

DescriptionMULTI-SOURCE IMAGE REAL TIME MIXING
AND ANTI-ALIASING

This invention relates generally to image display apparatus and method and, in particular, to the simultaneous display of images from a plurality of image sources.

This patent application is related to the following commonly assigned U.S. Patent Applications: S.N. 07/733,576, filed 22 July 1991, entitled "Look-Up Table Based Gamma and Inverse Gamma Correction for High-Resolution Frame Buffers" S. Choi et al.; and S.N. 07/733,950, filed 22 July 1991, entitled "High Definition Multimedia Display" S. Choi et al.

In a modern display system data from multiple image sources are displayed. These images are shown on the display using a "window", a technique in which an area of a display is assigned to an image source. However, problems arise when the outputs from multiple sources must be coordinated on the display. When there is more than one source for the display, a mechanism is required to coordinate the output to a single display monitor.

A straightforward solution is to design the system such that only one source can be displayed at a time. For example, if a High Definition Televisions (HDTV) sampler input is selected, then only the HDTV output is output to the display. However, with an increased demand for "windowing" in graphics systems, and with an increased emphasis on a multimedia environment, where there exists more than one video source, this relative-

ly simple solution is not adequate.

In order to accomplish windowing there must be provided a mechanism such that, within the total area defined by a monitor screen, different sub-areas are sourced by different video sources. Fig. 1 shows an example of a windowed graphics monitor screen. Area A may be sourced by a local host or workstation. Area B may be sourced by a remote graphics server connected via a high speed network. Area C may be sourced by a HDTV sampler. One technique to provide such a display is known in the art as pixel switching. That is, for a given display area, or window, the source of pixels for that area is selected from a specified image plane.

A more complex problem is presented when an arbitrarily shaped image from a source is overlaid on top of another image, rather than over a simple rectangular window. For example, an image of an automobile may be rendered using a graphics server, and the automobile image may be required to be overlaid upon a HDTV-generated background image. This requires that all pixel selection be accomplished on a pixel-by-pixel basis, since the shape of the foreground object, or automobile, is not rectangular.

One solution that is applicable to a two-image source system utilizes color keyed pixel switching, and allows pixel-by-pixel selection between two sources. This technique is described in commonly assigned U.S. Patent No. 4,994,912, entitled "Audio Visual Interactive Display", by L. Lumelsky et. al..

However, for the case where there are more than two video sources, such as is illustrated in Fig. 1, a different solution is required for displaying N , where ($N > 2$), image sources on a pixel-by-pixel basis.

Another problem that arises in overlaying arbitrarily shaped multiple source images is due to an aliasing effect resulting from image pixel switching. In that a pixel of one image source may not blend with a pixel from a second image source, aliasing results. Aliasing causes the resultant image to exhibit undesirable artifacts along the boundary between the foreground and the background images, such as stair-casing and color distortion. In order to eliminate the artifacts, an anti-aliasing technique is necessary. However, in that the foreground image does not contain information about the background image, the anti-aliasing should be accomplished in real-time at the video output. Thus, a technique for anti-aliasing N image sources in real-time is required. Furthermore, in order to anti-alias an arbitrarily shaped foreground object, the anti-aliasing must be accomplished on a pixel-by-pixel basis.

However, in modern high resolution displays the video data bandwidth and data rates are very high, thus placing severe timing constraints on any pixel processing that is to be accomplished in a real-time manner.

In U.S. Patent No. 5,001,469, issued March 19, 1991, entitled "Window-Dependent Buffer Selection" to Pappas et al. there is described window control hardware in a graphics sub-system in which multiple image sources are shown on to a single monitor. This is accomplished by defining each window as a separate frame buffer and defining for each window, i.e. frame buffer, a window

THIS SHEET HAS BEEN
REPLACED

identification, and a window size and location based on four values of top location, bottom location, left location, and right location. This system also employs a prioritizing scheme where N frame buffers ("windows") are prioritized from 0 to $N-1$, where 0 has the highest priority and $N-1$ has the lowest priority. The graphics sub-system includes N window detection logics, one for each frame buffer, which use comparators for window size and location values to determine if the associated window is active for a region of the screen. If the window is active an image source pointer and other information are sent to prioritizing logic which prioritizes N input signals to determine which "active" image source has a highest priority. An active image with the highest priority is chosen by the priority logic and shown on a monitor.

Pappas et. al. employ window size and location values to control multiple image sources, and an image frame buffer does not contain multiple windows. Furthermore, this system appears to be limited for use with only rectangularly shaped windows. Also, the problem of displaying multiple sources with differing image formats is not addressed.

It is thus an object of the invention to provide for the simultaneous display of video data from N independent image sources, where N may be greater than two, through the use of pixel switching and control on a pixel-by-pixel basis for the N image sources.

The object of the invention is solved by the features laid down in the independent claims.



THIS SHEET HAS BEEN
REPLACED

The invention particularly provides a method and apparatus for displaying video data from a plurality of image sources on a monitor using a combination of alpha mixing and pixel switching, on a pixel-by-pixel basis, based on pixel color keying and window identification.

The foregoing and other problems are overcome and the object of the invention is realized by method and apparatus for implementing a raster graphic display video data path that provides arbitrary mixing of a plurality of images. The video data path is highly parallelized, and employs parallel devices operating under the control of a set of look-up tables. The look-up tables are loadable from a controller, such as a host workstation. The raster graphic display video data path functions with unlimited screen resolutions, and also enables a variety of different pixel data formats from a potentially large number of different sources. Outputs from many image sources are mixed under the control of the host workstation, with pixel accuracy being based on (a) a combined transparency coefficient (alpha) of the images, for each image source, and (b) a window identification number assigned by the host workstation.

Pixel data format conversion to a common predetermined format provides coherency between pixel values generated by a number of different image sources, such as HDTV and high resolution graphics servers, such as a super-computer.

An anti-aliasing function is disclosed for use in eliminating artifacts between a foreground image's boundary with a background image. The anti-aliasing function may also be used for generating special



effects, such as showing a translucent image foreground from one source on top of a background image from another source. The preferred anti-aliasing approach employs pixel mixing.

In order to allow any number of independent video sources, a separate frame buffer is allocated for each of the sources. Thus, for a system with N separate video data sources, there are N frame buffers. Depending on display monitor bandwidth requirements, each of the N frame buffers may also be interleaved M ways.

The invention also pertains to an integrated circuit device that provides, in a high speed pipelined manner, a pixel mixing function and a pixel multiplexing function.

The above set forth and other features of the invention are made more apparent in the ensuing detailed description of the invention, when read in conjunction with the attached drawing, wherein:

- Fig. 1 illustrates an exemplary display screen having multiple video data sources each displayed within an associated window;
- Fig. 2 is a block diagram showing, in accordance with the invention, a plurality of hierarchically arranged frame buffers having look-up tables, pixel mixing, and pixel multiplexing circuitry interposed between frame buffers;
- Fig. 3 is a block diagram showing an N by M configuration of hierarchically arranged frame buffers having look-up tables, pixel mixing, and



pixel multiplexing circuitry interposed
between frame buffers;

Fig. 4 illustrates a local host video path;

Fig. 5 is a block diagram illustrating pixel mixing
and multiplexing logic embodied within an
integrated circuit device; and

Fig. 6 is a schematic diagram illustrating an alpha
control logic function from the pixel mixing
logic of Fig. 5.

Referring to Fig. 2 there is illustrated a block
diagram of a multi-source video display pixel mixing
system 10 that is constructed and operated in accor-
dance with the invention. System 10 receives video data
input from N sources, with each source having an
associated frame buffer (FB) memory (FB_1 to FB_N).
Also shown is a video output data path for a pixel
(x,y) of each FB. In that there are N overlay FBs,
there are N overlay pixels for a pixel displayed at
display 12 location (x,y), one from each FB. The FBs
are hierarchically ordered from 1 to N, where 1 is
considered to have a highest priority and where N is
considered to have a lowest priority. By preference,
but not as a limitation upon the practice of the
invention, FB_1 is associated with a host 14 and FB_N
is associated with, by example, an HDTV video source
that does not provide an indication of pixel translu-
cency, or alpha. The FBs are addressed in parallel so
that each operates in lock-step with the others.

In a multi-media system such as the one depicted in
Fig. 2 a coherency of pixel format cannot be assumed.

For example, an image generated by a supercomputer and/or a graphic server and received from a communication network may have a 24-bit Red, Green, Blue (R,G,B) format. However, the local workstation host 14 may have an 8-bit pixel format, and an HDTV sampled input may have a chrominance/luminance (Y/C) format. In order to mix pixels from these diverse sources a first step converts the pixel values to a common format. In the system 10 the R, G, B format is chosen as the common format due to its relative simplicity, although other formats are within the scope of the teaching of the invention. Thus, all pixel data is converted to R, G, B. This data conversion is accomplished at each FB through the use of a Look-Up-Table (LUT) 15. That is, knowing the particular format stored within a given FB, the associated LUT 15 is programmed such that when a pixel of the format to be converted is applied to the address inputs of the LUT 15, a corresponding R,G,B value is output. Each LUT 15 is preferably coupled to the local host 14 and is programmable thereby, although fixed-content LUTs may also be employed, depending upon the application. It should be appreciated that the host 14 may also be a source of video data to one of the FBs, such as is illustrated in Fig. 1 for the data displayed within window A. As an example, if the host 14 is a workstation then alphanumeric text information may be generated by the host 14 for display.

It is further noted that for some video sources, such as a HDTV source, the video signal may be expressed in an analog format. As such, an analog-to-digital converter is employed to first convert the analog video input to a digital format suitable for storage within the FB_N.

The use of the LUT 15 pixel data format converters thus overcomes the problem of coherency of the pixel data format provided by each independent image source, in that a common pixel format is provided prior to pixel mixing.

In order to perform mixing and anti-aliasing per pixel boundary, each of the FBs includes an alpha buffer memory plane, except for the FB_N, which has no downstream FB and, thus, no background image to mix with. The alpha buffer memory plane is accessed by the device which is also the source of the video data for the associated FB. For example, FB_2 may have a 32-bit memory plane organized as a 24-bit R, G, B plane and an 8-bit alpha plane, both of which are sourced by a graphics server over a communication network. When the graphics server renders an image it also generates an alpha value associated with each pixel rendered for the image.

As employed herein alpha is considered to be an indication of pixel translucency or transparency, and may assume values between zero and one. With alpha equal to zero the associated pixel is considered to be totally transparent. That is, any background pixel or pixels are visible. With alpha equal to one the associated pixel is considered to be totally opaque. Values between zero and one provide for gradations in pixel transparency and are employed to advantage at the boundaries of foreground and background images to anti-alias the boundary pixels to eliminate undesirable display artifacts.

The pixel data and alpha values are both transferred over the communication network into the associated

FB_2. The 8-bit alpha value is used for pixel mixing, via MIX logic 16 and multiplexer (MUX) logic 17, so as to mix R,G,B image pixel data passed up from the down-stream FB with the pixel from FB_2. The alpha FB plane thus enables real-time, pixel-by-pixel mixing of the foreground pixels with the background pixels. Anti-aliasing may also be accomplished in real-time, as will be described below.

In operation, the FB having the lowest priority (N) converts a pixel (x,y) to R, G, B pixel format, via the associated LUT 15, and passes the converted pixel value on for mixing with a corresponding pixel from the next highest priority frame buffer (FB_N-1), via the MIX 16. All FB_j, except for the N-th FB (lowest priority), may perform one of three operations. These operations are shown in TABLE 1.

TABLE 1

1. Discard the pixel passed from the j+1 FB and pass the associated pixel to the upstream j-1 FB, or to the display 12 if j=1. This is accomplished by selecting the A input of the associated MUX 17.
2. Discard its own pixel and pass the pixel received from the j+1 FB to the j-1 FB, or to the display 12 if j=1. This is accomplished by selecting the C input of the associated MUX 17.
3. Use the value of alpha, from the associated FB, to mix its own pixel value with the pixel received from the j+1 FB, and then pass the mixed pixel value to the j-1 FB, or to the display 12 if j=1.

This is accomplished by selecting the B input of the associated MUX 17.

The method set forth above provides a technique where, for a given display pixel (x,y), either a single pixel (x,y) from any FB is selected (pixel switching), or any mixed result of two or more pixels (x,y), from different FBs, is selected (pixel mixing).

As seen in Fig. 4, the invention employs a color keying method based on the pixel value of the local host 14. The local host 14 FB is comprised of three planes. One plane stores a local host 14 pixel value, or color index (CI). A second plane stores a local host 14 window identification number (WID). A third plane stores a local host 14 alpha value (WS_alpha). The system 10 employs control logic 19 for generating a plurality of signals based on inputs from the host 14 FB. These inputs include a CI input for "color keying" to determine display 10 operation. WID provides a mechanism where different windows of the display 12 may have one or more key colors associated therewith. This is accomplished through the use of a palette LUT 15 and a video control (VC) LUT 22. VC_LUT 22 has $2(N-1)$ outputs which are provided in pairs to each of the MUXes 17 for controlling the operation thereof, in accordance with TABLE 1.

For example, for a first window a red CI may be defined as "key" to select, via the VC_LUT 22, the HDTV input. For another window a red CI may be the "key" to cause, via VC_LUT 22, the mixing of a host 14 pixel with a HDTV background pixel, using WS_alpha for modifying the values of boundary pixels so as to perform anti-aliasing. For another window, a red CI may be a pixel that is

displayed on the monitor 12, via the palette LUT 20 where the CI is converted to 24-bit R,G,B format. This method advantageously enables any number of independent image sources to be coordinated, rather than being limited to only two image sources. Furthermore, this method also provides pixel mixing for functions such as real-time anti-aliasing or image blending. All of these functions may be accomplished on pixels contained within and/or bounding an object of interest, based on the value of WID. Furthermore, these functions are applied on pixel-by-pixel basis.

Preferably, video output controls are implemented using the local host 12 FB. For illustration, FB₁ is chosen as the local host FB, but any one of the FBs may be chosen instead. As seen in Fig. 4, for the local host 12 graphics work space there are a total of P bit planes for the FB. Of the P-bits output by these planes, C-bits are used as the color index (CI), W-bits are used as the window identifier (WID), and M-bits are used for (WS_alpha) for blending the local host 12 image with another image. Both CI and WID are employed as indexes (addresses) to the LUT 15 to provide 24 bit R, G, B data. Furthermore, these same bits are used as indexes (addresses) to VC_LUT 22 to provide the video path control bits (VID_CNTRL), which are used to coordinate the display output. The width of VID_CNTRL is a function of the total number of the image sources (N).

For each FB_j, where $1 \leq j < N$, two bits are required to choose one of the three operations described in TABLE 1. The definition of these two bits may be expressed as:

- 00 - Mixing Disabled. Select Input A for MUX;
- 01 - Mixing Disabled. Select Input C for MUX;
and
- 1x - Mixing Enabled. Select input B for MUX.

For FB_j , if bit 1 (MSB) is set, mixing is enabled and the result of the mixed pixel of FB_j and FB_{j+1} is passed to the $j-1$ FB. In this case bit 0 (LSB) is ignored. If bit 1 is reset, then alpha mixing is disabled, and either the pixel value from FB_j (if bit 0 is reset) or the pixel value received from FB_{j+1} (if bit 0 is set) is passed to FB_{j-1} .

In that the FB_N can only pass its output pixel value upstream, it does not require any control bits. Thus, the total number of bits required for VID_CNTRL , for N image sources, is $B = 2(N-1)$, where B is number of bits in VID_CNTRL and N is the number of independent image sources.

The VID_CNTRL assignment for FB_j are bits $2j-2$ for the least significant bit and $2j-1$ for the most significant bit. This provides a flexible mechanism for window control for a multi-source system. It should be noted that from the local host 12, by using the color keying, a user may cause any display pixel to be formed from a result of all corresponding pixels from all FBs. That is, the user defines a color key to represent a specific output format, and then employs that color key where a specific output is desired. Furthermore, through the use of WID the user is enabled to select the color keying as a function of window identification. For example, if the width of WID is 4-bits, then there may be up to 2^4 , or 16, windows that are simultaneously displayed and controlled. For this case the

VC_LUT 22 includes up to 16 different storage regions, each region containing different VID_CNTRL values. A specific one of the regions within VC_LUT 22 is selected by the value of WID, while a specific VID_CNTRL bit combination within the region is selected by the value of CI.

In summary, this mechanism provides a flexible output control for any number of FBs utilizing both color keying and mixing (anti-aliasing) techniques on a pixel-by-pixel basis. If FB mixing is enabled for each level, the equation for the resulting displayed pixel (R) is given by:

$$R = \alpha_1 P_1 + (1 - \alpha_1)(\alpha_2 P_2 + (1 - \alpha_2)(\alpha_3 P_3 + (1 - \alpha_3) \dots (\alpha_{N-1} P_{N-1} + (1 - \alpha_{N-1}) P_N) \dots)),$$

where P_j represents a pixel from the FB_j , and where α_j represents an alpha value from the FB_j .

A high resolution monitor, e.g. 2048 X 2048 pixels, requires a video bandwidth in excess of 360 Mhz. In order to provide the bandwidth necessary for a high resolution monitor, the serial video output from the FB VRAMs is required to be interleaved. For example, for a conventional VRAM, having a serial output bandwidth of 33 Mhz, the FB video output path needs to be, for a 60 Hz 2048 X 2048 resolution monitor, interleaved at least 11 ways and, thus, requires 11 independent data paths. As a further example of interleaving, for a monitor resolution of 1280 x 1024, the video bandwidth is 110 MHz. Thus, four way interleaving is sufficient for this resolution. However, for a monitor resolution of 2048 x 1536, the video bandwidth is 260 Mhz. This requires eight way interleaving, in that four way

interleaving provides only 4×33 Mhz, or 132 Mhz, while eight way interleaving provides 264 MHz.

The MIX logic 16 and the MUX logic 17 for each FB is replicated to match the interleave factor. In general, there is a MIX logic 16 and a MUX logic 17 for each of the interleaved, independent data paths. Thus, if the display output of the FB is interleaved M ways for N image sources, there are $((M \times N) - N)$ pairs of MIX and MUX logic blocks, in that the lowest priority FB level does not require mixing. This modular approach enables pixel mixing in real-time for any monitor resolution.

Fig. 3 shows a M X N embodiment for an interleaved video system. A Serializer 24 accepts the outputs from $j=1$ MIXERS 30 and shifts these outputs out to the display 12 at the video clock rate.

An integrated circuit implementation of the foregoing may be accomplished by several methods. One method is to provide a fast integrated circuit device after serializing the interleaved video data output from the FB VRAM's. However, this requires, for 260 Mhz display, a 3.85 nanosecond cycle time to perform any operation, such as blending two images. Another method is to provide multiple, slower devices at the output of each interleaved memory data path, as in Fig. 3. The second method is preferred over the first in that the first requires either ECL or GaAs gate arrays to accommodate the high video bandwidth. However, the same circuit architecture described below lends itself to either approach.

A most time critical section of the video data path flow is the blending function, in that this function

requires multipliers and adders. The blending function for the FB_j uses the equation:

$$R_j = (\alpha_j P_j) + ((1 - \alpha_j) R_{j+1}),$$

where R_j is the resulting pixel value output to the next FB , P_j is the pixel from the FB_j , R_{j+1} is the pixel value input from the previous FB_{j+1} , and α_j is the $P_j / (P_j + R_{j+1})$ pixel weight, wherein $(0 \leq \alpha \leq 1.0)$. This approach requires two multipliers and an adder. However, using an algebraic manipulation there is obtained,

$$\begin{aligned} R_j &= (\alpha_j P_j) + ((1 - \alpha_j) R_{j+1}), \\ R_j &= (\alpha_j P_j) + R_{j+1} - (\alpha_j R_{j+1}), \text{ and} \\ R_j &= (\alpha_j (P_j - R_{j+1})) + R_{j+1}. \end{aligned}$$

This latter expression for R_j requires but one multiplier and two adders.

Fig. 5 illustrates a mixer integrated circuit device 30 block diagram. Because of the required cycle time, the adders 32 and the multiplier 34 are pipelined, and all data values are delayed to maintain them in synchronous order, as will be described.

The following describes the function of the MIXER 30 of Fig. 5. Reference is also made to Fig. 3. The image data associated with each pixel arrives at the inputs simultaneously. In order to provide a maximum clock speed all inputs and outputs are registered (REG1-REG8) and clocked by the system clock.

The system clock is derived by dividing the video bandwidth by M for an M -way interleaved memory. For

example, in the case of a medium resolution monitor, the video bandwidth is 110 MHz. The FB memory is interleaved, by example, four ways. Therefore, the system clock is $110 \text{ Mhz}/4$, or 27.5 Mhz. In case of the high resolution monitor, the video bandwidth is 260 Mhz and the memory is interleaved eight ways. Thus, the system clock is $260 \text{ Mhz}/8$, or 32.5 Mhz.

In that it is difficult to economically achieve a full multiplier that operates at approximately 33 Mhz, the multiplier 34 is preferably pipelined. The depth of the pipelining depends on the selected chip technology. Using readily available integrated circuit technology the multiplier pipeline depth may be achieved with two stages, and the adder 34 achieved with one stage. To maximize the throughput, the output of each functional block is reclocked. Thus, the MIXER 30 is, totally synchronous with a single free running system clock.

The function of the MIXER 30 is described as follows. All inputs are simultaneously clocked into the chip registers REG1 through REG4. These inputs are VID_CNTRL_j , α_j , P_j and R_{j+1} . The output of REG3 and REG4 are fed into function block FUNC1 where the $(R3-R4)$ operation is performed. Meanwhile, to synchronize all other inputs, R1, R2, and R4 are delayed via registers REG5 and REG6. The function of (FUNC4) is discussed later. For the next two clock cycles, R5 and F1 are multiplied by FUNC2, and R6 is correspondingly delayed two clock cycles with REG7 and REG8. Finally, F2 and R8 are fed through the function block FUNC3 where the $(F2 + R8)$ operation is performed.

In order to obtain perfect mixing across N frame buffers, the full precision of R_j , which increases

by the width of P_j at each FB_j , should be propagated. However, this is not a practical solution in that it is not cost effective to carry the full resolution of R_j to the next stage. However, a simple truncation may be used and results in, on the average, a 1/2 bit error per mixing stage.

If rounding off is performed at the function block FUNC5 then, on average, the error is 1/4 bit, or one half of the 1/2 bit error if FUNC5 were not employed. Thus, FUNC5 provides that the resulting pixel R_j has the same number of bits as P_j . Therefore, the total cycle time required for MIXER 30 is six system clock periods.

MIXER 30 serves two functions. The first function is alpha mixing (or anti-aliasing). The second function is data path control. As shown in Fig. 2, for each FB, except the FB_N , there is provided a three-to-one MUX 17. To further reduce cost and complexity the function of MUX 17 is incorporated within MIXER 30, as described below.

When alpha is set to 1.0, R_j (the result of alpha mixing) is P_j . When alpha is set to 0.0, then R_j is R_{j-1} . In other words, the mixing function may also perform a multiplexer function, provided that the values of alpha are properly selected. This is accomplished in the block (FUNC4), as shown in Fig. 5. The logic for FUNC4, for an 8-bit wide alpha value, is shown in Fig. 6. The two bit VIDEO_CNTRL is employed to control the alpha value. When the alpha enable bit (VID_CNTRL_1) is asserted, then alpha-OUT is identical to the alpha input. When the alpha enable bit (VID_CNTRL_1) is deasserted, alpha_OUT_0 to alpha_OUT_6

are forced to a zero by AND6 through AND0. Furthermore, alpha_OUT_7 is set to a one, making alpha_OUT = 1.0, which causes the MIXER 30 to select P_{j+1} .

The total number of inputs and outputs to MIXER 30, assuming an 8-bit alpha value, are

for each R,G,B color:	8 outputs for R_j ;
	8 inputs for P_j ;
	8 inputs for R_{j+1} ;
common signals	8 inputs for alpha;
	2 inputs for VID_CNTRL; and
	1 input for SYSTEM CLOCK,

for a total of 83 pins, not including power, etc.

As was stated, one of the problems encountered with pixel mixing is that two images, generated from two different sources, may be different in format. For example, pixel sizes may be different between two images sources, where one image source pixel may be a square pixel while the other image source pixel is not a square pixel. Another example is that two image sources may not have the same resolution. Thus, in order to have a pixel-by-pixel correspondence for the entire FB structure, one image may need to be either scaled up, if that image source is smaller than the other, or scaled down, if that image source is larger than the other. One solution to these problems, which is accommodated by the teaching of the invention, is to over-sample on all of the FBs where the resolution of the image source for the FB is a common least multiple of the resolution of each of the image sources, in both the horizontal and vertical directions. The common least multiple FB approach is preferable, in that

over-sampling is not always readily accomplished, as in the case of an HDTV image sampler in which the resolution is fixed at 1920 X 1035 pixels. However, if the FB size is the size of a least common multiple, then pixels may be either simply replicated or interpolated as a means for over-sampling. Another approach is to employ digital signal processing in order to over-sample.

It should be noted however that the least common multiple approach may result in a very large FB. For example, the least common multiple of 480 scanlines and 1035 scanlines is 33,120 scanlines. Furthermore, the resulting pixel may not be a square pixel. Also, a time-base correction may be necessary to match the monitor bandwidth.

Fortunately, for many graphics or video systems the pixels are normally square (or almost square). Furthermore, due to windowing, image sources need not fill the entire FB, so that each pixel can be treated equally. Thus, pixel $P(x,y)_j$ may be considered as being both square and equal in size for all j ($1 \leq j \leq N$), where j represents the j -th FB and N is the number of frame buffers in the system.

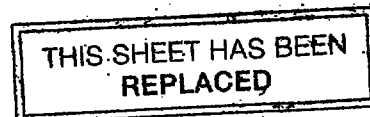
In summary, the invention addresses and solves the problems of combining multiple image sources at a display monitor. Overlay and blending techniques are disclosed for a system with N independent image sources.

The technique of the invention allows for any number of independent image sources (N), where each source has an associated FB. The architecture also enables the serial output of each FB to be interleaved so as to

match the video output. Thus, if the interleaving requires M arrays, then the total number of MIXERS 30 is $N \times M$, which enables mixing of any number of images with a pixel accuracy that is a function of the combined transparency coefficient of the images.

The teaching of the invention also employs video look-up tables to enable a variety of mixing operations to occur in a multi-tasking environment, based on an application specific window identification number.

The invention also provides real-time pixel switching and control on a pixel-by-pixel basis, for any number of independent image sources per window, using the WID value.



Claims

1. Image display apparatus, comprising:

a plurality of frame buffer means organized in a hierarchical manner so as to provide a highest priority frame buffer means and a plurality of lower priority frame buffer means, each of said frame buffer means storing image pixel data from an associated image source;

means, coupled to an output of each of said frame buffer means, for converting a format of the stored image pixel data to a predetermined format, each of said converting means preferably having an input coupled to control means for having data, representing the predetermined format, stored within and;

means, having inputs coupled to individual ones of said converting means outputs, for selectively combining an image pixel from each of said plurality of frame buffer means so as to form a composite image pixel.

2. Image display apparatus as set forth in claim 1, wherein said combining means includes a plurality of means, individual ones of which are interposed between hierarchically adjacent frame buffer means, for selectively generating an image pixel value and for providing the generated image pixel value to an input of a generating means at a next highest level, said generating means being responsive to a control means for (a) providing an image pixel value that is equal to an image pixel value





output by a first one of the frame buffer means that is coupled to the input thereof, (b) for providing an image pixel value that is equal to an image pixel value output by a second one of the frame buffer means that is coupled to the input thereof, or (c) for providing an image pixel value that is equal to a combination of the image pixel values output by each of the first and the second frame buffer means.

3. Image display apparatus as set forth in claim 1 or 2, wherein said combining means comprises, for each frame buffer means, mixer means for mixing together, in accordance with a pixel transparency value provided by the associated frame buffer means, a pixel value from the associated frame buffer means and a pixel value received from a mixer means associated with a frame buffer means at a next lower priority level.
4. Image display apparatus as set forth in claim 3, wherein said combining means further comprises multiplexer means having a plurality of inputs, wherein one of said inputs is coupled to an output of said mixer means, one of said inputs is coupled to the output of the associated converting means, and one of said inputs is coupled to an output of the mixer means associated with the frame buffer means at the next priority level, said multiplexer means being responsive to a signal generated by control means for selectively providing at an output a value corresponding to one of said three inputs, wherein said control means preferably comprises look-up table means for generating said control signal on a pixel-by-pixel basis in



THIS SHEET HAS BEEN
REPLACED

accordance with a pixel color code and a display means window identifier.

5. Image display apparatus as set forth in claim 4 wherein said pixel color code and said display screen window identifier are stored within one of said frame buffer means that is associated with said control means, wherein said frame buffer means that is associated with said control means preferably is a highest priority frame buffer means.
6. Image display apparatus as set forth in claim 5, wherein the output of said multiplexer means, that is associated with said highest priority frame buffer means, is coupled to said display means.
7. Image display apparatus as set forth in any one of claims 3 to 6, wherein said mixer means includes first means for subtracting the converted pixel value output by the associated frame buffer means from the pixel value received from the mixer means associated with the frame buffer means at the next lower priority level, second means for multiplying a difference value output by said first means times a transparency value associated with the pixel value output by the associated frame buffer means, and third means for adding a product value output by said second means to the pixel value received from the mixer means associated with the frame buffer means at the next lower priority level.

This Sheet Has Been Replaced



YO 991 084



* C L M S *



8. Image display apparatus according to any one of the preceding claims, wherein there are N frame buffer means corresponding to N priority levels, and wherein each of said N frame buffer means is interleaved M ways.
9. Image display apparatus according to any one of the preceding claims, wherein there are $(N \times M)$ converting means, and wherein there are $((N \times M) - M)$ combining means.
10. A method for generating video pixel data, comprising the steps of:

providing a plurality of frame buffer means organized in a hierarchical manner so as to provide a highest priority frame buffer means and a plurality of lower priority frame buffer means, each of said frame buffer means storing image pixel data from an associated image pixel source; and

selectively combining an image pixel from each of said plurality of frame buffer means so as to form a composite image pixel;

wherein the step of selectively combining preferably is executed on a pixel-by-pixel basis for each image pixel stored within each of the frame buffer means.
11. A method as set forth in claim 10, wherein the step of selectively combining comprises the steps of decoding a predetermined pixel color value in conjunction with a display window identifier so as





to generate a plurality of control signals for controlling the selective combination of image pixels, wherein the predetermined pixel color value and the display window identifier preferably are stored within a highest priority frame buffer means.

12. A method according to claim 10 or 11, wherein at least one of the frame buffer means stores, in conjunction with each pixel value stored therein, a transparency value, and wherein the step of selectively combining combines the transparency value with the associated pixel value and with the combined pixel value received from a next lower priority level frame buffer means.
13. A method according to any one of claims 10 to 12, wherein the step of providing comprises a step of converting a format of the stored image pixel data from each of the frame buffer means to a predetermined format.
14. An integrated circuit device for selectively mixing together image pixel data received from a first input and from a second input, said integrated circuit device including first means for subtracting a pixel value received from the first input from a pixel value received from the second input, second means for multiplying a difference value output by said first means times a pixel transparency value associated with the pixel value received from the first input, and third means for adding a product value output by said second means to the pixel value received from the second input, said integrated circuit device preferably further





comprising means for synchronously delaying the pixel value received from the second input by an amount of time required for the operation of the first and second means.

16. Image display apparatus, comprising:

a plurality of frame buffer means organized in a hierarchical manner so as to provide a highest priority frame buffer means and a plurality of lower priority frame buffer means, each of said frame buffer means storing image pixel data from an associated source; and

means, interposed between hierarchically adjacent frame buffer means, for generating an image pixel value and for providing the generated image pixel value to an input of another generating means at a next highest level, said generating means being responsive to control signals for (a) providing an image pixel value that is equal to an image pixel value output by a first one of the frame buffer means that is coupled to the input thereof, (b) for providing an image pixel value that is equal to an image pixel value output by a second one of the frame buffer means that is coupled to the input thereof, or (c) for providing an image pixel value that is equal to a combination of the image pixel values output by each of the first and the second ones of the frame buffer means, said image display apparatus preferably further including means, coupled to an output of each of said frame buffer means, for converting a format of the stored image pixel data to a predetermined format which preferably is R, G, B.





THIS SHEET HAS BEEN
REPLACED

17. Image display apparatus as set forth in claim 16, wherein said generating means is responsive to a pixel transparency value output by an associated one of said frame buffer means for mixing the image pixel values output by each of the first and the second frame buffer means in accordance with the pixel transparency value.
18. Image display apparatus as set forth in claim 16 or 17, wherein said control signals are generated by control means that includes look-up table means for generating the control signals on a pixel-by-pixel basis in accordance with a pixel color code and a display means window identifier, wherein said pixel color code and said display screen window identifier are stored within one of said frame buffer means that is associated with said control means.
19. Image display apparatus according to any one of claims 16 to 18, wherein said generating means includes first means for subtracting the pixel value output by the associated frame buffer means from the pixel value received from the generating means associated with the frame buffer means at the next lower priority level, second means for multiplying a difference value output by said first means times a transparency value associated with the pixel value output by the associated frame buffer means, and third means for adding a product value output by said second means to the pixel value received from the generating means associated with the frame buffer means at the next lower priority level.

