(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) Publication number: **0 482 746 A2**

(12) EUROPEAN PATENT APPLICATION

(21) Application number: **91307969.5**

(51) Int. Cl.⁵: **G09G 5/14**, G09G 5/06

(22) Date of filing: **30.08.91**

(30) Priority: **23.10.90 US 601835**

(43) Date of publication of application:
29.04.92 Bulletin 92/18

(84) Designated Contracting States:
DE FR GB IT

(71) Applicant: **International Business Machines Corporation**
Old Orchard Road
Armonk, N.Y. 10504(US)

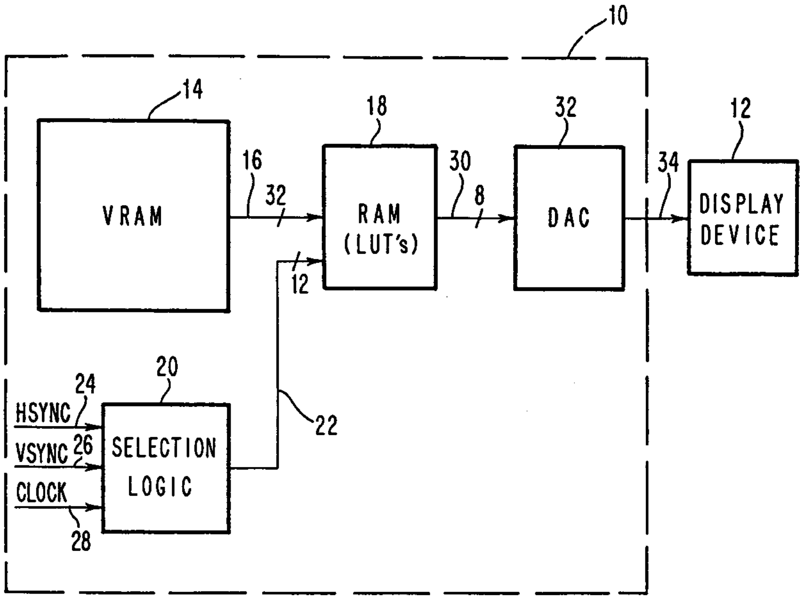
(72) Inventor: **Robertson, Taggart Huth**
381E John Joy Road
Woodstock, New York 12498(US)
Inventor: **Schanely, Paul Mark**
417 Old Route 209
Kingston, New York 12401(US)

(74) Representative: **Blakemore, Frederick Norman**
IBM United Kingdom Limited Intellectual Property Department Hursley Park
Winchester Hampshire SO21 2JN(GB)

(54) Multiple-window lookup table selection.

(57) A system for selecting from multiple palette lookup tables (LUTs) in a multiple-window system without requiring extra planes in video memory for storing palette information. X and Y position signals representing the current position on the screen are compared with the X and Y minima and maxima of each of the defined windows to determine whether the current position is within one or more windows. If the current position is within two or more windows, the window with the highest priority, as stored in a programmable priority register, is selected, and the lookup table is selected that is assigned to that window by a programmable lookup table register. Successive groups of pixels are processed in parallel to increase the speed of operation.

FIG. 1



EP 0 482 746 A2

Rank Xerox (UK) Business Services
(-/2.17/2.0)

This invention relates to a method and apparatus for selecting a data signal such as a palette identifier in a graphics system having multiple windows on a display screen.

In computer graphics systems, it is common to use an output device such as a cathode ray tube (CRT) to display data stored in digital form in a random access memory (RAM). For video graphics systems it is common to use a specialised RAM known as a video RAM, or VRAM (also referred to as a frame buffer) to facilitate the scanning out of serialised data. There is an entry in VRAM at a separate addressable location for each pixel, or picture element, corresponding to each dot or colour triad on the display screen. The VRAM data is converted by a digital-to-analog converter (DAC) to one or more analog outputs --typically, separate red (R), green (G) and blue (B) outputs in a colour system -- which are used by the display as intensity representations of the original VRAM data.

It is also common to provide intermediate translation of the VRAM data prior to the DAC operation. The translation of VRAM data from any value to any other value, within the limits of the system hardware, is accomplished by use of an additional RAM referred to as a lookup table (LUT), or palette. The address input to the LUT is provided by the pixel data originating from the VRAM. The new value for that pixel, which is stored in the LUT, is read out from the address location corresponding to the pixel data. The resulting output from the LUT is then used for the DAC operation. In a colour graphic system where red, green and blue are used as primary colours, the VRAM data can be converted, using a LUT, to a wide range of colour combinations.

A computer program or application controls the values stored or loaded in the LUT and does so independently from other applications. When more than one application is displayed on a CRT simultaneously, each application is confined to an area, or window, of display on the CRT. These windows are often overlapping in the sense that a given pixel may be common to two or more windows. In such a case, the windows are prioritised and it is the pixel data for the highest-priority window of two or more overlapping windows that is actually displayed on the screen. It is also common for a single application to use multiple windows. Multiple LUTs are used to allow the use of separate LUTs for separate application windows or for separate windows within a single application.

The assignment and selection of LUTs for each window on the CRT display can be accomplished by providing additional planes in the VRAM to store bits identifying the LUT to be used for each pixel. In such an arrangement, for each pixel stored in the VRAM there is an associated definition for the LUT to be used for that particular pixel. Because there is an associated cost for the additional planes that contain the LUT selection definition, it is apparent that a more cost-effective method of selecting LUTs is desirable.

The present invention now contemplates a method of selecting a data signal for a particular position on a display within a graphics system providing multiple windows on a display, each of said windows having defined boundaries on said display and a data signal associated therewith, said windows being ranked by relative priorities assigned thereto, the method comprising storing quantities representing the boundaries of each of said windows, generating signals representing said display position, selecting the windows whose boundaries as represented by said stored quantities include said display position, and selecting the data signal associated with the selected window with the highest priority.

The present invention also contemplates a graphics system providing multiple windows on a display screen, each of said windows having defined boundaries on said display and a data signal associated therewith, said windows being ranked by relative priorities assigned thereto, the system having apparatus for selecting one of said data signals for a particular position on said display such apparatus comprising means for storing quantities representing the boundaries of each of said windows, means for generating signals representing said display position, means responsive to said storing means and said generating means for selecting the windows whose boundaries include said display position, and means for selecting the data signal associated with the selected window with the highest priority.

In the drawings:-

Fig. 1 is a schematic block diagram of a graphics system and accompanying display in which the present invention is used.

Fig. 2 is a schematic diagram of the screen of the display shown in Fig. 1.

Fig. 3 is a fragmentary enlarged view showing a group of pixels that are processed simultaneously in the system shown in Fig. 1.

Fig. 4 is a schematic block diagram of the selection logic of the system shown in Fig. 1.

Fig. 5 is a schematic block diagram of a representative portion of the X Y compare logic of the selection logic shown in Fig. 4.

Fig. 6 is a schematic block diagram of the X Y counter portion of the selection logic shown in Fig. 4.

Fig. 7 is another view of the display screen, showing overlapping windows and the priorities and LUTs assigned to each.

Fig. 8 is a schematic block diagram of the LUT shown in Fig. 1.
Fig. 9 shows the arrangement of the priority registers shown in block form in Fig. 4.
Fig. 10 shows the arrangement of the LUT registers shown in block form in Fig. 4.
Fig. 11 shows the bit format of the X count signal provided by the X counter shown in Fig. 6.
Fig. 12 shows the bit format of the XMIN and XMAX signals provided by the XMIN and XMAX registers shown in Fig. 4.
Fig. 13 is a schematic block diagram of a portion of the window select logic shown in block form in Fig. 4.

Description of the Preferred Embodiment

Referring now to Fig. 1, a graphics system 10 incorporating the present invention is intended to be used with a display device 12. Display device 12 may be a cathode ray tube (CRT) display, liquid crystal display (LCD), gas plasma display, colour printer or any other suitable display device known to the art. The particular display technology used is not relevant to the present invention.

Referring now to Fig. 2, display device 12 provides a rectangular display comprising a plurality of pixels that are organised into rows and columns, as is conventional in the art. In the particular embodiment shown in Fig. 2, the display screen 40 may comprise 1024 rows, each of which contains 1280 pixels. These pixels are conventionally identified using X Y coordinates as shown in the figure so that the lower left pixel p1 has the X Y screen coordinates (0,0) while the upper right pixel p2 has the screen coordinates (1279,1023). Although the origin of the X Y coordinate system used to identify the pixels is conventionally located in the lower left corner of the screen 40, scanning of the pixels conventionally starts from the upper row (Y = 1023 in this embodiment) with each row being scanned from left to right before proceeding to the row beneath. Also shown in Fig. 2 is a window 42, more particularly, a rectangular window having an extent less than that of the entire screen. As shown in Fig. 2, window 42 has a left border at X = XMIN, a right border at X = XMAX, an upper border at Y = YMAX and a lower border at Y = YMIN. More particularly, XMIN is the X coordinate of the leftmost pixels (e.g., p3) in the window 42; XMAX is the X coordinate of the pixels (e.g., p4) immediately to the right of the window 42. YMIN is the coordinate of the lowermost pixels (e.g., p3) in the window 42; and YMAX is the Y coordinate of the pixels (e.g., p4) immediately above the window 42. If the window 42 borders the right or top edge of the screen 42, XMAX or YMAX is set equal to 1280 or 1024, respectively.

Referring back to Fig. 1, pixel data to be provided to display device 12 is stored in a video ram, or VRAM, 14, also referred to in the art as a frame buffer. In the particular embodiment shown each pixel stored in VRAM 14 contains 8 bits of data. This pixel data may be arranged in VRAM 14 by providing a number of planes equal to the number of bits in each pixel and by providing row addresses corresponding to the Y screen position (or 1023 - Y) and column addresses corresponding to the X screen position in Fig. 2. Other types of memory organisation alternatively may be used; the exact manner in which pixel data is organised in VRAM 14 is not as such relevant to the present invention.

VRAM 14 provides pixel data to a lookup table RAM 18 via lines 16. As shown in Fig. 8, RAM 18 comprises a plurality of groups LUT0-LUT7 of contiguous locations functioning as lookup tables (LUTs), one of which is selected in accordance with a selection signal provided on lines 22 originating from selection logic 20 that is the subject of the present invention. As shown in Fig. 8, each of lookup tables LUT0 through LUT7 comprises 256 consecutively addressable locations, each of which stores a word having a length of 8 bits, or 1 byte. The combined address signal representing the concatenation of the selection signal provided by selection logic 20 and the pixel data signal provided by VRAM 14 selects the particular location within RAM 18; selection logic 20 provides the three most significant bits on lines 22 to select one of lookup tables LUT0 through LUT7, while VRAM 14 supplies the 8 least significant bits to select a particular location within the selected table LUT0 through LUT7. Alternatively, fewer LUTs in RAM 18 may be provided if desired. Thus, tables LUT5 through LUT7 may be eliminated, and table LUT4 selected if the most significant bit of the three-bit selection signal is one, regardless of the value of the other two bits.

In the particular embodiment shown, pixel data is provided from VRAM 14 and LUT selection signals are provided from selection logic 20 four pixels at a time. Thus, referring now to Fig. 3, VRAM 14 provides pixel data simultaneously for a group of four adjacent pixels A0 through A3 along a given row, while simultaneously selection logic 20 provides selection signals on lines 22 for the same group of four pixels. Since each pixel requires eight bits of pixel data from VRAM 14 and three bits of selection data from selection logic 20, VRAM 14 provides RAM 18 on each read cycle with 32 bits of data, representing four adjacent pixels A0 through A3. In a similar manner, selection logic 20 provides a 12-bit control signal on each read cycle, 3 bits for each of pixels A0-A3. Since VRAM 14 is providing pixel data to RAM 18 four pixels at a time, the clock signal used to read out data from VRAM 14, as well as to read out the control signal from selection logic 20, is only one fourth the frequency at which pixels are scanned on the screen 40.

RAM 18 provides the words selected by the address signals on line 16 and 22 to a digital-to-analog converter (DAC) 32 one pixel at a time on lines 30. An output line 34 couples DAC 32 to display device 12. Although only a single VRAM 14, RAM 18 and DAC 32 are shown in Fig. 1, separate elements would typically be used for each of the three colour outputs (R G B) in a standard system. Although not shown in Fig. 1, display device 12 also receives suitable horizontal and vertical synchronisation signals from graphics system 10.

Selection logic 20 is preferably implemented as a single application-specific integrated circuit (ASIC), although other implementations are of course possible. Referring now to Fig. 4, in the selection logic 20, a plurality of extent registers 50 are used to define the X and Y extremes (minima and maxima) of each of up to eight windows WO-W7 (Fig. 7) to be displayed on the screen 40. More particularly, for each window W0-W7, there is an XMIN register 52 for storing the minimum X coordinate for the window, an XMAX register 54 for storing the maximum X coordinate of the window (actually, the X coordinate just to the right of the window), a YMIN register 56 for storing the minimum Y coordinate of the window, and a YMAX register 58 for storing the maximum Y coordinate for the window (actually, the Y coordinate just above the window). In the preferred system, a total of 32 registers 50 are used to define the extents of up to eight windows W0-W7. Each of the signals stored by extent registers 50 has the 11-bit format shown for XMIN and XMAX in Fig. 12.

X and Y counters indicated generally by the reference numeral 62 in Fig. 4 are shown in more detail in Fig. 6. As shown in that figure, an X counter 132 providing an X count on line 74 receives a reset input from an HSYNC line 24 and an incrementing count input from a CLOCK line 28. HSYNC line 24 carries a pulse at the beginning of each horizontal scan across the screen 40, while CLOCK line 28 carries a clock signal generated synchronously with the readout of pixel data from VRAM 14 (Fig. 1). Accordingly, X counter 132 provides an output on line 74 that emulates the horizontal sweep of the scanning beam (if a CRT display is used) across the display 40.

The CLOCK signal on line 28 is generated synchronously with the readout of pixel data from VRAM 14. Since, as noted above, data for contiguous groups of pixels A0-A3 are read out from VRAM 14 in parallel, four pixels at a time, the CLOCK signal frequency is one fourth the frequency at which successive pixels are displayed on screen 40. Accordingly, the X output from X counter 132 has the nine-bit format shown in Fig. 11, in which the least significant bit position (X8) represents, in effect, the 4's position of the X count.

A Y counter 134 providing a Y count on line 76 receives a load input from a VSYNC line 26 and a decrementing count input from the same HSYNC line 24 coupled to the reset input of counter 132. VSYNC line 28 carries a pulse at the beginning of the first (i.e., uppermost) left-to-right scan across screen 40. Upon receiving a VSYNC pulse at its load input, counter 134 is loaded with an input (not separately shown) corresponding to a Y coordinate (in this instance 1024) just above the uppermost line on screen 40. This count is decremented at the beginning of each successive scan by the HSYNC signal on line 24 applied to the decrementing count input of the same counter 134, so that the Y counter 134 emulates the downward vertical sweep of the scanning beam with a one-pixel resolution. Lines 74 and 76 from X and Y counters 132 and 134 of position counter logic 62 thus provide an indication of the current position on the screen 40 with a resolution of four pixels horizontally and a resolution of one pixel vertically.

X Y compare logic 60 simultaneously compares the current X and Y position as indicated by X Y counter registers 62 with the contents of each of the extent registers 50, the X position being compared with the contents of the X extent registers and the Y position being compared with the contents of the Y position registers. This allows window select logic 64 to determine which of the windows W0-W7, if any, includes the current position.

Referring now to Fig. 5, X Y compare logic 60 contains, for each window, an XMIN comparator 86, an XMAX comparator 88, a YMIN comparator 90 and an YMAX comparator 92. XMIN comparator 86 receives as inputs the nine-bit X position signal provided on line 74 from X counter 132 as well as the nine most significant bits of the XMIN signal (Fig. 12) provided on line 78 from the XMIN register 52 for that window. Similarly, XMAX comparator 88 receives as inputs the X position signal on line 74 as well as the nine most significant bits of the XMAX signal (Fig. 12) provided on line 80 from the XMAX register 54 for that window. XMIN comparator 86 provides an XMIN COMPARE signal on line 94 whenever the two inputs to the comparator match, while XMAX comparator 88 similarly provides an XMAX COMPARE signal on line 98 when its two inputs match. The two least significant bits of the XMIN signal (XMIN9 and XMIN10) are not fed to comparator 86 but instead are provided as outputs on lines 96. Similarly, the two least significant bits (XMAX9 and XMAX10) of the XMAX signal on lines 80 are not fed to XMAX comparator 88 but are instead supplied as output signals on lines 100.

YMAX comparator 92 receives as inputs the Y position signal on line 76 from Y counter 134 and the YMAX signal on line 84 from the YMAX register 58 for that window. Similarly, YMIN comparator 90 receives as input signals the Y position signal on line 76 from Y counter 134 as well as the YMIN signal on line 82 from the YMIN register 56 for that window. As noted above, the Y counter 134 is continually decremented from its maximum value as scanning progresses downwardly on the screen 40. When the Y position as indicated by the Y counter becomes equal to the upper window edge as indicated by the YMAX signal on line 84, YMAX comparator 92 provides an output setting a Y condition latch 102 for that window. Thereafter, when the Y position signal on line 76 is decremented to a value equal to the lower edge of the window as indicated by the YMIN signal on line 82, YMIN comparator 90 provides an output resetting the same Y condition latch 102.

Thus for each window, the Y condition latch 102 for that window provides a Y condition signal on the corresponding line 104 indicating that the current position is within the Y boundaries of the window. This signal is applied, together with the signals on lines 94, 96, 98 and 100 for that window, to the window select logic 64.

Fig. 13 shows a portion of the window select logic 64 that is replicated for each of the windows W0-W7. In the circuit shown in Fig. 13, pixel latch logic 106 responsive to the XMIN COMPARE and XMAX COMPARE signals on lines 94 and 98 as well as the XMIN9-XMIN10 and XMAX9-XMAX10 signals on lines 96 and 100 controls the setting and resetting of respective latches 108, 110, 112 and 114 associated with the respective pixels A0-A3 being concurrently processed. More particularly, each pixel latch indicates whether that pixel is within the X boundaries of the window as indicated by the XMIN and XMAX signals on lines 78 and 80 from the corresponding XMIN and XMAX registers 52 and 54. As noted above, the X count is incremented for every four pixels displayed on the screen 40. An XMIN COMPARE signal on line 94 from X Y compare logic 60 indicates the occurrence of a left window boundary (i.e., the leftmost pixel in the window) coincident with one of the pixels A0-A3, as indicated by the least significant bits XMIN9 and XMIN10 of the XMIN signal, provided on line 96. Pixel latch logic 106 responds to the XMIN COMPARE signal on line 94 by setting one or more of pixel latches 108-114 in accordance with the values XMIN9 and XMIN10. Thus, if both XMIN9 and XMIN10 are 0, then the left window boundary coincides with pixel A0, and all of the pixel latches 108-114 are set. On the other hand, if XMIN9 and XMIN10 are both 1, then the left window boundary coincides with pixel A3, and only pixel A3 latch 114 is set, since that is the only pixel within the window. In a similar manner, latches 110-114 are set if XMIN9 and XMIN10 are 0 and 1, while latches 112 and 114 are set if XMIN9 and XMIN10 are 1 and 0. Any latches not set on the current cycle of the CLOCK signal on line 28 are set on the next cycle, since the pixels processed on that cycle are even farther to the right of the left window boundary.

In a similar manner, in response to the appearance of an XMAX COMPARE signal on line 98, pixel latch logic 106 resets one or more of pixel latches 108-114 in accordance with the values of XMAX9 and XMAX10 on lines 100. Thus, if XMAX9 and XMAX10 are both 0, the right window boundary (i.e., the leftmost pixel to the right of the window) coincides with pixel A0. Pixel latches 108-114 are therefore all reset, since pixels A0-A3 are all outside of the window. On the other hand, if XMAX9 and XMAX10 are both 1, then pixels A0-A2 lie within the window, while pixel A3 lies without. Pixel logic 106 therefore resets A3 latch 114 immediately and resets the remaining latches 108-112 on the next clock cycle. Suitable means within pixel latch logic 106 are provided to handle the situation where the left and right window boundaries occur within the same group or adjacent groups of pixels A0-A3. Pixel latches 108-114 provide outputs to respective AND gates 116, 118, 120 and 122, each of which also receives an input from the Y CONDITION line 104. AND gates 116-122 provide outputs B0-B3 on lines 124, 126, 128 and 130, indicating whether the respective pixels A0-A3 are within the window in question. Since there are eight windows W0-W7, window select logic 64 provides a total of 32 outputs, eight binary outputs per pixel A0-A3.

Referring also to Fig. 9, respective programmable priority registers 70 store signals indicating the relative priorities P(0)-P(7) assigned to the eight windows W0-W7 provided for in the embodiment shown. Priorities P(0)-P(7) may each range between O and 7, with the higher numbers indicating higher priorities. Priority select logic 66 responds to the signals from window select logic 64 and from priority registers 70 to select, from the priorities P(i) associated with the selected windows W(i), the highest of such assigned priorities P(i). A 12-bit output from priority select logic 66 indicates the selected priority P(i) for each of the four pixels A0-A3 being concurrently processed. If none of the windows W0-W7 contains a particular pixel A0-A3, then the priority P(i) generated by priority select logic 66 for that pixel is the lowest assigned priority contained in priority register 70.

Windows W0-W7 are assigned default priorities to enable a unique selection if two or more windows have the same assigned priority stored in priority registers 70. Thus, window W7 is assigned the highest default priority, window W6 the next highest default priority and so on, with window W0 having the lowest default priority. These default priorities, as already noted, are only used in the event two or more windows inclusive of the current position have the same programmed priority.

Referring now also to Fig. 10, respective programmable LUT registers 72 store signals L(0)-L(7) identifying the lookup tables LUT0-LUT7 of RAM 18 respectively assigned to the windows W0-W7. Just as two or more windows Wi may have the same programmed priority P(i) stored in priority registers 70, two or more windows Wi may also have the same assigned lookup table LUTi in RAM 18. LUT select logic 68 responds to the signal P(i) from priority select logic 66, indicating the highest priority associated with a window inclusive of the current position, as well as to the signals from priority registers 70 and LUT registers 72 to generate a signal L(i) on line 22 identifying the lookup table LUTi associated with that highest priority. LUT select logic 68 accomplishes this by first using the priority signal P(i) from priority select logic 66 and the priority registers 70 to identify the corresponding window Wi by generating a cross-reference of window versus assigned priority. LUT select logic 68 then uses LUT register 72 to obtain the corresponding LUT identifier L(i), which it outputs on lines 22. This three-bit signal is supplied via lines 22 as additional address bits to RAM 18 to select the particular lookup table LUT0-LUT7 within the RAM. These operations are performed in parallel for each of the four pixels A0-A3, for a total of 12 bit outputs on lines 22 each cycle of the CLOCK signal on line 28.

As noted above, two or more windows W0-W7 can have the same assigned priority stored in priority registers 70. In such an instance, it may not be possible to associate a priority P(i) received from priority select logic 66 with a particular window Wi. In order to provide a unique LUT selection where the selected priority is associated with two or more windows, LUT selection logic 68 selects the LUT assigned to the highest-numbered window having the selected priority P(i).

Fig. 7 shows a typical situation in which screen 40 contains overlapping windows W0-W7. In the figure, the identifiers Pi indicate the priority associated with a particular window Wi, while the identifiers Li indicate the particular lookup table LUTi of RAM 18 associated with window Wi. Thus, in Fig. 7, window O has a priority P(0) of three and is associated with lookup table LUT1, while window W1 has a priority P(1) of two and is associated with lookup table LUT3 in RAM 18. These priorities and lookup tables are assigned to windows W0-W7 by storing the following values P(i) in the priority registers 70

i	P(i)
0	3
1	2
2	4
3	5
4	7
5	6
6	1
7	0

and by storing the following values in the LUT registers 72:

i	L(i)
0	1
1	3
2	3
3	4
4	5
5	2
6	1
7	0

Extent registers 50, priority registers 70 and LUT registers 72 are loaded under program control and altered at appropriate times, such as during the VSYNC pulse or during a blanking interval when pixels are not being scanned.

In the preferred system, X Y compare logic 60, window select logic 64, priority select logic 66 and LUT select logic 68 are each clocked logic circuits, timed by the CLOCK signal on line 28, with outputs following the inputs from which they are generated from one clock cycle. Each logic unit latches its outputs in registers before passing them to the next unit in the pipeline. It will be seen that as a result of this pipelined architecture and timing scheme, the LUT select signal on line 22 will trail the current position signals on lines 74 and 76 by four cycles of the clock signal 28 or 16 pixels. In order to ensure that the output signal on line 22 is properly synchronised with the readout of pixel data from VRAM 14, the counters 132 and 134 are so controlled as to compensate for the delay occasioned in the pipeline.

Variations on the embodiment described above will be apparent to those skilled in the art. Thus, while the embodiment shown provides for up to eight windows and lookup tables, a different number of each of these elements could be provided for if desired. Further, while the palette select logic preferably processes four pixels at a time, this parallel processing is not an essential element of the invention in its broadest form. Additionally, instead of having programmed priorities, the windows could have fixed priorities determined by their identifiers. Also, scanning could be performed vertically rather than horizontally.

Finally, it should be noted that while a preferred use of the selection logic described herein is to select a lookup table to which pixel data is applied, that is not the only use to which it can be put. More generally, the disclosed selection logic can be used to generate a data signal that identifies or otherwise provides information associated with the highest-priority window that includes a particular pixel. In such a case, the registers 72 would be loaded with the data signals associated with the windows.

Claims

1. A method of selecting a data signal for a particular position on a display within a graphics system providing multiple windows on a display, each of said windows having defined boundaries on said display and a data signal associated therewith, said windows being ranked by relative priorities assigned thereto, the method comprising storing quantities representing the boundaries of each of said windows, generating signals representing said display position, selecting the windows whose boundaries as represented by said stored quantities include said display position, and selecting the data signal associated with the selected window with the highest priority.

2. A method as claimed in Claim 1 in which said display position is defined by Cartesian X and Y coordinates, said generated signals representing the X and Y coordinates of said display position, each of said windows being a rectangular window having X and Y minima and maxima represented by said stored quantities.

3. A method as claimed in Claim 2 in which said window selecting step includes the step of comparing the X and Y coordinates of said display position with the X and Y minima and maxima of each of said windows.

4. A method as claimed in Claim 1, 2 or 3 in which said priorities assigned to said windows are stored in programmable registers.

5. A method as claimed in Claims 1, 2, 3 or 4 in which said data signals identify respective palette lookup tables associated with said windows, said method including the further step of selecting a lookup table in accordance with said selected identifier.

6. A method as in Claim 5 including the further step of applying pixel data corresponding to said display position to said selected lookup table.

7. A method as claimed in any preceding claim in which the data signals for a plurality of display positions are selected simultaneously.

8. A method as claimed in any preceding claim in which said second selecting step includes the steps of determining the priorities of the selected windows and selecting the highest of said determined priorities.

9. A method as claimed in Claim 8 in which said second selecting step includes the further steps of determining the window associated with said selected priority and selecting the data signal associated with said window.

10. A graphics system providing multiple windows on a display screen, each of said windows having defined boundaries on said display and a data signal associated therewith, said windows being ranked by relative priorities assigned thereto, the system having apparatus for selecting one of said data signals for a particular position on said display such apparatus comprising means for storing quantities representing the boundaries of each of said windows, means for generating signals representing said display position, means responsive to said storing means and said generating means for selecting the windows whose boundaries include said display position, and means for selecting the data signal associated with the selected window with the highest priority.

11. A system as claimed in Claim 10 in which said display position is defined by Cartesian X and Y coordinates, said generated signals representing the X and Y coordinates of said display position, each of said windows being a rectangular window having X and Y minima and maxima represented by said stored quantities.

12. A system as claimed in Claim 11 in which said window selecting means includes means for comparing the X and Y coordinates of said display position with the X and Y minima and maxima of each of said windows.

13. A system as claimed in Claims 10, 11 or 12 including respective programmable registers for storing said priorities assigned to said windows.

14. A system as claimed in Claims 10, 11, 12 or 13 in which said data signals identify respective palette lookup tables associated with said windows, said apparatus including a plurality of lookup tables and means for selecting one of said lookup tables in accordance with said selected data signal.

15. A system as claimed in Claim 14 including a video memory for storing pixel data corresponding to said display position and means for applying said pixel data from said memory to said selected lookup table.

16. A system as claimed in any one of Claims 10 to 15 in which said data signal selecting means selects the data signals for a plurality of display positions simultaneously.

FIG. 1

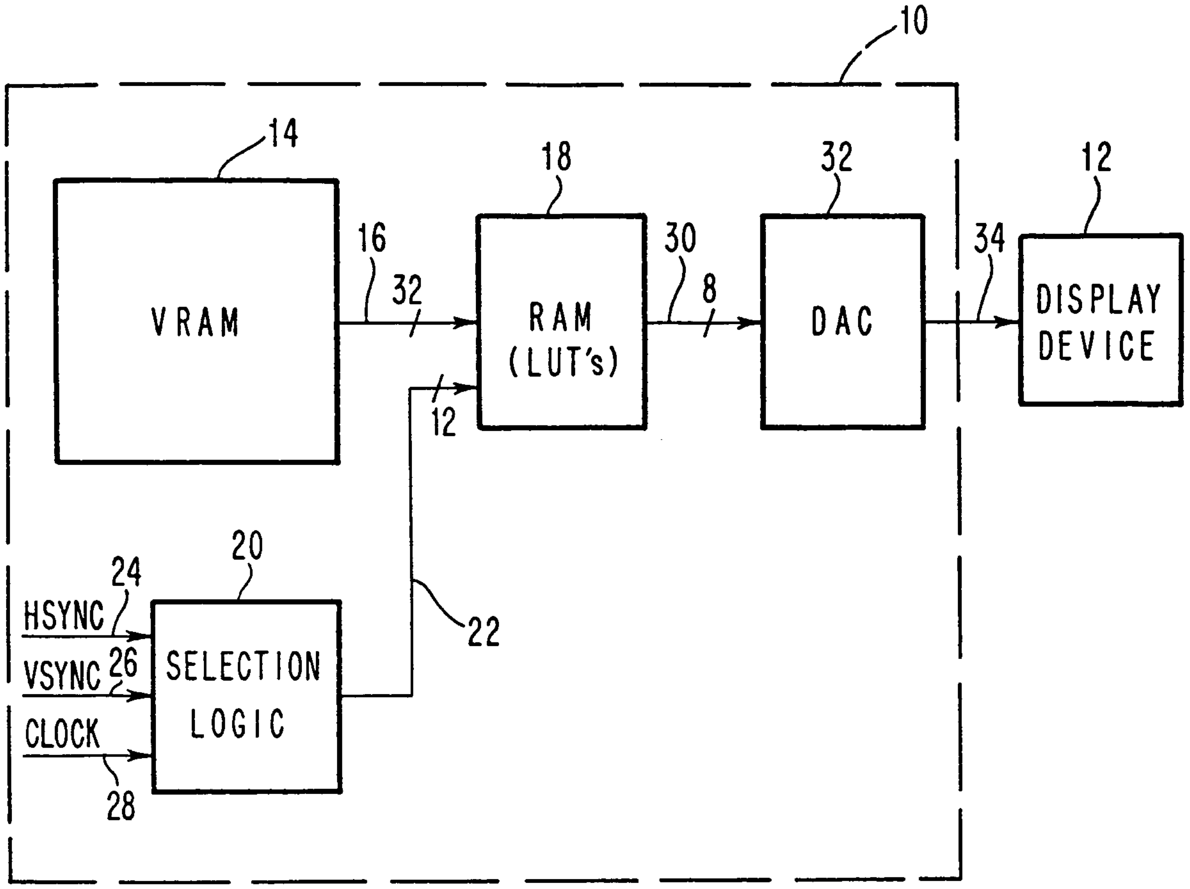

10
14
VRAM
16
32
18
RAM
(LUT's)
12
30
8
32
DAC
34
12
DISPLAY
DEVICE
20
24
HSYNC
VSYNC
26
CLOCK
28
SELECTION
LOGIC
22

FIG. 8

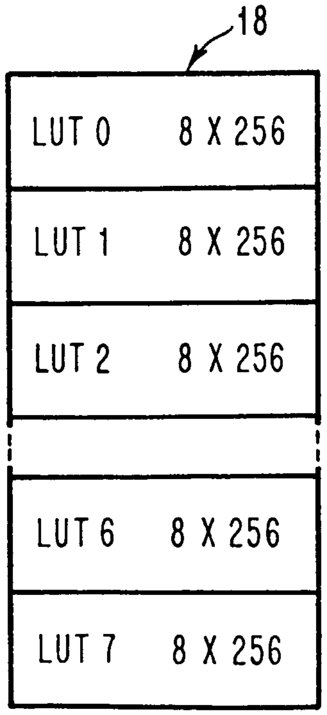

18
LUT 0 8 X 256
LUT 1 8 X 256
LUT 2 8 X 256
LUT 6 8 X 256
LUT 7 8 X 256

FIG. 2

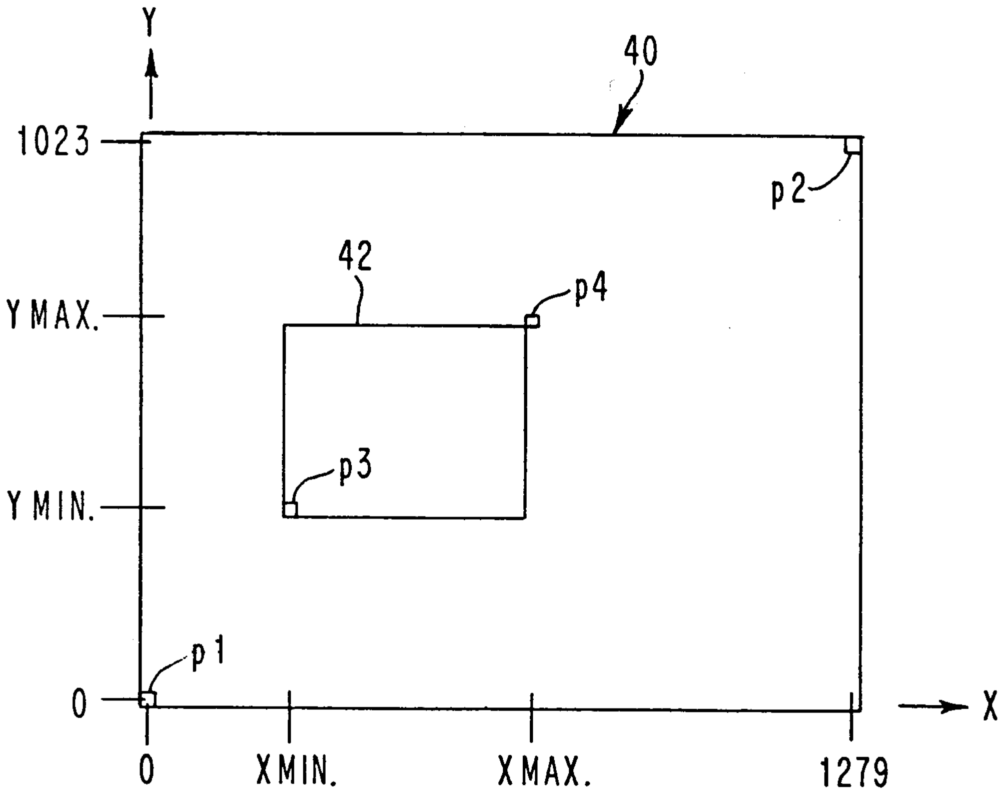

Y
40
1023
p2
42
p4
YMAX.
p3
YMIN.
p1
0
X
0
XMIN.
XMAX.
1279

FIG. 3

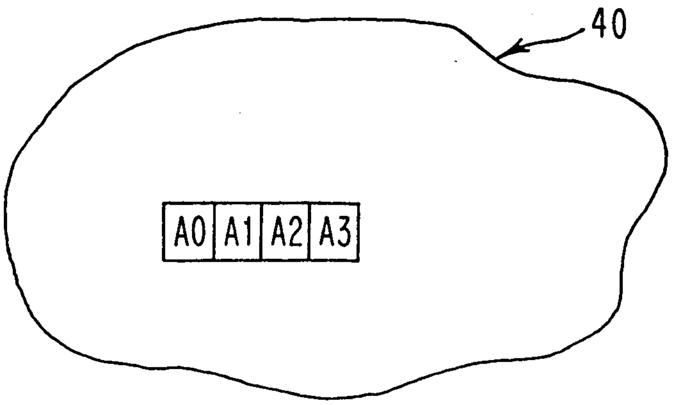

40
A0
A1
A2
A3

FIG. 4

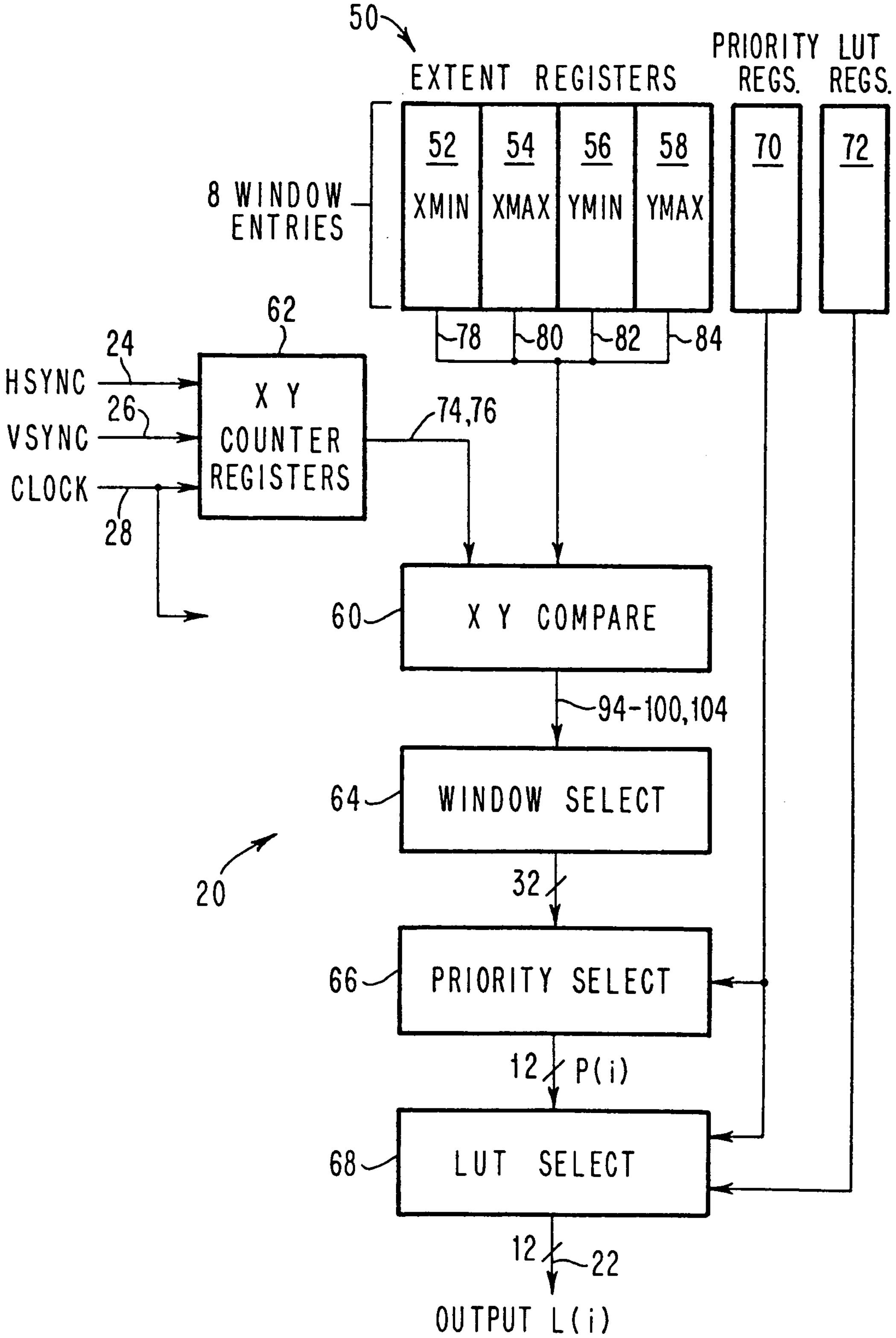

50
EXTENT REGISTERS
PRIORITY REGS.
LUT REGS.
8 WINDOW ENTRIES
52
XMIN
54
XMAX
56
YMIN
58
YMAX
70
72
78
80
82
84
62
24
HSYNC
26
VSYNC
CLOCK
28
X Y COUNTER REGISTERS
74,76
60
X Y COMPARE
94-100,104
64
WINDOW SELECT
20
32
66
PRIORITY SELECT
12
P(i)
68
LUT SELECT
12
22
OUTPUT L(i)

FIG. 5

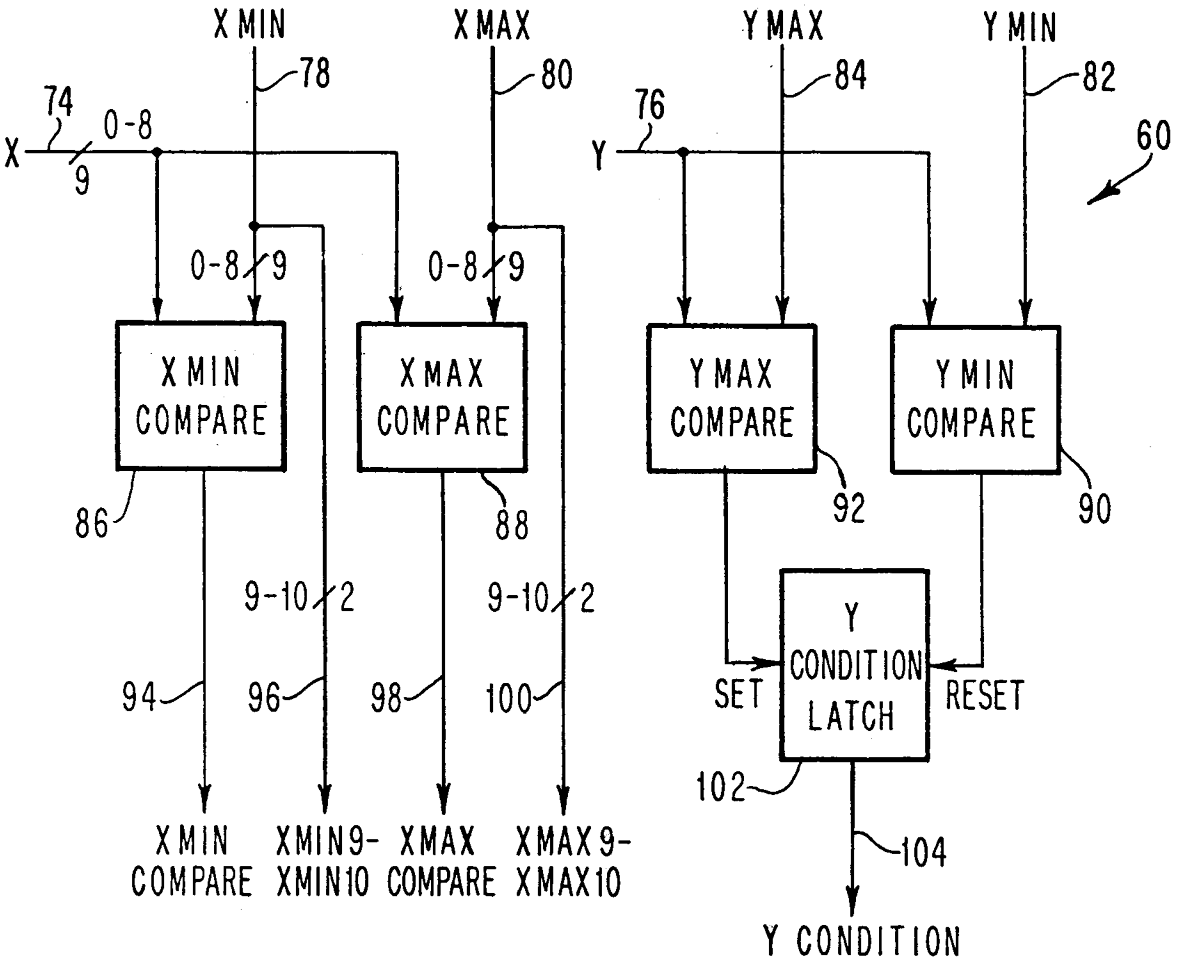

XMIN
XMAX
YMAX
YMIN
78
80
84
82
74
76
60
X
0-8
9
Y
0-8
9
0-8
9
XMIN COMPARE
XMAX COMPARE
YMAX COMPARE
YMIN COMPARE
86
88
92
90
9-10
2
9-10
2
Y CONDITION LATCH
SET
RESET
94
96
98
100
102
104
XMIN COMPARE
XMIN9-XMIN10
XMAX COMPARE
XMAX9-XMAX10
Y CONDITION

FIG. 6

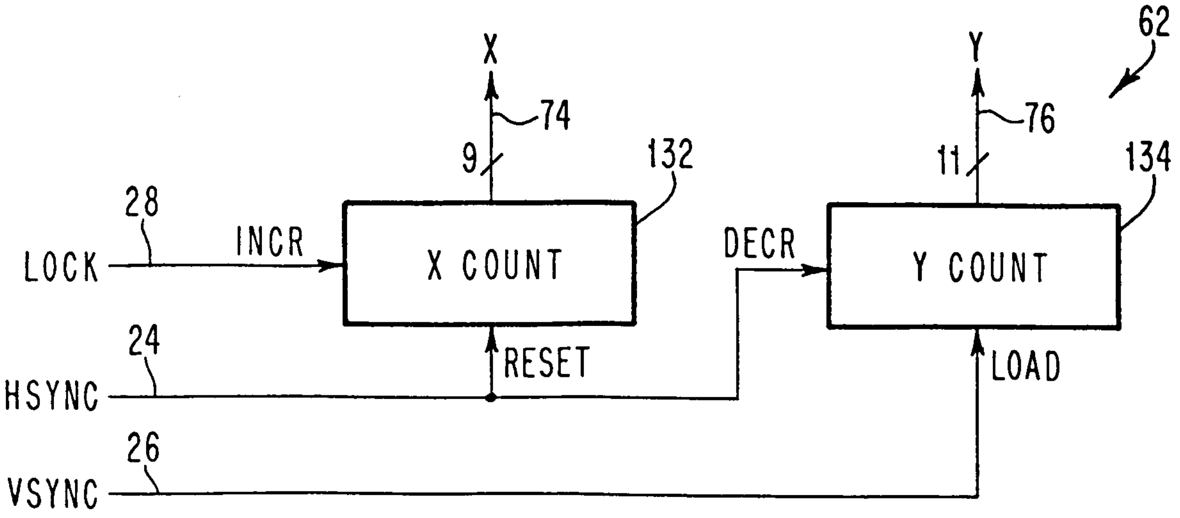
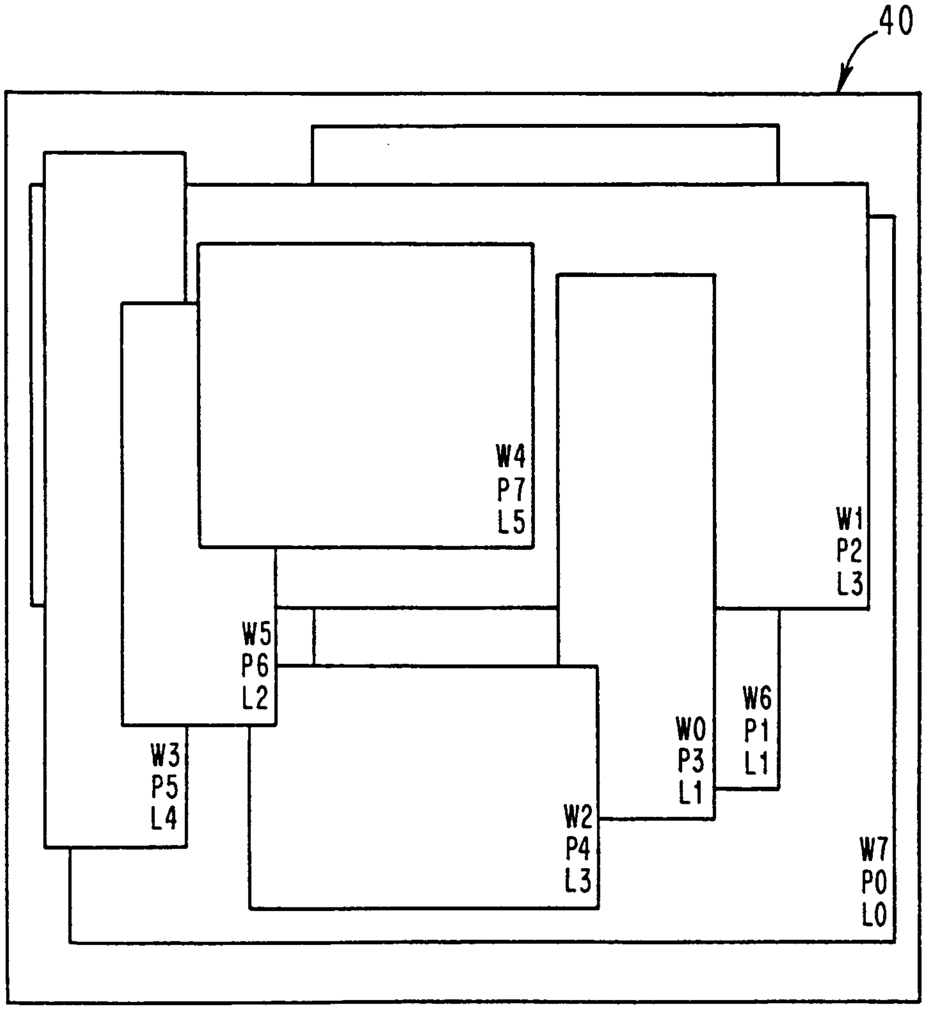

X
Y
62
74
76
9
11
132
134
28
LOCK
INCR
X COUNT
DECR
Y COUNT
24
HSYNC
RESET
LOAD
26
VSYNC

FIG. 7



CRT DISPLAY WITH WINDOWS.

EACH WINDOW IS OUTLINED.
THE ALPHANUMERICS REPRESENT:

WINDOW NUMBER (W_)
PRIORITY LEVEL (P_)
LUT SELECTION (L_)

FIG. 9

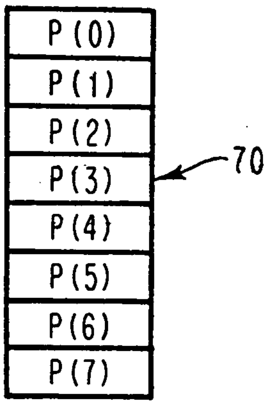


FIG. 10

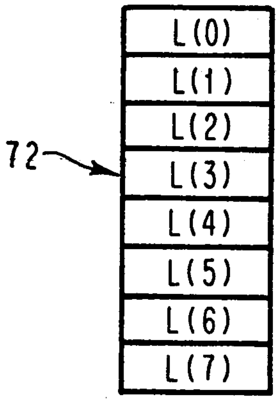


FIG. 11

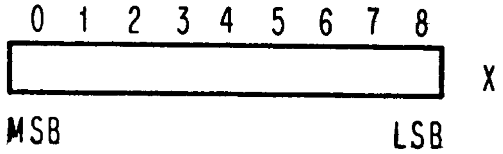


FIG. 12

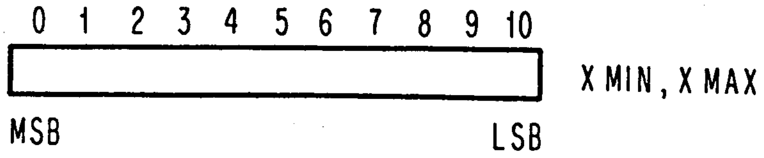
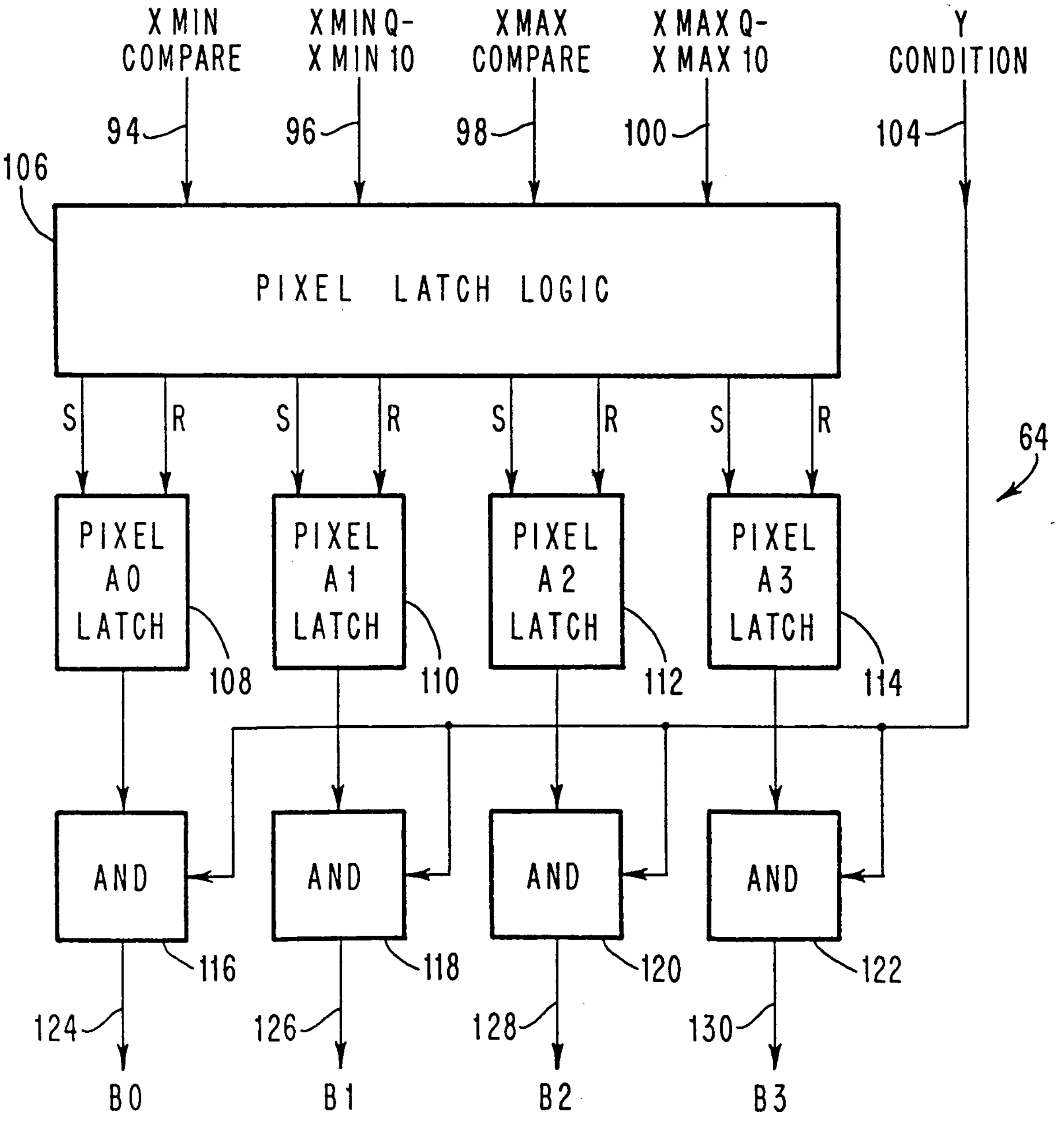


FIG. 13


X MIN COMPARE
X MIN Q-X MIN 10
X MAX COMPARE
X MAX Q-X MAX 10
Y CONDITION
94
96
98
100
104
106
PIXEL LATCH LOGIC
S
R
S
R
S
R
S
R
64
PIXEL A0 LATCH
PIXEL A1 LATCH
PIXEL A2 LATCH
PIXEL A3 LATCH
108
110
112
114
AND
AND
AND
AND
116
118
120
122
124
126
128
130
B0
B1
B2
B3