processing means for processing said parameters to produce a digital output characterising said video signal; and means for converting said digital output into said video signal.

Said graphical symbols may be stroke symbols (e.g., straight or curved lines) or may be symbols having an appreciable area.

Stroke symbols may, for example, be used to form alphanumerical characters. Alternatively, the stroke symbols may represent other information such as, for example, an aircraft flight path, or the position of the horizon relative to an aircraft.

Preferably, said digital data processing means comprises: a video store adapted to hold information characterising a portion of said video signal corresponding to at least one linescan of the raster; computing means for performing calculations on said parameters in respect of each linescan of the raster in turn to produce said information for loading into said video store; and read-out means for reading said information out of said video store, on completion of said calculations in respect of a linescan, to produce the portion of said digital output corresponding to that linescan.

In a preferred form of the invention, said input store is adapted to hold the parameters of a plurality of graphical symbols, and said computing means is adapted to perform, for each linescan of the raster, said calculations on the parameters of each of said graphical symbols in turn. Thus, it will be seen that the computing means is time-shared among the plurality of symbols, with a corresponding saving in the quantity of hardware required.

In one particular form of the invention, said video store is adapted to hold information characterising a portion of said video signal corresponding to a single linescan of the raster, the apparatus being so arranged that, in operation, said computing means performs said calculations in respect of a linescan during the linescan flyback period immediately preceding that linescan.

In another particular form of the invention, said video store is adapted to hold information characterising a portion of said video signal corresponding to two linescans of the raster, the apparatus being so arranged that, in operation, said computing means performs said calculations in respect of a linescan while information characterising the immediately preceding linescan is being read out of the video store. This has the advantage of allowing a much longer time for performance of the calculations, and hence, allows correspondingly more graphical symbols to be dealt with on a time-shared basis.

In a preferred arrangement, said data processing means comprises means for storing a number defining the position of a portion of a said symbol within a linescan of the raster, and means for performing a predetermined algorithm, according to said parameters, to update said number to its appropriate value for a succeeding linescan of the raster.

According to another aspect of the invention, there is provided a display system comprising a raster-scanned display device, and apparatus for generating a video signal for input to the device, in accordance with the first aspect of the invention.

One display system, incorporating apparatus in accordance with the invention, will now be described by way of example with reference to the accompanying drawings, of which:

FIG. 1 is a schematic block diagram of the display system;
FIG. 2 is an enlarged schematic view of a part of the picture displayed by the system;
FIGS. 3a and 3b, when arranged as indicated in FIG. 3c, constitute a schematic block diagram of part of the display system;
FIG. 4a is a schematic diagram illustrating the manner in which the video signal shown in FIG. 4b is stored in the display system;
FIG. 5 is a schematic block diagram of a modification of the display system;
FIGS. 6 and 7 are schematic circuit diagrams of parts of FIG. 3; and
FIGS. 8a, b and c are flow diagrams arranged as indicated in FIG. 8d, and representing the operating logic of the circuit of FIG. 7.

Referring to FIG. 1, the display system comprises an on-line digital computer 1, from which digital output data are fed, by way of an interface 2, to a vector generator 3 and a circle generator 4. These generators 3 and 4 produce respective vector output signals which are combined in a mixer 5 and fed to the brightness modulation input of a cathode ray tube 6. The cathode ray tube 6 is scanned in a television-type raster by means of scanning circuits 7. Line and frame synchronisation signals from the scanning circuits 7 are also fed to the vector and circle generators 3 and 4.

The raster pattern on the screen of the cathode ray tube 6 comprises 625 lines on a 4:3 aspect ratio picture area, and is of the interlaced field type. Of these lines, 576 are available for definition in the vertical (Y) direction. The corresponding definition in the horizontal X direction is approximately 700 video elements per linescan, corresponding to a video frequency of 14 megahertz.

Thus each frame of the picture can be considered as being composed of 700 × 576 video elements, each of approximately 1:1 aspect ratio. In the following description, the spacing between adjacent lines in the same field of the raster will be taken as the unit of length in the vertical direction, and the width of one video element will be taken as the unit of length in the horizontal direction. It will be appreciated that the vertical unit is therefore twice the horizontal unit.
The vector generator 3 is designed to generate a video signal which produces one or more straight lines (referred to herein as "vectors") on the screen of the tube 6. In this particular example it is arranged that the vectors can only be positioned on a restricted "vector writing area" of the screen, covering only 512 of the 700 elements in each linescan and only 512 lines in the vertical direction.

FIG. 2 shows a small portion of one frame of the raster (both odd and even interlaced fields) on a greatly enlarged scale, containing two vectors P and Q. Each vector is built up from a series of brightened up portions of the raster lines. It will be seen that the brightened-up portions of a vector such as vector P are displaced horizontally by a fixed amount $2 \tan \theta$ (where $\theta$ is the angle between the vector and the vertical) between one linescan and the next in the same field, (in terms of the above-defined horizontal unit of distance). Furthermore, in the interlaced odd field, the brightened up portions for a given vector are displaced horizontally by a distance of $\tan \theta$ with respect to the even field.

The video signal from the generator 3 is quantised in three discrete levels of brightness, referred to respectively as full bright-up, two-thirds bright-up, and one-third bright-up (or, for brevity, full B.U. and one-third B.U.). For vectors making an angle $\theta$ greater than $45^\circ$ with the vertical, such as vector P, the sequence of brightness levels in each line of the raster is as follows (in terms of the above-defined unit of distance):

- one-third B.U. for one-half $\tan \theta$,
- two-thirds B.U. for one-half $\tan \theta$,
- full B.U. for $\tan \theta$,
- two-thirds B.U. for one-half $\tan \theta$,
- one-third B.U. for one-half $\tan \theta$.

This progressive increase and decrease of brightness level ensures that the vector has a substantially unstepped appearance to the eye. This is especially important for near-horizontal lines. In addition, it will be seen that for vectors such as P the total horizontal width of the vector is proportional to $\tan \theta$ which ensures that the overall brightness of the vector is independent of the angle $\theta$.

For vectors making an angle $\theta$ smaller than $45^\circ$ with the vertical, such as the vector Q, this pattern of brightness levels clearly cannot be followed, since $\tan \theta$ becomes smaller than unity, and hence the duration of the full B.U. level becomes smaller than one video element. Therefore, such vectors have to be treated specially, as will be described.

Referring to FIG. 3, the vector generator 3 comprises a random access memory M1 having a capacity of 16 words, each 59 bits in length, and serving as an input data store. This memory is constructed from integrated circuit memory elements (e.g., Texas Instruments Limited, type no SN 7489 memory elements.)

Data for up to sixteen vectors can be stored in the memory M1, each vector being allocated one word of storage space.

During field flyback periods of the raster, the following data for each vector can be written into the memory M1 via the interface 2 from the computer 1:

- $X_0$: the X coordinate of the starting point of the vector, measured from the start of the "vector writing area";
- $Y_0$: the number of the linescan (counted from the top of the "vector writing area", counting lines of one field only) in which the starting point of the vector occurs;
- $Y_1$: the number of the linescan in which the end point of the vector occurs counted as for $Y_0$; and
- $\tan \theta$: as defined above.

It will be seen that modifications can be made to the parameters of any selected one of the sixteen vectors, without affecting the other vectors, by addressing the appropriate word of the memory M1 and writing in the modified parameters.

Each word of the memory M1 is subdivided into a number of data stores for the associated vector, as follows:

(S1) Eight bits, acting as a store for $X_0$.
(S2) Eight bits, acting as a store for $Y_0$.
(S3) Eight bits, acting as a store for $Y_1$.
(S4) Thirteen bits, acting as a store for the numerical value of $\tan \theta$.
(S5) One bit, to store the sign of $\tan \theta$.
(S6) One bit, to signify whether or not the magnitude of the integral part of $\tan \theta$ is smaller than unity.
(S7) Thirteen bits, acting as a store for $X$, the X-coordinate of the vector in the current linescan.
(S8) Four bits, acting as a "next address" store, to signify the address of the next word in the memory M1 to be processed. This facility allows the words in the memory M1 to be operated upon in any desired sequence.
(S9) One "in vector" bit, which is set to "1" when the linescan $Y_1$ is reached, and reset to "0" when the linescan $Y_1$ is reached.
(S10) Two "shading bits", the purpose of which will be explained below.

During field flyback of the raster, while data is being read into the memory M1, from the interface 2, a sort logic circuit 11 compares the value of $\tan \theta$ to be written into the store S4 of the memory M1 with the other $\tan \theta$ values already in the store S4, and modifies the contents of the "next address" store S8 of the memory M1 in such a manner that the "next address" store S8 of each word contains the address of the word containing the next lowest value of $\tan \theta$. This enables the vectors in the memory M1 to be processed in order of decreasing $\tan \theta$. The reason for this will be explained below.

As each value of $\tan \theta$ is written into the memory M1, the magnitude of its integral part is compared with unity by means of a comparator circuit 12 which sets the single bit in store S6 of the memory to "1" if the magnitude is smaller than unity.

The output from section S6 controls a data selector circuit 13 which provides an eight-bit binary output T. If the output from store S6 is equal to "1", indicating that the integral part of $\tan \theta$ is smaller than unity, the data selector 13 gives an output T=1. If the output is "0", the data selector 13 gives an output T=\tan \theta, derived from the store S4 of the memory M1.

After the end of a period allocated for reading-in data from the interface 2, but still during the field flyback, the value of $X_0$ in the store S1 each word is written into the "current X" store S7 of the same word, via a data selector 14 and a binary adder 15. For even fields of the raster, the adder 15 merely adds zero to the value of $X_0$. However, for odd fields of the raster, the adder adds $\pm \tan \theta$ to $X_0$ before writing the result into the "current X" store S7. The value of the $\tan \theta$ is ob-
tained from the tan θ store S4 of the memory M1 by way of a data selector 16. This causes the tan θ displacement between odd and even fields, as mentioned above in connection with FIG. 2.

The vector generator 3 also comprises a second random access memory M2, having a capacity of 512 words, each word having six bits B1-B6. The memory M2 serves as a video store to contain information characterising a linescan of the video signal. This memory is constructed from integrated circuit memory elements (e.g., Texas Instruments Limited Type No. SN 74,200 memory elements).

Each of the 512 words in the memory M2 corresponds to one of the 512 video elements in the vector writing area of a linescan of the raster. Initially, all the bits of the memory M2 are set to "O", A "1" occurring in any given bit position of a given word signifies that the start or finish point of one of the three brightness levels, one-third B.U., two-thirds B.U. and FULL B.U. occurs at the video element corresponding to that word, as follows:

<table>
<thead>
<tr>
<th>Bit</th>
<th>Video level</th>
</tr>
</thead>
<tbody>
<tr>
<td>B1 = 1</td>
<td>% B.U. ON</td>
</tr>
<tr>
<td>B2 = 1</td>
<td>% B.U. ON</td>
</tr>
<tr>
<td>B3 = 1</td>
<td>FULL B.U. ON</td>
</tr>
<tr>
<td>B4 = 1</td>
<td>% B.U. OFF</td>
</tr>
<tr>
<td>B5 = 1</td>
<td>% B.U. OFF</td>
</tr>
<tr>
<td>B6 = 1</td>
<td>FULL B.U. OFF</td>
</tr>
</tbody>
</table>

Thus, it will be seen that the memory M2 stores a complete linescan of the video signal in the form of the start and finish points of the three discrete brightness levels.

By way of illustration, FIG. 4e shows schematically a portions of the memory M2 storing the video signal for the linescan portion A—A in FIG. 2, while FIG. 4b shows the corresponding video signal.

It is possible that two or more vectors may cross each other on a linescan, in which case their brightened-up portions will overlap in that linescan. This possibility is allowed for by ensuring that, on reading out from the memory M2, a given brightness level is not switched off until an equal number of ON's and OFF's have been read for that level, and that higher brightness levels suppress lower levels (see below).

The video signal for each linescan is calculated from the input parameters in the memory M1, and written into the video store memory M2, during the period between leaving the vector writing area in the immediately preceding linescan and re-entering the vector writing area on the current linescan. It will be seen that this period comprises the line flyback period (12 microseconds) between those two linescans, plus a portion of each of the two linescans.

The calculation of the video signal is controlled by means of a video-store-load control circuit 17, and a scan counter 19. The counter 19 receives line and frame synchronisation signals from the scanning circuit 7 (FIG. 1) to produce an 8-bit binary output Y equal to the number of lines of the current field of the raster which have been executed at any given moment. The control circuit 17 is a conventional control logic circuit comprising conventional logic components.

The calculation is performed for each of the words (vectors) in the memory M1 in turn, in the order determined by the "next address" stores S8 of the words.

The calculation for each word (vector) proceeds as follows assuming that tan θ is negative as for vector P in FIG. 2.

(Stage 1)

The control circuit 17 reads the "in vector" bit from store S9 of the word, and at the same time compares the value of Y from the scan counter 19 with the values of Yx and Y1 in the memory M1. If the "in vector" bit is "I", the calculation proceeds with stage (2). If the "in vector" bit is "O", but Y = Yx, the calculation still proceeds with stage (2), and at the same time the "in vector" bit is set to "I". If Y = Y1, the calculation still proceeds, but the "in vector" bit is reset to "O". If none of the above conditions is satisfied, (i.e., if the "in vector" bit is "O" and Y does not equal Yx) then the vector does not appear in the current linescan, and therefore the calculation is discontinued for this vector and moves on to the next vector in the memory, as determined by the "next address" store S8 of the current vector. The calculation for the next vector will, of course, commence again at the beginning of stage (1).

(Stage 2)

a. The contents of the "current X" store S7 of the word are read into a 9-bit accumulator A2, by way of a data selector 21.

b. The contents of the "current X" store S7 are also fed by way of the selector 21 to a 10-bit digital adder 22, where they are added to T/2, and the result is read into a 10-bit accumulator A1. (T is the output of the selector 13). The division of T by four is performed in a data selector 23, which can be alternatively operated to divide T by two.

At this point, therefore, accumulator A2 contains X which is the video position of start of the first one-third B.U. period (see FIG. 4b), and A1 contains X + T/2, the video position of the end of the first one-third B.U. period.

c. The contents of the accumulator A2 are used to address the ON part (bits B1, B2, B3) of the video store M2, via data selectors 24 and 25. At the same time, the contents of the accumulator A1 are used to address the OFF part (bits B4, B5, B6) of the video store M2, via data selectors 26 and 27. The contents of the accumulators A1 and A2 are calculated to an accuracy better than one half of a unit of distance (i.e., better than one video element) and are rounded up or down to the nearest half unit (i.e., to the nearest video element) before being used to address the memory M2.

A check is made to determine whether either of bits B1 or B4 so addressed already contains a "1", which may be the case if a previously calculated vector intersects the present vector in the current linescan. If neither bit position contains a "1", a "1" is now written into both bit positions. If, on the other hand there is a "1" already present in either bit, writing into the memory M2 is inhibited for both bits.

Since the vectors have been sorted into order of decreasing tan θ the horizontal width of the current vector will always be equal to or smaller than that of the previously calculated vector. Therefore, when writing into the memory M2 is inhibited in this way, the effect is merely to eliminate a brightened-up portion which
lies completely within a previously calculated bright-
ened-up portion of the same level.

It is important to note that the ON and OFF bits are
written or inhibited together as a pair, so that the total
number of ONs in the memory M2 always remains
equal to the number of OFFs.

Stage (2) actually runs concurrently with Stage (1)
in order to reduce the total calculation time required.
If stage (1) is discontinued, Stage (2) will also be dis-
continued before any data is written into the memory
M2.

STAGE (3)

a. The contents of accumulator A1 are read into
accumulator A2, by way of data selector 21.
b. The contents of accumulator A1 are read via data
selector 21 to adder 22 where they are added to
T/2, the result being read back into accumulator
A1.

At this point, therefore, accumulators A2 and A1 re-
spectively contain X + T/2 and X + T, the start and fin-
ish positions of the first two-thirds B.U. period.
c. The contents of accumulators A2 and A1, suitably
rounded to the nearest half unit, are used to ad-
dress the memory M2, as in stage (2), and a "1" is
written into the bits B2 and B5 so addressed, if both are
empty.

STAGE (4)

a. The contents of accumulator A1 are read into ac-
cumulator A2.
b. The contents of accumulator A1 are added to T
and the result is read back into accumulator A1.

At this point, accumulators A2 and A1 respectively
contain X + T and X + 2T, the start and finish points of
the full B.U. period.
c. The contents of accumulators A2 and A1, suitably
rounded, are used to address the memory M2 as in
stage (2), and a "1" is written into the bits B3 and B6
so addressed, if both are empty.

STAGE (5)

a. The contents of accumulator A1 are read into ac-
cumulator A2.
b. The contents of accumulator A1 are added to T/2
and the result is read back into accumulator A1.

At this point, accumulators A2 and A1 respectively
contain X + 2T and X + 5/2T, the start and finish points of
the second two-thirds B.U. period.
c. The contents of accumulators A2 and A1, suitably
rounded, are used to address the memory M2 as in
stage (2), and a "1" is written into the bits B2 and B5
so addressed, if both are empty.

STAGE (6)

a. The contents of accumulator A1 are read into ac-
cumulator A2.
b. The contents of accumulator A1 are added to T/2
and the result is read back into accumulator A1.

At this point, accumulators A2 and A1 respectively
contain X + 5T/2 and X + 3T, the start and finish points of
the second one-third B.U. period.
c. The contents of accumulators A2 and A1, suitably
rounded, are used to address the memory M2 as in
stage (2), and a "1" is written into the bits B1 and B4
so addressed, if both are empty.

STAGE (7)

The memory M2 now contains complete information
characterising the video brightness pattern correspond-
ing to the relevant vector for the current linescan.

Concurrently with the previous stages (3) – (6) the
contents of the "current X" store S7 are fed to one
input of adder 15, via selector 14, and added to 2 tan \( \theta \),
which is applied to the adder 15 via selector 16. The re-
sult is read back into the "current X" store S7 to serve
as the value of X for the next line of the raster.

The above stages (1) – (7) are then repeated for the
next vector in the memory M1, as determined by the
"next address" store S8 of the preceding vector.

If the sign of tan \( \theta \) is positive (which means that the
vector will slope upwards from left to right, instead of
downwards as for the vector P in FIG. 2) the above se-
quence is modified as follows:

i. In part (b) of each of stages (2) – (6), the quantity
T/2 (or T) is subtracted instead of added in adder 22.

ii. In part (c) of each of stages (2) – (6), the contents
of accumulator A2, instead of being used to ad-
dress the ON part of the memory M2, are used to
address the OFF part, via data selectors 26 and 27.
Correspondingly, the contents of accumulator A1
are now used to address the ON part of memory
M2, via data selectors 24 and 25.

iii. In stage (7), tan \( \theta \) is subtracted from the contents
of the "current X" store S7 by the adder 15.

When the magnitude of the integral part of tan \( \theta \) is
less than unity, as is the case for vector Q in FIG. 2, the
output T from the data selector 13 will be equal to 1,
and therefore the values of the quantities X, X + T/2,
X + T, X + 2T, X + 5T/2 and X + 3T will, when
rounded up or down to the nearest video element, over-
lap to some extent. As mentioned, it is arranged that
where such an overlap occurs the higher brightness
level overrides the lower one. The result is as shown in
FIG. 2. During some linescans, the pattern of bright-
ness is as indicated in linescan B, i.e., it comprises one
video element of two-thirds B.U., followed by full B.U.
for another video element and one-third B.U. for a fur-
ther video element. In other linescans, the pattern of
brightness is as indicated in linescan C, i.e., it com-
prises one-third B.U. for one video elements, followed
by full B.U. for another video element and two-thirds
B.U. for a further video element.

As in the case of near-horizontal lines, this progres-
sive increase and decrease in brightness ensures that
the vector has a substantially unstopped appearance to
the eye.

The calculations described above are performed suf-
iciently rapidly so that, in the time allowed for calcu-
lation, up to eight of the sixteen vectors can be pro-
cessed. Thus, although the system has capacity for dis-
playing up to sixteen vectors in one frame of the raster,
only up to eight of these vectors may appear in any
given linescan.

After the above calculations have been performed
for all the vectors in the current linescan, the memory
M2 is read out in synchronism with the execution of the
linescan by the cathode ray tube.

Read-out of the memory M2 is controlled by a clock
28 running at 14 megahertz (the frequency corre-
sponding to the division of the vector writing area of a
linescan into 512 elements). Output pulses from the
clock 28 are counted by an address counter 29, which is reset to zero at the point of reentry into the vector writing area on the linescan.

The memory M2 is read out by addressing it with the contents of the address counter 29, via the data selectors 25 and 27. Thus, each of the 512 words in the memory M2 is addressed at the instant at which the corresponding element of the vector writing area on the screen is scanned.

The output from the memory M2 is fed to three up/down counters 30, 31, and 32 which correspond respectively to the three video levels one-third B.U., two-thirds B.U., and full B.U. These counters are all initially set to zero.

If a word addressed in the memory M2 contains a "1" at any one of its ON bit positions B1, B2 or B3, the corresponding up/down counter is incremented by one. Similarly, if a word addressed in the memory M2 contains a "1" at any one of its OFF bit positions B4, B5, or B6, the corresponding up/down counter is decremented by one. Thus, if the contents of one of the counters 30, 31 and 32 are zero, this indicates that the corresponding video level should be off, since an equal number of ON and OFF signals have been read from the memory M2. Conversely, if the contents of one of the counters 30, 31 and 32 is non-zero, this indicates that the corresponding video level should be on, since more ON signals than OFF signals have been read from the memory M2. (It will be recalled that when writing into the memory M2, precautions were taken to ensure that each ON bit was always associated with a corresponding OFF bit.) As each word in the memory M2 is read out, the contents of that word are reset to zero, to prepare the memory M2 for the next linescan.

The digital output from the up/down counters 30, 31 and 32 is converted by a video output circuit 33 into a video output signal having one of the three discrete levels one-third B.U., two-thirds B.U. and full B.U., according to which of the counters 30, 31 and 32 have non-zero outputs. If more than one of the counters have non-zero outputs, the higher video level will suppress the lower one. The video output signal from the output circuit 33 is fed via output lead 34 to the cathode ray tube 6 via the mixer 5 (FIG. 1).

Conveniently, the video store M2 may be divided into two halves, each containing 256 words of 6 bits in length. When writing into the memory, the two halves of the memory are addressed alternately, so that adjacent video elements are stored in different halves of the memory. Similarly, when reading out of the memory, the two halves are addressed alternately. This enables faster read out speeds to be achieved than would be possible with an unidivided memory.

In a modification of the vector generator 3, the video store M2 may be replaced by two stores, each of which is capable of storing a complete linescan of video. In operation, while the contents of one of these stores are being read out, the vector generator performs calculations to update the contents of the other store for the next linescan. In the alternate linescans, the roles of the two stores are interchanged. It will be seen that with this arrangement the time available for calculation is greatly increased, comprising a whole line-scanning period (64 microseconds) of the raster in addition to the flyback period. Therefore, many more vectors can be processed and displayed simultaneously (typically up to 40).

In another modification of the vector generator 3, the video store M2 may be replaced by an associative (i.e., contents-addressable) memory.

Referring to FIG. 5, the video store M2 in this modification comprises an associative memory divided into six stores, two for each level of bright-up. In the drawing, only the two stores 40, 41 for the one-third B.U. level are shown.

Each store contains a number of words, each eight bits in length. The contents of each word in the store 40 represent the X-coordinates of the starting point of a given one-third B.U. level, while the contents at the same address in the store 41 represent the X-coordinates of the end points of that one-third B.U. level.

During read-out of the video store M2, the output of an address counter 43 (corresponding to the counter 29 in FIG. 3) is applied to the "associate" inputs 48 of the stores 40 and 41. When a match occurs between the input word and the contents of a word of either store, an output signal appears at the "associate" output corresponding to that word, indicating that a one-third B.U. level starts or finishes at that instant.

The output from each word of the store 40 is fed to one side of a corresponding bistable circuit 44, while the output from the corresponding word in the store 41 is fed to the other side of the same bistable circuit 44. The bistable circuits 44 are clocked at the read-out rate (14 megahertz) of the system by means of a clock 46. Thus, the state of a given bistable circuit 44 at any instant signifies whether the corresponding one-third B.U. level is ON or OFF at that instant.

The outputs of all the bistable circuits 44 are combined in an OR gate 45. An output of "1" from the OR gate 45 indicates that the video element which is currently being scanned is at the one-third B.U. level.

The arrangement shown in FIG. 5 is repeated for the two-thirds B.U. level and the full B.U. level.

The use of an associative memory for the video store M2 has the advantage that it is not necessary, as in the arrangement of FIG. 3, to sort the input data into order of decreasing tan θ, since overlapping vectors do not present any special problem in this case, each vector being stored in a separate word of the video store. In addition, it is not necessary to use up/down counters for the output of the video store.

Referring again to FIG. 3, the two shading bits in store S10 of the input memory M1 are used to allow a one-third B.U. shading between two selected vectors. The two bits provide four different codes: one code is used to signify which vector turns the shading on, another to signify which vector turns the shading off. A third code signifies that there is no shading associated with the relevant vector.

If the shading bits of a given vector are set to the fourth code, the bright-up of that vector is suppressed completely. This provides a facility whereby all the necessary parameters for a vector can be stored in the input memory M1, but simply suppressed by means of one computer input, until it is required to display that vector.

The vector generator 3 can be used to generate curves, by "stringing" a series of vectors together i.e., by arranging for a series of vectors to follow on one from another, the finishing point of one being the starting point of the next. The larger the number of vectors, the smoother the curve.
In order to enable a large number of vectors to be generated for stringing together, the input memory M1 may be enlarged. However, provided the strung vectors do not double back on themselves they only take up approximately the calculation time required for a single vector.

Stringing of a large number of vectors can alternatively be achieved, without enlarging the input memory, by adjusting the value of \( \tan \theta \) of a vector during the course of a frame of the raster. However, this is only possible where data can be read in from the computer 1 during line scanning periods as well as during frame flyback.

The circle generator 4 (FIG. 1) is similar in principle to the vector generator 3, and comprises an input data store, into which parameters of one or more (up to three) circles can be written from the interface 2 during frame flyback periods, a video store in which a line of video can be stored, and digital calculating circuitry which calculates, during line flyback, from the input parameters of each circle, the video signal required to display that circle, the result being used to update the video store.

The input parameters for each circle comprise:

- \( X_0 \): the X-coordinate of the centre of the circle
- \( Y_0 \): the top line of the circle
- \( R \): the radius of the circle
- \( R^{1/2} \): the square root of R.

For each circle, calculations commence at linescan \( Y_0 \) and terminate at linescan \( Y_0 + 2R \). Each linescan containing the circle comprises two brightened-up portions, corresponding respectively to the left and right sides of the circle. The X-coordinates of these brightened up portions are equal to \( X_0 \)-offset and \( X_0 + \) offset respectively. The value of the "offset" is stored in a register, and is modified by the digital calculating circuitry between linescans.

The value of the offset is calculated for each line as follows:

Let \( R \) be the radius of the circle, measured in units of the distance between adjacent lines in the same (odd or even) field.

Let \( N \) be the number of the current linescan, counting from the linescan \( Y_0 \).

Then the new offset \( X_{N+1} \) for the current linescan can be calculated approximately from the offset \( X_N \) of the previous linescan by the equation:

\[
X_{N+1} = X_N + R - N/X_N
\]

(assuming that \( R \) is very much greater than unity).

The initial value of the offset in linescan \( Y_0 \) is taken as equal to \( R^{1/2} \).

In the circle generator, only one level of bright-up is used, it being found that this is adequate to produce a circle of unstepped appearance to the eye.

As in the case of the vector generator, the circle generator may be modified so that the video store can hold two linescans of video at a time. Calculations can then be performed to update one linescan while the previous linescan is being read from the video store. Therefore, the whole line scanning period is available for calculation, which increases the capacity of the generator from three to twelve circles.

Referring now to FIG. 6, this shows a possible circuit configuration for the video output circuit 34 of FIG. 3.

Each of the up/down counters 30, 31 and 32 is arranged to produce a "1" output when its count is non-zero (i.e., when the corresponding level is ON) and a "0" output when its count is zero (i.e., when the corresponding level is OFF). The three output signals from the up/down counters are fed to respective potentiometers 60, 61 and 62 which produce respective output voltages proportional to the one-third, two-thirds and FULL B.U. brightness levels. The voltages across these potentiometers are added together by way of resistors 63-66 and applied to an amplifier 67, the output of which is connected to the video output lead 34.

The presence of a "1" at the output of the two-thirds B.U. counter 31 is arranged to suppress any "1" from the one-third B.U. counter 30, by means of an inhibiting "0" from an inverter 68 applied to AND gate 69. Similarly, the presence of a "1" at the output of the FULL B.U. counter 32 is arranged to suppress any "1" from the one-third B.U. counter 30 or the two-thirds B.U. counter 31, by means of an inhibiting "0" from an inverter 70, applied to AND gate 69 and AND gate 71.

In this way, higher brightness levels are made to suppress lower brightness levels, as mentioned previously. Referring now to FIG. 7, the tan \( \theta \) sort logic circuit 11 (FIG. 3a) will be described in greater detail.

FIG. 7 shows the next address store 58 and the tan \( \theta \) store 54 of the data store M1, which have already been described. As previously mentioned, up to sixteen vectors are stored in the sixteen words of the data store M1, these vectors being arranged in a sequence in order of decreasing \( \tan \theta \), by virtue of the next address portion of each word which gives the address of the following vector in the sequence. One bit 8' of the next address store is designated the "next address valid" (NAV) bit, and is set to "1" for each vector in the store, except the last vector in the sequence, thus enabling that last vector to be readily identified. Any one of the sixteen words in the data store M1 can be accessed by applying its address to input line 100, and in this way the parameters of new vectors can be written into the store, or those of existing vectors can be updated.

As already described, when it is desired to wirte new parameters into the data store M1, these parameters are applied to interface 2 (FIG. 3a). These parameters appear on data highway 102. The address of the vector into which the parameters are to be written is entered into a register 104 (referred to as the N register), while the new value of \( \tan \theta \) for this vector is entered into a register 106 (referred to as the \( \tan \theta \) register), whence it can be written into the tan \( \theta \) store 54 of the vector.

Data can be applied to the address input line 100, by way of a data selector 108, from either register 110 (referred to as the address register), or from the N register 104.

The logic circuit 11 also contains the following registers, each of which has storage space for a vector address:

- a. Register 112 (referred to as the A register), which is used to hold the address of the vector in the store M1 that has the largest \( \tan \theta \) value, thus enabling the start of sequence of vectors to be readily identified. When it is required to update this register, the contents of N register 104 can be written into it, as will be described.
b. Register 114 (referred to as the S register), which can be written into from either the address register 110 or the N register 104.

c. Register 116 (referred to as the P register) which can be written into from either the address register 110 or the S register 114.

d. Register 118 (referred to as the T register) which can be written into from address register 110.

e. Register 120 (referred to as the L register) which can be written into from the address register 110 or from the next address store 58 of the vector that is currently addressed over line 100.

The contents of the address register 110 and N register 104 can be compared in address comparator 122, the output of which can be used to set bistable circuit G or H. Similarly, the contents of tan θ register 106 and the tan θ value of the currently addressed vector in the store M1 can be compared in tan θ comparator 124, the output of which can be used to set bistable circuits J or K.

The address register 110 can be written into by way of data selector 126 from either P register 116 or from the next address store of the currently addressed vector in M1. This next address store can, in turn, be written into via data selector 128, either from L register 120 or from N register 104. The NAV bit of the currently addressed vector can be used to set bistable Z.

Movement of data between the various registers in FIG. 7 is controlled by means of a control logic circuit 130, which is a conventional control logic circuit comprising conventional logic components (AND gates, OR gates, bistable circuits etc.).

Referring to FIGS. 8a, b and c, these show a flow diagram characterizing the control logic circuit 130, showing in detail each step of the operation of the tan θ sort logic circuit. In these Figures, the symbol := has its usual meaning thus, for example, the statement

\[ \text{P REG.} := \text{S REG.} \]

symbolizes that the contents of the S register are written into the P register.

Operation of the tan θ sort logic circuit 11 will now be described with reference to FIGS. 7, 8a b, and c.

Operation commences when new data appears over the highway 102, specifying new parameters, and specifying the address of a new vector, or of a vector to be updated. Box 200 (FIG. 8a) reads this address from the data highway into the N register 104, and box 202 then reads the tan θ value from the highway into the tan θ register 106. Box 204 then writes the contents of the A register 112 (i.e., the address of the vector currently having the largest tan θ in the memory M1) into the address register 110. Box 206 resets all the bistables G, H, J, K, and Z.

The following loop is then performed. Box 208 examines the NAV bit of the address register 110. Assuming that this bit equals “1” (as it will usually, initially there is only one vector currently in the store) control passes to box 210, which operates data selector 108 applying the contents of the address register 110 to input line 100 of memory M1, whereupon the first vector in the sequence (i.e., having the largest tan θ value) is addressed. Box 212 examines bistable J, and since this bistable has been reset, control passes to box 214, which operates the tan θ comparator 124 to compare the new value of tan θ in register 106 with the tan θ value of the vector in the store M1 which is currently addressed (in this case, the first vector). If the new value is smaller than the current value, box 216 sets bistable K, and at the same time loads the contents of the address register 110 into the S register 114. If, on the other hand, the new value is larger or equal to the current value, box 218 sets bistable J, and loads the contents of the address register 110 into the T register 118. Box 220 then operates data selector 126 to write the contents of the next address store of the currently addressed vector (including the NAV bit) into the address register 110, and returns control to the start of the loop. This loop is thus repeated, for each vector in the sequence in turn, in order of decreasing tan θ, until a vector is found which has a tan θ value smaller than or equal to the new tan θ value. When this happens, bistable J is set by box 218, and the loop is short-circuited at box 212. Finally, when the last vector in the sequence is reached, box 208 finds that NAV = 0, and the loop is therefore broken.

It will be appreciated that at this point, the sort logic circuit has determined the exact point in the sequence of vectors at which the new tan θ value in tan θ register 106 should be placed. Thus, at this point, the S register 114 contains the address of the vector in the sequence having the next highest value of tan θ, and the T register 118 contains the address of the vector having the next lowest value of tan θ. If bistable J is not set, this indicates that the new tan θ is smaller than all the existing tan θ values, while if bistable K is not set, this indicates that the new tan θ is larger than all the existing tan θ values.

Concurrently with execution of this loop, a second loop is performed, as follows: Box 222 examines bistable G. Initially, this bistable is reset, so that control passes to box 224, which operates address comparator 122 causing the contents of the N register 104 to be compared with the contents of the address register 110. If the two addresses are unequal (as will usually be the case), box 226 writes the contents of the address register 110 into the P register 116, and sets bistable H. If, on the other hand, the two addresses are equal (indicating that the new parameters are intended to update the parameters of the vector currently addressed vector) control passes to box 228. This box checks whether the NAV bit of the currently addressed vector is zero, and if this is the case, bistable Z is set by box 230, indicating that the new parameters are intended to update the parameters of the vector which is currently the last in the sequence. Generally, however, the box 228 will detect that the NAV bit is non-zero, whereupon control will pass to box 232, which sets the bistable G and writes the next address bits of the currently addressed vector into the L register 120.

This second loop is repeated, in synchronism with the first-mentioned loop, so that the address in the N register 104 is compared with the address of each vector in the sequence in turn until either (a) a match is found, whereupon bistable G is set and the second loop is short-circuited by box 222, or (b) the end of the sequence is reached when input line 100 of memory M1 is examined.

In the case where a match has been found, it will be seen that the P register 116 contains the address of the vector immediately preceding the vector which is to be updated, while the L register 120 contains the address of the vector which immediately follows the vector to be updated. If bistable Z is set, the vector to be updated is the last in the sequence, while if bistable H is not set,
the vector to be updated is the first in the sequence, both of which cases have to be treated specially as described below.

When eventually the last vector in the sequence has been reached, an exit is made from box 208 to box 234 (FIG. 8B). Box 234 loads the contents of the P register 116 into the address register 110 via data selector 126, and box 236 then loads the contents of the S register 114 into the P register 116, and the contents of the N register 104 into the S register 114.

Box 238 then examines bistable H. If this bistable is not set, indicating the special case where the first vector in the sequence is to be updated, control passes directly to box 240. Generally, however, bistable H will have been set, and control will pass to box 242, which examines bistable G.

If bistable G is set, indicating that one of the vectors to be updated (other than the first or the last vector) control passes to box 244, which causes the contents of the L register 120 to be written into the next address store of the currently addressed vector, via data selector 128. It will be recalled that the address memory 110 was loaded, by box 234 above, with the address of the vector immediately preceding the vector to be updated. Thus, box 244 effectively removes the vector to be updated from the sequence, and "closes up" the gap left by it, by writing the address of the following vector into the next address store of the preceding vector. Control then passes to box 240.

If, on the other hand, bistable G is not set, control passes to box 246 which examines bistable Z. Bistable Z being set indicates the special case where the last vector in the sequence is to be updated, and control therefore passes to box 248, which acts to reset bistable Z and to set the NAV bit of the vector which is currently addressed to zero. This effectively removes the last vector from the sequence, and terminates the sequence on the preceding vector, by setting the NAV bit of the preceding vector to zero. Control then passes to box 240, as it does if box 246 finds that bistable Z has not been set.

The following part of the operation deals with placing (or reacing, as the case may be) the new or updated vector into the sequence of vectors, at the appropriate point as determined by its new tan \( \theta \) value.

Box 240 is operative to write the contents of the P register 116 into the address register 110, via data selector 126, and to write the contents of the S register 114 into the P-register 116. It will be appreciated from the above that the register 116 now contains the address which was originally in the N register 104 (i.e., the address of the vector whose parameters are to be entered into the memory M1, or updated) and the address register 110 now contains the address which was originally in the S register 114 (i.e., the address of the vector having the next largest tan \( \theta \) value to the new tan \( \theta \) value).

Box 250 examines bistable K. Generally, bistable K is set, and control therefore passes to box 252, which acts to write the contents of the N register 104 into the next address store 58 of the currently addressed vector via data selector 128. In other words, the address of the new (or updated) vector is written into the next address store of the vector with the next largest tan \( \theta \) value, thus effectively breaking the sequence of vectors at the appropriate point, to insert the new vector. In addition, the NAV bit of the currently addressed vector is set to "1" (although, in general, this bit will already be "1"). Control then passes from box 252 to box 254.

If, on the other hand, box 250 finds that bistable K is not set (signifying the special case where the new tan \( \theta \) value is larger than all the tan \( \theta \) values currently in the tan \( \theta \) store S4) control passes from box 250 to box 256, which acts to write the contents of N register 104 into A register 112, thereby updating the record of the address of the vector having the largest tan \( \theta \) value. Box 256 also acts to set the NAV bit of this vector to "1" and to reset bistable G. Control then passes to box 254.

In order to complete the insertion of the new (or updated) vector, it still remains to write the address of the vector having the next lowest tan \( \theta \) value into the next address store of the new vector, unless, of course, the new vector is to be placed at the end of the sequence. This is performed as follows.

Box 254 acts to write the contents of the T register (i.e., the address of the vector having the next lowest tan \( \theta \) value to the new tan \( \theta \) value) into the L register, provided the bistable G is not set (in which case the L register will already have been written into by box 232). Box 254 also writes the contents of the P register (containing the address of the new or updated vector) into the address register 110, via data selector 126.

Box 258 (FIG. 8C) then tests whether bistable J is set. If this bistable is not set (indicating that the new tan \( \theta \) value is smaller than all the values currently in the memory M1) control passes to box 260, which acts to write the new tan \( \theta \) value from tan \( \theta \) register 106 into the tan \( \theta \) store of the currently addressed vector, and to set the NAV bit of that vector to zero. This completes the operation of the tan \( \theta \) logic circuit for this special case.

If the bistable J is set, control passes from box 258 to box 262, which tests bistable Z. If bistable Z is set (indicating that the new tan \( \theta \) value is updating the last vector in the sequence) control again passes to box 260, as above.

If bistable Z is not set, control passes to box 264, which tests the bistable G. If bistable G is set (indicating that the new tan \( \theta \) value is updating one of the vectors already in the store, other than the first or last) control passes to box 266, which acts to write the new tan \( \theta \) value from tan \( \theta \) register 106 into the tan \( \theta \) store of the currently addressed vector, and sets the NAV bit of this vector to "1". The next address store of this vector has, in this case, already been written into by box 244 above.

If the bistable G is not set, control passes from box 264 to box 268, which has a similar effect to box 266, but in addition causes the contents of the L register to be written into the next address store of the currently addressed vector, via data selector 128. This completes the entry of the new vector into the correct position in the sequence.

We claim:

1. Apparatus for generating a video signal for input to a raster-scanned display device to cause the device to display at least one graphical symbol, the apparatus comprising: an input store for holding, in digital form, input parameters of said at least one graphical symbol; digital data-processing means comprising means for calculating start and finish time positions of at least one video brightness level, for processing said parameters to produce a digital output characterizing said video
17 signal; and at least one up/down counter for counting up when said digital output indicates a start of said video brightness level, and counting down when said digital output indicates an end of said level, together with means for producing a video signal having that brightness level only when the count of the counter is greater than zero, for converting said digital output into said video signal.

2. Apparatus according to claim 1 wherein said digital data processing means comprises: a video store adapted to hold information characterising a portion of said video signal corresponding to at least one linescan of the raster; computing means for performing calculations on said parameters in respect of each linescan of the raster in turn to produce said information for loading into said video store; and read-out means for reading said information out of said video store, on completion of said calculations in respect of a linescan, to produce the portion of said digital output corresponding to that linescan.

3. Apparatus according to claim 2 wherein said input store comprises a plurality of storage locations for respectively holding parameters of a plurality of graphical symbols, and said computing means comprises means for addressing each of said storage locations in turn to enable said calculations to be performed on the parameters of each of said graphical symbols in turn.

4. Apparatus according to claim 2 wherein said video store comprises a plurality of words of storage space, respectively corresponding to successive segments of a linescan of the raster, and said read-out means comprises means for addressing each of said words in turn, at a predetermined rate, and means for converting the contents of a said word when so addressed into a digital output.

5. Apparatus according to claim 4 wherein said computing means comprises means for calculating in which of said parameters of the linescan start and finish time positions of at least one video brightness level and means for loading digital codes signifying "start" and "finish" into the words of said video store corresponding to those segments.

6. Apparatus according to claim 5, wherein said input store comprises a plurality of storage locations for respectively holding parameters of a plurality of graphical symbols, and said computing means comprises means for addressing each of said locations in turn, in order of decreasing extent of the symbol in the direction of the linescan.

7. Apparatus according to claim 6, wherein said symbols are straight vectors, and said computing means comprises means for addressing said locations in order of decreasing tan θ, where θ is the angle between the vector and the direction perpendicular to the lines of the raster.

8. Apparatus according to claim 1 wherein said data processing means comprises means responsive to synchronisation signals from said raster scanned display device, for initiating the processing of said parameters, in connection with a linescan of the raster, during the linescan flyback period immediately preceding that linescan.

9. Apparatus according to claim 1 wherein said data processing means comprises means for storing a number defining the position of a portion of said symbol within a linescan of the raster, and means for performing a predetermined algorithm, according to said parameters, to update said number to its appropriate value for a succeeding linescan of the raster.

* * * * *