

12

**(21) Application number : 93305893.5**

⑤① Int. Cl.<sup>5</sup>: **G09G 5/14**

②② Date of filing : 26.07.93

**(30) Priority : 27.07.92 GB 9215949**

④3 Date of publication of application :  
**02.02.94 Bulletin 94/05**

⑧ Designated Contracting States :  
**DE FR GB IT**

71 Applicant : **QUANTEL LIMITED**  
**Pear Tree Lane**  
**Newbury Berkshire RG13 2LT (GB)**

(72) Inventor : **Searby, Anthony David**  
**7 Montgomery Road**  
**Newbury, Berkshire (GB)**  
 Inventor : **Cawley, Robin Alexander**  
**Autumn Cottage, Penwood**  
**Burghclere, Newbury, Berkshire RG15 9EN**  
**(GB)**  
 Inventor : **Ingram, Martin Stewart**  
**113 Alresford Road**  
**Winchester, Hampshire SO23 8JZ (GB)**

74 Representative : Whitten, George Alan et al  
R.G.C. Jenkins & Co., 26 Caxton Street  
London SW1H 0RJ (GB)

**(54) An electronic image processing system.**

57) An electronic image processing system comprises an image store (11), a controller (14) and a monitor (13). The image store (11) stores data representing plural images as a multiplicity of pixels with the data for each image being stored as a one dimensional data list in a sequence of consecutive store addresses. The controller is provided to control the reading of data representing image portions from the store for display at respective display areas on the monitor (13) by identifying in the store data defining each of the image portions and by dividing the monitor into plural vertical zones (Y1, Y2). Each vertical zone is defined by a vertical start position and a vertical length (YLEN) and has one or more associated horizontal zones (X1, X2, X3) defined by a horizontal start position and a horizontal length (XLEN). Together the vertical and horizontal zones define the areas in which respective image portions are displayed.

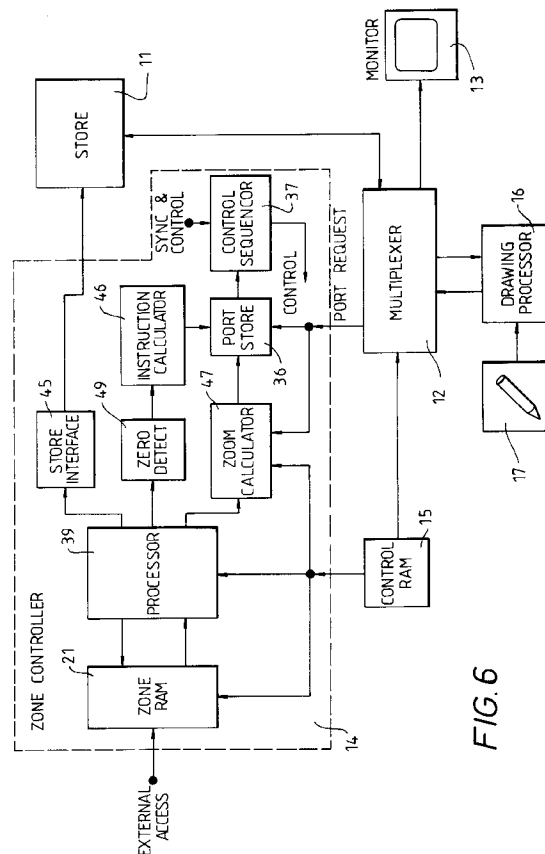


FIG. 6

The invention relates to an electronic image processing system.

In the art of electronic image processing, image stores are used to store data representing an image to be processed. The data usually represents a plurality of individual picture elements which together make up the picture or image to be processed. In early image processing apparatus a store would be provided having sufficient storage locations to store picture element data for only one image. The store was constructed to have the same number of horizontal and vertical storage locations as there were horizontal and vertical picture elements in the image. The limited number of storage locations available in early apparatus restricted the size of the image that could be stored therein.

As store technology has improved the cost of stores has fallen and at the same time the number of available storage locations has increased. Nowadays it is common practice to provide a store with a large number of storage locations for storing several images each represented by a relatively large number of picture elements.

Images to be processed are nearly always rectangles of one shape and size or another, and it is natural to consider the store to be rectangular and the respective areas within the store for each image to be of the same rectangular form as the image stored thereat. Thus, for example, an image defined by 1K x 2K picture elements would be stored in an area of 1K x 2K storage locations in the store with a direct correspondence between the positions of picture elements within the image and the respective positions of the corresponding storage locations at which the picture element data are stored in the store.

There is, however, a problem associated with the use of a large store to store several images, the problem being to optimise the way in which images are stored so that the maximum number of images are stored with the minimum amount of unused storage locations within the store. This problem will be appreciated from consideration of a simple example. Supposing a store comprises 3x3 storage locations and data representing an image of 2x2 image elements is to be stored therein. If the image data is stored in the store in the shape of the image then four adjacent storage locations will be used, say the first two adjacent locations in the first row and the first two adjacent locations in the next row of the store. Five locations will be available for the storage of other data, namely the last location in each of the first and second rows and all three locations in the third row. The available locations could be used to store data for say a 1x3 or 3x1 element image, but cannot be used to store data in the form of a second 2x2 image. Another store would have to be used to store data for the second image. Thus, it will be appreciated that the constraint of storing image data in a store in the same

form as the image tends to be wasteful of locations within the store.

Further discussion of this problem will be given with reference to Figure 1 of the accompanying drawings. In Figure 1 there is shown a store 1 in which there are stored several different images 2 to 8 of different size and rectangular shape. Store management programs are available which aim to store the images 2 to 8 in such a way as to reduce the number and size of unused areas of the store, such as areas 9, 10 and 11, created as a result of the difference in the sizes of the images 2 to 8. For example, the management program might be configured such that in the first instance images are placed as near as possible to the top left hand corner of the store 1 (as represented in Figure 1). The program might be further configured such that when an image is removed from the store the remaining images are moved around within the store to occupy the vacated locations. Thus, for example, the removal of say image 4 from the store 1 might cause the program to move image 7 and then move image 8 into the space previously occupied by image 7.

The problem of optimising storage locations is not one that can easily be solved. Where images of different shapes and sizes are stored in a single store there will often be gaps, such as gap 9 in Figure 1, between image data containing areas. The best utilization of store space for a given combination of different images can only be identified by first trying each and every possible combination of image positions within the store. There is no algorithm that will lead inevitably to the best solution though there are procedures, such as discussed hereinabove, available for providing a short cut to an optimum solution.

It is undesirable to spend a great deal of processing time in optimising the use of the store since that processing time could be more usefully spent in accessing the stored image data to produce the required processed images. A compromise between store size, store optimisation and processing time must therefore be reached in order to realise a store of acceptable cost, capacity and accessibility.

A related problem is that of controlling the transfer of data from an image store for display of plural images on a monitor. The problem of plural image display is considered in British patent application GB-A-2174277 (IBM) which describes a display system in which a display screen is divided into multiple display rows and information relating to the image or images to be displayed in each row, including a start address and a boundary indicator for each image, is held in a store. This system suffers from the disadvantage that a large amount of information needs to be stored to define a multiple image display on a display row by display row basis.

The problem is also considered in two British patent applications by Cadtrak Corporation, namely in

GB-A-2130857 and GB-A-2179227. These patent applications both describe a graphics display system in which multiple image displays are defined by a set of control word sequences each of which defines one scan line segment in one display area. Again this system suffers from the disadvantage that a large amount of data is required to define a multiple image display and this data in turn requires a large amount of storage space.

According to one aspect of the invention there is provided an electronic image processing system comprising: a store for storing image data representing a plurality of images as a multiplicity of pixels, the data for each image being stored as a one dimensional data list in a sequence of consecutive store addresses; a display screen having a multiplicity of display locations for displaying images represented by image data provided thereto; defining means for defining plural areas on the display screen in terms of data identifying plural vertical screen portions each having at least one associated horizontal screen portion and for identifying for each display area data for an image portion in terms of a start address in the store; calculating means for calculating for each display location from said horizontal and vertical screen portion data and said start address an address in the store at which data representing a pixel for display is stored; and reading means for reading the pixel data from the calculated store addresses and outputting the same to the display screen.

According to another aspect of the invention there is provided a method of image processing comprising: storing image data representing a plurality of images as a multiplicity of pixels, the data for each image being stored as a one dimensional data list; defining plural areas on the display screen in terms of data identifying plural vertical screen portions each having at least one associated horizontal screen portion; identifying for each display area data for an image portion in terms of a start address of said stored data; calculating from said horizontal and vertical screen portion data and said start address an address of the stored data at which data representing a pixel for display is stored; and reading from the calculated address pixel data and outputting the same for display.

The invention also provides an image processing system for displaying multiple images, the system comprising a controller for controlling the reading of data representing plural image portions from an image data store for display of the portions at respective areas on a monitor, the controller being arranged to identify in the image data store the data defining each of said image portions and to divide the monitor into a plurality of vertical zones each defined by a vertical start position and a vertical length and each having at least one associated horizontal zone defined by a horizontal start position and a horizontal length, the

horizontal and vertical zones together defining said areas in which respective ones of said multiple image portions are to be displayed.

By dividing the monitor into a plurality of vertical zones each defined by a vertical start position and a vertical length and each having at least one associated horizontal zone defined by a horizontal start position and a horizontal length, considerably less data is required to define a multiple image display as compared to the above discussed systems.

The above and further features of the invention are set forth with particularity in the appended claims and together with advantages thereof will become clearer from consideration of the following detailed description of an exemplary embodiment of the invention given with reference to the accompanying drawings.

In the drawings:

Figure 1 shows the layout of an exemplary store as already discussed hereinabove;

Figure 2 is a schematic diagram of an image processing system embodying the invention;

Figure 3 shows an exemplary image situation and the manner in which image data therefore is stored;

Figure 4 shows a second exemplary display situation and the manner in which data therefor is stored;

Figure 5 shows another exemplary display situation and the manner in which data therefor is stored;

Figure 6 is a schematic diagram showing the system of Figure 2 in greater detail;

Figure 7 illustrates the manner in which data is stored within the store controller; and

Figure 8 illustrates one sequence of operations performed by the store controller.

Turning now to Figure 2 of the accompanying drawings there is shown an image processing system 10 comprising a dynamic random access memory image store 11 for storing data relating to one or more images. The store 11 is configured as a one dimensional array of storage locations each uniquely identified by a numerical address ranging from 0 (zero) to a maximum  $2^n - 1$ , for example  $2^{32} - 1$  for a 32-bit addressing system. Image data from the store 11 is read for display of the image via a multiplexer 12 on a monitor 13. The reading of the data from the store 11 is controlled by a zone controller 14 and associated control RAM 15. The zone controller 14 controls access to the store 11 by controlling the addressing of storage locations within the store. The zone controller 14 comprises several ports, for example sixteen ports, each able to control access to a selected storage location within the store 11 for the reading of data therefrom and thereto.

The system 10 also comprises independently controlled drawing processor 16, similar to that de-

scribed in our British patent number 2,089,625, the teachings of which are incorporated herein by reference, and embodied in an image processing system marketed by us under the name PAINTBOX (Registered Trade Mark). The drawing processor 16 is thus responsive to user manipulations of a stylus/touch tablet device 17. The drawing processor 16 requests via the zone controller 14 data from the store 11 at locations determined by the position of the stylus on the touch tablet, modifies the data in order to change an image characteristic, e.g. colour, in the portion of the image represented thereby, and writes the modified data back to the same locations replacing the data previously stored there. The read-modify-write circuitry accesses the store via selected ports in the zone controller 14.

The system may be required to display on the monitor 13 a picture showing several images represented by the data stored in the store 11. Typically, the monitor 13 will be capable of displaying images to a high definition television standard (HDTV), for example 1250 lines at 50Hz. An image to be displayed may be any size. It may for example be significantly smaller, i.e. comprise fewer picture elements than display locations on the monitor screen, it may be represented by a number of picture elements identical to the number of display locations available on the monitor screen, or it may comprise significantly more picture elements (e.g. 10,000 x 8,000 picture elements) than there are display locations on the monitor screen. In other words the image may have a resolution which is less than, equal to, or greater than the resolution of the monitor screen.

It should thus be appreciated that several permutations of display are possible. For example, the whole image may be displayed at a reduced resolution over the whole or a portion of the monitor screen; a portion of the image may be displayed at full resolution over the whole or a portion of the monitor screen; or a portion of the image may be displayed at less than full resolution over the whole or a portion of the monitor. This list is far from exhaustive.

Turning now to Figures 3(a) and (b) of the accompanying drawings data representing an image 18 of width  $W_1$  pixels and height  $H_1$  pixels is stored in raster scan order in the store 11 at a sequence of storage locations beginning at start address  $S_1$  and ending at address  $(S_1 + H_1) \cdot (W_1 - 1)$ . The zone controller 14 in Figure 2 generates addresses so that image data is scanned out of the store 11 in raster order for display on the monitor 13. The simplest task required of the zone controller is to scan out the data for display of the whole image on the monitor 13. In this case the zone controller 14 will first address store location  $S_1$  (as shown in Figure 3(a)) which contains data for the pixel in the top left hand corner of the image (as shown in Figure 3(b)) and will then step through locations  $S_1 + 1$ ,  $S_1 + 2$ ,  $S_1 + 3$  until it reaches location

$(S_1 + H_1) \cdot (W_1 - 1)$  after which it will return to location  $S_1$  and repeat the scan.

Since the addressing of data is simply a matter of incrementing a counter by a fixed amount each time it is easy to reverse the addressing by incrementing the counter by a negative number. The effect of reversing the addressing in this way is to turn the image upsidedown when displayed on the monitor.

In the case where the image 18 is much larger, i.e. has a greater resolution, than the display screen of the monitor 13 then a cut-out image portion 19 of width  $XLEN$  and height  $YLEN$  pixels may be defined for display on the monitor 13. The cut-out 19 has a start address  $S_1'$  in the store 11 at which data corresponding to the pixel at the top left-hand corner of the cut-out is stored. The data at address  $S_1'$  is read first and then the address is incremented to  $S_1' + 1$  and then  $S_1' + 2$  and so on until the data representing the last pixel in the first line in the cut-out has been read from address  $S_1' + XLEN - 1$ . The data for the first pixel in the second line is stored at location  $S_1' + W$  in the store 11 and so this is the next location to be addressed in the store. (It should be noted here that the address is not  $S_1' + XLEN$ . This is because it is the data for the whole image 18 that is stored in scan order and not the data for the cut-out 19). Thus, each line of the cut-out 19 starts at an address one full width  $W_1$  of the whole image 18 away from the address of the start of the previous line in the cut-out. This process of reading line portions corresponding to the cut out 19 is repeated until the data for the bottom right hand pixel of the cut out has been read from store location  $S_1' + YLEN W_1 + XLEN - 1$ . All of the information necessary to perform the scan of the cut-out 19 is loaded into a memory within the zone controller as will be described in greater detail hereinafter.

It is often necessary to display multiple images simultaneously at respective portions of the monitor screen. For example, in our product PAINTBOX (Registered Trade Mark) a pop-up palette or menu area can be selected by the user to facilitate painting and other operations available in the machine. When the palette is selected the image is displayed in the top  $\frac{7}{8}$ ths of the screen, and the bottom  $\frac{1}{8}$ th of the screen contains the palette. Thus, the two parts of the screen contain wholly unrelated images. The simultaneous display of an image or image portion 20 and a palette or menu 20a at respective portions of the monitor screen is illustrated in Figure 4(a) of the accompanying drawings.

In order to handle the display situation shown in Figure 4(a) the scan is divided into two Y-zones  $Y_1$  and  $Y_2$ . The Y-zones have widths  $XLEN_1$  and  $XLEN_2$  and heights  $YLEN_1$  and  $YLEN_2$ . The widths  $XLEN_1$  and  $XLEN_2$  represent the number of display locations across the width of the monitor 13 and therefore  $XLEN_1$  and  $XLEN_2$  are equal to each other. The heights  $YLEN_1$  and  $YLEN_2$  together equal the

heights, i.e. number of lines, of the Y-zones on the monitor 13. The stored images 20, 20a to be displayed have start addresses  $S_1$ ,  $S_2$ , widths  $W_1$ ,  $W_2$  and heights  $H_1$ ,  $H_2$  and each stored image 20, 20a occupies its own sequence of addresses in the store 11 as illustrated in Figure 4(b) of the accompanying drawings. The zone controller is arranged to process the Y-zones in order, starting with zone Y1 and going on to zone Y2. More than two Y-zones may be defined if required. Assuming that the stored images 20, 20a are of the same size as their respective display areas, then the zone controller first scans the store starting at address  $S_1$  for display location [0, 0] and continuing through to address  $(S_1+H_1).(W_1-1)$  for display location [XLEN1-1, YLEN1-1], and then scans from address  $S_2$  for display location [0, YLEN1] to address  $(S_2+H_2).(W_2-1)$  for display location [XLEN1+XLEN2-1, YLEN1+YLEN2-1].

If, however, the stored images 20, 20a are larger than the areas in which they are to be displayed then a portion only of each of the two images will be required for display. Note that under these circumstances the values XLEN1, XLEN2, YLEN1 and YLEN2 will not all be the same as the corresponding sizes of the stored images  $W_1$ ,  $W_2$ ,  $H_1$ ,  $H_2$  respectively. The first pixel of the image 20 to be displayed will be identified by a separate start address  $S_1'$  and the first pixel of the image 20a will be identified by a separate address  $S_2'$  and the zone controller will scan the store in a similar manner to that described with reference to Figure 3 hereinabove. Thus, to produce the display shown in Figure 4(a) the zone controller starts scanning from  $S_1'$  for XLEN pixels, then proceeds to  $S_1'+W_1$  for a further XLEN pixels and then to  $S_1'+2W_1$  and so on until YLEN1 lines of pixels have been output. The zone controller then scans from  $S_2'$  for a further XLEN1 pixels (since  $XLEN2=XLEN1$ ) then moves to  $S_2'+W_2$  and so on for the remainder of the display.

Image data representing a picture will usually define the picture to a high resolution and thus the image will appear crisp and sharp when viewed. The same level of resolution is however not required for defining images representing menus, palettes, etc. and it therefore makes sense to define such images at low resolution in order to free storage space within the store 11 for other uses. The zone controller is therefore arranged to be able to control the outputting of image data from the store such that the displayed image can be an enlarged version of the stored image, multiplied by a magnification factor M. Each Y-zone has its own magnification factor M and is arranged so that  $M \times W$  for each zone is equal to the width of the display screen.

As will become clearer from the description hereinafter, the zone controller 14 of Figure 2 is further arranged so that one or more Y-zones can be divided horizontally into a number of X-zones. Referring now to Figure 5(a) of the accompanying drawings there is

shown an exemplary display configuration in which the screen of the monitor 13 is divided vertically in to two Y-zones Y1 and Y2 which extend across the width of the screen and have respective heights of YLEN1 and YLEN2 display lines. The first Y-zone Y1 is divided horizontally into three X-zones X1, X2, X3 having respective widths of XLEN1, XLEN2, XLEN3 display points. The second Y-zone is formed as a single X-zone having a width of XLEN4 display points corresponding to the width of the screen. Therefore the display point at the top left-hand corner of the first X-zone X1 is positioned at the top left-hand corner or origin [0,0] of the screen. The display point at the top left-hand corner of the second X-zone X2 is positioned at screen coordinate [XLEN1, 0] and similarly the display point at the top left-hand corner of the third X-zone X3 is positioned at screen co-ordinate [XLEN1+XLEN2, 0]. The bottom right hand corner of the third X-zone X3 is positioned at screen coordinate [XLEN1+XLEN2+XLEN3-1, YLEN1-1]. Likewise, the bottom right-hand corner of the fourth X-zone X4 is positioned at screen coordinate [XLEN4-1, YLEN1+YLEN2-1].

As shown in Figure 5(b) of the accompanying drawings each image or image portion is stored in the store as blocks of scan-sequential data and the zone controller 14 controls the reading of the data from the memory for display of the pixels represented thereby in the correct order to achieve the required display arrangement. There may be a one-to-one relationship between the displaying of the image and the data representing the image or the displayed image may be an enlarged version of the stored data in which case each pixel and each line will be replicated  $M_n$  times where  $M_n$  is the zoom factor M for the image occupying the nth X-zone. Likewise, each displayed image may be the whole of the respective shared image, similar to the case described hereinabove, with reference to Figure 4, or it may be a portion of a larger image, similar to the case described hereinabove with reference to Figure 3.

Returning briefly to Figure 3, it should be appreciated that in order to display the cut-out image 19 within the larger image 18 as shown, a total of five display areas 18a, 18b, 19, 18c and 18d must be defined by way of three Y-zones the middle one of which has three X-zones. This is because an X-zone 18b, 18c is required on each side of the cut-out image 19, (even though the two X-zones 18b, 18c contain part of the same picture as zones 18a and 18d) in order to divide the display correctly to enable the cut-out 19 to be inserted. It should be noted that the zoom values in zones 18b, 19 and 18c are the same because it is impractical to change the Y zoom in a line. Since zones 18b and 18c are part of the same picture as zones 18a and 18d and the picture must be continuous between all zones in which it is shown then the zoom factor must be the same for all five zones 18a,

18b, 19, 18c, 18d.

Figure 6 of the accompanying drawings shows the zone controller 14 in greater detail. As will have been appreciated from the foregoing description by those skilled in the art, the zone controller generates addresses which are used to access storage locations in the store 11 such that a screen full of pixels distributed in the store is assembled for display on the monitor. Each address is calculated either by adding an offset value to a base or start address  $S_n$  or by replacing the previous base address  $S_n$  with a new address  $S_{n+1}$ . Data defining the size of each zone and the location in the store 11 of respective images or image portions is stored in the control RAM 15.

As has already been mentioned hereinabove, the zone controller 14 comprises a plurality of ports, e.g. sixteen ports, in parallel. The ports are operable individually such that for example control data can be transferred to one port whilst another port controls access to the store 11. In the presently preferred configuration four ports, namely ports zero, one, eight and nine are used exclusively as video ports through which image data is transferred for display of the desired image or combined images on the monitor 13. Two of the ports, say ports zero and one, are used for the display of odd-numbered display fields and the other two ports, say ports eight and nine, are used for the even-numbered fields. This enables one port to be used for example to control the reading of data for one scan cycle while the other is being loaded with control data to control the reading of data in the next scan cycle. Other ports are selected as required for example to control the transfer of data in read-modify-write operations, with one port being used in the reading of data and another port being used in the writing of modified data. In order to reduce costs, ports are paired so as to share some common hardware. This means that both ports in a pair cannot be used at the same time, but they can be used in double-buffering operations, e.g. port two controlling loading of data while port ten controls the writing of data to the store 11 and vice versa. For the sake of clarity only the elements of one port are shown in Figure 6.

The zone controller 14 comprises an internal zone RAM 21 in which are stored sets of data for each port. The data sets contain all the information necessary to enable the above discussed and other display situations to be realised. The data sets are divided into separate blocks which in turn comprise plural data words which identify locations in the zone RAM 21 where other data words can be found and locations in the store 11 at which image data to be output to the monitor 13 is located.

During an initialising or set up procedure external circuitry (not shown) causes at least three data blocks, referred to herein as port block 22, Y-zone block 23 and X-zone block 24 as shown in Figure 7, to be loaded into the zone RAM 21. The data in the

zone RAM is then ready for use by the zone controller when a port is selected. There will be a Y-zone data block 23 for each Y-zone defined in the image and an X-zone data block 24 for each X-zone defined in the image. The port data block 22 comprises three fields of information 25-27. The first field 25 identifies the address in the zone RAM 21 at which the first Y-zone data block is located and the length of that Y-zone (YLEN1 in Figure 5). The second field 26 identifies the address in the zone RAM 21 at which the current Y-zone data block is located and the length of the current Y-zone, and the third field 27 identifies the address in the zone RAM 21 at which the current X-zone is located together with the length of the current X-zone (XLEN in Figure 5).

The Y-zone data block 23 only contains two data fields 28, 29 because it serves merely as a pointer to the data block for the X-zone. The first field 28 of the Y-zone block 23 identifies the address in RAM 21 at which the first X-zone data block for that Y-zone is located, together with the length of the first X-zone. The other field 29 identifies the address in the zone RAM 21 at which the next Y-zone data block is stored and the length of the next Y-zone.

An X-zone data block, e.g. block 24, contains all of the information needed to display an X-zone except for the zoom and scroll data which is defined separately. Thus the X-zone data block 24 contains one field 30 which points to the address in RAM 21 at which the next X-zone data block is held, together with data identifying its length. The second field 31 of the X-zone data block identifies an increment value which determines the size of address increments when addressing the store to move along the line of data. The third field 32 contains the current address, that is to say the storage location in the store 11 currently being addressed by the zone controller. The fourth field 33 contains the start address of the X-zone and the fifth and sixth fields 34, 35 respectively identify the length of each line in the picture to be displayed (XLEN in Figure 5) and the address in the store 11 at which the current line in the X-zone starts.

The zone RAM 21 provides up to 1k storage locations and thus has space for storing many separate port, X-zone and Y-zone data blocks for each of the ports provided in the zone controller. As shown in figure 6, the zone controller 14 comprises a port store 36 in which is stored, at the end of a port operation, instruction data identifying the tasks to be performed by the port next time it is requested. When a port is requested the port store 36 transfers its data to a control sequencer 37 which responds to the instruction data by outputting control signals which control operation of the elements forming the zone controller 14 thereby causing the zone controller to execute the required task.

Thus, it is the control sequencer 37 that enables the zone controller to control the outputting of image

data from the store. The control sequencer 37 is a microcontroller comprising a small microcode RAM (not shown) in which commands relating to different instruction sequences are stored. The instruction sequence executed by the control sequencer is selected depending on the length values in the various fields of the data blocks 22, 23, 24 discussed hereinabove with respect to Figure 7 of the accompanying drawings.

The task of the zone controller 14 is to supervise the accessing of storage locations within the store 11 and thereby to control the reading of data from the store 11 to the monitor 13 to achieve the desired display of multiple images thereon. In operation the zone controller 14 outputs, via the store interface 45, data representing the storage location of the present pixel required for display whilst calculating concurrently the storage location of the next pixel required for display.

Operation of the zone controller is defined by a total of eight different instruction sequences under the control of the control sequencer 37. The eight instruction sequences are described in greater detail hereinafter and comprise a start sequence, an X-step sequence, a new X-zone sequence, an initialize X-zone sequence, a next-line sequence, a new Y-zone sequence, an end sequence, and a repeat-line sequence.

#### START SEQUENCE

By way of example, the start sequence executable by the zone controller is represented by a sequence of data transfers in Figure 8 of the accompanying drawings. Turning now to Figure 8, the start, or initialising, sequence causes data to be loaded to the correct locations in the zone RAM 21 for subsequent tasks. The first operation 38 performed in this initialising or start sequence is to cause a processor 39 (see Figure 6) to read the data from the RAM 21 identifying the address of the first Y-zone data block 23 from field 25 of the port block 22. In step 40 the processor reads the data identifying the address of the first X-zone data block 24 from field 28 of the Y-zone block 23 in RAM 21. Once the data from field 28 has been read to the processor 39 the XLEN value in field 28 in the zone RAM is decremented by one in readiness for the next time that field is accessed by the processor 39. The YLEN value in field 33 is also decremented in readiness for the next time that field is accessed. In step 41 the data read from the field 25 is written into the second field 26 so that the current Y-zone is identified as the first Y-zone. Next, as represented by step 42, the start address in field 33 of the X-zone data block 24 is read into the processor 39. The start address is written to each of fields 32 and 35 in steps 43a and 44 so that the current X address value in the port block 22 and the current ad-

dress and the address in the X-zone block 24 are all set to the start address value of the first X block. The operation in step 43a of writing to the current address field 32 also causes the processor 39 to output the current address value to a store interface 45 (see Figure 6).

The store interface 45 is connected to the store 11 and provides addresses to enable image data to be read from addressed locations to the multiplexer 12 for modification by the drawing processor 16 or for display of the image represented thereby on the monitor 13.

Thus, the start sequence represented by Figure 8 initialises the data structures in the zone RAM at the start of a new display or drawing cycle. The start sequence only affects data relating to the first X and first Y-zones. Where there are other X and/or Y-zones the data for the or each other zone is initialised by other sequences as will be described in greater detail hereinafter.

It will be appreciated that the other operations that can be executed by the zone controller 14 are controlled by similar microprogramme controlled data transfers to and from the zone RAM 21. The construction of such microprogramme controlled sequences are well within the ability of those possessed of the appropriate skills and accordingly the following description of the other seven operation sequences performed by the zone controller will not be given in such great detail.

#### X-STEP SEQUENCE

The X-step sequence causes the zone controller to generate a series of image store addresses so as to travel along a line of image points in a single X-zone. In other words, the control sequencer 37 causes the zone controller to add the value XINC (field 31 in X-zone block 24) to the address output via the store interface 45 to the image store and at the same time decrements the X-length value (field 27) to enable the end of the X-zone to be detected when the value reaches zero. During this operation the control sequencer causes the XLEN value to be read from the zone RAM 21 and thereafter to be output from the processor 39 to a zero detect circuit 49 (see Figure 6). The control sequencer 37 causes the zero detect circuit 49 to monitor for an XLEN value of zero and when such a value is detected this means that at least the end of a line in an X-zone has been reached.

The zero detect circuit 49 also receives data relating to the YLEN (field 26 of port block 22) together with indications as to whether or not the current X and Y-zones are the last X and Y-zones. The zero detect circuit 49 passes the data it receives to an instruction calculator circuit 46 which analyses the data to determine which sequence should be executed next time by the zone controller.

### NEW X-ZONE SEQUENCE

If the XLEN value is zero and the current X-zone is not the last X-zone in the current Y-zone then here is another X-zone to be addressed and the instruction calculator 46 causes the port store 36 to load the control sequencer 37 with microcode so that the new X-zone sequence is executed by the zone controller. The new X-zone sequence is arranged to look at the next X-zone pointer (field 30 in X-zone block 24) and set that value as the current X-zone value. It also adds the value LineInc (field 34) to the X-zone current address (field 27) so that the current address is incremented by one lines worth of storage locations in the store 11. Thereafter, the above discussed X step sequence is executed by the zone controller under the control of the control sequencer 37.

### INITIALIZE X-ZONE SEQUENCE

Where the address to be generated by the zone controller corresponds to the start of the first line of a new X-zone in a new Y-zone the above discussed new X-zone sequence is not used. Instead, an X-zone initialising sequence is executed which, instead of adding the value of LineInc to the current address, copies of the start address from the X-zone data block (field 33) into the current address field 32 ready for use in the above discussed X step sequence.

From time to time the length data in Fields 26 and 27 is decremented. The XLEN is decremented with each step along a line and the YLEN is decremented with each line step. Thus, the length data provides an indication of the length remaining in the current X or Y-zone.

### NEXT LINE SEQUENCE

When the value of XLEN reaches zero in the previous cycle and the current X-zone is the last X-zone in the current line, this means that the end of the line has been reached. The zero detect circuit 49 and the instruction calculator 46 together detect this condition and cause the port store 36 to transfer to the control sequencer 37 microcode for generating addresses for the next line. The next-line control sequence identifies the first X-zone of the current Y-zone (from field 28) and then updates the value of the line address in field 35 from the address of the start of the previous line to the address of the start of the current line by adding to it the value LineInc from field 34.

### NEW Y-ZONE SEQUENCE

When, in addition to the XLEN value of X length being zero and the current X-zone being the last X-zone, the value of YLEN is also zero, then the end of the current Y-zone has been reached. The zero de-

tect circuit 45 and instruction calculator 46 respond by causing the port store 36 to load the control sequencer with microcode for setting up the zone controller for a new Y-zone. The new Y-zone sequence identifies the next Y-zone by reading field 29 in Y-zone data block 23 and writing this data to field 25 of block 22. The next-Y data points to a new Y-zone data block 23' which contains a first X field that points to the start address of the data block 24' of the line first X-zone in the new Y-zone. Thus, the new Y-zone sequence identifies the new Y-zone and its first X-zone and sets these up as the current Y and X-zones. The sequence also initialises data for the first X-zone by copying the start address in field 33 to the current X address field 27 and initialises the data for all X-zones in the Y-zone in a similar manner as the store addresses progress along this display line.

### END SEQUENCE

In the event that the values of XLEN and YLEN are both zero, the current X-zone is the last X-zone and the current Y-zone is the last Y-zone then the end of a display scan has been reached. At this time the zone controller does not know whether the same port or another port will be requested next time round and so an end sequence is executed which saves the contents of the current Y and current X fields 26, 27 and places the port in a standby condition until such time as the port is again requested, forcing the start sequence to be run.

### REPEAT-LINE SEQUENCE

There is one further sequence controlled by the control sequencer, namely, a repeat-line sequence which is executed during a zoom operation. The zone controller comprises a zoom calculator 47 which calculates when to repeat a line for a pixel-replicate vertical zoom by causing lines of addresses to be repeated. The zoom calculator 47 also calculates for a horizontal zoom whether new pixel data is needed or whether the previous pixel data is to be used again. The current XLEN is decremented and if new pixel data is required the current X address is recalculated.

The repeat-line sequence executed by the sequencer 37 overrides the next-line sequence which would otherwise be run. When the end of a line is reached the instruction calculator 46 decides that the next sequence to be performed should be the next-line sequence and causes the port store to pass the appropriate microcode to the sequencer. If, however, the zoom calculator 47 decides that the previous line should be repeated then the microcode selected by the instruction calculator is replaced by microcode defining the repeat-line sequence. The effect of the repeat-line sequence is to get the previous start of line value (field 35) from the zone RAM 21 and make



that the current address (field 32), and to decrement the current YLEN value in field 26.

The zone controller 14 co-operates with the multiplexer 12 (see Figure 2) in order to achieve horizontal pixel-replication zoom by sending horizontal scroll data from the control ramp to the multiplexer. The multiplexer 12 contains an accumulator (not shown) comprising a register which is loaded with a scroll value at the beginning of a line. The accumulator repeatedly adds a zoom value to the register and depending on the result decides whether or not the same pixel should be displayed more than once. Zoom calculations are carried out in the processor 39 at the beginning of a control sequence while the first values are being read from the zone RAM because during this period the processor is not occupied with zone calculations and therefore is free to be used for zoom calculations. An X-zoom is processed at the beginning of an X-step sequence, and a Y-zoom is processed at the beginning of a next-line sequence with the next line sequence being replaced by a repeat-line sequence if appropriate.

Together the above described instruction sequences dictate the operation of the zone controller 14 such that it controls the addressing of pixel data in the one dimensional data lists in the store 11 for display at display locations on a two dimensional display.

When a read-modify-write operation is to be performed by the drawing processor 16, the zone controller 14 serves simply to generate a patch of address values from the instantaneous position of the stylus on the touch tablet 17. Data from the control RAM is sent to the multiplexer 12 to tell it when to start and stop reading and writing to the addresses identified by the zone controller. None of the above discussed complex image data manipulations such as zoom need be performed by the zone controller during a read-modify-write operation and therefore there is no need to latch the control data in the zone controller.

Having thus described the present invention by reference to a preferred embodiment it is to be well understood that the embodiment in question is exemplary only and that modifications and variations such as will occur to those possessed of appropriate knowledge and skills may be made without departure from the spirit and scope of the invention as set forth in the appended claims and equivalents thereof.

## Claims

1. An electronic image processing system comprising:

a store (11) for storing image data representing a plurality of images as a multiplicity of pixels, the data for each image being stored as a one dimensional data list in a sequence of consecutive store addresses;

a display screen (13) having a multiplicity of display locations for displaying images represented by image data provided thereto;

defining means (15, 21) for defining plural areas on the display screen in terms of data identifying plural vertical screen portions each having at least one associated horizontal screen portion and for identifying for each display area data for an image portion in terms of a start address in the store;

calculating means (36, 37, 39, 46, 47, 49) for calculating for each display location from said horizontal and vertical screen portion data and said start address an address in the store at which data representing a pixel for display is stored; and

reading means (45) for reading the pixel data from the calculated store addresses and outputting the same to the display screen.

2. A system as claimed in claim 1, wherein said defining means (15, 21) is arranged to define said plural areas such that at least one of said vertical screen portions has a plurality of associated horizontal screen portions.
3. A system as claimed in claim 1 or 2, wherein said defining means (21) comprises a memory for storing said horizontal and vertical screen portion data for said plural areas and data representing said start address for each of said image portions.
4. A system as claimed in any preceding claim wherein said calculating means comprises a processor (39).
5. A system as claimed in claim 4, wherein said reading means comprises a plurality of ports each having an associated processor (39) and memory (21) to enable each port to calculate address values for reading of data from the store independently of other ports.
6. A system as claimed in any preceding claim, wherein the calculating means is arranged to calculate for each instantaneous screen position in a screen scan from said horizontal and vertical screen portion data an offset value identifying the instantaneous screen position and to calculate from said offset value and said start address the store address at which the pixel for display at the instantaneous screen position is stored.
7. A system as claimed in any preceding claim, wherein each image portion comprises a plurality of image lines and data representing each image line is stored as a respective data list in said store.

8. A system as claimed in claim 7, further comprising indicating means (49) for providing an indication when the calculated address in the store corresponds to the end of a line in an image portion; and determining means (36, 37, 46) responsive to said indication for determining the next calculation to be performed by the calculating means.
9. A system as claimed in claim 8, wherein said determining means comprises an instruction calculator (46), a control sequencer (37) and an instruction store (36) for storing data defining sequences of operations controlled by the control sequencer, the instruction calculator being responsive to said indications to select and transfer data from said instruction store to said sequencer.
10. A system as claimed in any preceding claim, further comprising a zoom calculator (47), said calculating means being responsive to the zoom calculator for calculating addresses so as to enable the reading means to read data representing a magnified version of the image portion from the store.
11. A system as claimed in any preceding claim, wherein at least one image portion corresponds to substantially the whole of an image defined by data in the store.
12. A system as claimed in any preceding claim, further comprising a drawing processor (16) and a user operable input device (17) for effecting user determined modifications to image data in the store, the input device being arranged to generate position data in response to user operation thereof and the calculating means being arranged to generate from the position data a patch of store addresses containing data for modification by the drawing processor.
13. A system as claimed in claim 12 as dependent on claim 5, wherein one of said ports is operable to calculate said patch of addresses.
14. A system as claimed in claim 12 or 13, further comprising a multiplexer (12) for multiplexing data between said store (11), said display screen (13), said defining means and said calculating means (14), and said drawing processor (16).
15. A method of image processing comprising:  
     storing image data representing a plurality of images as a multiplicity of pixels, the data for each image being stored as a one dimensional data list;  
     defining plural areas on the display screen
- in terms of data identifying plural vertical screen portions each having at least one associated horizontal screen portion;  
     identifying for each display area data for an image portion in terms of a start address of said stored data;  
     calculating from said horizontal and vertical screen portion data and said start address an address of the stored data at which data representing a pixel for display is stored; and  
     reading from the calculated address pixel data and outputting the same for display.
16. A method as claimed in claim 15, wherein said plural areas are defined such that at least one of said vertical screen portions has a plurality of associated horizontal screen portions.
17. A method as claimed in claim 15 or 16, further comprising storing said horizontal and vertical screen portion data for said plural areas and data representing said start address for each of said image portions in a memory.
18. A method as claimed in any of claims 15 to 17, further comprising calculating for each instantaneous screen position in a screen scan from said horizontal and vertical screen portion data an offset value identifying the instantaneous screen position and calculating from said offset value and said start address the address at which the pixel for display at the instantaneous screen position is stored.
19. A method as claimed in any of claims 15 to 18, wherein each image portion comprises a plurality of image lines and data representing each image line is stored as a respective data list.
20. A method as claimed in claim 19, further comprising providing an indication when the calculated address corresponds to the end of a line in an image portion; and  
     in response to said indication determining the next calculation to be performed.
21. A method as claimed in any of claims 15 to 20, further comprising calculating addresses repeatedly such that data representing a magnified version of an image portion is output for display.
22. A method as claimed in any of claims 15 to 21, wherein at least one image portion corresponds to substantially the whole of an image defined by the stored data.
23. An image processing system for displaying multiple images, the system comprising a controller

(14) for controlling the reading of data representing plural image portions from an image data store (11) for display of the portions at respective areas on a monitor (13), the controller (14) being arranged to identify in the image data store (11) the data defining each of said image portions and to divide the monitor (13) into a plurality of vertical zones (Y1, Y2) each defined by a vertical start position and a vertical length (YLEN) and each having at least one associated horizontal zone (X1, X2, X3) defined by a horizontal start position and a horizontal length (XLEN), the horizontal and vertical zones together defining said areas in which respective ones of said multiple image portions are to be displayed.

5

10

15

20

25

30

35

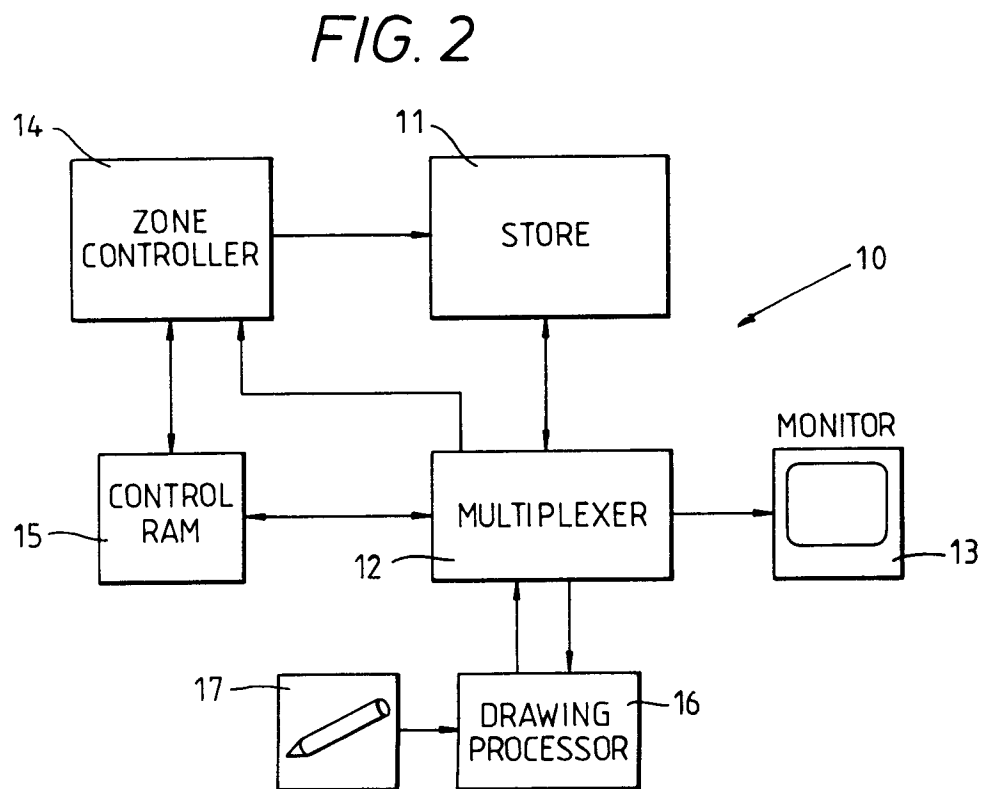
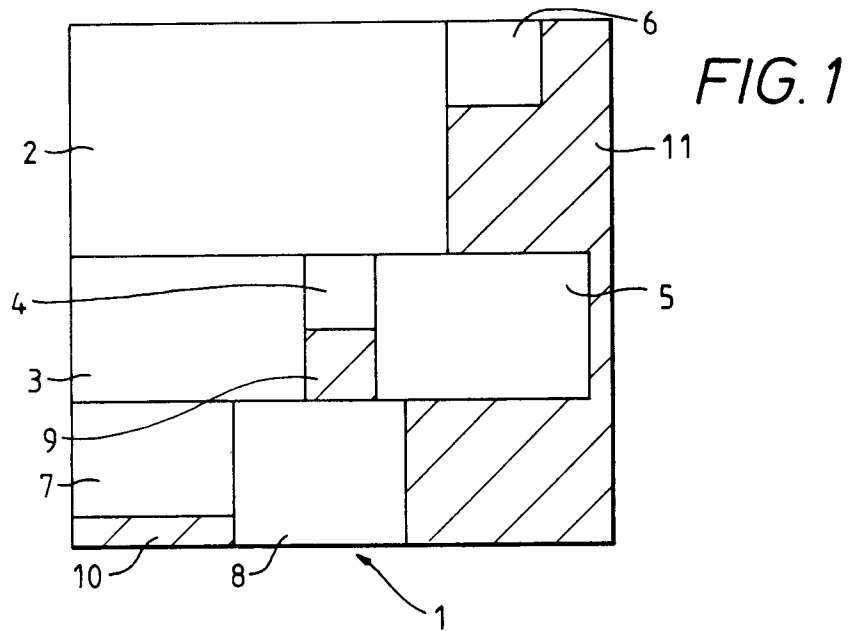
40

45

50

55

11



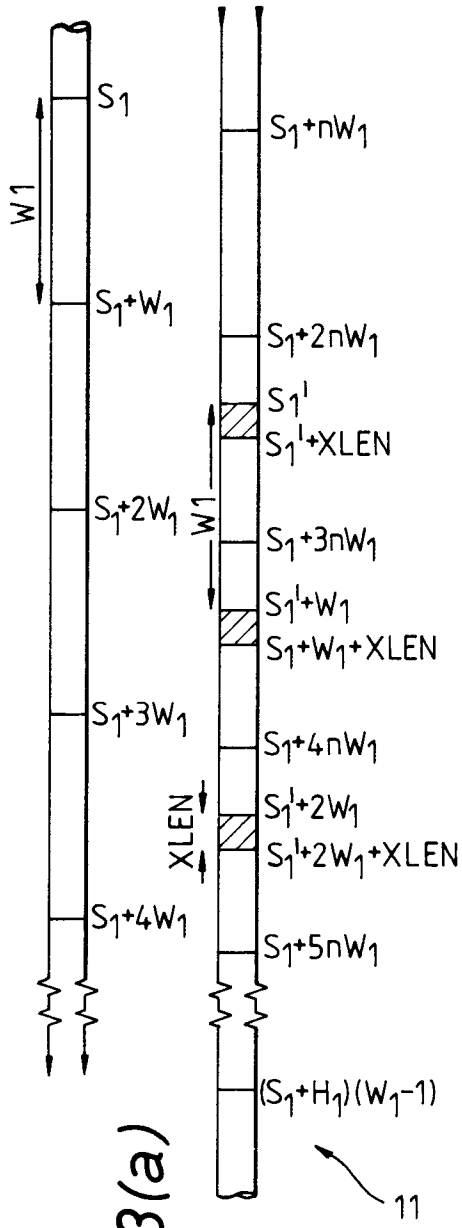


FIG. 3(a)

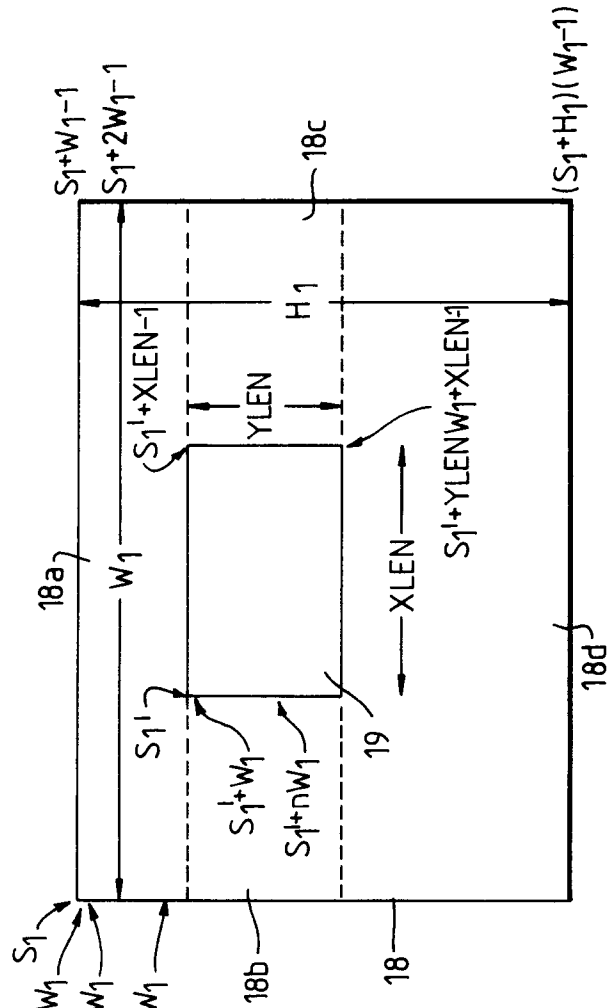
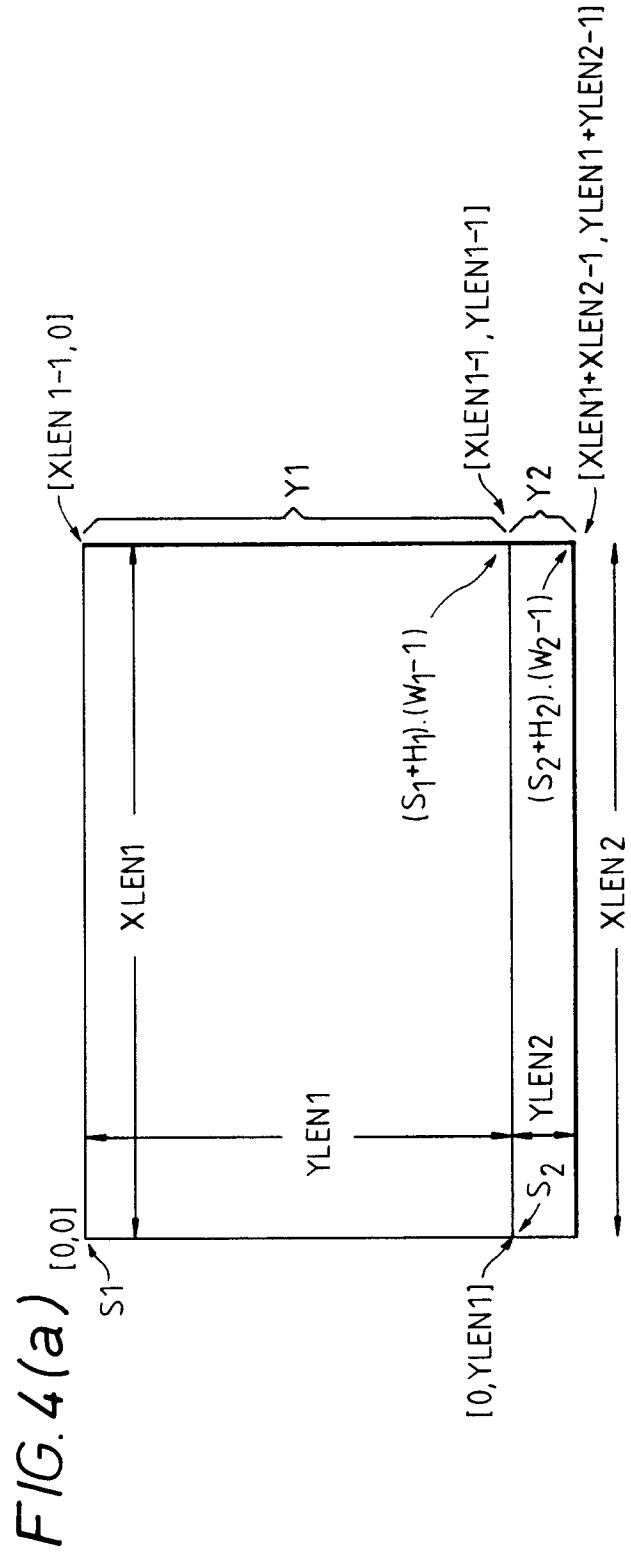
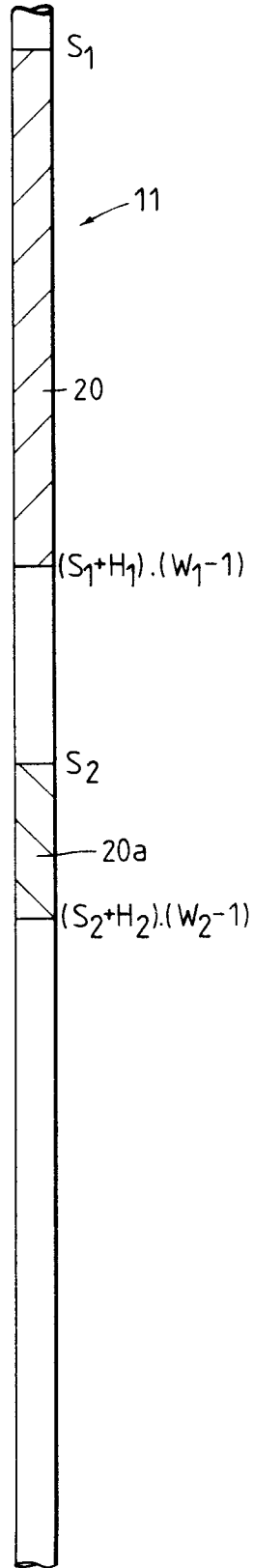
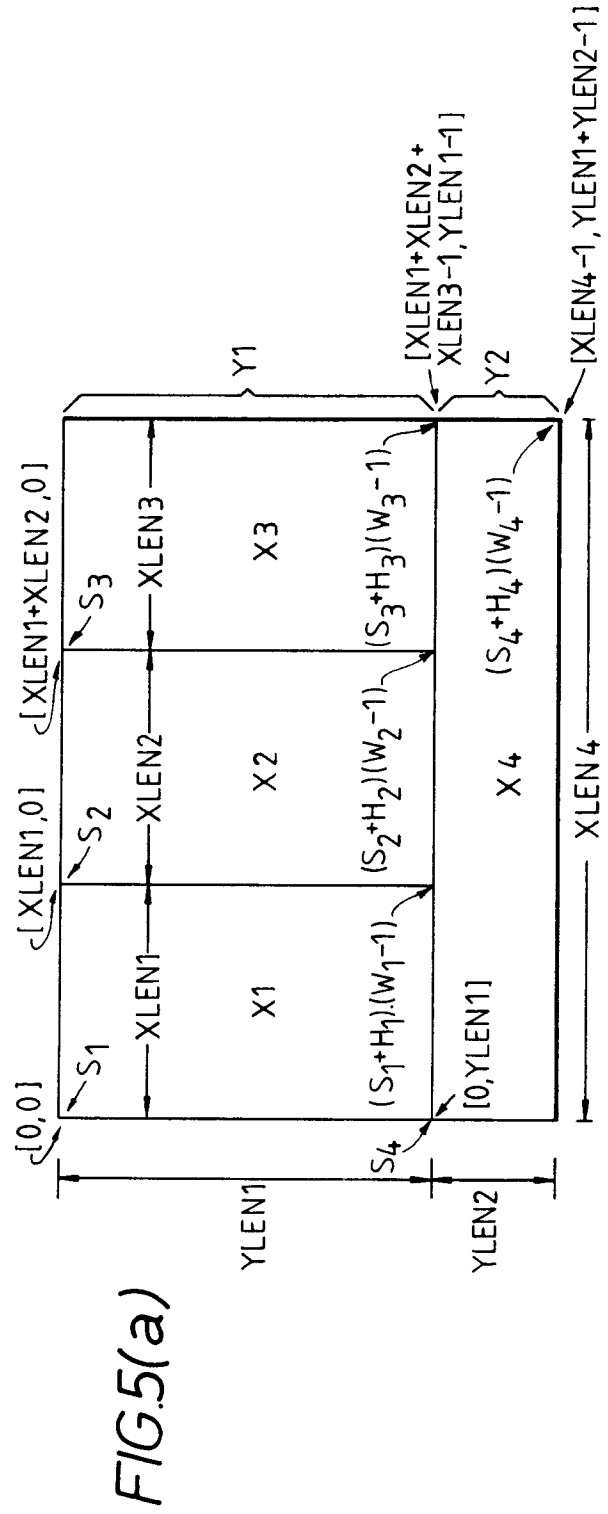
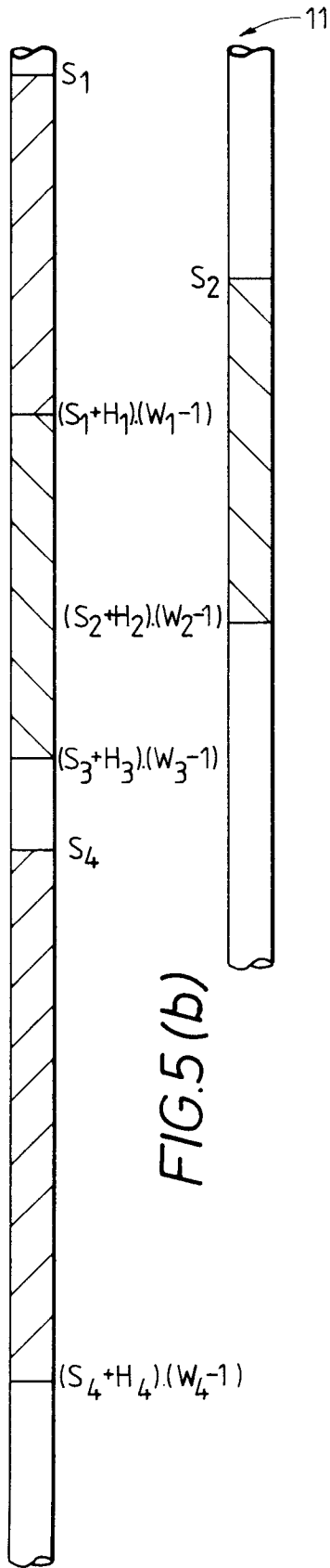


FIG. 3(b)





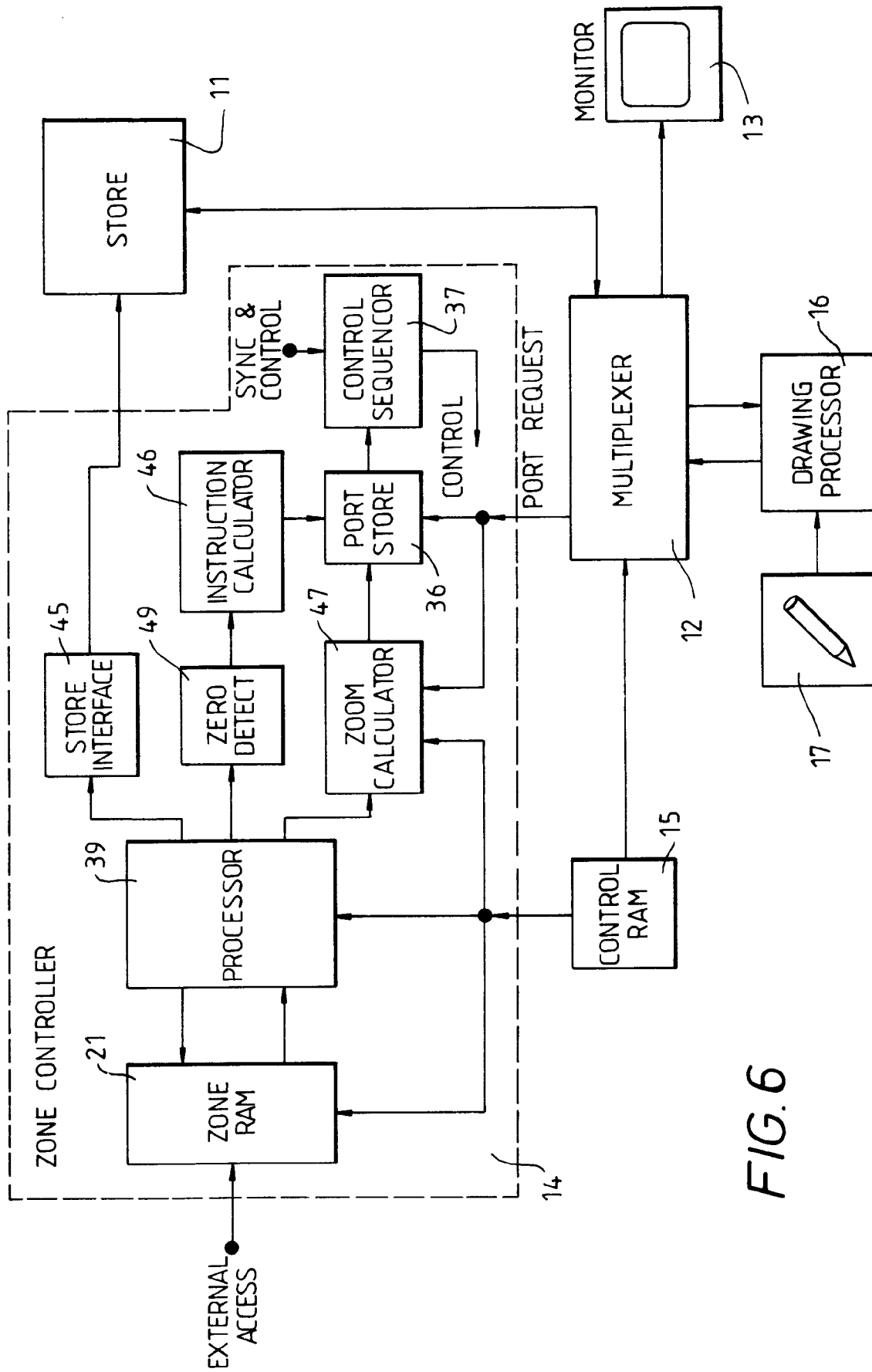


FIG. 6



FIG. 7

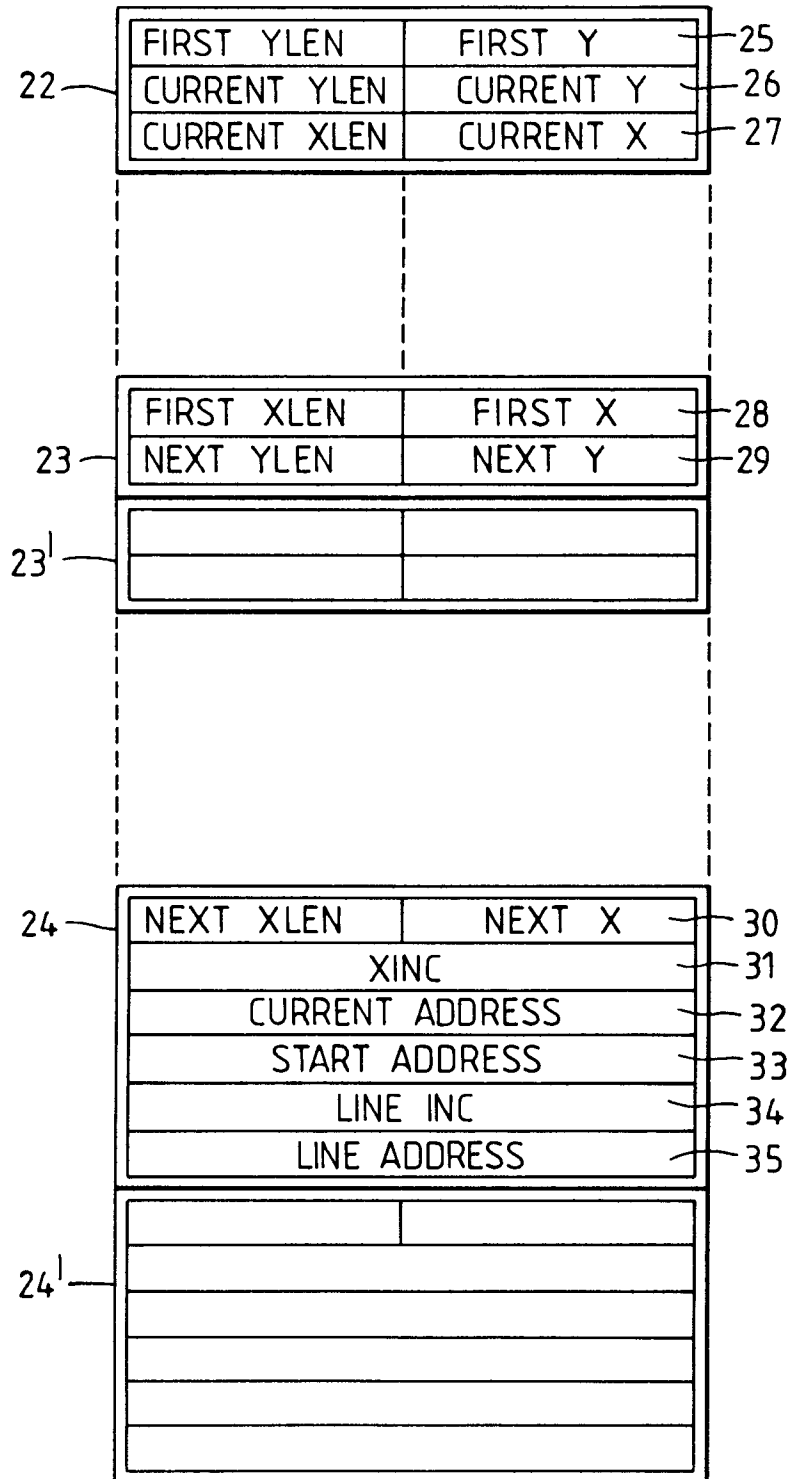


FIG. 8

