

TIME MULTIPLEXING PIXEL FRAME BUFFER VIDEO OUTPUT

BACKGROUND OF THE INVENTION

Field Of The Invention

This invention relates to computer systems, and, more particularly, to a RAMDAC (random access memory-digital-to-analog converter) used to transfer and process data from a frame buffer to an output display device.

History Of The Prior Art

One of the significant problems involved in increasing the operational speed of desktop computers has been in finding ways to increase the rate at which information is transferred to an output display device. Many of the various forms of data presentation which are presently available require that large amounts of data be transferred. For example, if a computer output display monitor is operating in a color mode in which 1280 X 1024 pixels are displayed on the screen at once and the mode is one in which thirty-two bits are used to define each pixel, then a total of over forty million bits of information must be transferred to the screen with each individual picture (called a "frame") that is displayed. Typically, sixty frames are displayed each second so that over one and one-half billion bits must be transferred each second in such a system. This requires a very substantial amount of processing power.

In order to provide such a large amount of information to an output display device, computer systems typically utilize a frame buffer which holds the pixel data which is to be displayed on the output display.

REPLACED SHEETS		SG
☒	Description : paged ..7... to paged ..7....	
☒	Claim(s) : paged ..11... to paged ..25....	
☐	Drawing(s) : paged to paged	
☐	Abstract : paged to paged	
DATE OF RECEIPT		

Typically a frame buffer offers a sufficient amount of random access memory to store one frame of data to be displayed. The information in the frame buffer is transferred to the display from the frame buffer sixty or more times each second. After (or during) each transfer, the pixel data in the frame buffer is updated with the new information to be displayed in the next frame.

Various improvements have been made to speed access in frame buffers. In DRAM frame buffers, pixel data may be read from the same port as data is written. This approach severely reduces the time available for rendering graphics data to the frame buffer. VRAM frame buffers add a separate video data port so that the main pixel port remains free for rendering. Two-ported video random access memory (VRAM) or frame buffer random access memory (FBRAM) has been substituted for dynamic random access memory so that information may be transferred from the frame buffer to the display at the same time other information is being loaded into the frame buffer.

One of the problems which all frame buffers have faced is caused by the method by which data is transferred from the frame buffer to an output display device. Typically, the display device is a cathode ray tube which renders the pixel data stored in the frame buffer on a screen in a series of rows. A typical display is comprised of 1024 horizontal rows, each of which includes as many as 1280 individual pixels. A frame is described on the display by writing individual rows of pixels starting at the upper left

corner of the display. Each row of ³pixels is rendered from left to right across the display before a next row in sequence is begun, When a row is completed, the next row below is begun at the left side of the screen. Each row is rendered in order until the last row at the bottom of the screen is completed. This completes one frame. Then the process starts over from the beginning with the next frame at the upper left corner of the display. As explained above, in the typical display sixty individual frames are presented each second.

In order to cause each of the pixels stored in the frame buffer to be presented at the appropriate position on the display, it is necessary to read the data for each pixel and transfer that data to the circuitry which controls its rendering on the output display device.

Frame buffers exist today that time multiplex the pixel data output of a RAM in order to pack 24-bit pixels onto a 32-bit data bus. This invention differs from such prior art approaches in that full 32-bit pixels are used, and the purpose is to allow a whole 32-bit pixel to live in a single RAM chip.

VRAMs of Samsung (Samsung WRAM) select 16 pins per RAM for their video port. However, this approach does not suggest that a whole 32-bit pixel be stored in the frame buffer, nor does it suggest that a 32-bit pixel be time multiplexed to get the pixel out of the frame buffer.

The data from the frame buffer is input to circuitry which converts the data from the frame buffer to a form usable by the

output display device. Figures ⁴1 and 2 each show a computer system in which the present invention may be utilized where data in a memory 11 from a host CPU 12 is placed on host bus 13 and passed by rendering controller 14 to the frame buffer memory shown in Figures 1 and 2 as VRAMs 15a-15d, although FBRAMs could be used as well. A RAMDAC 21 is coupled to the host bus through the rendering controller and to the frame buffer and includes a look-up table (or LUT which is the RAM part of the RAMDAC) and other elements for translating 16 bit data from VRAMs 15a-15d to a 64 or 128 bit digital RGB signal which is converted by a digital to analog converter (DAC) to three analog signals representing voltage levels for red, blue and green which when combined at a pixel location in monitor 25 create a desired color at that pixel. The particulars of the frame buffer, rendering controller and monitor components are well known in the art and will not be described herein except as necessary for a proper understanding of the invention. In this connection, for the most part, the present invention is directed to certain improvements to RAMDAC 21 which provide the enhanced capabilities of the invention.

SUMMARY OF THE INVENTION

For many graphics operations optimal performance is achieved by storing an entire 32-bit pixel in a single RAM chip. These operations may be Z-buffering, blending, and raster operations using an XOR function. When displaying video data from a frame buffer, pixels must be read out serially from the frame buffer at real-time speeds. The problem to be solved is how to get 32-bit

pixels out of a frame buffer RAM chip with the fewest pins. Pins add cost, so limiting pins provides a lower cost solution.

As noted above, VRAM or FBRAM frame buffers add a separate video data port so that the main pixel port remains free for rendering. The number of pins used for this second port will affect the frame buffer's RAM, board and digital to analog components cost.

In this connection, according to the present invention, a frame buffer memory with 16 pins for serial video output is used. An entire 32-bit pixel is stored in a single RAM chip. For convenience of notation, the 32-bit pixel is designated as containing four byte (8-bit) quantities: X, B, G and R.

On the first clock cycle, the X and B bytes are made available on the 16 pins of the frame buffer. On the next clock cycle, the G and R bytes are made available. Thus, over two cycles the entire 32-bit pixel is output from the frame buffer.

Another component called a DAC (from digital to analog converter) samples the X and B bytes on 16 input pins. The DAC stores these X and B bytes in an internal register. On the next clock cycle it samples the G and R bytes. The DAC then reassembles the X, B, G and R bytes into a single 32-bit pixel for conversion into video.

With this invention, 32-bit pixels are communicated across a 16-bit pixel data bus. A 16-bit data bus saves a total of 32 pins (16 at the RAM for sending and 16 at the DAC for receiving) over a

non-multiplexed 32 bit data bus. The 32 pins saved results in a lower frame buffer cost.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is a block diagram showing a system having a 64 bit frame buffer memory in which the present invention may be utilized.

Figure 2 is a block diagram showing a system having a 128 bit frame buffer memory in which the present invention may be utilized.

Figure 3 is a detailed block diagram of a RAMDAC which employs the invented time multiplexing of pixel data hardware.

Figure 4 is a timing diagram showing 2:1 single buffered interleaved pixel format.

Figure 5 is a timing diagram showing 2:1 double buffered interleaved pixel format.

Figure 6 is a timing diagram showing 4:1 single buffered interleaved pixel format.

Figure 7 is a timing diagram showing 4/2:1 single buffered interleaved pixel format.

Figure 8 is a timing diagram showing 4/2:1 double buffered interleaved pixel format.

Figure 9 is a timing diagram showing 8/2:1 single buffered interleaved pixel format.

Figure 10 is a timing diagram showing timing for the pixel port.

Figure 11 shows the SC pixel clock input to the pixel port.

Figure 12 is a timing diagram similar to Figure 10 showing further detail of pixel port timing.

several functional blocks as follows: CPU port, interface logic, address pointers and data registers 31, pixel port, pixel input registers and serialization 33, shadow and RAM look-up tables, transfer control and overlay/underlay logic 35, color model selection 37, cursor logic serialization 39, monitor serial port 41, diagnostic registers and control logic 43, digital-analog converters (DAC) 45a-45c and PLL clock synthesizer, pixel clock divider and video timing generator 49. The invention lies mainly in an implementation of the pixel port, pixel input registers and serialization 33 component of the RAMDAC. Therefore, the following description will be limited to ~~the pixel~~ ^{the pixel} port, pixel input registers and serialization, with information pertaining to the other components of the RAMDAC provided only as needed for an understanding of the present invention. Although the other components shown in Figure 3 may vary between RAMDACs of different manufacturers, persons skilled in the relevant art will recognize these various components and know how they or their equivalents may be implemented

The pixel port is a synchronous input port which accepts interleaved pixel data. Several interleaving formats are required. Selection among these utilizes register programming and is done as part of a boot time configuration process. RAMDAC 21 has two pixel ports, labeled A and B, with a programmable interleaving factor. This configuration accommodates double buffered operation for animation. Here, as in all other cases, the interleaving selection is made during configuration. The selection of port A or port B is



made by decoding a window attribute field of port A. In 4:1 and 8/2:1 pixel formats an X field comes from ports A and B. The contents of the X data field are interpreted as either a Window Identification (WID) index or as an Overlay Color. The Overlay Color case and selecting the particular interpretation of the X data field is discussed below.

For the case where the X field is interpreted as a WID, Window ID's (WIDs) are index addresses into a WID look up table which serve to select the pixel source, e.g. port A or B, and to associate the pixel with a particular color model. The X field is a component of every pixel and its content may differ in contiguous pixels. Therefore, port and color model selection must be performed for each individual pixel.

The described interleaving formats are divided into two broad categories. These are the single buffered interleaving format, and the double buffered interleaving format.

In 2:1 and 4/2:1 input formats, the X field from port A is used. The X field from port B is ignored.

In 4:1 and 8/2:1 input formats, the X field from each pixel is used.

The X field does not directly control port and color model selection. The contents of the lower five bits of the X field, X[04:00], constitute the address to the active WID LUT; hereafter called WID[05:00]. It is contained in the locations corresponding to these addresses and is used to effect the port and control

Figure 13 is a circuit diagram of an implementation of the pixel port input registers and serialization according to the present invention.

Figure 14 shows a pixel port interleaving format circuit for pixel 0.

Figure 15 shows a pixel port interleaving format circuit for pixel 1.

Figure 16 shows a pixel port interleaving format circuit for pixel 2.

Figure 17 shows a pixel port interleaving format circuit for pixel 3.

Figure 18 shows a pixel port interleaving format circuit for pixel 4.

Figure 19 shows a pixel port interleaving format circuit for pixel 5.

Figure 20 shows a pixel port interleaving format circuit for pixel 6.

Figure 21 shows a pixel port interleaving format circuit for pixel 7.

Figure 22 is a circuit timing diagram for the pixel port for pixel 0.

Figure 23 illustrates the interleaving format circuits 51 routing table.

DETAILED DESCRIPTION OF THE INVENTION

Figure 3 shows the components of a RAMDAC 21 which can be utilized to implement the present invention. The RAMDAC includes

color model selection according to definitions shown in Table 1,
 "Color Model Table Data Entry Codes," below.

Selected Input Port and Color Model	Color Model Control					
	5*	4	3	2	1	0
Input Port B - 24-Bit Non-Linear True Color	1	1	1	0	x	x
Input Port B - 24-Bit Linear True Color	1	1	0	1	x	x
Input Port B - 24-Bit Direct Color	1	1	0	0	x	x
Input Port B - 8-Bit Non-Linear Grey Scale from B Channel	1	0	1	0	1	1
Input Port B - 8-Bit Non-Linear Grey Scale from G Channel	1	0	1	0	1	0
Input Port B - 8-Bit Non-Linear Grey Scale from R Channel	1	0	1	0	0	1
Input Port B - 8-Bit Non-Linear Grey Scale from X Channel (8/2:1 or 4:1 only)	1	0	1	0	0	0
Input Port B - 8-Bit Linear Grey Scale from B Channel	1	0	0	1	1	1
Input Port B - 8-Bit Linear Grey Scale from G Channel	1	0	0	1	1	0
Input Port B - 8-Bit Linear Grey Scale from R Channel	1	0	0	1	0	1
Input Port B - 8-Bit Linear Grey Scale from X Channel (8/2:1 or 4:1 only)	1	0	0	1	0	0
Input Port B - 8-Bit Pseudo Color from B Channel	1	0	0	0	1	1
Input Port B - 8-Bit Pseudo Color from G Channel	1	0	0	0	1	0
Input Port B - 8-Bit Pseudo Color from R Channel	1	0	0	0	0	1
Input Port B - 8-Bit Pseudo Color from X Channel (8/2:1 or 4:1 only)	1	0	0	0	0	0
Input Port A - 24-Bit Non-Linear True Color	0	1	1	0	x	x
Input Port A - 24-Bit Linear True Color	0	1	0	1	x	x
Input Port A - 24-Bit Direct Color	0	1	0	0	x	x
Input Port A - 8-Bit Non-Linear Grey Scale from B Channel	0	0	1	0	1	1
Input Port A - 8-Bit Non-Linear Grey Scale from G Channel	0	0	1	0	1	0
Input Port A - 8-Bit Non-Linear Grey Scale from R Channel	0	0	1	0	0	1
Input Port A - 8-Bit Non-Linear Grey Scale from X Channel	0	0	1	0	0	0
Input Port A - 8-Bit Linear Grey Scale from B Channel	0	0	0	1	1	1
Input Port A - 8-Bit Linear Grey Scale from G Channel	0	0	0	1	1	0
Input Port A - 8-Bit Linear Grey Scale from R Channel	0	0	0	1	0	1
Input Port A - 8-Bit Linear Grey Scale from X Channel	0	0	0	1	0	0
Input Port A - 8-Bit Pseudo Color from B Channel	0	0	0	0	1	1
Input Port A - 8-Bit Pseudo Color from G Channel	0	0	0	0	1	0
Input Port A - 8-Bit Pseudo Color from R Channel	0	0	0	0	0	1
Input Port A - 8-Bit Pseudo Color from X Channel	0	0	0	0	0	0

Table 1, "Color Model Table Data Entry Codes,

Pixel Display Ordering

In each of the following described formats, pixel data port pin group 0 always has the leftmost pixel as viewed on the screen of all pixels coming in to the pixel port on a clock. Higher-

numbered bits in each pixel are the more significant bits of the pixels, i.e. cause a larger change in the DAC output voltage when selected for color palette bypass.

Pixel Port Signals

RED, GREEN, BLUE and Window Attribute Field Pixel Inputs

These are the video data and window attribute inputs. To facilitate discussion, assume that the pixel inputs are divided into two ports, labeled A and B which consist of four groups per port. Furthermore, each group is divided into an upper byte and a lower byte. Thus, the pixel port comprises a total of 128 pixel bits contained in groups 0 through 7. Table 2 illustrates these assignments.

Pixel Port	Group	Group Bits	Device Bits
		[15:8]	PB(63-56)
	7	[7:0]	PB(55-48)
		[15:8]	PB(47-40)
	6	[7:0]	PB(39-32)
B		[15:8]	PB(31-24)
	5	[7:0]	PB(23-16)
		[15:8]	PB(15-08)
	4	[7:0]	PB(07-00)
		[15:8]	PA(63-56)
	3	[7:0]	PA(55-48)
		[15:8]	PA(47-40)
	2	[7:0]	PA(39-32)
A		[15:8]	PA(31-24)
	1	[7:0]	PA(23-16)
		[15:8]	PA(15-08)
	0	[7:0]	PA(07-00)

Table 2. Pixel Port Naming Convention

The arrangement of data arriving at the pixel port is hereafter referred to as an interleaving format. RAMDAC 21 accommodates five interleaving formats which are selected by

configuration register programming performed at boot time. The five interleaving formats are defined below.

Pixel Formats Supported

Each of the pixel formats is explained and illustrated in the following sections. It should be noted that the serialized pixel detail is not intended to show a cycle relationship with data coming in. The pixel format is selected by programming the Video Format Control Register. The mapping of this register is shown in Table 3.

Bit(s)	Field	Reset Value	Description
15-4	Reserved	0x00	
3	Transparent Overlay Enable (0) Disabled (1) Enabled	0	When set to a logical zero, the overlay disabled state, the enable bit causes the multiplexer to select the Underlay path, i.e., WID[05:00] are passed to CMC[05:00]. When set to a logical 1, the overlay enabled state, the action of the multiplexer is controlled by the result of the equality comparison.
2	Double Buffer Enable (0) Single Buffered (1) Double Buffered	0	This field is valid only when in the 4/2:1 or 2:1 pixel format. Other formats require that this bit be set to 0.
1,0	Pixel Format Control (00) 2:1 (01) 4:1 (10) 4/2:1 (11) 8/2:1	00	Selects the pixel interleaving format. The LD frequency for each multiplex rate is: $f_{LD} = f_p/2$ MHz, $f_{LD} = f_p/4$ MHz, $f_{LD} = f_p/2$ MHz, $f_{LD} = f_p/4$ MHz.

Table 3. Video Format Control Register

2:1 - Single and Double Buffered Interleaving Formats

These formats are applicable when operated at pixel frequency, f_p , ≤ 135 MHz, with a LD frequency, $f_{LD} = f_p/2$ MHz.

These formats are illustrated in Figures 4 and 5. The function of the X field was explained above.

4:1 - Single Buffered Interleaving Format

This mode is valid for $f_p \leq 220$ MHz. LD frequency, $f_{LD} = f_p/4$ MHz. This format is illustrated in Figure 5.

4/2:1 - Single and Double Buffered Interleaving formats

This mode is applicable when operated at pixel frequency, f_p , ≤ 135 MHz. LD frequency, $f_{LD} = f_p/2$ MHz.

This format is illustrated in Figure 6 and Figure 7.

8/2:1 - Single Buffered Interleaving Format

This format is applicable when operated at frequencies f_p , ≤ 220 MHz. LD frequency, $f_{LD} = f_p/4$ MHz.

This format is illustrated in Figure 8.

Pixel Port Timing

The design incorporates circuitry to insure correct entry of pixel port data as the phase relationship of LD and pixel clock is varied between certain limits. This circuitry performs the required internal adjustments either during every vertical blanking interval or when invoked by an external mechanism. The mode of operation is controlled by register programming. The timing relationships of SC, LD, pixel clock and pixel data are specified Figures 10-12.

Table 4 provides a description of the various signals utilized by the RAMDAC.

Signal Name	I/O/Z	Description
D(7:0)	I/O/Z	CPU Data Bus. Bidirectional data. The CPU port will zero fill unused bits on data reads.
C(1:0)	I	CPU Control Bus Input. These signals serve a dual purpose. They define major divisions in the RAMDAC address space and the access type.
R/W	I	CPU Read/Write Control Input. Defines the transaction direction.
LB*	I	CPU Low Byte Control.
CE*	I	CPU Chip Enable Input. When negated, this signal causes the RAMDAC to ignore all CPU interface signals with the exception of reset.
P(A,B)(63:0)	I	Pixel Port Inputs. These inputs have internal pullup resistors that cause the logic level to be high if they are left unconnected.
LD	I	Pixel Port Load Clock. The rising edge of this signal captures input pixel data.
PVLD	I	Pixel Port Data Valid. This input is captured on the rising edge of LD, along with pixel data.
SC	O	Serial Clock Output. This signal is produced by the Pixel Clock Divider. It is meant to be used as the clock for the serial port of the video memory. Please refer to the description of the Pixel Clock Divider for details.
SCEN	O	Serial Clock Enable Output. This signal is produced by the timing generator and is meant to control the serial port of the video memory.
STSCAN	O	Horizontal scan line indicator. This signal is produced by the timing generator and is meant for use by external circuitry for the purpose of indexing the serial port of the video memory.
FIELD	I/O	Odd Field Indicator. This signal is produced by the timing generator and is meant for use by external circuitry for the purpose of indexing the serial port of the video memory.
XTAL1, XTAL2	I,O	PLL Reference Crystal.
COMP, COMP2		Compensation for internal reference amplifier.
CSYNC*	O	Composite Sync Output.
MON(3:0)	I	Monitor Serial Port Data
RESET*	I	Reset Input. This is the Reset signal. Its assertion causes a number of actions, these are described in following paragraphs.

Table 4

Referring now to Figure 13, the pixel port of the present invention may be implemented using interleaving format circuits 51, the specifics of which are described with reference to Figures 14-

21, multiplexor 53 (MPX1), pipeline register 55 (D REG), multiplexor 57 (MPX 2) and shift register 59 (SHIFT REG).

Figure 13 depicts the flow of signals and elements involved in converting video pixels provided in parallel into a serial stream of single pixels. Here, the various interleaving formats are accommodated and the selection of display buffer, in double buffer modes, is made.

Pixels are received from interleaving format circuits block 51 from the frame buffer memory, in several allowed parallel formats. These formats are described in Figures 4-9. The interleaving format circuits block 51 performs the task of undoing the interleaving and providing complete, 32 bit pixels at its output.

The interleaving format circuits block utilizes eight sub-blocks, each one manipulating incoming data to assemble one pixel. The circuits comprising these blocks are illustrated in Figures 14-21 for pixels 0-7 respectively. Note that these circuits are not identical but that they do have elements in common. These elements are flip-flop M2, flip-flop M3B and flip-flop M3C in the diagrams for pixels 0-3 and flip-flops M2, M3A and M3B in the diagrams for pixels 4-seven (the mnemonics differ but the functions are identical). These elements deal with the time multiplexed interleaving formats 4/2:1 and 8/2:1. Figure 22 depicts the action of the pixel 0 circuit in the 4/2:1 case, which is identical to the remaining cases in every respect except the period of LD and LD/2. Figure 22 shows the manner in which a

complete 32 bit pixel is assembled from two LD clock cycles each containing half of the pixel information. When either the 4/2:1 or 8/2:1 interleaving format is selected, bit 1 of the video format control register is set to logic 1. This level causes multiplexer M4 to pass the output of flip-flop M3B to shift register M5; this is the lower half of a pixel and comprises the GREEN and RED components of the pixel. The output of flip-flop M3C also connects to shift register M5. This connection conveys the upper half of the pixel, comprising the X and BLUE components of the pixel. The manner in which pixel 0 is assembled in the 2:1 and 4:1 interleaving formats is not described in the timing diagram. This subject is discussed in subsequent paragraphs.

As previously noted, the format circuits differ. They do so as an artifact of the design which utilizes simple circuitry to implement a seemingly complex task. That task is the re-organization of the incoming data, not only to satisfy the time multiplexing requirement, but also to accommodate single and double buffered operation as well as modes which are not time multiplexed. All of this is accomplished by routing the various groups of incoming pixels to the appropriate interleaving format circuit. This routing is depicted in Figure 23.

Returning to Figure 13, portions of the output of the interleaving format circuits are passed to two blocks. The first of these, titled D REG 55, is nothing more than a pipeline register. It accepts P0 through P7 from the interleaving format circuits. The second block multiplexor 53 is titled MPX 1. It

accepts P0 through P3. Multiplexor MPX 1 is used to select the appropriate buffer when the system is operated in 2:1 double buffered mode. The multiplexer is controlled by bits 1 and 0 of the video format control register as well as bit number 5 of the X components of P0 and P1. Not shown in the diagram is the connection to bit 2 of the user control register which enables or disables the double buffered mode. The combined action of these signals is as follows. If the double buffered mode is enable, (user control register) and if the 2:1 mode has been selected (video format control register) and bit 5 of the X component of P0 (for example) is 1 then the multiplexer passes P2, which is P0 of buffer B. If bit 5 were 0 instead of 1 than the multiplexer would pass P0 from buffer A. If the double buffered mode is not selected, or if the 2:1 mode is not selected, the multiplexer passes P0 and P1.

The output of the pipeline register, D REG, is treated in a manner similar to that described in the previous paragraph. However, in this instance, multiplexor MPX 2 deals with the 4/2: 1 double buffered mode. The control of this multiplexer is similar to that described, however it is the 4/2: 1 mode (from the video format control register) which forms part of the qualifier instead of the 2:1 mode.

The final element of the circuit is the shift register which receives P0 through P7 in parallel and produces a serial output consisting of one 32 bit pixel per pixel clock, starting with the location occupied by P0 in the illustration. That is the device

shifts in the direction of the lowest numbered pixel occupying the register. Although the register is shown to have eight levels, it does not always shift eight pixels. Indeed, eight pixels are only shifted in the 8/2:1 mode. Four pixels are shifted in the 4:1, 4/2: (single and double buffered) and two pixels are shifted in the 2:1 mode. This variation in depth is not accomplished by special control circuitry but rather by the nature of the PAR(ALLEL) LOAD clock driven by LD/n. The circuit which produces LD/n is not shown but its operation is described as follows. The state of bits 1 and 0 of the video format control register control a divider which acts to divide the input, LD, by two when in 8/2: 1 mode or 4/2: 1 mode. When in 2:1 or 4:1 mode, LD is not altered but is simply passed to the output LD/n. The effect of this circuit is to make the period of its output LD/n equal to the period occupied by m pixels, where m is equal to the interleaving factor.

Attached hereto as Appendices 1, 2 and 3 are specifications containing further information relating to the above-described invention.



CLAIMS

1. An apparatus for multiplexing pixel data from a frame buffer for use by a RAMDAC for display on a display device comprising:

a) interleaving format circuits for coupling to a frame buffer and receiving interleaved pixel data;

b) logic circuits coupled to the interleaving format circuits for processing predetermined portions of the received interleaved pixel data to produce pixel data for processing by said RAMDAC.

2. The apparatus defined by Claim wherein said interleaving format circuits include circuits which undo the interleaving performed by said frame buffer and assemble said pixel data as received from said frame buffer into corresponding sets of pixel data.

3. The apparatus defined by Claim 2 wherein said undoing and assembling circuits comprises sets of registers and multiplexors which are coupled together to form a plurality of sub-blocks which cooperatively operate so that each produces a complete pixel formed by a predetermined number of bits.

4. The apparatus defined by Claim 3 wherein predetermined ones of said sub-blocks receive signals which define the interleaving format performed by said frame buffer.

5. The apparatus defined by Claim 1 wherein the interleaved pixel data is in one of:

2:1 single buffered interleaving format;

2:1 double buffered interleaving format;

4:1 single buffered interleaving format;

4/2:1 single buffered interleaving format;

4/2:1 double buffered interleaving format;

8/2:1 single buffered interleaving format.

This Sheet Has Been Replaced





6. The apparatus defined by Claim 3 wherein said received pixel data comprises 128 pixel bits divided into two sets containing 64 bits each, the bits in said first set being designated PA(63:00), the bits in said second set being designated PB(63:00), wherein one of said sub-blocks comprises:

- a first register which receives bits PA(15:00) of said pixel data;
- a second register which receives bits PA(31:16) of said pixel data;
- a third register which receives bits PA(15:00) of said pixel data;
- a fourth register coupled to said first register;
- a multiplexor coupled to said second and third registers;
- a fifth register coupled to said fourth register and said multiplexor.

7. The apparatus defined by Claim 3 wherein said received pixel data comprises 128 pixel bits divided into two sets containing 64 bits each, the bits in said first set being designated PA(63:00), the bits in said second set being designated PB(63:00), wherein one of said sub-blocks comprises:

- a first register which receives bits PA(31:24) of said pixel data;
- a second register which receives bits PA(63:32) of said pixel data;
- a third register which receives bits PA(31:24) of said pixel data;
- a fourth register coupled to said first register;
- a multiplexor coupled to said second, third and fourth registers;
- a fifth register coupled to said multiplexor.

8. The apparatus defined by Claim 3 wherein said received pixel data comprises 128 pixel bits divided into two sets containing 64 bits each, the bits in said first set being designated PA(63:00), the bits in said second set being





designated PB(63:00), wherein one of said sub-blocks comprises:
a first register which receives bits PA(47:32) of said pixel data;

a second register which receives bits PB(31:00) of said pixel data;

a third register which receives bits PA(47:32) of said pixel data;

a fourth register coupled to said first register;

a multiplexor coupled to said second, third and fourth registers;

a fifth register coupled to said multiplexor.

9. The apparatus defined by Claim 3 wherein said received pixel data comprises 128 pixel bits divided into two sets containing 64 bits each, the bits in said first set being designated PA(63:00), the bits in said second set being designated PB(63:00), wherein one of said sub-blocks comprises:

a first register which receives bits PA(63:48) of said pixel data;

a second register which receives bits PB(63:32) of said pixel data;

a third register which receives bits PA(63:48) of said pixel data;

a fourth register coupled to said first register;

a multiplexor coupled to said second, third and fourth registers;

a fifth register coupled to said multiplexor.

10. The apparatus defined by Claim 3 wherein said received pixel data comprises 128 pixel bits divided into two sets containing 64 bits each, the bits in said first set being designated PA(63:00), the bits in said second set being designated PB(63:00), wherein one of said sub-blocks comprises:

a first register which receives bits PB(15:00) of said pixel data;

a second register which receives bits PB(15:00) of said pixel data;





a third register coupled to said first register;
a fourth register coupled to said second register and said third register.

11. The apparatus defined by Claim 3 wherein said received pixel data comprises 128 pixel bits divided into two sets containing 64 bits each, the bits in said first set being designated PA(63:00), the bits in said second set being designated PB(63:00), wherein one of said sub-blocks comprises:

a first register which receives bits PB(31:16) of said pixel data;

a second register which receives bits PB(31:16) of said pixel data;

a third register coupled to said first register;

a fourth register coupled to said second register and said third register.

12. The apparatus defined by Claim 3 wherein said received pixel data comprises 128 pixel bits divided into two sets containing 64 bits each, the bits in said first set being designated PA(63:00), the bits in said second set being designated PB(63:00), wherein one of said sub-blocks comprises:

a first register which receives bits PB(47:32) of said pixel data;

a second register which receives bits PB(47:32) of said pixel data;

a third register coupled to said first register;

a fourth register coupled to said second register and said third register.

13. The apparatus defined by Claim 3 wherein said received pixel data comprises 128 pixel bits divided into two sets containing 64 bits each, the bits in said first set being designated PA(63:00), the bits in said second set being designated PB(63:00), wherein one of said sub-blocks comprises:

a first register which receives bits PB(63:48) of said pixel data;





a second register which receives bits PB(63:48) of said pixel data;
a third register coupled to said first register;
a fourth register coupled to said second register and said third register.

14. The apparatus defined by Claim 1 wherein said processing means comprises:

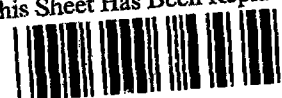
a first multiplexor coupled to said interleaving format circuits;
a pipeline register coupled to said first multiplexor and said interleaving format circuits;
a second multiplexor coupled to pipeline register;
a shirt register coupled to said pipeline register and said second multiplexor.

15. The apparatus defined by Claim 14 wherein said interleaving format circuits assemble said pixel data as received from said frame buffer into corresponding sets of pixel data and said first multiplexor is coupled to said interleaving format circuits so that a first set of said pixel data is a first selectable input to said first multiplexor; a second set of said pixel data is a second selectable input to said first multiplexor; a third set of said pixel data is a third selectable input to said first multiplexor; and a fourth set of said pixel data is a fourth selectable input to said first multiplexor.

16. The apparatus defined by Claim 15 wherein said pipeline register is coupled to said first multiplexor to receive first and second selected outputs from said first multiplexor as first and second inputs and is coupled to said interleaving format circuits to receive:

as a third input, said third set of pixel data;
as a fourth input, said fourth set of pixel data;
as a fifth input; a fifth set of pixel data;
as a sixth input, a sixth set of pixel data;
as a seventh input, a seventh set of pixel data;

This Sheet Has Been Replaced





as an eighth input, an eighth set of pixel data.

17. The apparatus defined by Claim 16 wherein said second multiplexor is coupled to said pipeline register to receive as a first selectable input, said first through fourth sets of pixel data output from said pipeline register, and as a second selectable input, said fifth through eighth sets of pixel data output from said pipeline register.

18. The apparatus defined by Claim 17 wherein said shift register is coupled to said multiplexor so that first through fourth sets of pixel data input to said shift register are a selected output of said second multiplexor, and said shift register is coupled to pipeline register so that fifth through eighth sets of pixel data input to said shift register are fifth through eighth sets of pixel data output from said pipeline register.

19. A method for multiplexing pixel data from a frame buffer for use by a RAMDAC for display on a display device comprising the sets of:

- a) undoing the interleaving performed by said frame buffer and assembling said pixel data as received from said frame buffer into corresponding sets of pixel data;
- b) processing predetermined portions of the received interleaved pixel data to produce pixel data for processing by said RAMDAC.

20. The method defined by Claim 19 wherein said undoing and assembling produces a plurality of complete pixel sets, each formed by a predetermined number of bits.

21. The method defined by Claim 20 wherein said undoing and assembling steps produces first through eighth sets of complete pixel data.

22. The method defined by Claim 21 wherein said processing

This Sheet Has Been Replaced





step comprises the steps of:

first selecting first and second sets of pixel data from among said first through fourth sets of pixel data;

passing said first and second sets of selected pixel data and said third through eighth sets of pixel data;

second selecting first through fourth sets of said pixel data from among said first through fourth sets of said pixel data on the one hand and from among said fifth through eighth sets of said pixel data on the other hand;

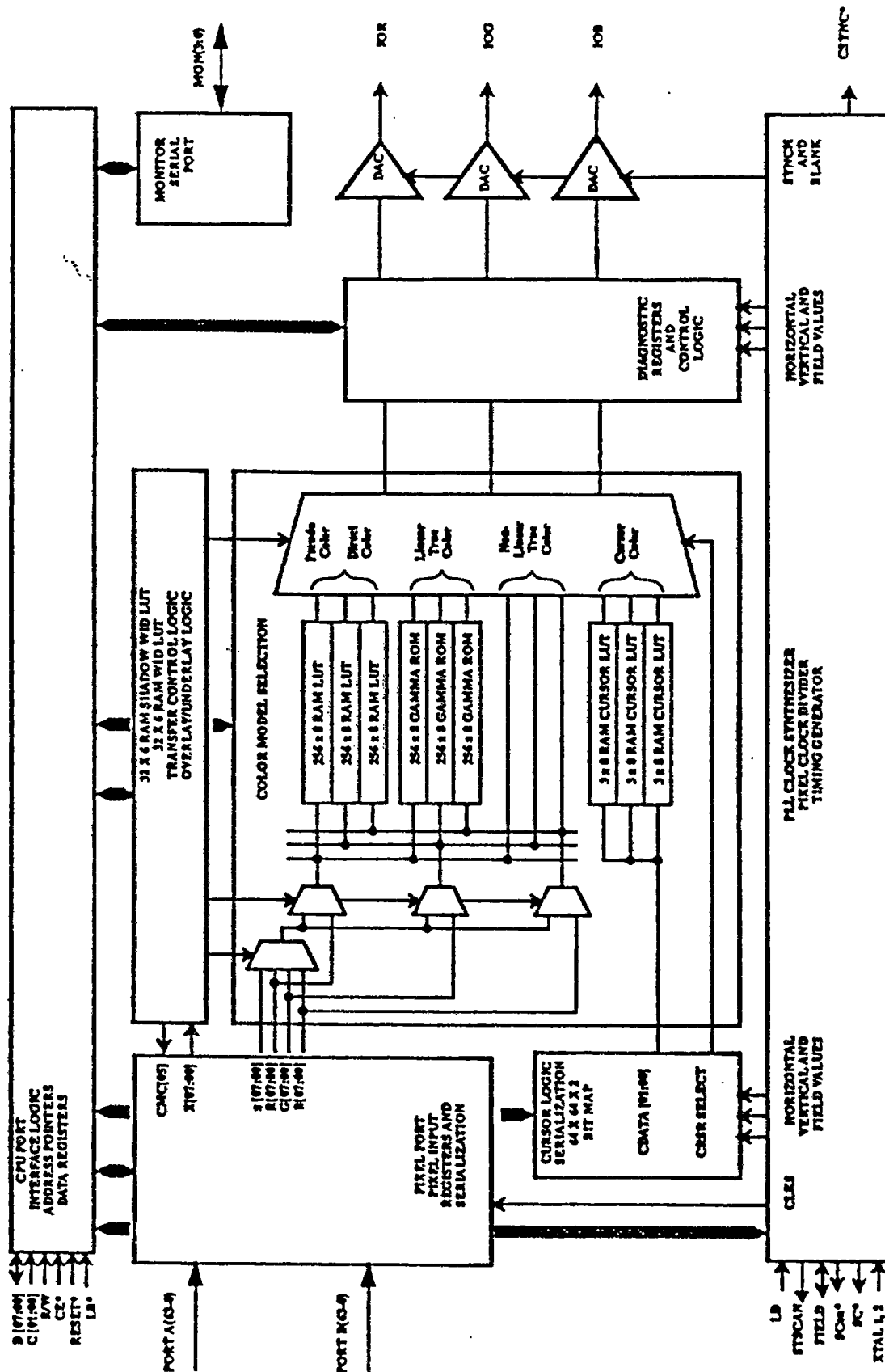
passing said second selected sets of pixel data as first through fourth sets of pixel data and said fifth through eighth sets of pixel data.



TIME MULTIPLEXING PIXEL FRAME BUFFER VIDEO OUTPUT
ABSTRACT OF THE DISCLOSURE

A method and apparatus for multiplexing pixel data from a frame buffer to a RAMDAC to reduce the number of pins required. For many graphics operations optimal performance is achieved by storing an entire 32-bit pixel in a single RAM chip. When displaying video data from a frame buffer, pixels must be read out serially from the frame buffer at real-time speeds. A frame buffer memory with 16 pins for serial video output is used. An entire 32-bit pixel is stored in a single RAM chip. For a 32-bit pixel containing four byte (8-bit) quantities designated X, B, G and R, on the first clock cycle, the X and B bytes are made available on the 16 pins of the frame buffer. On the next clock cycle, the G and R bytes are made available. Thus, over two cycles the entire 32-bit pixel is output from the frame buffer to a RAMDAC which samples the X and B bytes on 16 input pins. The RAMDAC stores these X and B bytes in an internal register. On the next clock cycle it samples the G and R bytes. The DAC then reassembles the X, B, G and R bytes into a single 32-bit pixel for conversion into video. In this manner, 32-bit pixels are communicated across a 16-bit pixel data bus.

(Figure 3)



Function		Version A	Version B
Maximum Pixel Frequency	135 MHz	●	
	220 MHz		●
Pixel Port Data Size	64 bit	●	
	128 bit		●
Pixel Port Interleaving	$8/4 : 1$		●
	$4/2 : 1$ port B		●
	$4/2 : 1$ port A	●	●
	$4 : 1$	●	●
	$2 : 1$	●	●
WID LUT, RAM	32×6	●	●
Shadow WID LUT, RAM	32×6	●	●
Color LUT, RAM	$256 \times 8 \times 3$	●	●
LUT bypass path		●	●
Cursor Color LUT, RAM	$3 \times 8 \times 3$	●	●
Cursor Bit Map	$32 \times 32 \times 2$	●	●
PLL Pixel Clock Synthesizer	programmable	10 to 135	10 to 220
Pixel Clock Divider	programmable	$1/4, 1/2$	$1/4, 1/2$
Timing Generator	programmable	●	●
Cursor Control Logic		●	●
DACs, Three, R, G, B	8 bits per chan.	●	●
JTAG Port	5 bit	●	●
CPU Port	14 bit, parallel	●	●
Monitor ID/Control Port	4 bit, serial	●	●

2 CPU Port

The CPU port is the path of communication between the system and the various control registers and state within the RAMDAC. This communication is regulated by an interface protocol which comprises electrical and operational characteristics. The protocol defined here is an extension of the protocol used for the Brooktree Bt458.

2.1 Interface Signals

2.1.1 D[07:00]

Bidirectional data

2.1.2 C[01:00]

Input. These signals serve a dual purpose. They define major divisions in the RAMDAC address space and the access type. Please refer to the address map for specifics.

2.1.3 R/W

Input. Defines the transaction direction.

2.1.4 CE

Input. Chip enable. When negated, this signal causes the RAMDAC to ignore all CPU interface signals with the exception of reset.

2.1.5 RS

Input. This is the Reset signal. Its' assertion causes a number of actions, these are described in following paragraphs.

2.2 Interface Type

2.2.1 This is an asynchronous interface, providing bidirectional communication.

2.3 Interface Timing and Electrical Characteristics

2.3.1 Timing and electrical characteristics are defined in tables and diagrams which appear at the end of this document. It is important to note that the electrical characteristics must be consistent with LVTTTL specifications operating at $V_{CC} = 3.3$ vdc. Furthermore, no output or input, with the possible exception of PECL clocks, shall be referenced to any voltage which is less than zero volts or greater than 3.3 volts.

2.4 Operational Characteristics and Features

The RAMDAC address space is larger than can be accommodated by an eight bit address. Furthermore, some of the state is greater than eight bits in width. In order to improve efficiency, transactions (read and write) are defined to be of two kinds. One kind, the indirect access, requires an address pointer, the other kind, the direct access, does not. The address pointer can be written or read, as a direct access, via the CPU port and provides an auto-increment function in the byte and word dimensions. Direct and Indirect transactions are differentiated by encoding of interface bits C[01:00]. An address map will be provided in a forthcoming revision of this document.

2.4.1 All versions of this RAMDAC include cursor control logic, a cursor bit map and a cursor color LUT. These versions require a second and wholly independent address pointer for all cursor state. All cursor state will be identified by a unique value or set of values of the interface bits C[01:00].

2.4.2 When asserted, system reset causes all address pointers to assume the reset state i.e., pointer value equals zero.

3 Pixel Port

This synchronous input port accepts interleaved pixel data. Several interleaving formats are required. Selection among these shall be controlled by means of register programming and is done as part of a boot time configuration process. Versions B and C of the RAMDAC specify two pixel ports, labeled A and B, with an interleaving factor of $4/2 : 1$. This configuration accommodates double buffered operation for animation. Here, as in all other cases, the interleaving selection is made during configuration, while the selection of port A or port B is made by decoding the Window Attribute field of port A. It is important to note that X values are congruent to pixels. Refer to the paragraph on color model and port selection for a description of the window attribute feature.

3.1 Pixel Port Signals

3.1.1 RED, GREEN, BLUE and Window Attribute Field Pixel Inputs

These are the video data and window attribute inputs. RAMDAC versions A and B differ in that version B has twice the number of input pins. To facilitate discussion, assume that the pixel inputs of version A are a subset of version B. Also assume that these inputs are divided into two ports, labeled A and B which consist of four groups per port. Furthermore, each group is divided into an upper byte and a lower byte. Thus, the pixel port for version A comprises a total of 64 pixel bits contained in groups 0 through 3, while versions B comprises a total of 128 pixel bits contained in groups 0 through 7. The following paragraphs illustrate these assignments in table form.

3.1.1.1 Port B, applicable to versions B and C.

group7, pixel bits [15:08]

group7, pixel bits [07:00]

group6, pixel bits [15:08]

group6, pixel bits [07:00]

group5, pixel bits [15:08]

group5, pixel bits [07:00]

group4, pixel bits [15:08]

group4, pixel bits [07:00]

3.1.1.2 Port A, applicable to all versions.

group3, pixel bits [15:08]

group3, pixel bits [07:00]

group2, pixel bits [15:08]

group2, pixel bits [07:00]

group1, pixel bits [15:08]

group1, pixel bits [07:00]

group0, pixel bits [15:08]

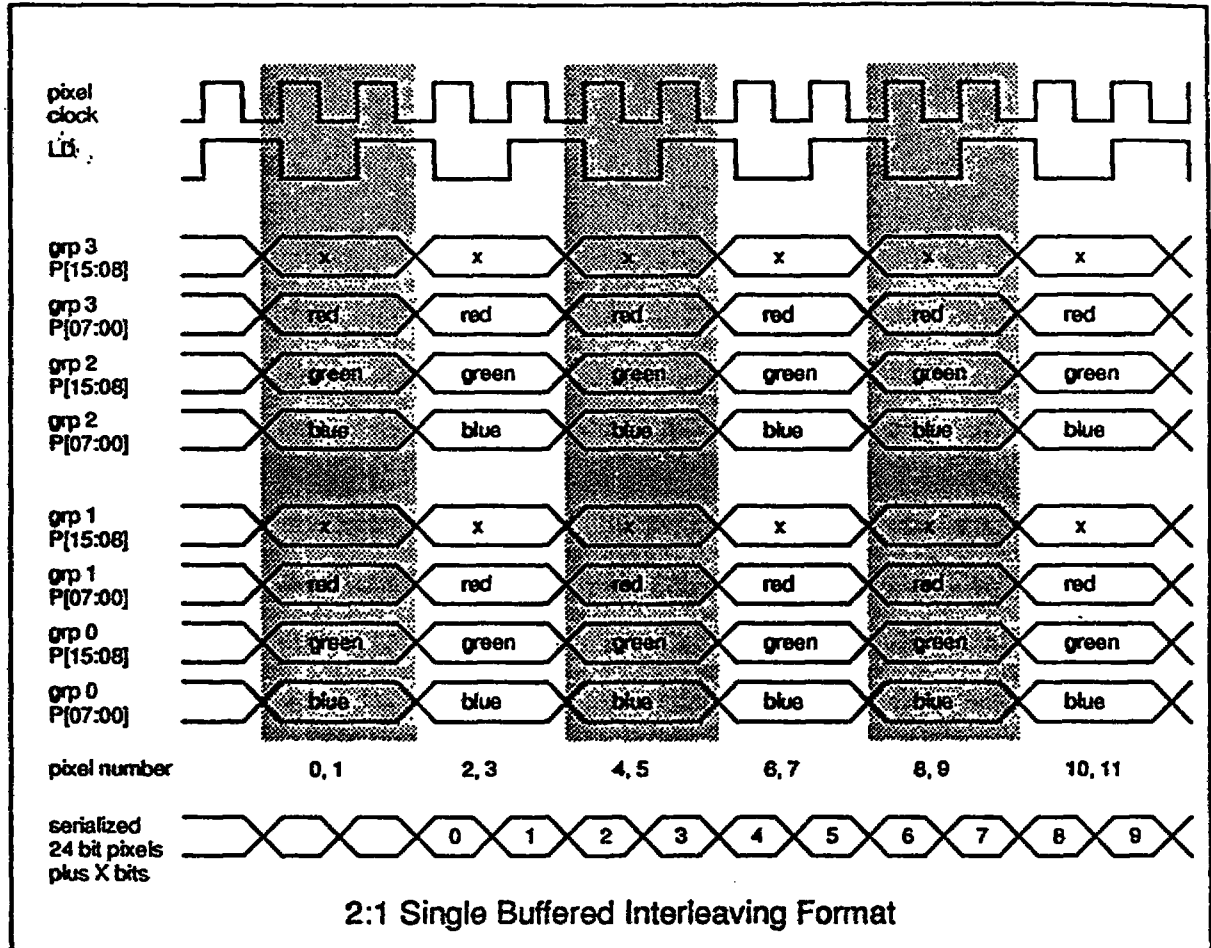
group0, pixel bits [07:00]

3.1.2 The arrangement of data arriving at the pixel port is hereafter referred to as an interleaving format. The RAMDAC shall accommodate five interleaving formats which are selected by means of configuration register programming performed at boot time. The five interleaving formats are defined in the following paragraphs.

- 3.1.2.1 2:1, single buffered interleaving format, applicable to all versions when operated at pixel frequency, $f_p \leq 135$ MHz.

LD frequency, $f_{LD} = f_p/2$ MHz.

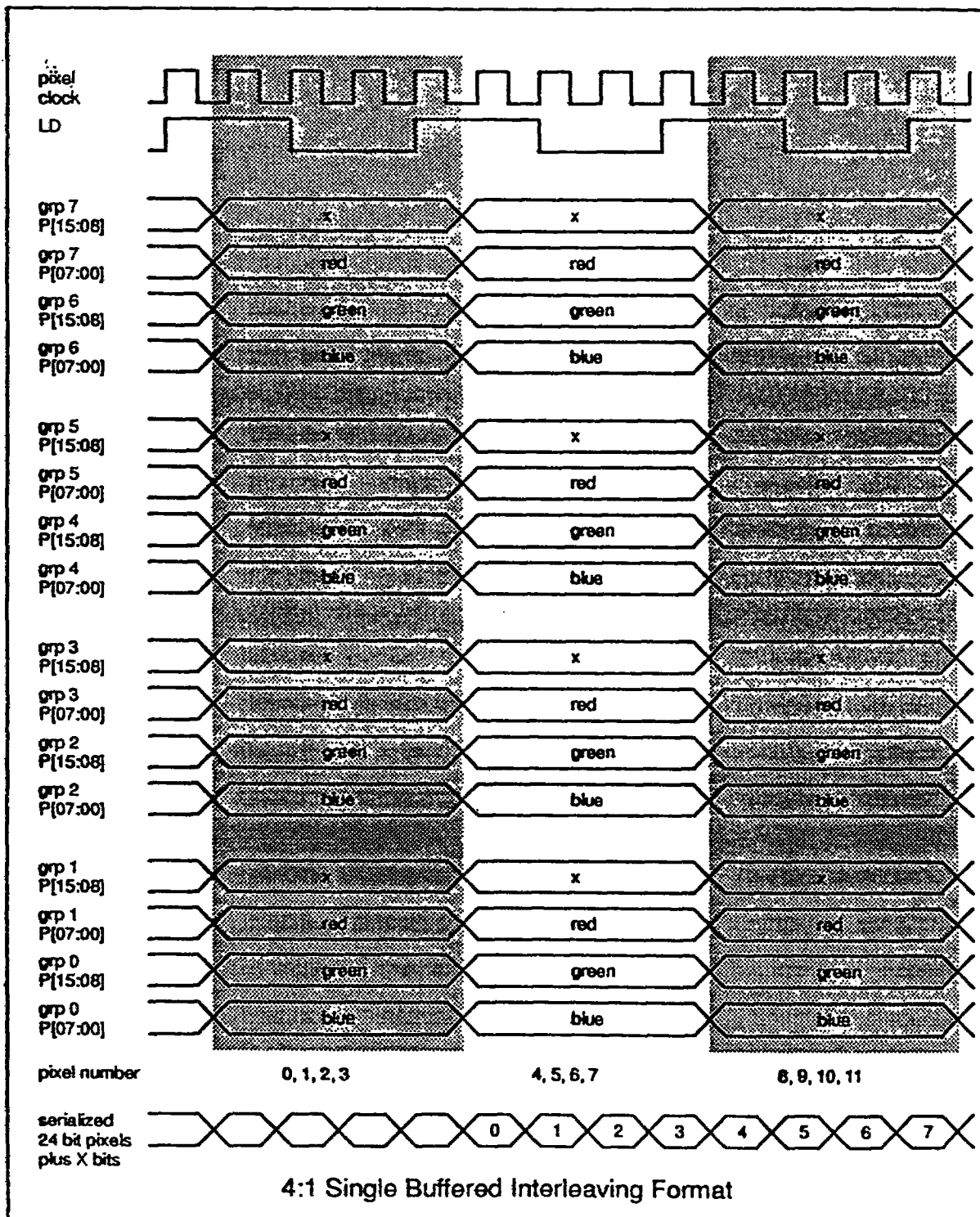
The function of the X field is explained in the paragraph which discusses color model selection.



3.1.2.2 4:1, single buffered interleaving format, applicable to:

Versions B. Pixel frequency, $f_p \leq 220$ MHz. LD frequency, $f_{LD} = f_p/4$ MHz.

The function of the X field is explained in the paragraph which discusses color model selection.

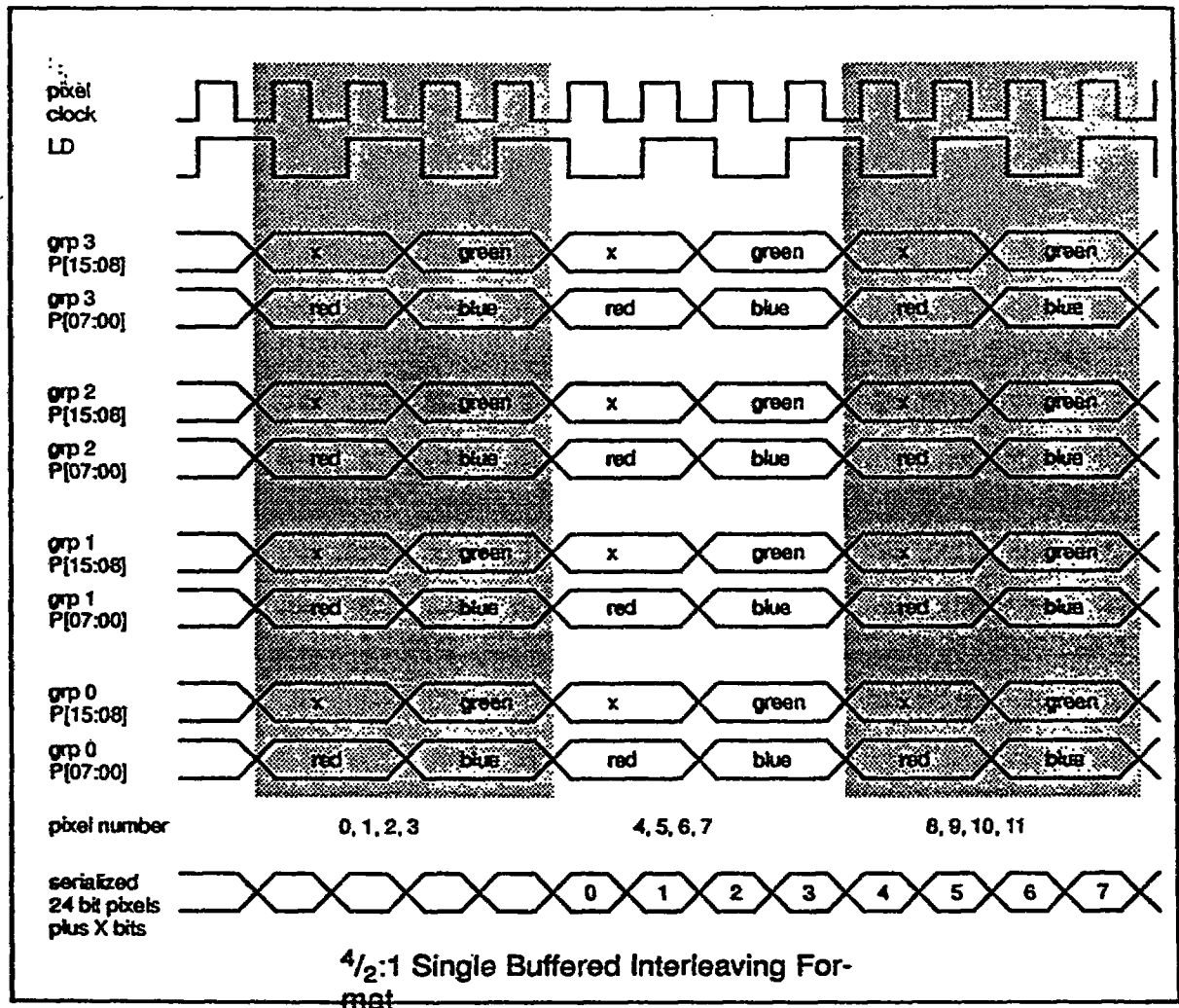


3.1.2.3 $4/2 : 1$, single buffered interleaving format, applicable to:

Port A of all versions when operated at pixel frequency, $f_p \leq 135$ MHz. LD frequency, $f_{LD} = f_p/2$ MHz.

Ports A and B of versions B when operated in double buffered mode at pixel frequency, $f_p \leq 135$ MHz. LD frequency, $f_{LD} = f_p/2$ MHz.

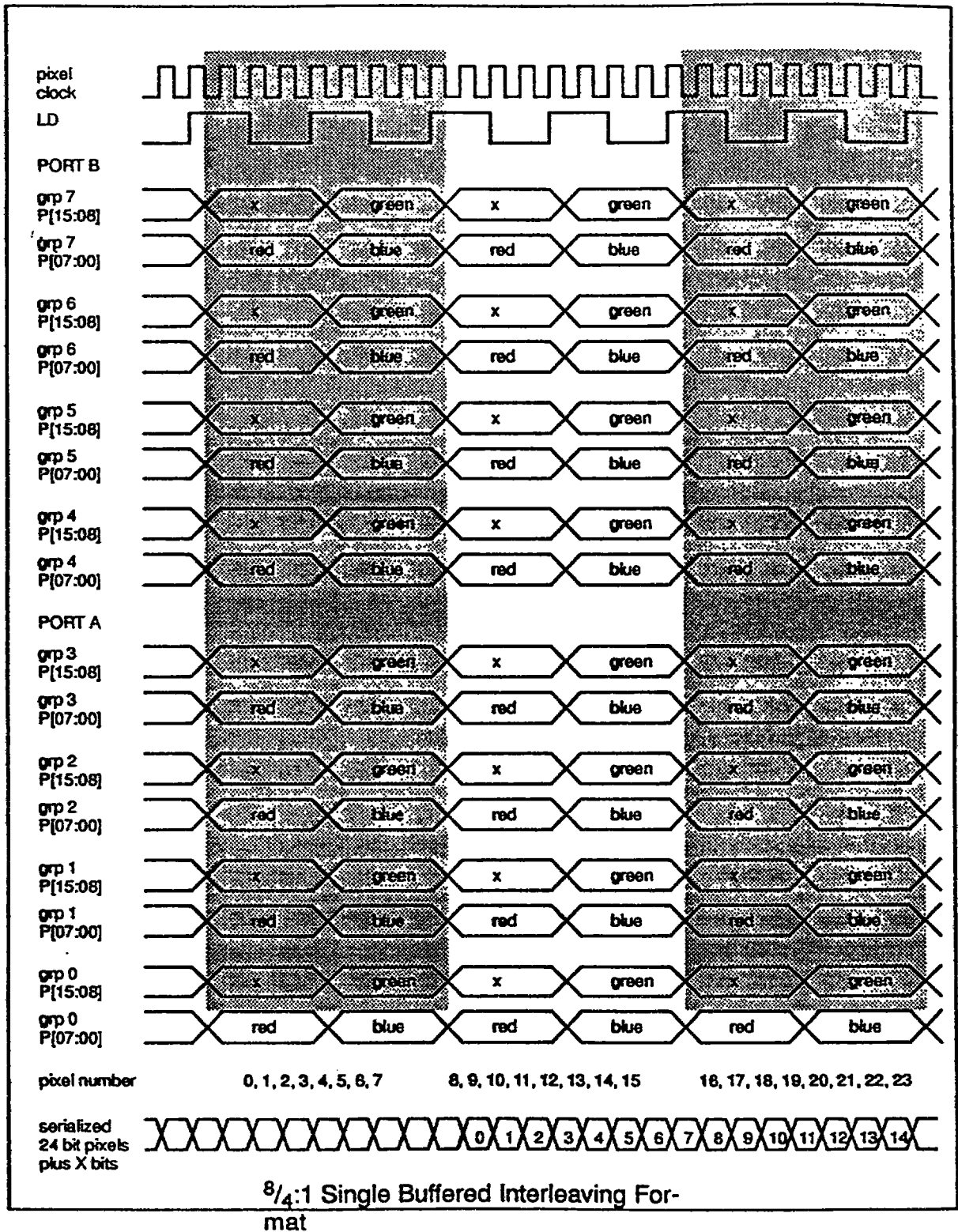
The function of the X field is explained in the paragraph which discusses color model selection.



3.1.2.4 $8/4 : 1$, single buffered interleaving format, applicable to:

Versions B. Pixel frequency, $f_p \leq 220$ MHz. LD frequency, $f_{LD} = f_p/4$ MHz.

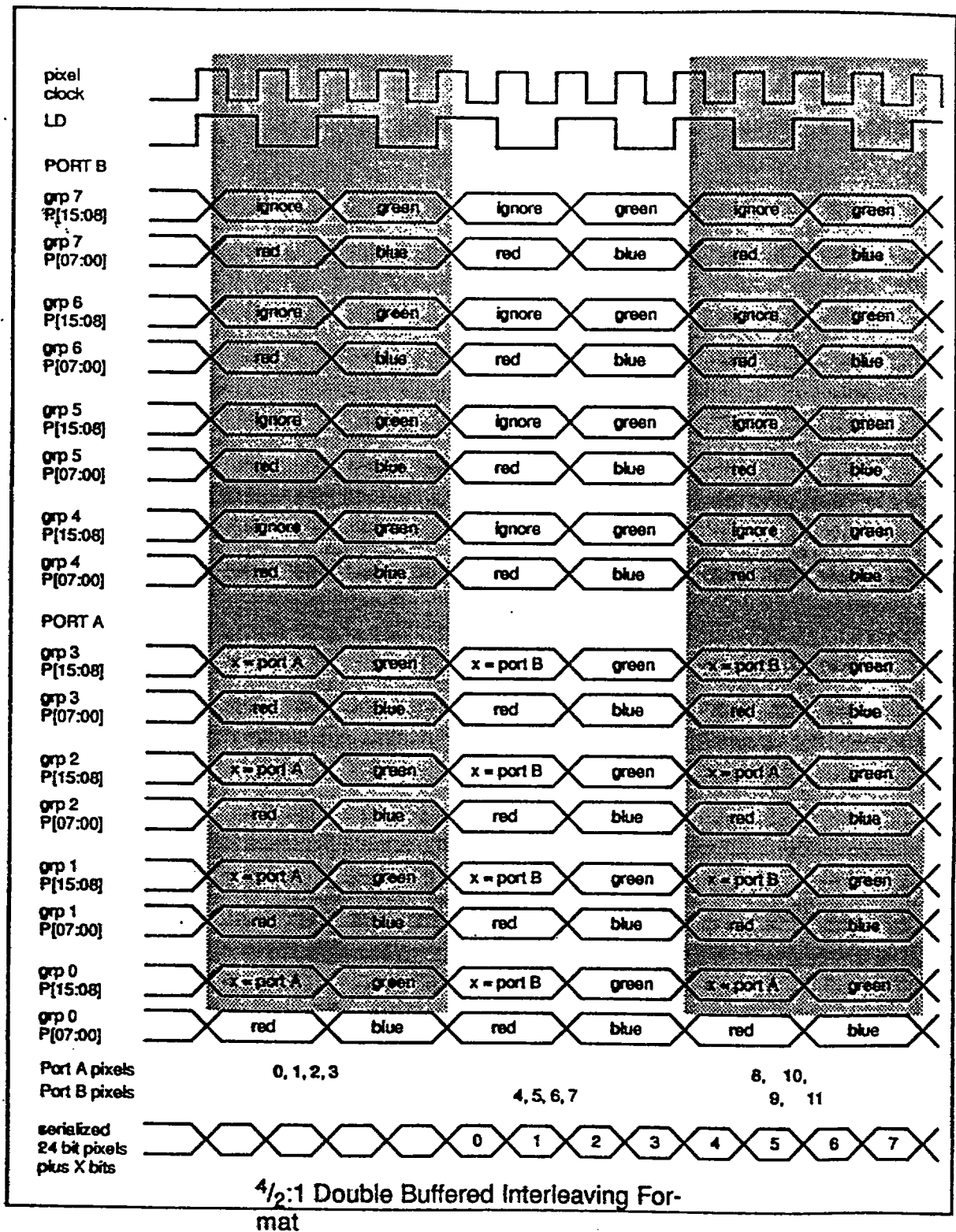
The function of the X field is explained in the paragraph which discusses color model selection.



3.1.2.5 $4/2 : 1$, double buffered interleaving format, applicable to:

Versions B. Pixel frequency, $f_p \leq 135$ MHz. LD frequency, $f_{LD} = f_p/2$ MHz.

The function of the X field is explained in the paragraph which discusses color model selection.



3.1.3

LD

Input clock. Pixel data are stored on the rising edge of this clock. Please refer to the section describing the clock generator for additional information.

3.1.4 VB, HB, FLD

Outputs. These are Vertical Blanking, Horizontal Blanking and Field. These signals are produced by the timing generator and are meant for use by external circuitry for the purpose of indexing the serial port of the video memory. Please refer to the description of the timing generator for details.

3.1.5 SCen

Output. This signal is produced by the timing generator and is meant to control the serial port of the video memory. Please refer to the description of the timing generator for details.

3.1.6 SC

Output. This signal is produced by the Pixel Clock Divider. It is meant to be used as the clock for the serial port of the video memory. Please refer to the description of the Pixel Clock Divider for details.

3.2 Pixel Port Timing

- 3.2.1 The design shall incorporate circuitry to insure correct entry of Pixel Port data as the phase relationship of LD and pixel clock is varied between certain limits. This circuitry shall perform the required internal adjustments either during every vertical blanking interval or when invoked by an external mechanism. The mode of operation shall be controlled by means of register programming. The timing relationships of SC, LD, pixel clock and pixel data are specified in diagrams and tables which appear at the end of this document.

4 Color Model Selection, Transparent Overlays and the Window Attribute LUT

- 4.1 The diagrams describing the various interleaving formats include a data field named X. The contents of this field are interpreted as either Window Attribute Data (alternatively called Window Identification or WID data) or as an Overlay Color. The case where the X field is interpreted as Window Attribute data is discussed first. The Overlay Color case and the means of selecting the particular interpretation of the X data field is discussed in following paragraphs.

Window Attribute data, after transformation by the WID look up table, serve to select the pixel source, e.g. port A or B, and to associate the pixel with a particular color model. The X field is a component of every pixel and its content may differ in contiguous pixels. Therefore, port and color model selection shall be performed for each individual pixel.

- 4.2 The described interleaving formats are divided into two broad categories. These are the Single Buffered interleaving format, applicable to all versions of the RAMDAC, and the Double Buffered interleaving format which applies only to version B. It is important to note that versions B shall accommodate both categories. Of relevance here is the difference in the location of the X field.

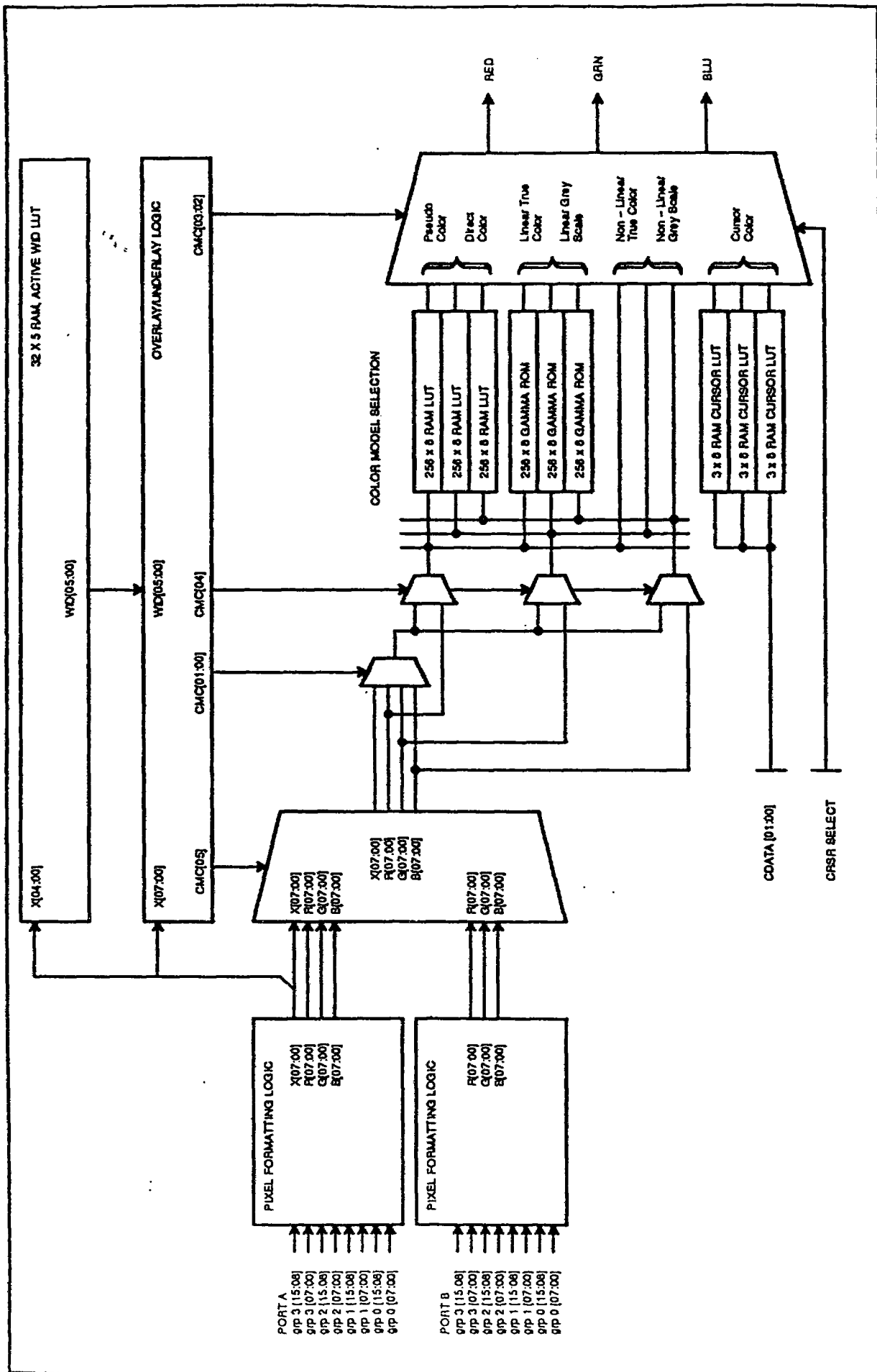
- 4.2.1 All interleaving formats of the Single Buffered category shall include the X field in the upper byte of each pixel group, as indicated in preceding illustrations.

- 4.2.2 The Double Buffered category shall include the X field in the upper byte of all of the port A pixel groups. The corresponding locations of port B shall be ignored by the RAMDAC when operating in the double buffered mode. Refer to the illustration of the double buffered category.

- 4.3 The X field shall not directly control port and color model selection. The contents of the lower five bits of the X field, X[04:00], constitute the address to the active WID LUT; the data, hereafter called WID[05:00], contained in the locations corresponding to these addresses, shall be used to effect the port and control color model selection according to definitions given in the following table.

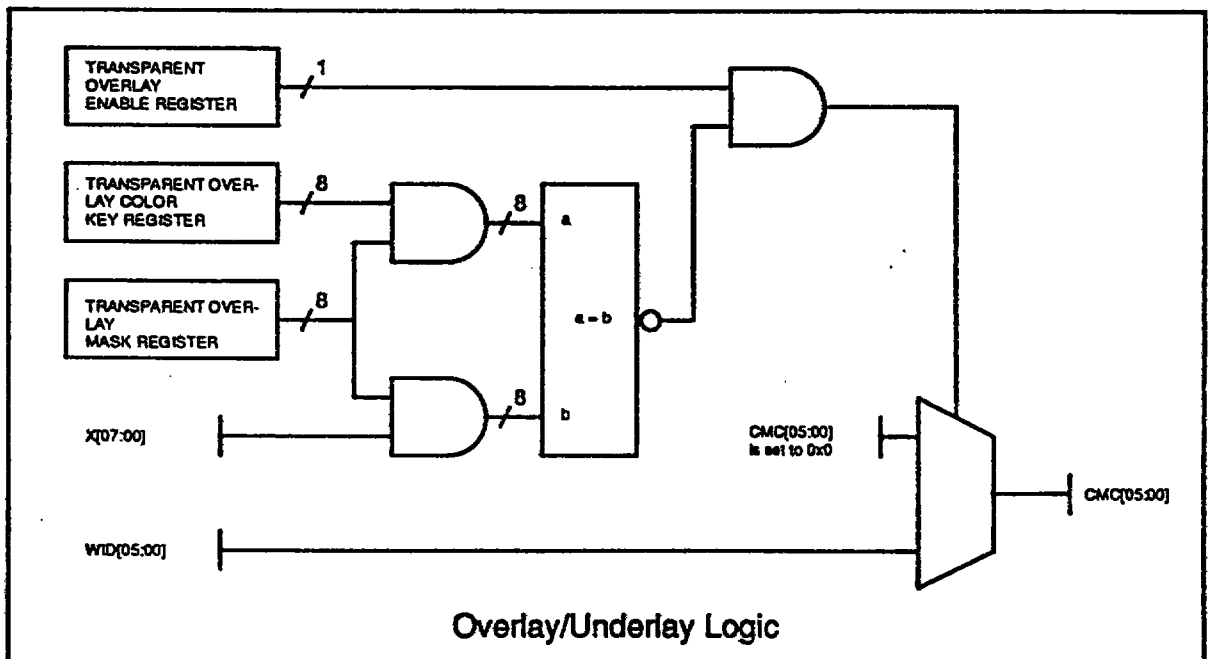
Selected Input Port and Color Model	Color Model Control					
	5	4	3	2	1	0
Input Port B – 24 Bit Non – Linear True Color	1	1	1	0	x	x
Input Port B – 24 Bit Linear True Color 1	1	0	1	x	x	
Input Port B – 24 Bit Direct Color	1	1	0	0	x	x
Input Port B – 8 Bit Non – Linear Grey Scale From B Channel	1	0	1	0	1	1
Input Port B – 8 Bit Non – Linear Grey Scale From G Channel	1	0	1	0	1	0
Input Port B – 8 Bit Non – Linear Grey Scale From R Channel	1	0	1	0	0	1
Reserved	1	0	1	0	0	0
Input Port B – 8 Bit Linear Grey Scale From B Channel	1	0	0	1	1	1
Input Port B – 8 Bit Linear Grey Scale From G Channel	1	0	0	1	1	0
Input Port B – 8 Bit Linear Grey Scale From R Channel	1	0	0	1	0	1
Reserved	1	0	0	1	0	0
Input Port B – 8 Bit Pseudo Color From B Channel	1	0	0	0	1	1
Input Port B – 8 Bit Pseudo Color From G Channel	1	0	0	0	1	0
Input Port B – 8 Bit Pseudo Color From R Channel	1	0	0	0	0	1
Reserved	1	0	0	0	0	0
Input Port A – 24 Bit Non – Linear True Color	0	1	1	0	x	x
Input Port A – 24 Bit Linear True Color 0	1	0	1	x	x	
Input Port A – 24 Bit Direct Color	0	1	0	0	x	x
Input Port A – 8 Bit Non – Linear Grey Scale From B Channel	0	0	1	0	1	1
Input Port A – 8 Bit Non – Linear Grey Scale From G Channel	0	0	1	0	1	0
Input Port A – 8 Bit Non – Linear Grey Scale From R Channel	0	0	1	0	0	1
Input Port A – 8 Bit Non – Linear Grey Scale From X Channel	0	0	1	0	0	0
Input Port A – 8 Bit Linear Grey Scale From B Channel	0	0	0	1	1	1
Input Port A – 8 Bit Linear Grey Scale From G Channel	0	0	0	1	1	0
Input Port A – 8 Bit Linear Grey Scale From R Channel	0	0	0	1	0	1
Input Port A – 8 Bit Linear Grey Scale From X Channel	0	0	0	1	0	0
Input Port A – 8 Bit Pseudo Color From B Channel	0	0	0	0	1	1
Input Port A – 8 Bit Pseudo Color From G Channel	0	0	0	0	1	0
Input Port A – 8 Bit Pseudo Color From R Channel	0	0	0	0	0	1
Input Port A – 8 Bit Pseudo Color From X Channel	0	0	0	0	0	0

The diagram on the following page describes the color model selection mechanism.



4.4

As noted in a previous paragraph, the data in the X field are interpreted as either Window Attribute data or as an Overlay color. The purpose in this is to accommodate transparent overlays without incurring the expense of additional planes in the frame buffer memory and additional look up tables in the RAMDAC. This purpose is accomplished by using the 256 x 8 x 3 RAM LUT to contain overlay colors and by circuitry which regulates the selection of overlay or underlay data for display. This regulatory function appears in the color model selection diagram as the Overlay/Underlay Logic block. The details are diagramed below. The principle of operation is described as follows. The contents of two registers, the mask register and the color key register, define a transparent overlay color key value. The color key value is compared to the value (modified by the overlay mask register) contained in X[07:00]. If the two values are equal, the overlay is said to be transparent and the underlay color is displayed, i.e., the multiplexer passes WID[05:00] to CMC[05:00]; therefore, the color model and the color source are chosen by the contents of the Window Attribute word. If the two values are not equal, the overlay color is displayed, i.e., the multiplexer passes 0x00 to CMC[05:00]; therefore, the color model and color source are chosen by a constant. This implementation defines the constant as 0x00, which causes the selection of the 8 bit pseudo color model, with color data originating in the X inputs of port A (refer to the color model selection table). Consequently, the data contained in the X field serve to address the color look up table and thereby select the overlay color.



4.4.1

Transparent Overlay Enable Register

Although a register is called for in the diagram, this single bit quantity may be implemented as a field of some register. When set to a logical zero, the overlay disabled state, the enable bit causes the multiplexer to select the Underlay path, i.e., WID[05:00] are passed to CMC[05:00]. When set to a logical 1, the overlay enabled state, the action of the multiplexer is controlled by the result of the equality comparison. The transparent overlay enable register shall be accessed via the CPU port. The system reset signal, when asserted, shall cause the register to assume the overlay disabled state.

4.4.2 Transparent Overlay Mask Register

This eight bit quantity operates on the contents of the overlay color key register and the X data to define the extent (in the bit dimension) of the color key value. Overlay Mask register bits set to logical one allow the corresponding bits of the color key register and the X field to participate in the color key comparison. Overlay mask bits set to logical zero preclude the corresponding bits of the color key register and the X field from comparison. The transparent overlay mask register shall be accessed via the CPU Port. Register initialization is not required.

4.4.3 Transparent Overlay Color Key Register

This eight bit quantity defines the color key value, subject to the contents of the mask register. The color key register shall be accessed via the CPU port. Register initialization is not required.

4.5 The WID LUT and the SHADOW WID LUT

All versions of the RAMDAC shall incorporate two 32 x 6 LUTs which form part of the color model selection circuitry. These are the WID and SHADOW WID LUTs. Operating in conjunction with a control register and a vertical blanking synchronizing mechanism, these LUTs provide the data necessary to control the path taken by each pixel through the RAMDAC. The WID LUT is addressed by the X[04:00] field of each pixel as described in the previous sections. Its outputs are connected to the port selection mechanism, the transparent overlay mechanism and, ultimately, to the various multiplexors which control color model selection. The following paragraphs define the requirements and describe the shadow mechanism.

The Purpose

The RAMDAC contains a number of LUTs which transform pixel data as they flow through the RAMDAC and, ultimately, to the video monitor. This transformation occurs at some fraction of the pixel frequency but the data which selects the particular transformation resides in the WID LUT and originates in the host system. Thus, the WID LUT constitutes a resource which is shared by two systems operating at unrelated clock rates. Furthermore, the effects of transformation are immediately apparent at the display, so that if the host accesses the WID LUT during the active video interval, the display will be disrupted. One way to prevent this disruption is to limit host access to the vertical blanking interval, which is approximately 500×10^{-6} seconds in duration. The success of this scheme depends on the ability of the host system to start and complete the data transfer within this interval. Unfortunately, this cannot be guaranteed because the response time of the host's operating system is large and variable. Clearly, some mechanism which isolates host accesses from the pixel transformation process is needed. Such a mechanism is provided by the SHADOW WID LUT and associated control circuits.

Description

The SHADOW WID LUT serves as a buffer between host accesses and the active WID LUT which actually selects transforms. The problem of isolating host accesses from the selection process is solved by alternatively assigning ownership of SHADOW WID LUT to the host and the RAMDAC and by checking for completion of transfer operations. This process is briefly described as follows. The host writes data into the SHADOW WID LUT or reads data from it. These actions differ only in the direction of the flow of data between the host and the RAMDAC and between the shadow and active tables. In both cases, the host issues a command to the RAMDAC which causes it to transfer data between the shadow and active tables. The RAMDAC does not execute the command immediately upon reception, it defers execution until the arrival of the next vertical blanking interval, at which time, the transfer of data between the active and shadow tables takes place. The transfer operation is guaranteed to reach completion within some

small fraction of the vertical blanking interval. The host monitors the status of operations by reading the contents of a Command/Status register. Obviously, write and read operations are sequentially different. The host is required to write the shadow table before issuing the write transfer command and it is required to issue the read transfer command and receive acknowledgement before reading the shadow table. The following paragraphs describe the required functions.

4.5.1 The SHADOW WID LUT may be written and read, at any time, via the CPU Port.

4.5.2 A register shall be provided for the purpose of initiating and monitoring such data transfers. The Command/Status Register is written and read via the CPU Port. The register functions are defined as follows.

4.5.2.1 Transfer Direction

read/write

1 = shadow to active

0 = active to shadow

Directs RAMDAC hardware to either transfer the contents of the shadow table to the active table or to transfer the contents of the active table to the shadow table.

4.5.2.2 Transfer Command

read/write

1 = transfer

0 = no action

Directs RAMDAC hardware to execute the transfer operation immediately upon, but not before, the arrival of the vertical blanking interval. This bit is set by software and may be cleared by either of two events. Either the transfer command bit is cleared by RAMDAC hardware when the transfer operation has been completed or it is cleared by software, indicating the command has been withdrawn.

4.5.2.3 Device Status

read only

1 = busy, table unavailable

0 = table available

Indicates whether or not the resource is available. This bit is set by RAMDAC hardware when it detects that software has posted the transfer command. The set bit indicates the resource is busy i.e., either the transfer is pending or it is in progress. The status bit is reset by either of two conditions. The first condition is the completion of a transfer operation. This event causes RAMDAC hardware to reset the status bit to indicate the resource is available. The second condition results because the protocol permits software processes to withdraw transfer commands initiated by other processes or indeed by the same process. RAMDAC hardware resets the device status bit if software withdraws the transfer command and the transfer operation has yet to begin. The status bit is not reset if the transfer command is withdrawn after the transfer operation has actually begun.

4.5.3 The transfer operation, shadow to active or active to shadow, shall be completed in less than ten microseconds.

4.5.4 When asserted, the system reset signal forces the Command/Status register to indicate no transfer pending, table available and transfer mode is shadow to active. The reset signal also forces the transfer mechanism to assume the idle state.

5 Other Look Up Tables

The RAMDAC shall incorporate a number of LUTs which serve to transform pixel data, thus accommodating a variety of color models, performing gamma correction and compensation for video monitor characteristics. Selection of a particular color model is done for each pixel under the control of the window attribute input for that pixel. A forthcoming version of this document will specify the various WID LUT to

color model mappings. For the moment, refer to the RAMDAC block diagram appearing on page 2. This diagram illustrates the principle of color model selection, it does not dictate a particular implementation.

Extant RAMDAC designs incorporate an "anti-sparkle" feature which minimizes the disruption of the video display when LUTs are altered during the active portion of a video frame. This RAMDAC shall apply "anti-sparkle" protection to the Color LUT. The WID LUT, the Cursor Bit Map and the Cursor Color Map require different mechanisms. Refer to following paragraphs and to the paragraphs describing Cursor Logic for descriptions of these alternate mechanisms. The Color Correction and Gamma LUTs do not require any "anti-sparkle" circuitry.

5.1 The Color LUT

The RAMDAC shall incorporate three 256 x 8 RAM based look up tables. These tables shall be accessed in two ways. First, table values may be written or read via the CPU Port. Second, pixels entering the Pixel port are routed to these tables by the color model selection mechanism. These accesses are not mutually exclusive. Consequently, "anti sparkle" circuitry shall be incorporated to minimize disruption of the video display when both CPU and Pixel Ports are accessing the same location within the table.

The Color LUT accommodates the Pseudo and Direct Color models.

5.2 The GAMMA LUT

The RAMDAC shall incorporate three 256 x 8 ROM based look up tables. These tables shall be accessed by data routed to them from the pixel port under the control of the color model selection mechanism. The data contained in the GAMMA LUT transform non-linear color values into linear color values. These values shall be provided in a forthcoming version of this document.

The GAMMA LUT accommodates the Linear True Color and Linear Grey Scale models.

5.3 The LUT Bypass Path

Although no look up table is required for this function, the LUT Bypass Path is described in this section because it implements the Non-linear True Color and Non-linear Grey Scale models. When selected, this path connects pixel data entering the Pixel Port to the address inputs of the Calibration LUT or the digital to analog converters depending on the RAMDAC version under discussion. It is important to note that the timing of pixel data taking this path must be the same as for any other path for a given RAMDAC version.

5.4 The Calibration LUT

Version C of the RAMDAC shall incorporate three 256 x 8 RAM based look up tables. These tables shall be accessed in two ways. First, table values may be written and read via the CPU Port. Second, these tables are not subject to the actions of the color model selection mechanism. Pixels of all color models serve as addresses to these tables. These tables provide a means for achieving consistent video display characteristics over a range of video monitor types.

6 PLL Pixel Clock Synthesizer

A programmable, multiplying, phase locked loop synthesizer, operating from an external frequency reference, shall provide the pixel clock. The frequency range, the reference frequency (or frequencies), phase noise figures, as well as programming details will be provided in a forthcoming revision of this document. Note that programming of the PLL shall be done via the CPU Port. Here are some general features required of the design.

6.1 Two options exist for generating the reference frequency. The design should accommodate either case by means of connection or register programming.

6.1.1 An external crystal reference is connected to a RAMDAC resident oscillator which produces the reference signal.

6.1.1.1 An external, crystal controlled oscillator provides the reference in the form of a differential PECL (operating from $V_{cc} = ?$) signal.

6.1.2 A means for bypassing the internal clock generator and internal reference oscillator is required. When bypass is active, an externally generated PECL differential clock is applied to the RAMDAC which uses it as pixel clock.

7 Pixel Clock Divider

A programmable divider, producing either the pixel frequency f_p , $f_p/2$ or $f_p/4$ for internal and external timing, is required. Externally the output of the divider, SC, may be used as the video memory serial port clock and, when connected to the LD input, as the Pixel Port clock. A forthcoming revision of this document will specify driver characteristics. Note that programming of the divider shall be done via the CPU Port.

8 Timing Generator

A programmable generator, operating from the pixel clock, or a fraction thereof, and providing video display and video memory timing reference signals, is required. The number of registers or the size of memory required to store timing generator parameters depends on the design. These elements will be specified once more is known about the implementation.

The timing generator produces signals which are described below. In all cases the points of assertion and negation shall be programmable. The timing generator shall operate in two dimensions, these are horizontal, corresponding to the horizontal scan of a video display format, and vertical, corresponding to the vertical scan of a video display format. Events in the horizontal dimension, such as horizontal synch, shall be placed on boundaries defined by $1/f_p$, $2/f_p$ or $4/f_p$ periods. Events in the vertical dimension, such as vertical synch, shall be placed on boundaries defined by a whole number of horizontal scans in the case of sequential displays, or by half horizontal scans in the case of interlaced displays. The number of $1/f_p$, $2/f_p$ or $4/f_p$ periods comprising a horizontal scan and the number of horizontal scans comprising a video frame is variable. The design of the generator shall be such that when generator outputs are observed at the outputs of the RAMDAC, they are correctly placed relative to the video data. For example, if the generator is programmed to assert some event congruent to pixel number p , then it will be seen, at the RAMDAC output, to occur at a position corresponding to pixel number p . The timing generator shall incorporate a rudimentary frame synchronization feature. This feature is described in the following paragraphs. Finally, the synch generator shall incorporate a control register, possibly operating in conjunction with system reset, which permits software, via the CPU port, to individually enable or disable horizontal and vertical synch. The following paragraphs describe additional synch generator requirements.

8.1 Frame Synchronization

The synch generator shall provide a means for video frame synchronization of several displays. The synch generator shall be capable of operating either as a master or a slave as specified by a control register bit. The controlling function will be fully defined in a forthcoming revision to this document.

8.1.1 When the synch generator is operated as a master, the signal FLD is an output producing the periodic functions described in paragraph 8.2.4.

8.1.2 When the synch generator is operated as a slave the signal FLD is an input which serves to set the horizontal and vertical elements of the generator so that they correspond to the start of vertical synch. Synch generators operated in the slave mode shall not drive the FLD signal.

8.1.3 The phase relationship of vertical synch emitted by the master generator and vertical synch emitted by any slave generator will be defined in a forthcoming revision to this document. For the moment, assume that the vertical synch emitted by any slave generator shall lag the vertical synch emitted by the master gen-

erator by less than seven microseconds (that number being approximately one half of the horizontal scan period of the 1280 x 1024 @ 76 Hz video format).

8.2 System Reset

When asserted, the system reset signal, RS, causes three effects. These are described as follows.

8.2.1 A registered function named Video Enable shall be forced to the video disabled state by the assertion of system reset. This condition shall persist after the negation of system reset and until it is overwritten via the CPU Port. Video Enable, when in the disabled state, asserts composite blanking within the RAMDAC so that the video monitor remains black while the timing generator is being programmed.

8.2.2 When asserted, system reset forces the timing generator to assume a known state. Should the implementation require it, a registered function named Synch Gen Enable shall be incorporated. Synch Gen Enable shall be forced to the synch gen disabled state by the assertion of system reset. This condition shall persist after the negation of system reset and until it is overwritten via the CPU Port. The purpose is to hold the timing generator in a known state while it is being programmed.

8.2.3 When asserted, system reset causes the synch generator to be in the slave mode. The FLD signal shall be placed in the high impedance mode when the synch generator is in the slave mode.

8.2.4 Timing Generator Signals

8.2.5 CS

Composite Synch. This signal is conveyed to the video monitor as a discreet signal or, optionally, it is added to the Green channel DAC in the usual manner. The "synch on green" and other composite synch options shall be controlled by register programming. Composite Synch is the video synchronization signal and comprises horizontal and vertical, optionally serrated, synch components. For interlaced formats, such as NTSC and PAL, these components include horizontal equalization pulses. HDTV requires multilevel synch pulses. In this case, the timing generator produces signals which are converted to the required analog levels by the RAMDAC. A forthcoming version of this document will specify output levels for HDTV and other cases as well as driver characteristics.

8.2.6 CB

Composite Blanking. Internal use only. This is the video blanking signal comprising vertical and horizontal elements. When asserted, composite blanking causes the RAMDAC analog outputs, R, G, and B, to assume the voltage level which corresponds to black.

8.2.7 VB

A Pixel Port output. VB is the vertical blanking signal. External circuitry interprets this signal as "top of frame" and uses it to index the video memory.

8.2.8 HB

A Pixel Port output. HB is the horizontal blanking signal. External circuitry interprets this signal as "start of horizontal scan" and uses it to index the video memory.

8.2.9 FLD

A pixel port bidirectional signal. FLD is an input when the synch generator is operated in the slave mode; it is an output when the generator is operated in the master mode. Refer to paragraph 7.1.

When the generator is operated in the master mode, FLD is used in stereo and interlaced display formats to indicate which field is to be displayed. In interlaced formats, FLD differentiates odd and even fields. In the stereo format, FLD differentiates left and right views. In sequential display formats, FLD simply indicates the start of the frame. External circuitry uses FLD to index the video memory and, in the stereo format, to control LCD shutters used in stereographic viewing. FLD shall change state congruent to vertical synch plus an offset. A forthcoming revision of this document will fully describe the characteristics of the FLD signal.

A Note to SUN Engineers

The FLD signal will be an LVTTTL signal referenced to $V_{cc} = 3.3$ volts. At present, LCD shutters used with stereographic displays specify standard TTL levels referenced to $V_{cc} = 5$ volts. Consequently, it may be necessary to supply an external (to the RAMDAC) level translator. This matter is under investigation.

8.2.10. SCen

A Pixel Port output. Serial Clockenable is used by external circuitry to regulate the extraction of data from the serial port of the video memory. Assertion and negation of this signal are controlled by two factors, these are the horizontal scan line number and the horizontal pixel number. The latter factor becomes a pixel group number in cases where the timing generator is operated at some fraction of the pixel frequency.

8.3 The following is a partial list of display formats which the timing generator shall support. A forthcoming version of this document will specify the entire complement of formats.

8.3.1 The following apply to Version B and are a part of the premium feature set.

8.3.1.1 HDTV [applicable only to Version B]

8.3.1.2 1600 x 1280 @ 76 Hz, $f_p = 216$ MHz [applicable only to Version B].

8.3.1.3 1280 x 1024 @ 76 Hz, $f_p = 135$ MHz, double buffered [applicable only to Version B].

8.3.2 The following apply to all versions and are a part of the standard feature set.

8.3.2.1 1280 x 1024 @ 76 Hz, $f_p = 135$ MHz, single buffered

8.3.2.2 1152 x 900 @ 71 Hz, $f_p = 108$ MHz

8.3.2.3 960 x 680 @ 108 Hz, $f_p = 100$ MHz, stereographic display

8.3.2.4 1024 x 768 @ 60 Hz, $f_p = 64$ MHz

8.3.2.5 768 x 576 @ 50 Hz, PAL

8.3.2.6 640 x 480 @ 60 Hz, NTSC

8.3.3 A note on the interlaced formats NTSC, PAL and HDTV. The timing generator shall comply with the timing specifications for these formats with one exception. Each of these formats assert and negate vertical blanking on half line boundaries so that each field consists of $n + 1/2$ horizontal scans. The generator shall incorporate circuitry which alters the timing of blanking for each field so that $n + 1$ horizontal scans are displayed in each field. This feature and HDTV requirements will be fully described in a forthcoming version of this document.

9 Cursor

9.1 All versions of the RAMDAC shall incorporate cursor control logic, a 32 x 32 x 2 cursor bit map and a 3 x 8 x 3 cursor color LUT. These elements are described below. Cursor circuitry shall support all of the display formats listed in paragraph 7.3. However, there is no requirement to support a stereo cursor. In this mode, the cursor is displayed in the manner of any sequential scan display format e.g., 1280 x 1024. Finally, it is important to note that all cursor state requires a separate address pointer as described in the CPU Port Paragraph. Address mapping shall be provided in a forthcoming version of this document.

9.1.1 A 3 x 8 x 3 color look up table shall be provide for cursor use. The table shall be accessed in two ways. First, color values are written and read via the CPU Port. Second, cursor pixels, originating in the cursor bit map, address the table which introduces the associated 24 bit color value into the video data stream. The depth of the cursor bit map allows four combinations of color and control. An example mapping is shown below, however more than one mapping may be required for the transparent color. In that case, the RAMDAC shall provide a register bit to select the mapping of the transparent color. Transparent color

mapping selection is not required to operate at the pixel rate. This selection is done as part of a configuration procedure which is performed at boot time. A forthcoming version of this document will specify these mappings.

11 selects cursor color three.

10 selects cursor color two.

01 selects cursor color one

00 selects transparent color.

This defines the display precedence. For all but the transparent color of the cursor pixel, the RAMDAC shall replace the 24 bit RGB color datum with a 24 bit cursor color datum. When the cursor color is transparent, the 24 bit RGB color datum shall pass unimpeded. It is important to note that this selection occurs for each pixel.

9.1.2 Cursor Control Logic

A cursor coordinate address is supplied to a cursor control logic register via the CPU Port. This address shall be partitioned into sufficiently large X and Y fields to accommodate the display format having the greatest number of pixels and horizontal scans. The cursor control logic continuously compares the cursor coordinate address to the X and Y addresses produced in the Timing Generator. When equality is detected, the control logic extracts the contents of the cursor bit map for processing by the cursor color table. Each cursor bit map address produces a 32 x 2 quantity which describes one horizontal scan of the cursor image. Note that this extraction implies a serialization of the bit map output in the x dimension, occurring in step with video data, while bit map word addresses change in step with vertical scan lines. It is important to note that the cursor image shall invariably be written into the bit map without special accommodations for interlaced formats. Cursor logic shall correctly interface the output of the bit map in the Y dimension when operating in the interlaced display mode. The mode information is available in a Timing Generator control register. Finally, a means, other than system reset, for resetting the cursor control logic and disabling the display of the cursor image shall be provided in the form of a one bit extension to the cursor coordinate address.

9.1.3 System Reset

Upon assertion, the system reset signal, RS, shall force the cursor control logic into a known state.

9.1.4 Cursor Bit Map

This 32 x 32 x 2 memory contains the cursor image. It is accessed in two ways. First, the cursor image shall be written or read via the CPU Port. Second, the cursor control logic addresses the bit map in the manner described in 9.1.2.

9.1.5 Cursor Color LUT

Refer to paragraph 9.1.1.

10 Hardware Aids to System Level Diagnostics

10.1 The RAMDAC design shall incorporate a mechanism for capturing digital pixel data at a point immediately prior to analog conversion. The number of pixels to be captured and the means for identifying the pixels to be captured have not yet been determined. A forthcoming revision of this document shall specify this function.

11 Monitor Identification and Control Interface

11.1 The RAMDAC shall incorporate a four bit register for use as the hardware portion of a software driven, synchronous, half duplex serial interface. This interface provides the means for reading video monitor identification codes and for controlling certain monitor functions from the host system. The host system

shall communicate with the interface register via the CPU port. Register inputs and outputs shall also be connected to four RAMDAC pins which form part of the physical interface to the monitor. The functions of these pins are described below. It is important to note that no interface control circuitry is required. All of the interface controls, the interface clock and the transmit data are formed by writing the appropriate three bit field to the register.

11.1.1 CLK
Output. This is the serial interface clock.

11.1.2 TxD
Output. This is the transmit data bit.

11.1.3 RxD
Input.
This is the receive data bit.

11.1.4 DSBL
Output. Interface enable.

12 JTAG

12.1 As a minimum requirement, the RAMDAC design shall incorporate circuitry to implement boundary scan testing as defined in IEEE Standard Test Access Port and Boundary-Scan Architecture (IEEE Std 1149.1-1990). The Tap Access Port (TAP) shall include the TRST* (test reset) signal. Please refer to sections 3.1.1, 3.2, 3.3, 3.4, 3.5, 3.6 and 5.3 of the cited document for a complete definition of the Test Access Port.

12.2 Although not required by SUN, the use of internal scan to facilitate testing of circuits such as the synch generator is strongly recommended.

12.3 The RAMDAC shall incorporate a Device Identification Register which can be accessed via the Tap Access Port as defined in sections 7.10 and 7.11 of IEEE Standard Test Access Port and Boundary-Scan Architecture (IEEE Std 1149.1-1990). Furthermore, the contents of the Device Identification Register shall be accessible via the CPU Port. The register shall be a read only device and shall uniquely identify the device in accordance with requirements and principles set forth by the JEDEC organization and in the SUN Microsystems document SUN Standard JTAG Encodings (rev 2/20/90).

13 Address Mapping

13.1 Address Map – Control Field Definition

Function	Access Type	Control Field	
		Control bit 1	Control Bit 0
Cursor Functions	Indirect	1	1
Cursor Address Pointer	Direct	1	0
Configuration Functions	Indirect	0	1
Configuration Address Pointer	Direct	0	0

13.2 Address Map – Cursor Functions

Function	Size ¹	C[01:00]	Pointer Address
Undefined		0x03	0xFF
			0x86
Cursor Coordinate – Active Register	01 x 32	0x03	0x85
Cursor Coordinate – Shadow Register	01 x 32	0x03	0x84
Cursor Control and Status Register	01 x 08	0x03	0x83
Cursor Color Lut	03 x 08 x 03	0x03	0x82
			0x80
Reserved			0x7F
			0x40
Cursor Bit Map – Plane 1	32 x 32	0x03	0x3F
			0x20
Cursor Bit Map – Plane 0	32 x 32	0x03	0x1F
			0x00
Cursor Address Pointer	01 x 08	0x02	–
Note 1: The size of an element is expressed as follows: (the number of words) x (the word size, in bits) x (the number of tables, if applicable)			

13.3 Address Map – Configuration Functions

Function	Size ¹	C[01:00]	Pointer Address
Reserved			0x0153
			0x0152
Video Format Control Register 2	01 x 16	0x01	0x0151
Video Format Control Register 1	01 x 16	0x01	0x0150
Reserved			0x014F
			0x014C
VCO Range Register	tbs	0x01	0x014B
PLL N Register	tbs	0x01	0x014A
PLL M Register	tbs	0x01	0x0149
Clock and Pre-scaler Control Register	01 x 08	0x01	0x0148
Reserved			0x0147
			0x0144
Transparent Overlay Mask Register	01 x 08	0x01	0x0143
Transparent Overlay Color Key Register	01 x 08	0x01	0x0142
Pixel Port Configuration Register	01 x 08	0x01	0x0141
Window Attribute – Active LUT	32 x 06	0x01	0x0140
			0x0121
Window Attribute – Shadow LUT	32 x 06	0x01	0x0120
			0x0100
Color LUT	256 x 08 x 03	0x01	0x00FF
			0x0000
Configuration Address Pointer	01 x 16	0x00	–
Note 1: The size of an element is expressed as follows: (the number of words) x (the word size, in bits) x (the number of tables, if applicable)			

Continuation of Address Map – Configuration Functions

Function	Size ¹	C[01:00]	Pointer Address
Reserved	01 x 12	0x01	0x016F
			0x016A
FLD Offset Register	01 x 12	0x01	0x0169
Equalization Interval Negation Point Reg.	01 x 12	0x01	0x0168
Equalization Interval Assertion Point Reg.	01 x 12	0x01	0x0167
V-blank. Negation Point Register	01 x 12	0x01	0x0166
V-blank. Assertion Point Register	01 x 12	0x01	0x0165
V-synch. Negation Point Register	01 x 12	0x01	0x0164
V-synch. Assertion Point Register	01 x 12	0x01	0x0163
SCen, V Negation Point Register	01 x 12	0x01	0x0162
SCen, V Assertion Point Register	01 x 12	0x01	0x0161
Vertical Counter Modulus Register	01 x 12	0x01	0x0160
Reserved			0x015F
H-serration. Negation Point Register	01 x 12	0x01	0x015E
H-serration. Assertion Point Register	01 x 12	0x01	0x015D
H-equalization. Negation Point Register	01 x 12	0x01	0x015C
H-equalization. Assertion Point Register	01 x 12	0x01	0x015B
H-blank. Negation Point Register	01 x 12	0x01	0x015A
H-blank. Assertion Point Register	01 x 12	0x01	0x0159
H-synch. Negation Point Register	01 x 12	0x01	0x0158
H-synch. Assertion Point Register	01 x 12	0x01	0x0157
SCen, H Negation Point Register	01 x 12	0x01	0x0156
SCen, H Assertion Point Register	01 x 12	0x01	0x0155
Horizontal Counter Modulus Register	01 x 12	0x01	0x0154
Note 1: The size of an element is expressed as follows: (the number of words) x (the word size, in bits) x (the number of tables, if applicable)			

Continuation of Address Map – Configuration Functions

Function	Size ¹	C[01:00]	Pointer Address
Undefined			0x01FF
			0x0173
Device Identification Register	01 x 32	0x01	0x0172
Signature Analysis Register	1bs	0x01	0x0171
Monitor Serial Interface CNTL and DATA	01 x 08	0x01	0x0170
Note 1: The size of an element is expressed as follows: (the number of words) x (the word size, in bits) x (the number of tables, if applicable)			

14 Electrical Specifications

14.1 Operating Conditions – All Versions, except as noted

Signal	Parameter	min	typ	max	symbol	units
	Power supply, 3.3 volt, $\pm 10\%$	3.14		3.47	V_{dd1}	V
	Power supply, 5.0 volt, $\pm 10\%$	4.75		5.25	V_{dd2}	V
	Ambient operating temperature	0		70	T_a	$^{\circ}\text{C}$
	Junction temperature @ 100 LFPM airflow			105	T_{jmax}	$^{\circ}\text{C}$
	Version A power dissipation			≤ 1.4	P_{max}	W
	Version B power dissipation			≤ 2.0	P_{max}	W

14.2 t_r shall be specified in a forthcoming revision to this document.

14.3 t_f shall be specified in a forthcoming revision to this document.

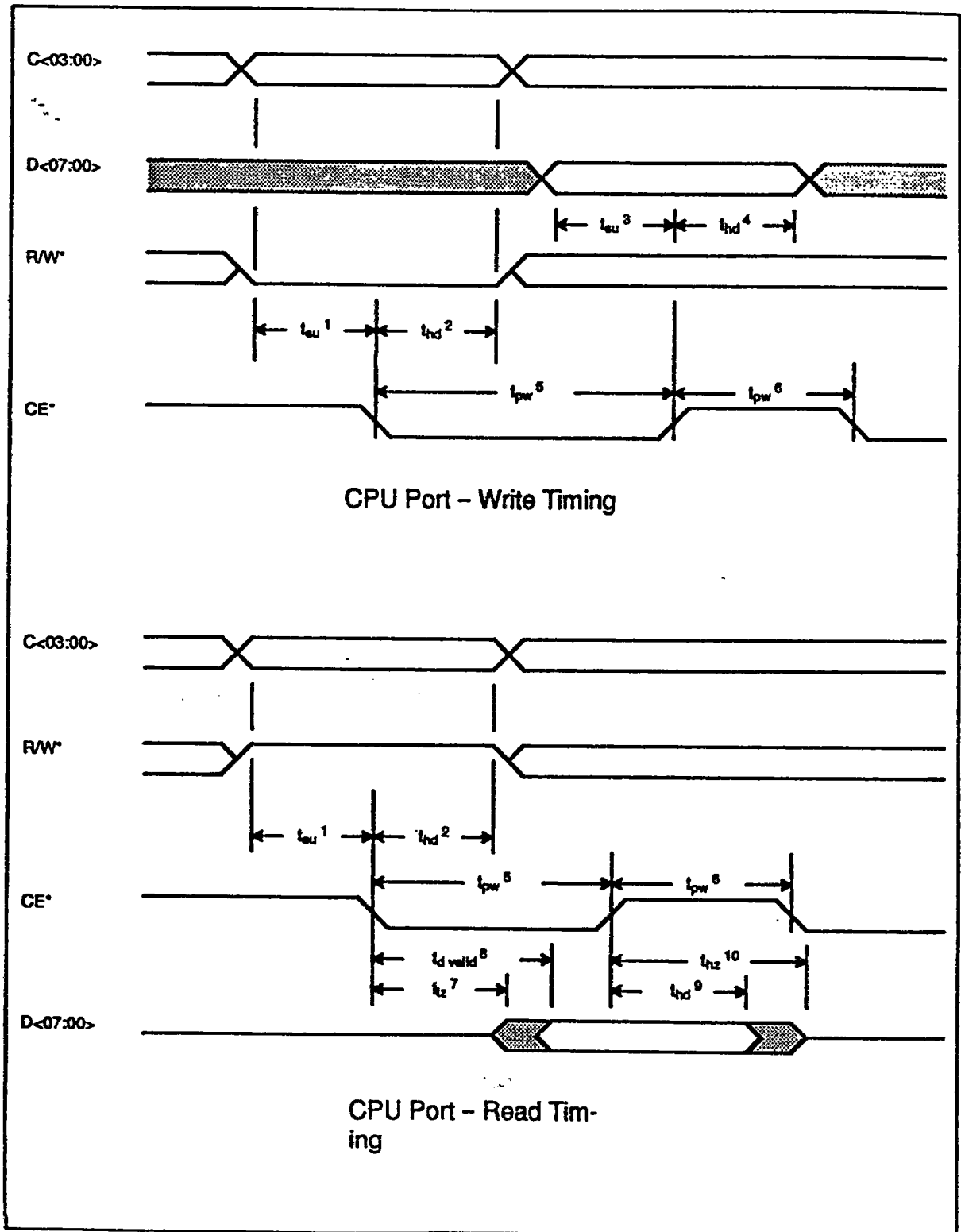
14.4 DC Characteristics – All Versions

Signal	Parameter	min	typ	max	symbol	units
D<07:00>, input, LVCMOS referenced to V_{dd1}	input high voltage	2.0		V_{dd1}	V_{ih}	V
	input low voltage	V_{ss}		0.8	V_{il}	V
	input high current	tbs		tbs	I_{ih}	μA
	input low current	tbs		tbs	I_{il}	μA
	capacitance at pin			7	C_0	pF
D<07:00>, output, LVCMOS referenced to V_{dd1}	output high voltage	2.4		V_{dd1}	V_{oh}	V
	output low voltage	V_{ss}		0.4	V_{ol}	V
	output high current	tbs		tbs	I_{oh}	μA
	output low current	tbs		tbs	I_{ol}	μA
	capacitance at pin			7	C_0	pF
	hi Z leakage current			10	I_{oz}	μA
All other digital inputs except XTAL 1, 2 LVCMOS referenced to V_{dd1}	input high voltage	2.0		V_{dd1}	V_{ih}	V
	input low voltage	V_{ss}		0.8	V_{il}	V
	input high current	tbs		tbs	I_{ih}	μA
	input low current	tbs		tbs	I_{il}	μA
	capacitance at pin			5	C_0	pF
XTAL 1, 2, inputs, PECL referenced to V_{dd2}	input high voltage	4.0		5.5	V_{Kih}	V
	input low voltage	V_{ss}		3.4	V_{Kil}	V
	input high current	tbs		tbs	I_{Kih}	μA
	input low current	tbs		tbs	I_{Kil}	μA
	capacitance at pin			5	C_{K0}	pF
All other digital outputs LVCMOS referenced to V_{dd1}	output high voltage	2.4		V_{dd1}	V_{oh}	V
	output low voltage	V_{ss}		0.4	V_{ol}	V
	output high current	tbs		tbs	I_{oh}	μA
	output low current	tbs		tbs	I_{ol}	μA
	capacitance at pin			5	C_0	pF

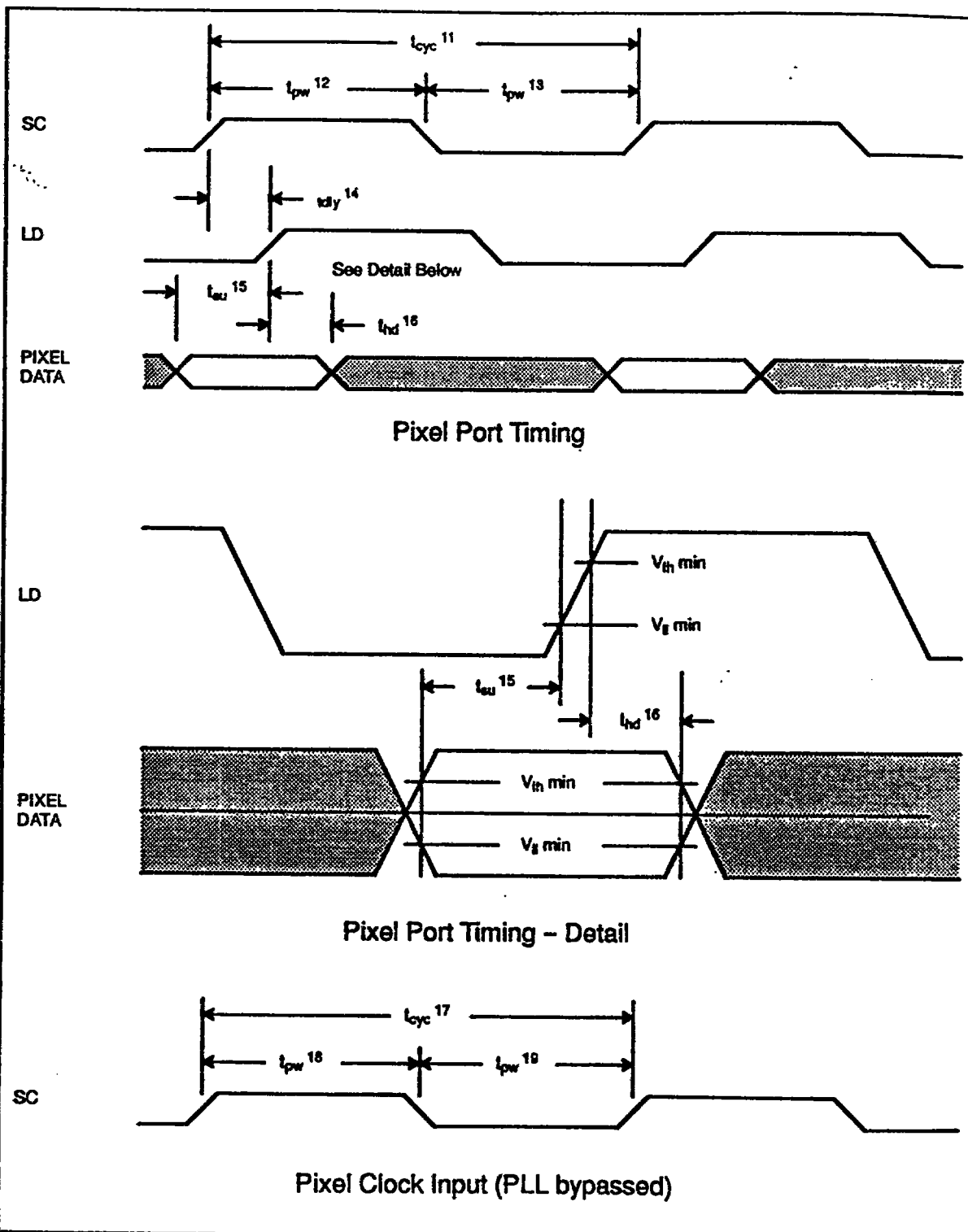
14.5 CPU and Pixel Ports
AC Characteristics – ALL Versions, except as noted

Signal Name	Diagram Reference	Parameter	min	typ	max	load	units
R/W*	1	T_{su} , R/X*, C<03:00> to CE* ↓	0				ns
	2	T_{hd} , CE* ↓ to R/W*, C<03:00>	10				ns
D<07:00>	3	T_{su} , D<07:00> to CE* ↑	20				ns
	4	T_{hd} , CE* ↑ to D<07:00>	15				ns
CE*	5	T_{pw} , low	35				ns
	6	T_{pw} , high	15				ns
D<07:00>	7	T_{lz} , CE* ↓ to low Z state	7				ns
	8	T_d valid, access time from CE* ↓			35		ns
	9	T_{hd} , data hold from CE* ↑	0				ns
	10	T_{hiz} , high Z state from CE* ↑			50		ns
SC	11	T_{cyc} , shift clock period	13.3				ns
	12	T_{pw} , high	6				ns
	13	T_{pw} , low	6				ns
	14	T_{dly} , SC ↑ to LD ↑	tbs		tbs		ns
Pixel Data	15	T_{su} , data valid to LD ↑	2				ns
	16	T_{hd} , data valid after LD ↑	0				ns
Pixel Clock PLL Bypassed	17	T_{cyc} , pixel clock period version A	4.54				ns
	18	T_{pw} , high version A	2				ns
	19	T_{pw} , low version A	2				ns
	17	T_{cyc} , pixel clock period version B	7.41				ns
	18	T_{pw} , high version A	3				ns
	19	T_{pw} , low version A	3				ns

15 Timing
15.1 CPU Port



15.2 Pixel Port



16 Packaging and Thermal Specifications

16.1 The RAMDAC shall be packaged in a surface mount package; TAB or PQFP packages are preferred. RAMDAC power dissipation shall not exceed 2 watts. For thermal calculations assume the following conditions.

16.2 At sea level assume 100 linear feet per minute at a temperature of 60° C.

16.3 At 10,000 feet assume 200 linear feet per minute at a temperature of 60° C.

17 Items And Issues To Be Resolved

17.1 Logic Levels

17.1.1 This document requires all digital inputs (with the exception of the PECL level pixel clock reference input) and outputs to conform to LVCMOS levels. Some designs which utilize this ramdac may not be able to guarantee that signals driving ramdac inputs will be referenced to 3.3 volts. The ability of ramdac inputs to withstand input levels higher than those specified for LVCMOS must be determined. Should the input cells prove incapable of withstanding these higher input levels as a normal operating condition, modifications to ramdac input cell design should be explored.

17.2 CPU Port

17.2.1 Ordering of data in the byte and bit dimensions for CPU Port write and read operations requires specification.

17.2.2 This document specifies the (defacto) standard CPU interface. The performance of this interface and some aspects of its' protocol do not fit our application well. A modification to the standard interface should be explored.

17.3 Pixel Port

17.3.1 Ordering of data for the various interleaving formats requires specification.

17.3.2 Ordering of bits within a pixel group requires specification.

17.3.3 There is a possibility that the RAMDAC will be required to support an 8 bit, packed pixel mode. Should this requirement be made, the interleaving format and controls will require specification.

17.4 Cursor

17.4.1 Ordering of cursor bit map data, relative to the CPU Port as well as display, requires specification.

17.4.2 The index point of the cursor bit map relative to display requires specification.

17.5 Gamma LUT

17.5.1 Values for the ROM based Gamma look up table require specification.

17.6 Video Timing Generator

17.6.1 A complete list and specification of all video formats is required.

17.7 Timing Specifications

17.7.1 Electrical loads and other test conditions require specification.

17.8 Register Definitions

17.8.1 Definition of all state and control registers is required.

17.9 Analog Section

17.9.1 The electrical characteristics of the analog section require specification.

17.10 PLL

17.10.1 The programming variables require definition.

17.10.2 A method of characterizing phase noise is required. Phase noise limits require specification.

17.11 Thermal Performance

17.11.1 The airflow and ambient temperature figures given in this document are subject to revision as new information becomes available.

Color Model Selection, Transparent Overlays and the Window Attribute LUT

The diagrams describing the various interleaving formats include a data field named X. The contents of this field are interpreted as either a Window Identification (WID) index or as an Overlay Color. The case where the X field is interpreted as a WID is discussed first. The Overlay Color case and the means of selecting the particular interpretation of the X data field is discussed in following paragraphs.

Window ID's (WIDs) are index addresses into the WID look up table, serve to select the pixel source, e.g. port A or B, and to associate the pixel with a particular color model. The X field is a component of every pixel and its content may differ in contiguous pixels. Therefore, port and color model selection shall be performed for each individual pixel.

The described interleaving formats are divided into two broad categories. These are the Single Buffered interleaving format, and the Double Buffered interleaving format.

In 2:1 and $4/2$:1 input formats the X field from port A shall be used. The X field from port B is ignored.

In 4:1 and $8/2$:1 input formats the X field from each pixel is used. Bit CMC[05] is don't care for 4:1 and $8/2$:1 input formats.

The X field shall not directly control port and color model selection. The contents of the lower five bits of the X field, X[04:00], constitute the address to the active WID LUT; the data, hereafter called WID[05:00], contained in the locations

corresponding to these addresses, shall be used to effect the port and control color model selection according to definitions shown in Table 5, "Color Model Table Data Entry Codes," on page 15.

Table 5. Color Model Table Data Entry Codes

Selected Input Port and Color Model	Color Model Control					
	5*	4	3	2	1	0
Input Port B - 24-Bit Non-Linear True Color	1	1	1	0	x	x
Input Port B - 24-Bit Linear True Color	1	1	0	1	x	x
Input Port B - 24-Bit Direct Color	1	1	0	0	x	x
Input Port B - 8-Bit Non-Linear Grey Scale from B Channel	1	0	1	0	1	1
Input Port B - 8-Bit Non-Linear Grey Scale from G Channel	1	0	1	0	1	0
Input Port B - 8-Bit Non-Linear Grey Scale from R Channel	1	0	1	0	0	1
Input Port B - 8-Bit Non-Linear Grey Scale from X Channel ($\frac{8}{2}$:1 or 4:1 only)	1	0	1	0	0	0
Input Port B - 8-Bit Linear Grey Scale from B Channel	1	0	0	1	1	1
Input Port B - 8-Bit Linear Grey Scale from G Channel	1	0	0	1	1	0
Input Port B - 8-Bit Linear Grey Scale from R Channel	1	0	0	1	0	1
Input Port B - 8-Bit Linear Grey Scale from X Channel ($\frac{8}{2}$:1 or 4:1 only)	1	0	0	1	0	0
Input Port B - 8-Bit Pseudo Color from B Channel	1	0	0	0	1	1
Input Port B - 8-Bit Pseudo Color from G Channel	1	0	0	0	1	0
Input Port B - 8-Bit Pseudo Color from R Channel	1	0	0	0	0	1
Input Port B - 8-Bit Pseudo Color from X Channel ($\frac{8}{2}$:1 or 4:1 only)	1	0	0	0	0	0
Input Port A - 24-Bit Non-Linear True Color	0	1	1	0	x	x
Input Port A - 24-Bit Linear True Color	0	1	0	1	x	x
Input Port A - 24-Bit Direct Color	0	1	0	0	x	x
Input Port A - 8-Bit Non-Linear Grey Scale from B Channel	0	0	1	0	1	1
Input Port A - 8-Bit Non-Linear Grey Scale from G Channel	0	0	1	0	1	0
Input Port A - 8-Bit Non-Linear Grey Scale from R Channel	0	0	1	0	0	1
Input Port A - 8-Bit Non-Linear Grey Scale from X Channel	0	0	1	0	0	0
Input Port A - 8-Bit Linear Grey Scale from B Channel	0	0	0	1	1	1
Input Port A - 8-Bit Linear Grey Scale from G Channel	0	0	0	1	1	0
Input Port A - 8-Bit Linear Grey Scale from R Channel	0	0	0	1	0	1
Input Port A - 8-Bit Linear Grey Scale from X Channel	0	0	0	1	0	0
Input Port A - 8-Bit Pseudo Color from B Channel	0	0	0	0	1	1
Input Port A - 8-Bit Pseudo Color from G Channel	0	0	0	0	1	0
Input Port A - 8-Bit Pseudo Color from R Channel	0	0	0	0	0	1
Input Port A - 8-Bit Pseudo Color from X Channel	0	0	0	0	0	0

Figure 8, on page 16, describes the color model selection mechanism.

*Ignored in $\frac{8}{2}$:1 or 4:1 input formats.

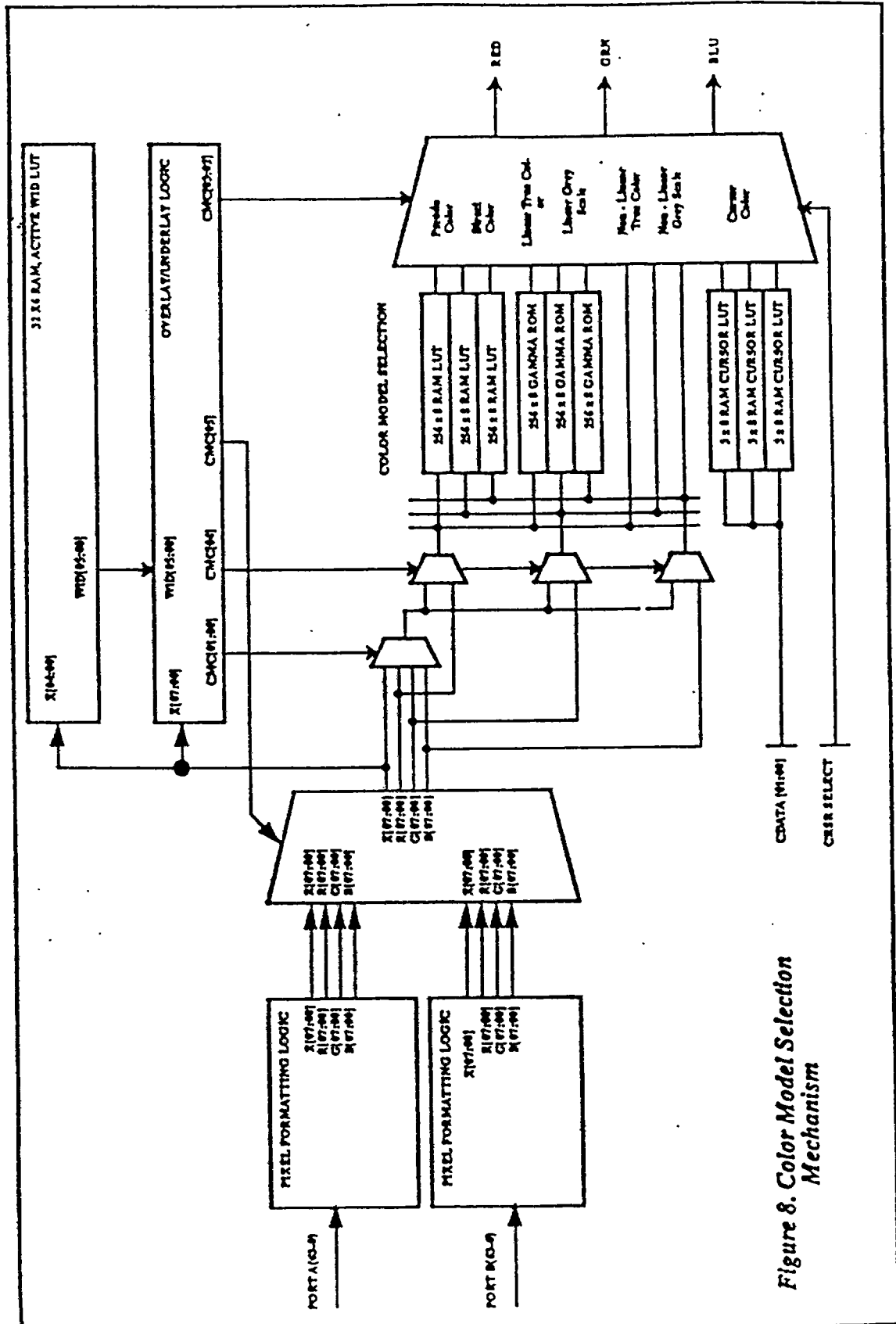


Figure 8. Color Model Selection Mechanism

Overlay/Underlay Operation

As noted in a previous paragraph, the data in the X field may be interpreted as either a Window Identification (WID) index or as an Overlay color. The purpose of this is to accommodate transparent overlays without incurring the expense of additional planes in the frame buffer memory and additional look up tables in the RAMDAC. This is accomplished by using the 256 x 8 x 3 RAM LUT to contain overlay colors and by circuitry which regulates the selection of overlay or underlay data for display.

Given that transparent overlay is enabled, this function may be explained as follows: If the X value of the current pixel equals the transparent overlay color key register in the bits specified by the mask register, then the current pixel will be processed according to the window data received from location X of the active window ID LUT. Otherwise, the X value will be used as an 8-bit pseudocolor pixel.

This function appears in the color model selection diagram as the Overlay/Underlay Logic block of Figure 8, on page 16. The details are diagramed in Figure 9.

The principle Overlay/Underlay operation is described in Table 6, "Overlay/Underlay Operation," on page 17 as follows.

Table 6. Overlay/Underlay Operation

Transparent Overlay Enable Bit	Overlay Mask Bit [n]	X[07:00]	Overlay Color Key Bits [n]	CMC[05:00]
0	x	x	x	WID[05:00] (underlay)
1	0	x	x	Determined by bits where Mask[n] = 1 ^a
1	1	0	0	WID[05:00] (underlay)
1	1	0	1	000000 (overlay)
1	1	1	0	000000 (overlay)
1	1	1	1	WID[05:00] (underlay)

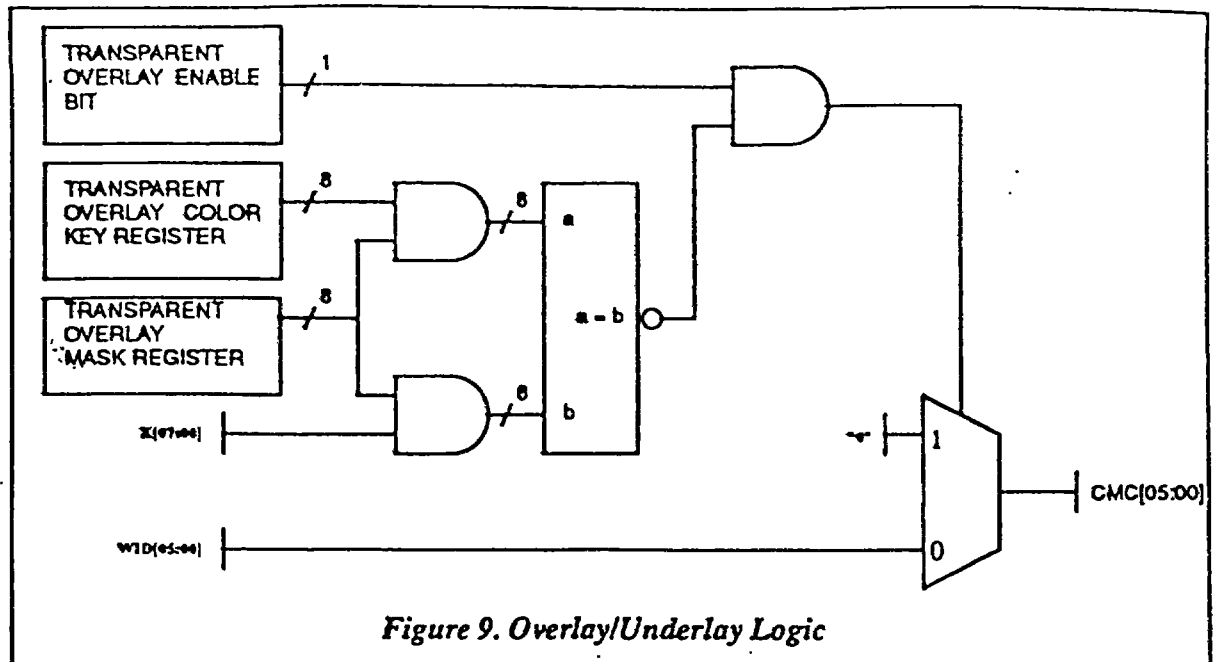
a. If any of the Overlay Mask Register bits [07:00] are set to zero, then the result of the compare operation on the corresponding bits for X[07:00] and the Overlay Color Key [07:00] will be ignored. For the special case where all the bits in Overlay Mask[07:00] are set to zero, the WID[05:00] will be used for color model control (i.e. CMC(5:0)).

The contents of two registers, the mask register and the color key register, define a transparent overlay color key value. The color key value is compared to the value (modified by the overlay mask register) contained in X(7-0). If the two values are equal, the overlay is said to be transparent and the underlay (i.e. normal WID processed) color is displayed, i.e., the multiplexer passes WID(5-0) to CMC(5-0); therefore, the color model and the color source are chosen by the contents of the Window Identification Index. If the two values are not equal, the overlay color is displayed, i.e., the multiplexer passes the value 0x00 to CMC(5-0); causing the color model processing to interpret the X field as an 8-bit pseudocolor pixel.

This implementation defines the constant as 0x00, which causes the selection of the 8 bit pseudo color model, with color data originating in the X field inputs (refer to Table 5, "Color Model Table Data Entry Codes," on page 15). Consequently, the data contained in the X field serve to address the color look up table and thereby select the overlay color.

Transparent Overlay Enable Control

The Transparent Overlay Enable Bit is contained within the Video Format Control Register (See Table 3, "Pixel Format Control Register," on page 5). When the transparent overlay control is disabled, (i.e., set to a logical zero), the window ID data (WID(5-0)) will be used for color model control (i.e. CMC(5-0)). When the transparent overlay control is enabled, (i.e., set to a logical 1), the action of the multiplexer is controlled by the result of the equality comparison.



Transparent Overlay Mask Register

This eight bit quantity operates on the contents of the overlay color key register and the X data to define the extent (in the bit dimension) of the color key value. Overlay Mask register bits set to logical one allow the corresponding bits of the color key register and the X field to participate in the color key comparison. Overlay mask bits set to logical zero preclude the corresponding bits of the color key register and the X field from comparison. Register initialization is not required.

Transparent Overlay Color Key Register

This eight bit quantity defines the color key value, subject to the contents of the mask register. Register initialization is not required.

The WID LUT and the SHADOW WID LUT

Pacifica incorporates two 32 x 6 LUTs which form part of the color model selection circuitry. These are the WID and SHADOW WID LUTs. Operating in conjunction with a control register and a vertical blanking synchronizing mechanism, these LUTs provide the data necessary to control the path taken by each pixel through the RAMDAC. The WID LUT is addressed by the X[04:00] field of each pixel as described in the previous sections. Its outputs are connected to the port selection mechanism, the transparent overlay mechanism and, ultimately, to the various multiplexors which control color model selection. The following paragraphs define the requirements and describe the shadow mechanism.

The Purpose

The RAMDAC contains a number of LUTs which transform pixel data as they flow through the RAMDAC and, ultimately, to the video monitor. This transformation occurs at some fraction of the pixel frequency but the data which selects the particular transformation resides in the WID LUT and originates in the host system. Thus, the WID LUT constitutes a resource which is shared by two systems operating at unrelated clock rates. Furthermore, the effects of transformation are immediately apparent at the display, so that if the host accesses the WID LUT during the active video interval, the display will be disrupted. One way to prevent this disruption is to limit host access to the vertical blanking interval, which is approximately 500×10^{-6} seconds in duration. The success of this scheme depends on the ability of the host system to start and complete the data transfer within this interval. Unfortunately, this cannot be guaranteed because the response time of the host's operating system is large and variable. Clearly, some mechanism which isolates host accesses from the pixel transformation process is needed. Such a mechanism is provided by the SHADOW WID LUT and associated control circuits.

Description

The SHADOW WID LUT serves as a buffer between host accesses and the active WID LUT which actually selects transforms. The problem of isolating host accesses from the selection process is solved by alternatively assigning ownership of SHADOW WID LUT to the host and the RAMDAC and by checking for completion of transfer operations. This process is briefly described as follows. The host writes data into the SHADOW WID LUT or reads data from it. These actions differ only in the direction of the flow of data between the host and the RAMDAC and between the shadow and active tables. In both cases, the host issues a command to the RAMDAC which causes it to transfer data between the shadow and active tables. The RAMDAC does not execute the command immediately upon reception, it defers execution until the arrival of the next VSYNC[®] signal, at which time, the transfer of data between the shadow and active table will take place. The transfer operation is guaranteed to reach completion within some small fraction of the vertical blanking interval. The host monitors the status of operations by reading the contents of a Command/Status register. Obviously, write and read operations are sequentially different. The host is required to write the shadow table before issuing the write transfer command and it is required to issue the read transfer command and receive acknowledgement before reading the shadow table. The following paragraphs describe the required functions.

The SHADOW WID LUT may be written and read, at any time, via the CPU Port. A register shall be provided for the purpose of initiating and monitoring such data transfers. The Command/Status Register is written and read via the CPU Port. The register bits are defined in Table 7, "Window Transfer Control Register," on page 19.

Table 7. Window Transfer Control Register

Bit(s)	Field	R/W	Description
7-3	Reserved	0	Reserved for future expansion
2	Transfer Event 1 = next frame 0 = next field	R/W	If the Transfer Command bit is set, then this bit controls the actual occurrence of the transfer. If this bit is set to 0, then the transfer will occur on any transition of the field signal (0 -> 1 or 1 -> 0). If this bit is set to 1, the transfer occurs on the falling edge of field signal (1 -> 0).
1	Transfer Command 1 = transfer 0 = no action	R/W	Directs RAMDAC hardware to execute the transfer operation immediately upon, but not before, the arrival of the transfer event interval. This bit is set by software and may be cleared by either of two events. Either the transfer command bit is cleared by RAMDAC hardware when the transfer operation has been completed or it is cleared by software, indicating the command has been withdrawn.
0	Device Status 1 = busy, table unavailable 0 = table available	R/O	Indicates whether or not the resource is available. This bit is set by RAMDAC hardware when it detects that software has posted the transfer command. The set bit indicates the resource is busy i.e., either the transfer is pending or it is in progress. The status bit is reset by either of two conditions. The first condition is the completion of a transfer operation. This event causes RAMDAC hardware to reset the status bit to indicate the resource is available. The second condition results because the protocol permits software processes to withdraw transfer commands initiated by other processes or indeed by the same process. RAMDAC hardware resets the device status bit if software withdraws the transfer command and the transfer operation has yet to begin. The status bit is not reset if the transfer command is withdrawn after the transfer operation has actually begun.

The transfer operation, shadow to active, shall be completed in less than one LD cycle.

When asserted, the system reset signal forces the Command/Status register to indicate no transfer pending, table available and transfer mode is shadow to active. The reset signal also forces the transfer mechanism to assume the idle state.

The Color LUT

The RAMDAC shall incorporate three 256 x 8 RAM based look up tables. These tables shall be accessed in two ways. First, table values may be written or read via the CPU Port. Second, pixels entering the Pixel port are routed to these tables by the color model selection mechanism. These accesses are not mutually exclusive. Consequently, "anti sparkle" circuitry shall be incorporated to minimize disruption of the video display when both CPU and Pixel Ports are accessing the same location within the table.

The Color LUT accommodates the Pseudo and Direct Color models.

The GAMMA LUT

The RAMDAC shall incorporate three 256 x 8 ROM based look up tables. These tables shall be accessed by data routed to them from the pixel port under the control of the color model selection mechanism. The data contained in the GAMMA LUT transform non-linear color values into linear color values. The data in the gamma ROM LUT will not be accessible through the CPU port. The alternate method to read the ROM values would be using the SAR in Data_Strobe mode. The gamma ROM values are shown in Table 8, "Gamma ROM Values," on page 20.

Table 8. Gamma ROM Values

Address	Data
Hex	Hex
0	0
1	15
2	1d
3	22
4	27
5	2b
6	2f
7	32
8	36
9	39
a	3b
b	3e
c	40
d	43
e	45
f	47
10	49
11	4b
12	4d
13	4f
14	51
15	53
16	55
17	56
18	58

Table 8. Gamma ROM Values (Continued)

Address	Data
Hex	Hex
19	5a
1a	5b
1b	5d
1c	5e
1d	60
1e	61
1f	63
20	64
21	66
22	67
23	68
24	6a
25	6b
26	6c
27	6d
28	6f
29	70
2a	71
2b	72
2c	74
2d	75
2e	76
2f	77
30	78
31	79
32	7a
33	7c
34	7d
35	7e
36	7f
37	80
38	81
39	82
3a	83
3b	84
3c	85
3d	86
3e	87
3f	88
40	89

Table 8. Gamma ROM Values (Continued)

Address	Data
Hex	Hex
41	8a
42	8b
43	8c
44	8d
45	8e
46	8e
47	8f
48	90
49	91
4a	92
4b	93
4c	94
4d	95
4e	96
4f	96
50	97
51	98
52	99
53	9a
54	9b
55	9b
56	9c
57	9d
58	9e
59	9f
5a	a0
5b	a0
5c	a1
5d	a2
5e	a3
5f	a3
60	a4
61	a5
62	a6
63	a7
64	a7
65	a8
66	a9
67	aa
68	aa

Table 8. Gamma ROM Values (Continued)

Address	Data
Hex	Hex
69	ab
6a	ac
6b	ac
6c	ad
6d	ae
6e	af
6f	af
70	b0
71	b1
72	b1
73	b2
74	b3
75	b4
76	b4
77	b5
78	b6
79	b6
7a	b7
7b	b8
7c	b8
7d	b9
7e	ba
7f	ba
80	bb
81	bc
82	bc
83	bd
84	be
85	be
86	bf
87	bf
88	c0
89	c1
8a	c1
8b	c2
8c	c3
8d	c3
8e	c4
8f	c5
90	c5

Table 8. Gamma ROM Values (Continued)

Address	Data
Hex	Hex
91	c6
92	c6
93	c7
94	c8
95	c8
96	c9
97	c9
98	ca
99	cb
9a	cb
9b	cc
9c	cc
9d	cd
9e	ce
9f	ce
a0	cf
a1	cf
a2	d0
a3	d0
a4	d1
a5	d2
a6	d2
a7	d3
a8	d3
a9	d4
aa	d4
ab	d5
ac	d6
ad	d6
ae	d7
af	d7
b0	d8
b1	d8
b2	d9
b3	d9
b4	da
b5	db
b6	db
b7	dc
b8	dc

Table 8. Gamma ROM Values (Continued)

Address	Data
Hex	Hex
b9	dd
ba	dd
bb	de
bc	de
bd	df
be	df
bf	e0
c0	e0
c1	e1
c2	e1
c3	e2
c4	e2
c5	e3
c6	e4
c7	e4
c8	e5
c9	e5
ca	e6
cb	e6
cc	e7
cd	e7
ce	e8
cf	e8
d0	e9
d1	e9
d2	ea
d3	ea
d4	eb
d5	eb
d6	ec
d7	ec
d8	ed
d9	ed
da	ee
db	ee
dc	ef
dd	ef
de	f0
df	f0
e0	f1

Table 8. Gamma ROM Values (Continued)

Address	Data
Hex	Hex
e1	f1
e2	f2
e3	f2
e4	f2
e5	f3
e6	f3
e7	f4
e8	f4
e9	f5
ea	f5
eb	f6
ec	f6
ed	f7
ee	f7
ef	f8
f0	f8
f1	f9
f2	f9
f3	fa
f4	fa
f5	fa
f6	fb
f7	fb
f8	fc
f9	fc
fa	fd
fb	fd
fc	fe
fd	fe
fe	ff
ff	ff

The GAMMA LUT accommodates the Linear True Color and Linear Grey Scale models.

The LUT Bypass Path

Although no look up table is required for this function, the LUT Bypass Path is described in this section because it implements the Non-linear True Color and Non-linear Grey Scale models. When selected, this path connects pixel data entering the Pixel Port to the digital to analog converters. It is important to note that the timing of pixel data taking this path must be the same as for any other path.

SPARCstation 5 S24 Graphics Technology

A White Paper



© 1994 Sun Microsystems, Inc.
2550 Garcia Avenue, Mountain View, California 94043-1100 U.S.A

All rights reserved. This product and related documentation is protected by copyright and distributed under licenses restricting its use, copying, distribution and decompilation. No part of this product or related documentation may be reproduced in any form by any means without prior written authorization of Sun and its licensors, if any.

Portions of this product may be derived from the UNIX® and Berkeley 4.3 BSD systems, licensed from UNIX Systems Laboratories, Inc. and the University of California, respectively. Third party font software in this product is protected by copyright and licensed from Sun's Font Suppliers.

RESTRICTED RIGHTS LEGEND: Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 and FAR 52.227-19.

The product described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications.

TRADEMARKS

Sun, Sun Microsystems, the Sun logo, SunSoft, Solaris, OpenWindows, the XIL Imaging Library, the XGL Graphics Library, the SunPHIGS Graphics Library, and the Direct Xlib Graphics Library are trademarks or registered trademarks of Sun Microsystems, Inc. UNIX and OPEN LOOK are registered trademarks of UNIX System Laboratories, Inc.. All other product names mentioned herein are the trademarks of their respective owners.

IrisGL and OpenGL are trademarks of Silicon Graphics, Inc. NTH GL is a trademark of Nth Graphics. PostScript and Display PostScript are trademarks of Adobe Systems. Figaro PHIGS is a trademark of Liant Software. GRAFPAK GKS is a trademark of Advanced Technology Center. HOOPS is a trademark of Ithaca Software. Prior GKS is a trademark of Gallium Software.

All SPARC trademarks, including the SCD Compliant Logo, are trademarks or registered trademarks of SPARC International, Inc. SPARCstation, SPARCstation 10SX, SPARCserver, SPARCengine, SPARCworks, and SPARCcompiler are licensed exclusively to Sun Microsystems, Inc. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

The OPEN LOOK® and Sun™ Graphical User Interfaces were developed by Sun Microsystems, Inc. for its users and licensees. Sun acknowledges the pioneering efforts of Xerox in researching and developing the concept of visual or graphical user interfaces for the computer industry. Sun holds a non-exclusive license from Xerox to the Xerox Graphical User Interface, which license also covers Sun's licensees who implement OPEN LOOK GUIs and otherwise comply with Sun's written license agreements. X Window System is a trademark and product of the Massachusetts Institute of Technology.



Please
Recycle

Contents

1. Overview	1
Introduction	1
Advanced Graphics Integration.....	2
Markets for Image Manipulation.....	3
Color Prepress.....	3
Graphic Arts	4
Technical Imaging.....	4
Technical Document Imaging	5
Commercial Document Imaging.....	5
Geographic Information Systems (GIS).....	6
Earth Resources.....	6
Desktop Video and Multimedia	6
2. SPARCstation 5 S24 Architectural Overview.....	9
SPARCstation 5 S24 Architectural Design Goals	9
SPARCstation 5 S24 High-Level Architecture	10

S24 High-Level Architecture	13
Software	14
The AFX Bus	15
The S24 ASIC.....	17
The RAMDAC.....	18
The S24 Monitor Interface	19
Frame Buffer Memory	20
SPARCstation 5 S24 Performance.....	20
SPARCstation 5 S24 Packaging.....	22
3. SPARCstation 5 S24 Software.....	25
SPARCstation 5 S24 Software Interfaces.....	25
The XIL Imaging and Video Library.....	25
The XGL Geometry Library.....	27
The OpenWindows Environment.....	27
Direct Xlib vs. Xlib Graphics Libraries.....	28
4. Summary	31
References	33

Introduction

Desktop computers incorporating color displays are common today — so much so that it is easy to forget how recently they replaced the monochrome systems of just a few years ago. Some may even recall the relatively recent debate concerning the merits of color versus monochrome, and whether the perceived benefits justified the additional cost and lower window system performance.

Today, most desktop systems employ color by default, with the majority providing 8 bits, 256 colors, for display of windows and graphics. 8-bit color represents a compromise between the millions of colors that the human eye can detect and generally acceptable rendering performance. While techniques such as dithering may be applied to create more realistic appearance, the results appear grainy, and are often unsuitable for quality-oriented imaging.

Multiple images on an 8-bit system also compete for limited color resources, resulting in annoying colormap flashing as the user moves from one image to another. This effect also frequently makes text unreadable outside the selected window and defeats one of the more important benefits of using a windowing environment.

The SPARCstation™ 5 S24 was designed to ease the transition from standard 8-bit color to 24-bit color without compromising either window system performance or display resolution¹. And the SPARCstation 5 S24 provides this capability at an industry-leading price point generally associated with lower-cost 8-bit devices such as Sun's TurboGX™ graphics accelerator.

The S24 supports the display of 8-bit applications side-by-side with 24-bit ones. This capability is in distinct contrast to common PC and Macintosh® video cards which force the user to choose between 8-bit and 24-bit mode for the entire display.

Advanced Graphics Integration

As the acceptance of color has grown, so too has the increased integration of low-cost graphics devices. This integration, including increased hardware and firmware functionality, together with 24-bit capability has resulted in the ability of affordable desktop systems to address what were previously strictly high-end applications such as visualization, color prepress, imaging, and high-quality rendering.

24-bit capability in particular is likely to have a dramatic impact on the types of applications possible on desktop systems. The ability to effectively generate photorealistic images, manipulate large image files in a professional publishing environment, and perform computationally intense technical tasks such as medical imaging and visualization are likely to bring previously unavailable facilities to entirely new classes of users.

Finally, the continuing emergence of multimedia, video technologies, and personal color publishing, combined with increasingly sophisticated graphics arts and imaging applications, is setting the stage for the common availability of 24-bit true color capabilities on the desktop. And like the debate over monochrome versus 8-bit color, the basic requirement for truecolor will become established relatively soon.

The SPARCstation 5 S24 represents the newest in a generation of extremely powerful, low cost true color systems from Sun Microsystems. Complementing Sun's popular multiprocessor-capable SPARCstation 20 SX workstations, the SPARCstation 5 S24 brings the features of high-end imaging

1. 24-bit devices often provide this capability only at the expense of lower display resolution.

systems to the performance demanding, cost-sensitive user. Further, these two systems reflect Sun's on-going commitment to delivering 24-bit capabilities to the generic desktop — thus addressing the evolving needs of users and their emerging imaging, graphics, and video applications.

The SPARCstation 5 S24 brings equivalent high-level integration characteristic of GX™ technology to the desktop while simultaneously delivering the features and benefits of 24-bit color. Utilizing high-speed CPU and memory buses and the high-performance of the microSPARC-II™ RISC CPU while consciously making effective trade-offs in graphics functionality and cost also enables the SPARCstation 5 S24 to raise the base level capabilities of workstation systems.

The SPARCstation 5 S24's advanced frame buffer was designed to deliver cost effective 24-bit support for X Windows at prices comparable to less powerful and more limited Apple® Macintosh and Power Macintosh® machines. Equally, the SPARCstation 5 S24's sophisticated architecture enables a workstation platform to provide 24-bit graphics capability at performance levels generally associated with simpler 8-bit devices while reducing annoying screen flashing problems resulting from sharing colormap entries across separate 8-bit applications.

Markets for Image Manipulation

Among the key trends computing users face today are the move toward running multiple, varied applications and the growing demand for high-performance graphics and integrated video. Multitasking operating systems, window-based graphical user interfaces, and powerful multimedia applications are just some of the factors influencing the trend toward more powerful highly-integrated graphics on the desktop.

The sections that follow describe the key primary and secondary market segments, the characteristics of their applications, and how the SPARCstation 5 S24 architecture provides support for them.

Color Prepress

A typical customer in this field wants to scan in high-resolution color photographs and prepare them for printing on high-quality color output devices. The first step in this preparation might be to perform simple image processing operations on the images, such as sharpening, blurring, rotation, or cropping. Finally, the images must be converted to the CMYK color space,

which provides the digital equivalent of photographic color separation. Morphological dilation of selected image features (known as "trapping") may be employed to avoid registration problems.

Because color prepress is concerned with guaranteeing image fidelity, it is important that the image data be digitally maintained at the highest possible resolution. The SPARCstation 5 S24 system's ability to efficiently manipulate large image rasters, coupled with its support for color space manipulation, make it ideal for color prepress requirements. The high level of integration between the S24 frame buffer, microSPARC CPU and system memory also insures that image processing operations can be performed quickly on multiple images.

Graphic Arts

Graphics arts applications perform the creation and manipulation of line art and images as used in a variety of areas including publishing, illustration, and advertising. Common operations include compositing, color correction, image enhancement, and scaling. As a cost-sensitive application area, graphic arts users have traditionally been restricted to 8-bit color display, leaving more sophisticated image enhancement to expensive high-end systems.

Applications addressing the graphic arts market would use the SPARCstation 5 S24 for a number of tasks. In addition to line art, the S24 can provide painting and retouching images, color correction, and image processing. The low entry cost of the SPARCstation 5 S24, combined with its extensive upgrade capability makes the platform even more attractive to users in this market.

Technical Imaging

Technical imaging applications include remote sensing, medical imaging, machine vision, and other areas where imaging is used to distinguish physical characteristics from digitized imagery. Technical imaging applications require broad base-level functionality for image manipulation, display, compression, and color. These applications frequently require advanced image processing capabilities, including transformations, filtering, and image enhancement.

The SPARCstation 5 S24 system is especially well suited for the wide range of image sizes, pixel depths, and compression characteristics common in technical imaging markets. The ability to manipulate and display sequences of images is extremely important to technical imaging users in areas such as medicine and remote sensing — and the SPARCstation 5 S24 system's high-speed access to

physical memory ensures that high frame rates can be achieved. In addition, many common image manipulation facilities (such as copy, pan, scroll, fill, clear, BLIT and stipple) are directly accelerated by the SPARCstation 5 S24 system.

Technical Document Imaging

This market handles fewer images than the commercial market, although they are generally much larger. Potential applications in the technical document imaging market include textile design, cartography, facilities layout, construction, and architecture. Applications in this market need to process very large images, provide image-manipulation capabilities such as 'de-speckling' for removal of scanning artifacts, realignment to correct for inaccurate scanning registration, and warping to deal with the attributes of old hard-copy drawings printed on unstable materials.

The SPARCstation 5 S24 system was designed to accelerate the common low-level image raster operations specifically characterized by the technical document imaging market. Many common image operations are directly supported through individual and multiple optimized operations, and also take advantage of higher performance integer and floating point capabilities of the system's CPU. The memory access speed between the S24 frame buffer and the system's physical memory further ensures that the copying and manipulation of multiple source images occurs very quickly.

Commercial Document Imaging

Banks and other financial institutions, airlines, and insurance companies process enormous numbers of paper documents during a typical day. Applications addressing this market must read digital versions of documents from scanning devices, compress the images into CCITT Group 3 or 4 facsimile format, index, and store them. Later, it may be necessary to retrieve the documents in a rapid manner for additional image processing, scaling, and enhancement.

The SPARCstation 5 S24 system balances imaging operation execution through accelerated frame buffer and CPU-based support of the manipulation and processing of images including compression/decompression capabilities, data conversion, resizing (including zooming), and enhancement facilities. The

SPARCstation 5 S24 system's ability to manipulate image raster data very quickly is especially important to this market segment, because of the extremely large volume of images that must be previewed and managed.

Geographic Information Systems (GIS)

In general, GIS applications vary broadly in data format, and manipulate large amounts of data. As a result, GIS applications depend equally upon both integer and floating point data manipulation and rendering.

The SPARCstation 5 S24 system is well suited for GIS applications for a variety of reasons. Copy speed from main memory to the screen and large in-memory support provide capabilities important to GIS. Where data format or quantity is particularly important, the SPARCstation 5 S24 architecture enables graphics data to be processed very quickly, through faster data fetching. The microSPARC-II CPU additionally ensures optimized floating point and integer performance where required. Finally, the 24-bit rendering capabilities of the SPARCstation 5 S24 system, together with its ability to manipulate very large rasters make it attractive to GIS.

Earth Resources

Like geographic information systems, the earth resources market typically deals with large volumes of data. Earth resources market requirements include 2D image, 2D geometry data, and occasionally 3D geometry data. With such varied requirements, earth resources applications need hardware with the flexibility to accelerate geometry with a range of coordinate types, and with the capability of manipulating images efficiently.

The SPARCstation 5 S24 system is an ideal platform for earth resources applications. Its direct support of video memory and fast access to display memory enables earth resources applications to manipulate and render large data sets easily.

Desktop Video and Multimedia

In recent years, multimedia and video capabilities have become important to users in both stand-alone and workgroup environments. Multimedia and video require simultaneous support for multiple data types including text, geometry graphics, images, audio, and video.

The SPARCstation 5 S24 delivers this capability through support for coincident 8- and 24-bit data, geometry graphics on par with Sun's low-cost accelerators, and real-time video with the addition of Sun's SunVideo products. The SPARCstation 5 S24's 24-bit capabilities also improve video display through its fast raster data movement and elimination of colormap flashing which is particularly troublesome on most 8-bit systems.

SPARCstation 5 S24 Architectural Overview

2 

SPARCstation 5 S24 Architectural Design Goals

The SPARCstation 5 S24 frame buffer is part of a highly integrated graphics architecture utilizing an innovative enhancement to the microSPARC-II CPU used in Sun's low-cost desktop systems. The microSPARC-II CPU chip is specifically designed with the AFX 64-bit graphics bus and memory controller logic built-in to reduce latency in the system.

The SPARCstation 5 S24 processor is the first device to take advantage of the microSPARC-II's integrated AFX bus controller. The S24 is designed to provide extremely low-cost, high-performance 24-bit graphics capability to geometry and imaging applications. Among the key features of the S24 processor are:

- Support for three types of 24-bit pixels
- 8-bit indexed color support
- Hardware accelerated block transfer (BLIT) of pixels for operations such as screen-to-screen copy, panning, and scrolling
- Write buffer logic for queuing of up to four individual write operations
- Integrated 32x32 bit hardware cursor support compatible with Sun's GX and TurboGX graphics processors
- Programmable resolution capability for support of up to 1152x900 pixel resolution displays
- Read cache to nearly double read performance

The S24 was designed to utilize an integrated graphics bus with direct access to system memory for several reasons. By designing the S24 to interact with the CPU and memory through a high-speed dedicated bus architecture, the system is able to deliver cost-effective 24-bit rendering capabilities at performance levels generally associated with simpler 8-bit devices.

Because pixel processing speed is, in general, a limiting factor for many application areas, any significant acceleration is an attractive proposition — particularly where low-cost and 24-bit are important considerations. Further, higher integration with the CPU helps reduce component cost while removing bottlenecks associated with slower I/O interfaces.

The memory interface on most systems typically has the highest bandwidth of any data path in the system. The S24's direct connection to the CPU makes it an ideal candidate for providing a path for access to additional graphics performance and functionality. Figure 2-1 illustrates the S24 frame buffer.

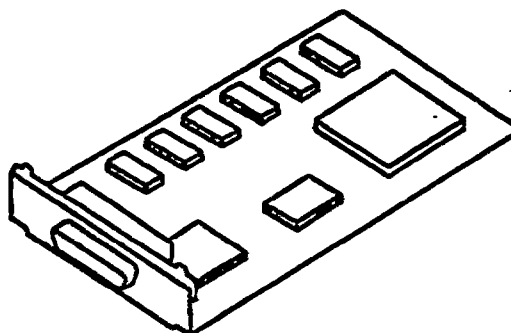


Figure 2-1 The S24 frame buffer

SPARCstation 5 S24 High-Level Architecture

Graphics performance is a function of more than just the acceleration provided by a graphics processor. It is equally dependent on overall system throughput including CPU processing performance, I/O, memory to CPU speed, display rendering, and networking. System performance is even more significant when associated with frame buffers and graphics devices which interact in a closely-coupled fashion to the workstation's CPU and memory.

From a high-level perspective, the SPARCstation 5 S24 implements performance-critical functionality in its highly integrated modular design. Among the key architectural features of the SPARCstation 5 S24 system are:

- **High-performance, low cost microSPARC-II CPU**
The microSPARC-II processor is a highly integrated, 0.5 micron CMOS based processor consisting of more than 2.3 million transistors.

The CPU consists of a SPARC Version 8 compliant Integer Unit, a SPARC Reference Memory Management Unit (MMU), a Floating Point Unit (FPU), Instruction and Data Caches, DKAM controller, and an SBus Controller.
- **64-bit memory bus**
Supported directly by the on-chip memory management unit (MMU), this wide memory bus reduces the overhead in fetching data from memory and sending data to the S24 frame buffer.

This capability is particularly important in light of the need to move 24-bit pixel data (32-bit increments) to the frame buffer. The resulting low-latency significantly improves rendering performance, placing it on par with that of 8-bit devices.
- **On-chip caches**
Supporting 8K direct mapped data and 16K direct mapped instruction caches, the microSPARC-II processor ensures that CPU data and instruction fetches occur with a minimum of cache misses. Additionally, the CPU supports a technique known as cache streaming which further accelerates transfer of data and instructions to the on-chip integer and floating point units.
- **SBus controller**
Providing the subsystem that handles input and output between external devices and the system's CPU, the on-chip SBus controller improves I/O bandwidth through Direct Virtual Memory Access (DVMA) for pipelining of bus arbitration and bus grants for previous DVMA writes. Using this approach, I/O bandwidth is increased by 30% over previous microSPARC implementations.
- **High-performance AFX Bus interface**
The SPARCstation 5 is equipped with a 100-pin microstrip connector that implements the AFX bus. Providing a memory-level interface between the system CPU, memory, and graphics subsystems, the AFX bus delivers high

bandwidth, low latency, slave-only access to the S24 frame buffer by placing the graphics interface directly on the system memory bus in a uniprocessor environment.

The AFX bus consists of a 64-bit wide data path, providing 28-bit physical addressing for the S24 frame buffer. The AFX bus further supports synchronous operation (except for interrupts) and 1, 2, 4, or 8-byte data transfers, thus permitting optimized movement of data from the CPU to the frame buffer.

The microSPARC-II directly implements the AFX bus protocol on-chip, allocating 256 MB of address space for graphics use. The 70MHz microSPARC-II processor has 186 MByte/second write bandwidth across the AFX bus, increasing to 226 MBytes/second with the 85 MHz version of the CPU. As such, the AFX bus scales with the CPU's performance and clock rate.

Figure 2-2 illustrates the microSPARC-II CPU architecture, and Figure 2-2 shows the high-level architecture of the SPARCstation 5 S24 system.

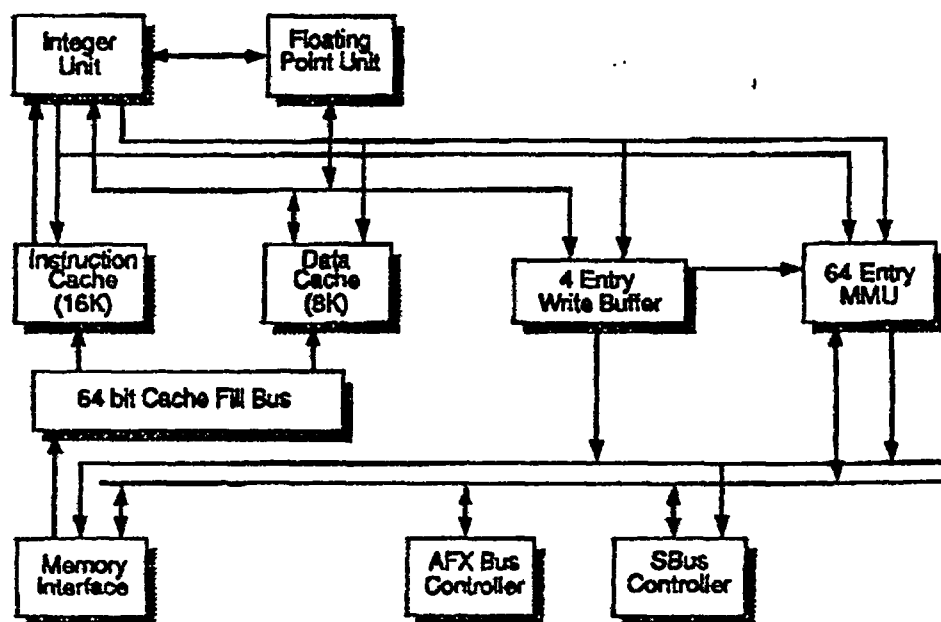


Figure 2-2 microSPARC-II processor architecture

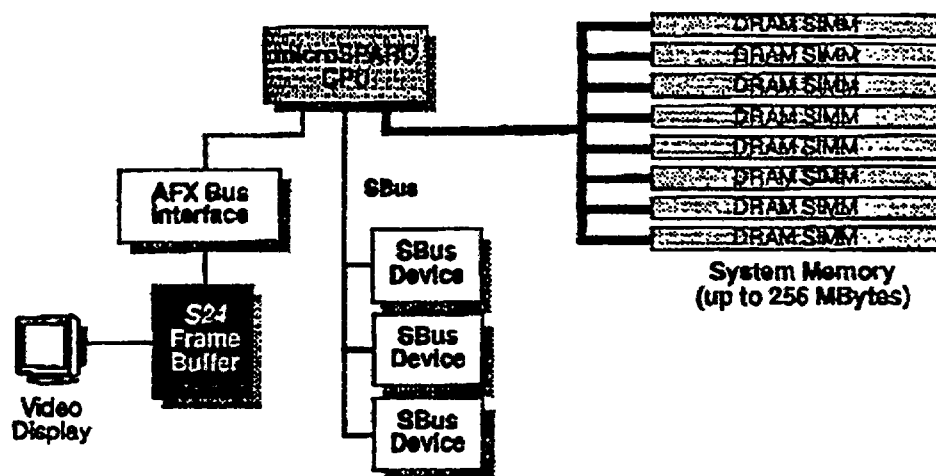


Figure 2-3 SPARCstation 5 S24 system high-level architecture

S24 High-Level Architecture

In designing the S24, Sun's engineers analyzed existing graphics and frame buffer architectures, market trends, and component cost/capabilities to establish a set of trade-offs in building a high-volume 24-bit device. Among the key considerations in the design included low cost to enable broader usage of truecolor for emerging desktop applications ranging from prepress and personal publishing to graphics arts and technical imaging.

To achieve acceptable performance levels at acceptable enduser pricing, the S24 had to provide 24-bit truecolor rendering at levels traditionally associated with 8-bit (256 color) devices. Further, since the S24 would target low-cost desktops such as the SPARCstation 5, it was critical that the overall frame buffer architecture leverage design capabilities and overall system performance.

To achieve the level of integration required to sustain necessary performance levels, as well as meet cost requirements, the S24 required the development of an entirely new digital-to-analog converter (RAMDAC). At the time of the S24's design, no commercial component vendor offered a product capable of simultaneous 8 and 24-bit display in a low-cost fashion. As a result, Sun began working with commercial component vendors to develop a RAMDAC which

met these requirements while also leveraging architectural characteristics of the overall S24 design. Today, the same RAMDAC is available commercially through AT&T Microelectronics to the general public.

Once the RAMDAC design considerations were addressed, Sun's engineers examined how they would achieve their cost targets for the remaining board logic. The design of a single highly integrated ASIC providing all the necessary control functions between the VRAM, frame buffer logic, and AFX bus allowed the reduction of dependencies on external logic, which further reduced cost.

Design trade-offs also included determining which graphics operations were supported in hardware, and which would occur in software. Given the clear importance of providing highest possible image fidelity at a low price, the S24 design heavily leverages the architecture of the microSPARC-II processor, and the SPARCstation 5 desktop system. At the same time, it was clear that high throughput of pixel data from the CPU and memory coupled with maximum rendering performance were more important to image-oriented users than pure vector performance. Figure 2-4 provides a block diagram of the high-level architecture of the SPARCstation 5 S24.

Software

Software is a significant part of the development time and cost associated with an accelerated frame buffer. The S24 architecture is stateless, allowing for the elimination of overhead and complexity of providing context switching. This approach greatly simplifies software development, including porting of low-level graphics and imaging software interfaces as well as the devices' interaction with the window system.

The S24 is also register compatible with the features it has in common with Sun's 8-bit GX accelerator family. This compatibility allowed software to be tested early on the GX, so that when functional hardware became available, the window system was already ported and working.

The reliance on the SPARCstation 5 desktop and microSPARC-II CPU enabled the designers to focus on the integration of the S24 and its relationship to the system's memory and CPU. As a result, the S24 provides extremely cost-effective 24-bit capability at performance levels of Sun's original GX graphics processor, and at the same time supporting a broad range of pixel types displayed simultaneously on the same screen, as shown in Table 2-1.

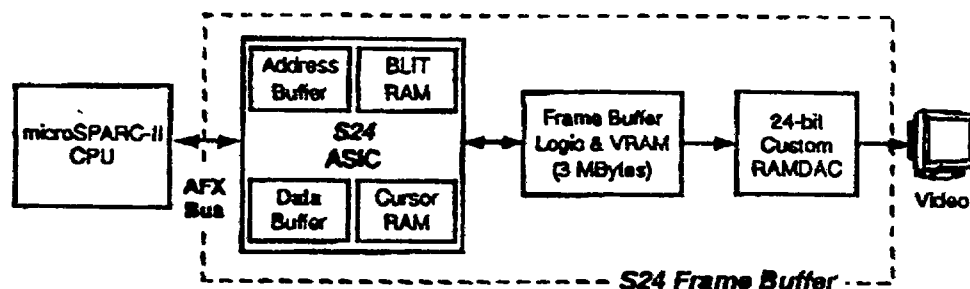


Figure 2-4 High-level S24 architecture block diagram

The AFX Bus

The AFX Bus is a new high-speed bus designed to allow a low-end 24-bit frame buffer to match the pixel performance of an 8-bit frame buffer on the SBus. Address and data line for the AFX bus are shared with the memory interface to save pins on the CPU chip. The S24's use of the AFX bus provides high bandwidth, low latency, slave-only access to graphics hardware by placing display and/or accelerator devices directly on the system memory bus in a uniprocessor environment.

The AFX bus architecture enables the S24 board to achieve write bandwidth of more than six times that of SBus graphics devices. Read bandwidth is more than twice that of SBus-based boards.

Among the principal features of the AFX bus are:

- Bus frequency range of 23 MHz to 42 MHz
- 64-bit wide data path
- 28-bit physical address per slave device
- Synchronous operation, except for interrupts
- 1, 2, 4, or 8-byte data transfers
- One interrupt line shared with other slave devices
- 105-pin MBus-style connector supporting graphics boards with a form factor of 6.25-inches by 3.25-inches (0.5-inches longer than a standard Sbus card)

The AFX bus and system memory share memory address and data lines. As a result, the AFX bus controller and system memory must arbitrate their usage for these common lines. Figure 2-5 illustrates the AFX Bus and its relationship to the host system's CPU and memory, and the S24 frame buffer.

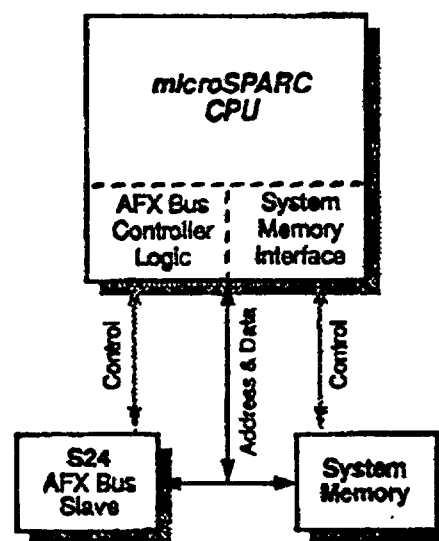


Figure 2-5 AFX Bus block diagram

The AFX Bus Controller

The AFX bus controller is the single master of the AFX bus and is responsible for initiating each AFX bus cycle. The controller is closely integrated with the CPU, and system memory controller.

The S24 as an AFX Bus Slave

The AFX bus slave, in this case, the S24 graphics processor, responds to requests as they are issued by the AFX bus controller. The S24 can be programmed to generate an interrupt at the start of the monitor's vertical retrace.

The architecture of the AFX bus permits multiple slave devices, which coexist and communicate with the controller, without specific knowledge of one another. The SPARCstation 5 does not however currently support multiple AFX bus devices.

The S24 ASIC

The S24 ASIC provides the core logic on the S24 board. It includes frame buffer control, read and write buffers, memory for BLIT operations, cursor RAM, and interface logic.

A four element first-in first-out (FIFO) buffer provides write queuing which complies with AFX bus specification. Control logic on the A24 ASIC handles video timing, memory control, and hardware cursor control.

Hardware Cursor

The S24 ASIC contains a dual plane 32 x 32 pixel programmable hardware cursor. The two cursor planes define three possible index values for the cursor color as well as transparent (i.e., allow the frame buffer contents to display). The cursor index values are applied to an overlay color lookup table in the RAMDAC.

BLIT RAM

The S24 frame buffer supports a stateless BLIT operation capable of copying from 1 to 32 pixels from any given pixel location to another. The BLIT RAM consists of a pair of 16 x 32-bit RAMs to store and retrieve up to 32 26-bit pixels, two at a time.

Read Cache

Reads are transparently accelerated by a read cache which automatically fetches the other 2 pixels in a 4 pixel group. Cache hits provide read data in a minimum 2 clock AFX read cycle. This approach nearly doubles the S24's read performance.

Stipple

The S24 provides 1 bit to 8- or 24-bit pixel expansion for fill, clear, and text operations. Up to 32 pixels can be drawn with a single stateless stipple command. Stipple takes advantage of the video RAM's block write feature which is up to four times faster than conventional writes.

The RAMDAC

RAMDACs convert digital data (in frame buffer memory) to the analog signals required to drive the CRT display. The term "RAM" refers to the color lookup table RAM for 8-bit index pixels.

One of the principle design goals of the S24 processor was support for simultaneous 8- and 24-bit visuals. At the start of the S24 development effort, there were no RAMDACs available that supported 8- and 24-bit pixels on the same screen. Higher-end Sun graphics accelerators such as the SX and ZX use a large ASIC in front of the RAMDAC to support this capability. Cost goals for the S24 necessitated the development of a custom RAMDAC to eliminate the need for an ASIC in front of the RAMDAC.

The S24 employs a Sun designed, AT&T manufactured low-cost RAMDAC. The S24 (AT&T Part ATT20C567) RAMDAC provides compatibility with other commercially available RAMDACs including the popular Brooktree Bt458 used in many of Sun's CG3-compatible 8-bit frame buffers. The S24 RAMDAC provides 24-bit truecolor, 24-bit direct color using the same look-up table (LUT) as an 8-bit index, 24-bit linear color using gamma correction provided by integrated read-only memory (ROM), and 8-bit index color with binary compatibility to the Bt458 RAMDAC.

Pixel Display Mode

8-bit Indexed

24-bit Truecolor (non-linear)

24-bit Truecolor (linear, gamma corrected)

24-bit Direct (permits user-defined gamma)

Table 2-1 Pixel display modes of the S24 frame buffer

Lookup Tables

The S24 RAMDAC contains two lookup tables to transform pixel data to accommodate multiple color models, perform gamma correction, and provide compensation for video monitor characteristics. The lookup tables include the Pixel Color RAM, Gamma, and ROM LUTs. Additionally, a bypass path is provided to allow data to be sent directly to the DACs.

Pixel Color RAM Lookup Table

The Pixel Color RAM lookup table is the principle LUT for the S24. It consists of three 256 x 8-bit RAMs.

The Color RAM LUT receives pixels associated with direct color modes such as those from the X Window system. This table is also used for all 8 and 24-bit direct color pixels, while allowing for applications to change the LUT values.

Gamma Lookup Table

The GAMMA ROM is implemented for linear pixels used in synthetic images such as 3D graphics and ray tracing. It compensates for the logarithmic response of the CRT monitor. The Gamma LUT is accessed via regular color model selection mechanisms.

LUT Bypass Path

The LUT Bypass Path provides a means to send frame buffer data directly to the DACs to implement a nonlinear truecolor or greyscale color model. Because the most common 24-bit mode is non-linear, this bypass mechanism helps avoid the overhead of passing data unnecessarily through the gamma and Pixel Color RAM LUTs.

The S24 Monitor Interface

The S24 frame buffer is designed to support Sun's current multisync monitor offerings, with programmable resolution of up to 1152x900 pixels. The pixel clock frequency may also be programmed to up to 108 MHz. The monitor connector contains a sense bus to identify the monitor connected to the S24. The OpenBoot PROM code reads the monitor's ID and loads the appropriate resolution and timing parameters into the S24 ASIC. Table 2-2 lists the standard resolution and clock frequencies of the S24 frame buffer.

Screen Resolution	Display Frequency	Pixel Clock Frequency
1152 x 900	76 Hz	108 Hz
1152 x 900	66 Hz	94.5 Hz
1024 x 768 (SVGA) [†]	70 Hz	84.375 Hz

[†] Not currently supported.

Table 2-2 Supported S24 resolutions, and display and clock frequencies

Frame Buffer Memory

The S24 uses conventional video RAM (VRAM) for meet its strict performance goals. The video display overhead is only 3 percent compared with more than 50 percent for lower cost dynamic RAM (DRAM). The S24 also takes advantage of the VRAM's block write feature which allows fill, clear, and text operations to run up to four times faster than normal. For example, the S24 can fill a 500x500 pixel 24-bit window over 700 times per second. A full screen clear takes less than 6 milliseconds.

VRAM is expensive, so the S24 design uses it as efficiently as possible. The 1152x900 resolution of standard Sun displays employs 99 percent of the one megapixel frame buffer memory. Additionally, only 26 bits per pixel are stored instead of 32. 24 bits of these are used for the pixel data and two are used for the pixel type (8- or 24-bit). The S24's high performance 24-bit capability make overlay planes unnecessary, thereby reducing cost.

SPARCstation 5 S24 Performance

Due to its high level of integration and direct CPU connection, the S24 processor delivers broad performance to a large range of low-level graphics and imaging functions. These functions, illustrated in Tables 2.3, 2.4, and 2.5, represent common operations for both geometry and raster-based graphics rendering, which are accelerated through Sun's low-level APIs including XGL, XIL, and Direct Xlib libraries. Additional, 3rd-Party APIs layered above the XGL, XIL, and Direct Xlib, will provide performance corresponding to the level of optimization at which they were ported. In general, most layered APIs

experience some performance benefits using the S24 over common unaccelerated frame buffers. Chapter 3 describes the SPARCstation 5 S24 system graphics software interfaces in more detail.

Operation xbench (XGL and XIL)	SPARCstation 5 70 MHz	SPARCstation 5 85 MHz
Graphics Performance		
2D Vectors (8-bit) [†]	473 K/sec	541 K/sec
2D Vectors (24-bit) [†]	307 K/sec	360 K/sec
3D Vectors [‡]	28.9 K/sec	32.8 K/sec
3D Triangles [‡]	44 K/sec	52.4 K/sec
3D Quads [§]	7.2 K/sec	8.4 K/sec

[†] X11perf v1.3, 10 pixel line

[‡] 10 pixel line, jagged

[§] 80 pixel, flat shaded

[§] 100 pixel Gouraud shaded, unit

Table 2-3 SPARCstation 5 S24 graphics performance (xbench)

Window System Performance X11perf v1.3	70 MHz 8-bit	70 MHz 24-bit	85 MHz 8-bit	85 MHz 24-bit
Xmark (xmark93)	2.58	1.98	2.84	2.18
Dot	3.27M/sec	3.05M/sec	3.95M/sec	3.66M/sec
Rectangle (10x10)	132K/sec	131K/sec	154K/sec	153K/sec
Rectangle (500x500)	707/sec	706/sec	718/sec	713/sec
Line (10 pixel)	473K/sec	307K/sec	541K/sec	360K/sec
Scroll 500x500 rectangle)	55.9/sec	55.8/sec	56.0/sec	55.6/sec
Copy (500x500 window)	56.5/sec	56.3/sec	56.4/sec	56.1/sec
PutImage (500x500)	88.5/sec	29.4/sec	97.5/sec	33.3/sec
80 Character (6x13 pixels)	231K/sec	230K/sec	281K/sec	277K/sec

Table 2-4 SPARCstation 5 S24 window system performance

While the S24 is essentially an accelerated frame buffer, 2D and 3D vector performance (assuming CPU floating-point calculation of transformations) will also render at better rates than 8 and 24-bit dumb frame buffers connected via the SBus. Additional performance measurements for a range of general geometry and image operations are illustrated below.

Operation [†] (xlibench)	SPARCstation 5 70 MHz	SPARCstation 5 85 MHz
Color Image [‡]		
Rotate	0.68 M/sec	0.79 M/sec
Scale	1.60 M/sec	1.73 M/sec
Copy	10.48 M/sec	11.43 M/sec
Convolve	0.18 M/sec	0.21 M/sec
Color Convert	0.166 M/sec	0.19 M/sec
Set Value	18.43 M/sec	21.39 M/sec
Add	1.19 M/sec	1.43 M/sec
Add Constant	1.08 M/sec	1.35 M/sec
Blend	0.44 M/sec	0.49 M/sec
Grayscale Image [‡]		
Rotate	0.78 M/sec	0.92 M/sec
Scale	3.29 M/sec	3.82 M/sec
Copy	30.50 M/sec	33.32 M/sec
Convolve	0.59 M/sec	0.69 M/sec
Set Value	55.01 M/sec	65.03 M/sec
Add	2.69 M/sec	3.27 M/sec
Add Constant	2.55 M/sec	3.08 M/sec
Blend	1.28 M/sec	1.49 M/sec
Lookup	5.76 M/sec	5.96 M/sec

[‡] Performance in millions of pixels per second

[†] 512x512x3 pixel image

[‡] 600x600x1 pixel image

Table 2-5 SPARCstation 5 S24 performance (xlibench)

SPARCstation 5 S24 Packaging

The S24 graphics board is an separate card designed for use in AFX bus-based desktop workstations. Currently the S24 board may be purchased as a standard part of the SPARCstation 5 S24 system, or as an upgrade option.

The S24 board utilizes the AFX bus connector located adjacent to the left-most SBus slot (when viewed from the front) of the SPARCstation 5 motherboard. Figure 2-6 illustrates the location of the S24 board in the SPARCstation 5 desktop.

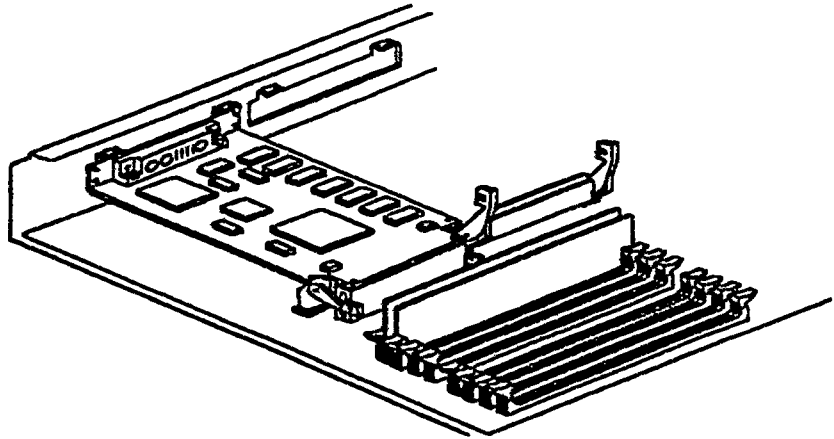


Figure 2-6 SPARCstation 5 motherboard with S24

SPARCstation 5 S24 Software Interfaces

The SPARCstation 5 S24 system is supported by all Solaris graphics and window systems APIs, including the XGL, XIL libraries, PostScript, and the OpenWindows Version 3.x (X11 R5) window system. Industry-standard X-extension libraries, such as Xlib and PEXlib, are also available and are accelerated via the XGL and XIL foundation graphics libraries. Emerging standards such as the X Imaging Extension (XIE), and Display PostScript (DPS), together with their related APIs (XIELib, DPSlib), will be supported in the future.

As both an 8- and 24-bit rendering device, the SPARCstation 5 S24 system supports and accelerates each of the APIs mentioned above. The sections that follow briefly describe the foundation graphics interfaces and the functions accelerated on the SPARCstation 5 S24 frame buffer.

Figure 3-1 illustrates the Solaris graphics and windowing APIs and their relationship to the SPARCstation 5 S24 system.

The XIL Imaging and Video Library

The XIL library is a foundation-level imaging interface providing the common functions required by many imaging and video applications. As a foundation software layer for imaging applications, the XIL library defines how imaging operations, such as display, image manipulation, and compression and decompression, are carried out.

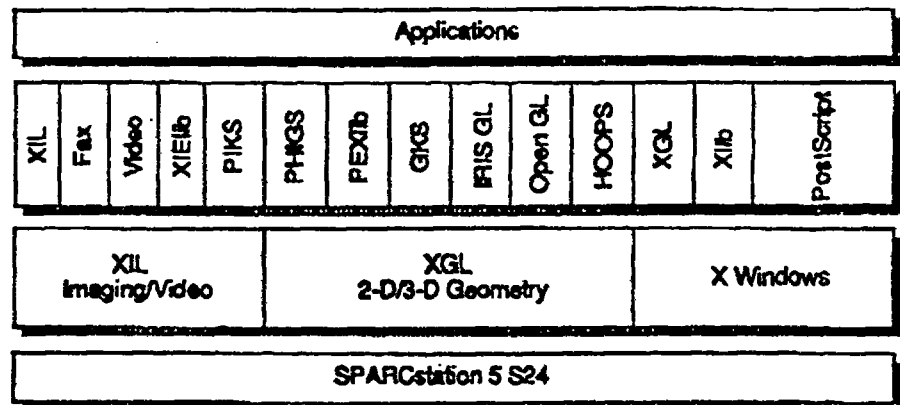


Figure 3-1 Solaris Foundation Graphics Libraries and Layered Interfaces

The XIL library contains three primary components: a programming interface specification for basic imaging functionality, a high-performance implementation of the specification, and a standard hardware interface specification, which enables third-party hardware developers to readily support the XIL library and applications built on it.

The XIL library is an object-oriented interface. As such, it hides the details of its implementation from the programmer. This enables the internals of the library to be extended without changing the interface in an incompatible way. C language bindings provide compatibility with other tools available under the Solaris environment including a large number of software development products. Figure 3.1 illustrates the XIL library in relationship to other foundation-level interfaces.

All XIL integer functions, whether they are 8-, 16-, or 32-bits, are supported, and many are accelerated by the SPARCstation 5 S24 system. Floating-point operations are handled by the microSPARC CPU's integrated floating-point processor. The SPARCstation 5 S24 system accelerates several classes of functions, including:

- Pan and zoom
- Raster manipulation (flip, BLIT, copy)
- Stipple rendering
- Colormap conversion
- Text and image scrolling

The XIL imaging library employs a building-block approach to image operations through the use of simple operations known as *atoms*. More complex operations, called *molecules*, can be implemented using XIL to optimize applications to use low-level hardware features. As a result, the SPARCstation 5 S24 system supports the acceleration of many of XIL's imaging operations.

The XGL Geometry Library

The XGL library is the Solaris foundation geometry library providing the functionality and performance required by applications requiring geometry manipulation and display. The XGL library provides for optimization of 2D and 3D rendering, including high-quality lighting and shading, and advanced primitives (such as NURBS and meshes, texture mapping, antialiasing, and transparency). The XGL library includes sophisticated state-of-the-art features, such as dynamic tessellation of NURBS surfaces, as well as flexible pipelines for lighting and transformation.

Like XIL, the XGL library has an object-oriented interface that hides the low-level details of implementation from the application developer. As such, it is architected so that it may evolve as an interface, permitting new algorithms to be implemented within the XGL library without incompatibly changing the application program interface bindings. Table 3-1 illustrates the 2D XGL operations accelerated by the SPARCstation 5 S24 system.

Operation	Description
2D	- Dots - Solid and patterned thin lines, 8 and 24 bit - Stipple line and fill - Pan and copy - Text scrolling

Table 3-1 XGL Operations Accelerated by the SPARCstation 5 S24 System

The OpenWindows Environment

The OpenWindows environment is an MIT X Window System (X11 R5) compliant server. Applications that use the X protocol, running both locally or remotely, will receive acceleration from the SPARCstation 5 S24 System.

Many window management functions have optimized to take advantage of the SPARCstation 5 S24 architecture, including the following functions:

- Hide/expose window
- Resize window
- Copy pixmaps to and from windows
- Text scrolling
- Get and put image

Many 2D rendering primitives are also accelerated with the SPARCstation 5 S24 platform, including those associated with text, vectors, raster operations and others.

Direct Xlib vs. Xlib Graphics Libraries

The Xlib interface is the basic low-level 2D graphics library shipped as part of the X11 Window System. The Direct Xlib library is a supplemental product that transparently enhances graphics rendering performance for graphics intensive Xlib applications in the OpenWindows environment. The Direct Xlib library was developed to provide an optimized Xlib interface for applications that typically run locally (client application and X Windows server on the same machine).

Direct Xlib allows most X Windows graphics rendering operations to occur directly in the client application process. This helps avoid the overhead associated with the copying of client application data to the X Windows request buffer, client/server context switching, and server parsing and processing of the request buffer for these operations. The key to the Direct Xlib library is the use of Sun's Direct Graphics Access (DGA) technology, which provides a synchronization mechanism between the Direct Xlib library and the OpenWindows server.

The principal benefit in using the Direct Xlib library is rendering performance. In general, most applications experience two times the rendering performance using the Direct Xlib library versus the Xlib interface. Use of the standard Xlib library is appropriate when graphics performance is not a factor, and when the

application must be statically linked. The following common 2-D rendering operations in the Xlib interface and the OpenWindows environment that are accelerated by the SPARCstation 5 S24 graphics system include:

- Copy
- Solid, stippled, textured fill and Raster Operation (ROP)
- Vectors
- Text

Today's desktop computing environment is becoming the focus of converging technologies supporting window systems, video, imaging, CAD, multimedia and other rendering-intensive applications. Until the SPARCstation 5 S24, this environment was driven by the lower cost almost exclusively associated with 8-bit display devices. As a result, the volume acceptance of 24-bit was limited to high-end CAD and publishing use.

The SPARCstation 5 S24, together with its accelerated counterpart, the SPARCstation 20SX, represents a important breakthrough in delivering low-cost 24-bit imaging capability to a high-volume performance-sensitive marketplace. By employing closely coupled CPU-to-memory designs while leveraging overall system throughput, these platforms are able to deliver performance to image-intensive applications, while simultaneously providing moderate geometry-based acceleration.

The key benefits of these architectures include lower cost per seat across the applications spectrum, from EDA and GIS markets to color publishing, imaging and visualization. By using the host system's dynamic memory for image manipulation, these desktop platforms not only outperform other 24-bit systems, but do so at lower cost.

Sun's SPARCstation 5 S24 and SPARCstation 20SX technology represent a clear industry direction in embedded graphics design. In the future, 24-bit support will become commonplace, as will the use of highly-integrated closely-coupled CPU/memory/graphics architectures which provide for the simultaneous processing of floating-point data and non-rendering application code while dedicated processor or framebuffer logic manages rendering.

References



- Solaris XGL 3.1 Reference Manual*, SunSoft, 1994.
- Solaris XGL 3.1 Programmer's Manual*, SunSoft, 1994.
- Solaris XIL 1.1 Imaging Library Programmers Guide*, SunSoft Inc.
- Xlib Reference Manual*, O'Reilly & Associates. Inc.
- Xlib Programming Manual*, O'Reilly & Associates. Inc.
- OpenWindows 3.0.1 User's Guide*, SunSoft, 1994.
- OpenWindows 3.0.1 Programmers Guide*, SunSoft, 1994.
- An Introduction to Computer Graphics Concepts*, Addison-Wesley, 1991.
- XGL Geometry Graphics for Solaris White Paper*, SunSoft Inc., 1994.
- Solaris VISUAL White Paper*, SunSoft Inc., 1993.
- XIL Imaging for Solaris White Paper*, SunSoft Inc., 1994.
- The microSPARC-II Processor White Paper*, Sun Microsystems Computer Corporation, 1994.
- The SPARCstation 5 Architecture White Paper*, Sun Microsystems Computer Corporation, 1994.

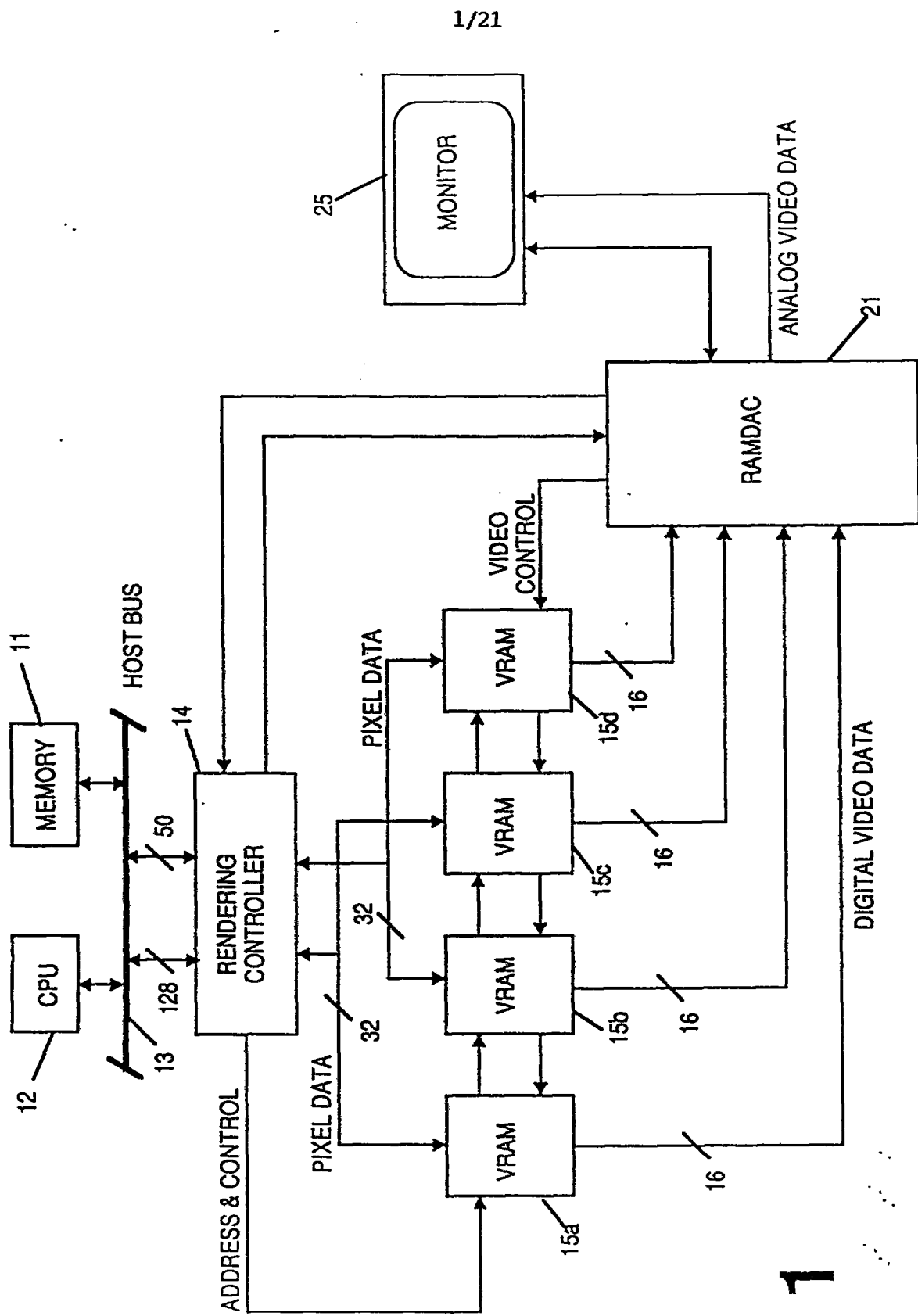
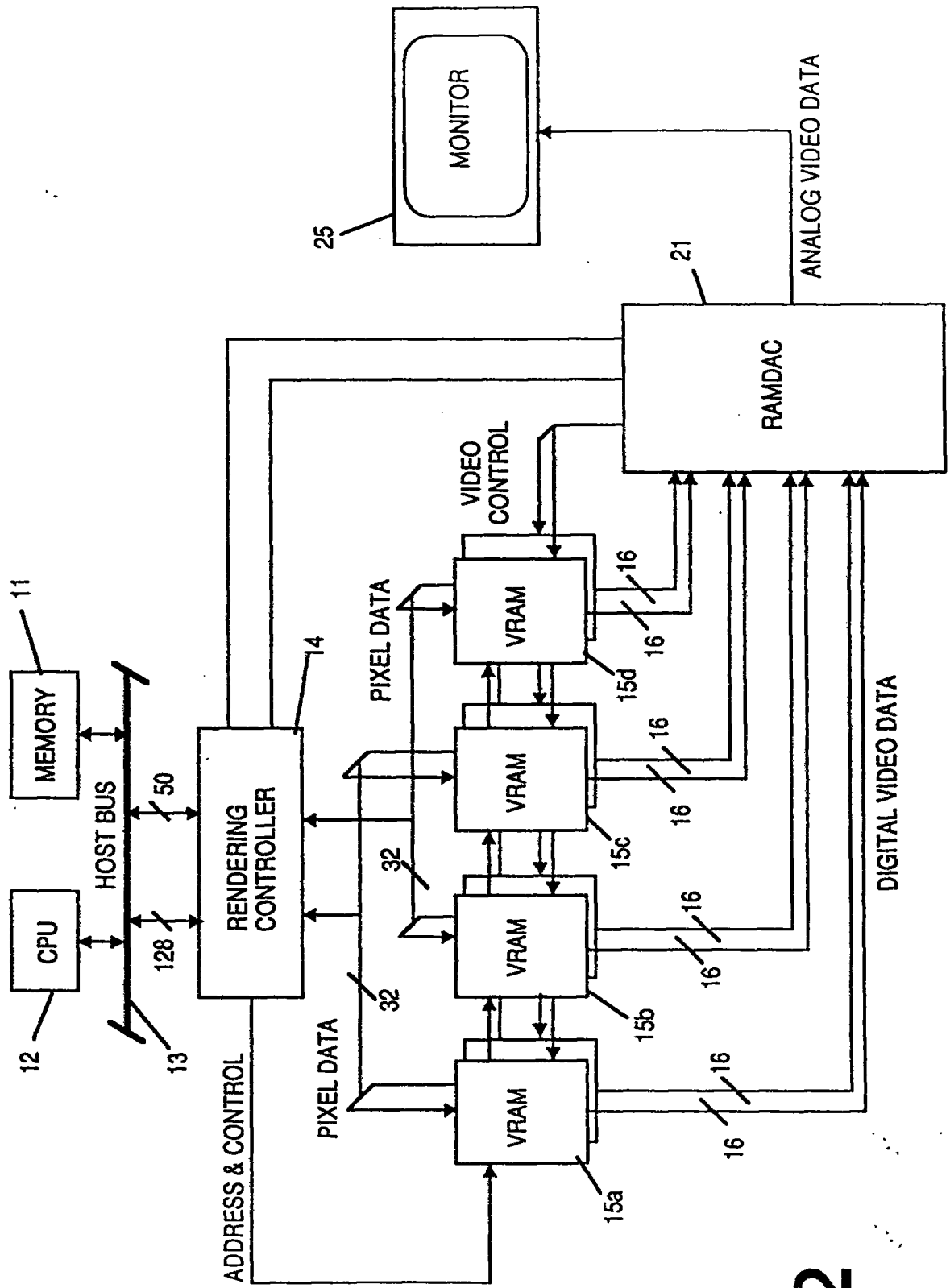


Fig. 1

**Fig. 2**

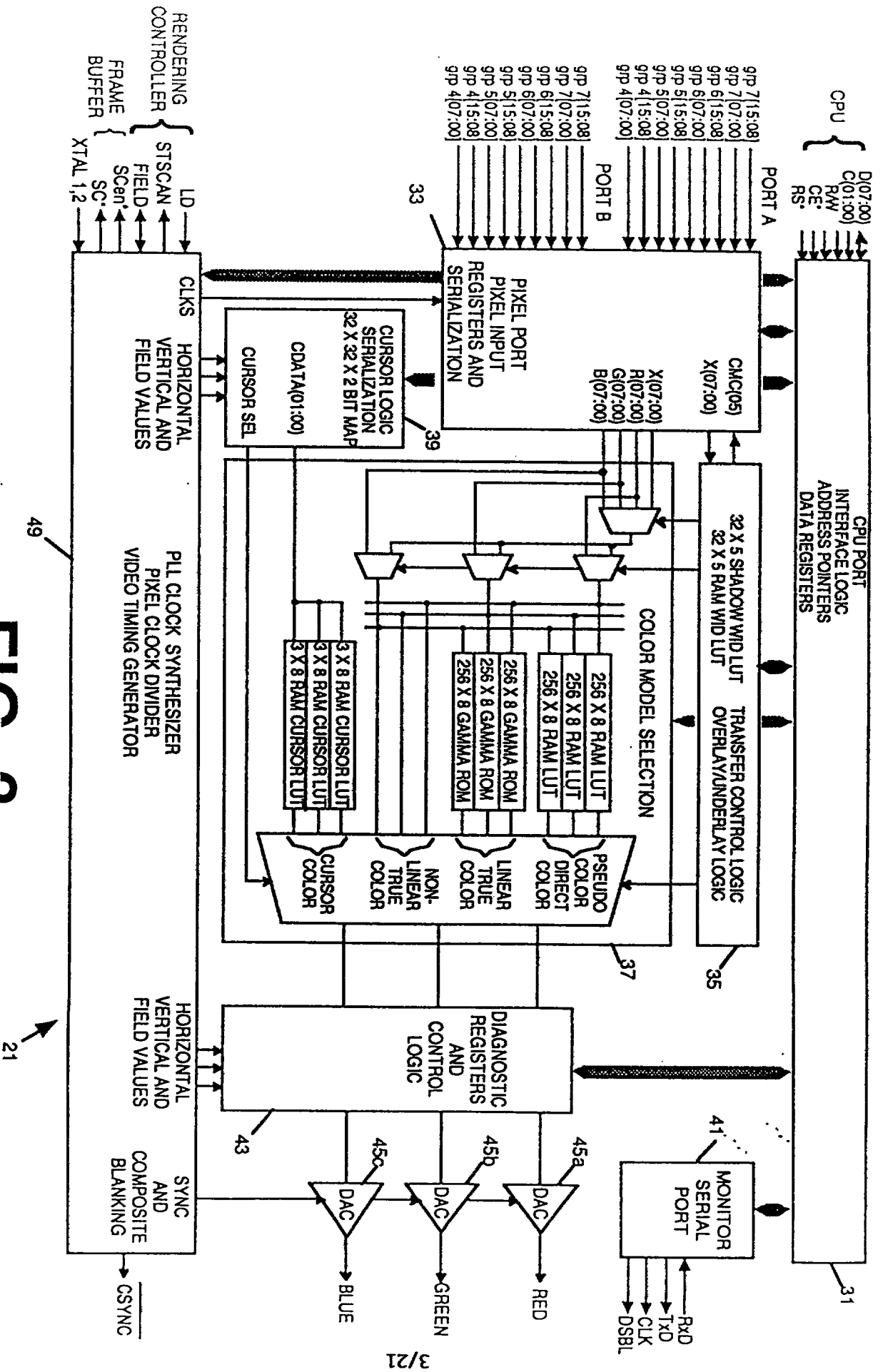
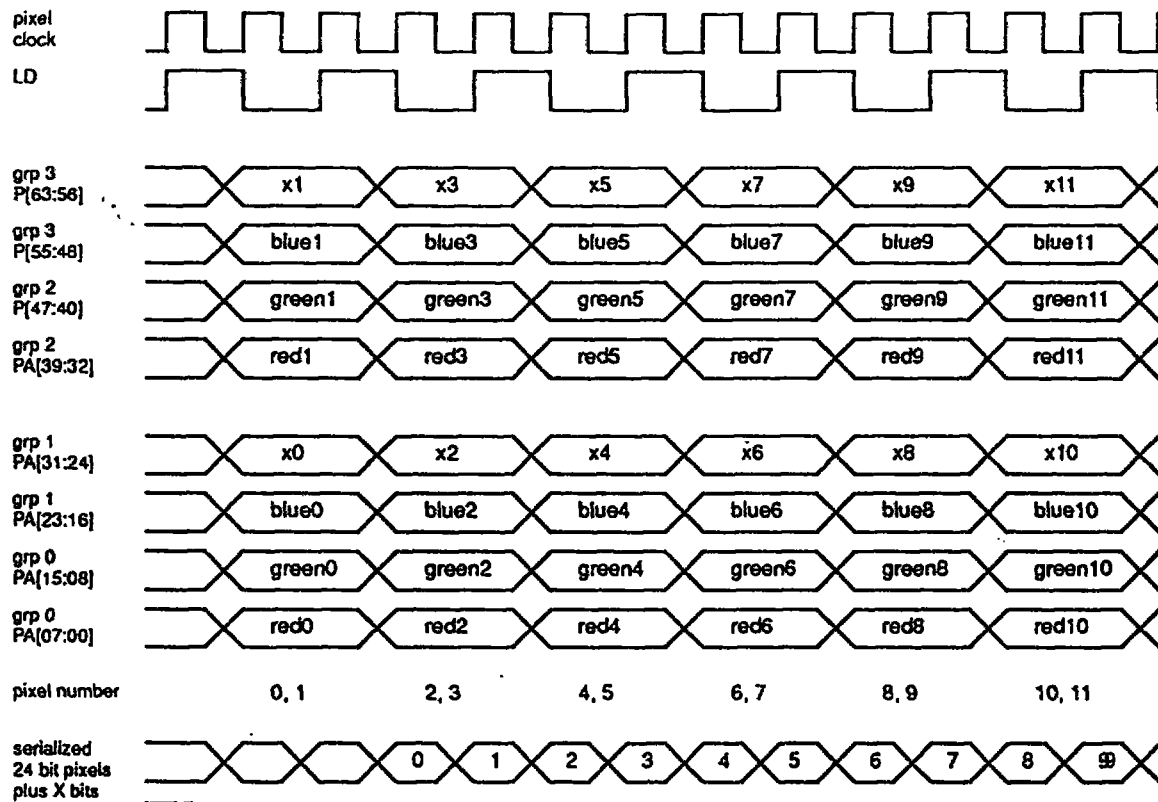


FIG. 3

**Fig. 4****2:1 Single Buffered Interleaved Pixel Format**

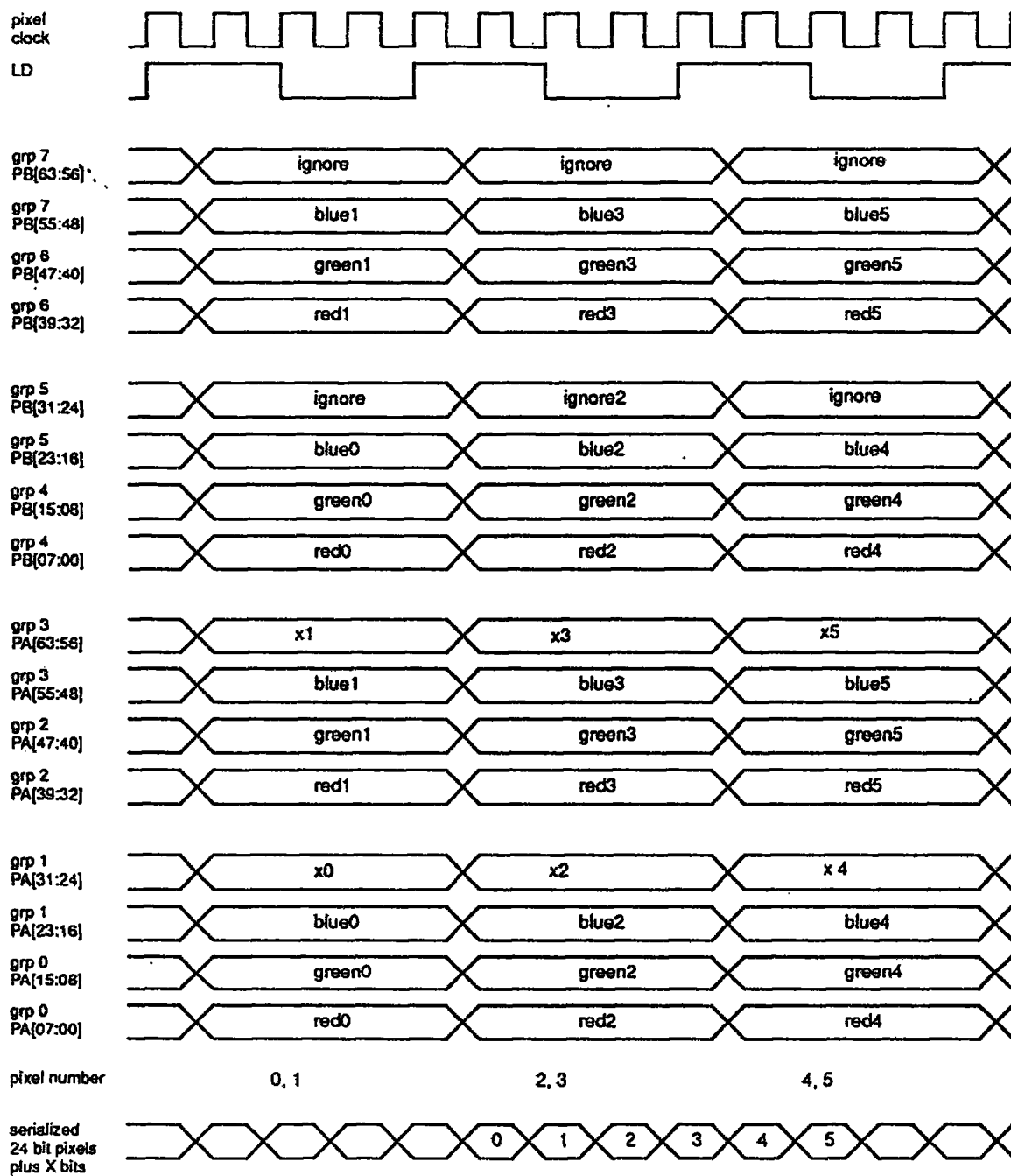


Fig. 5 2:1 Double Buffered Interleaved Pixel Format

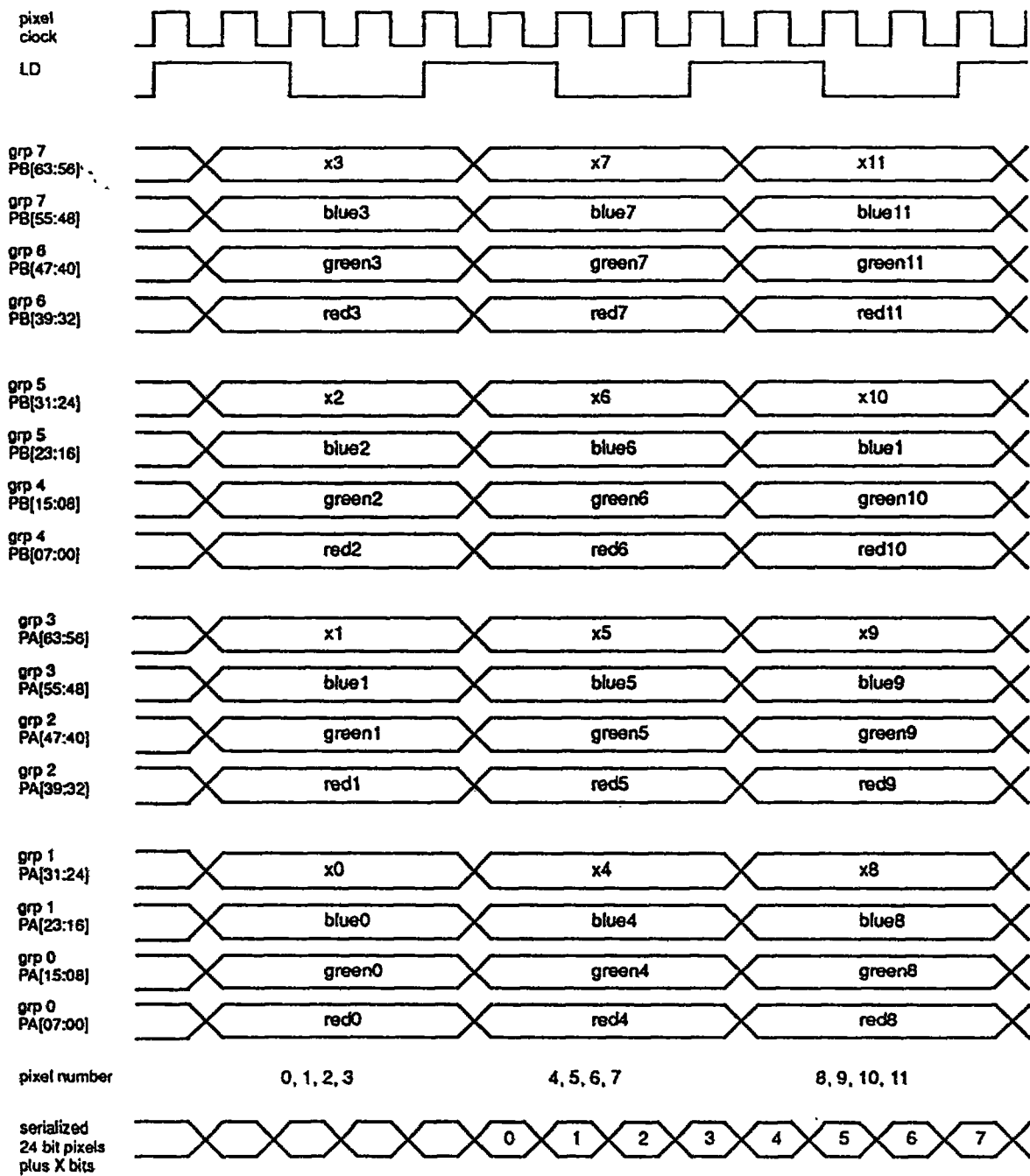
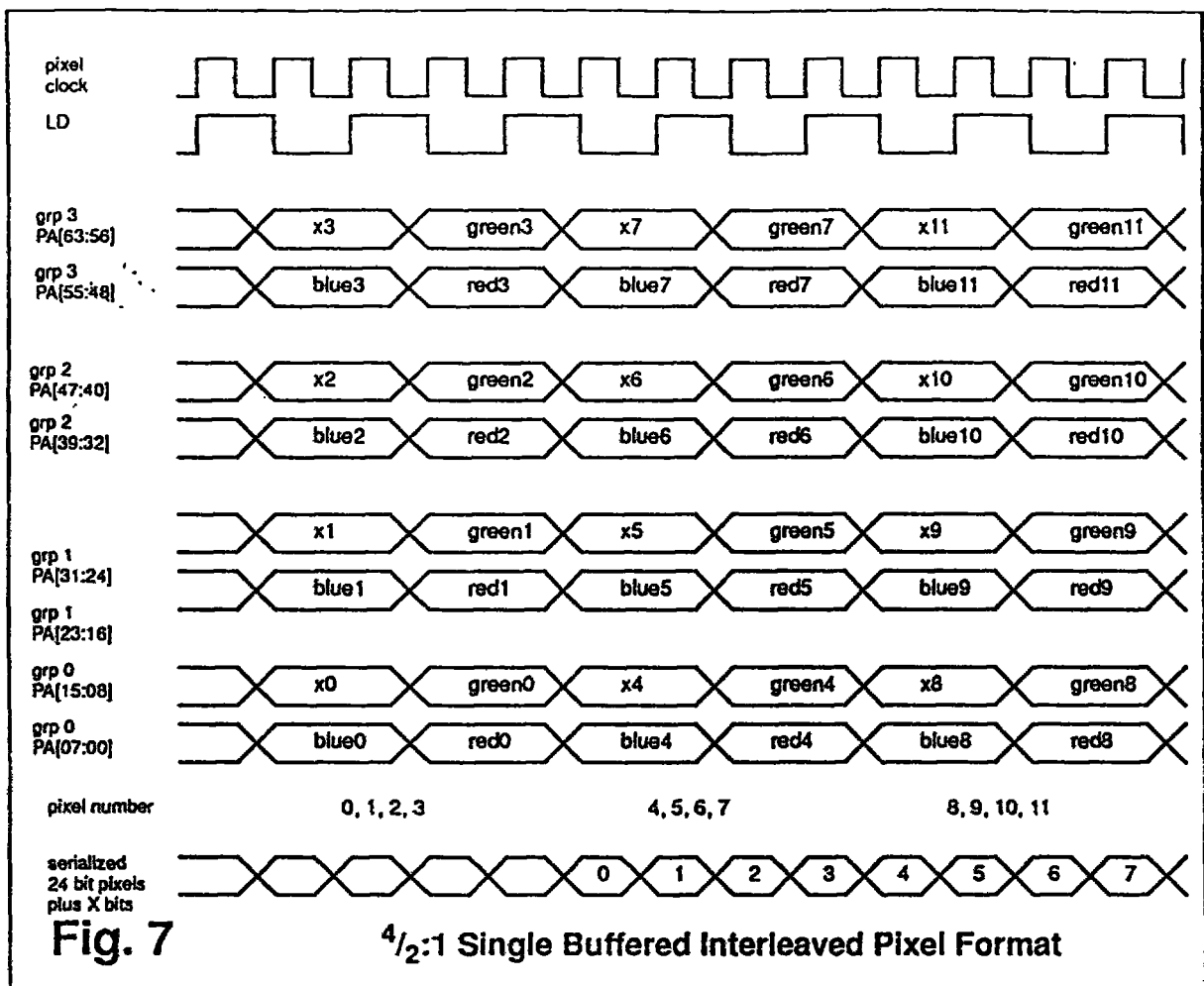


Fig. 6 4:1 Single Buffered Interleaved Pixel Format



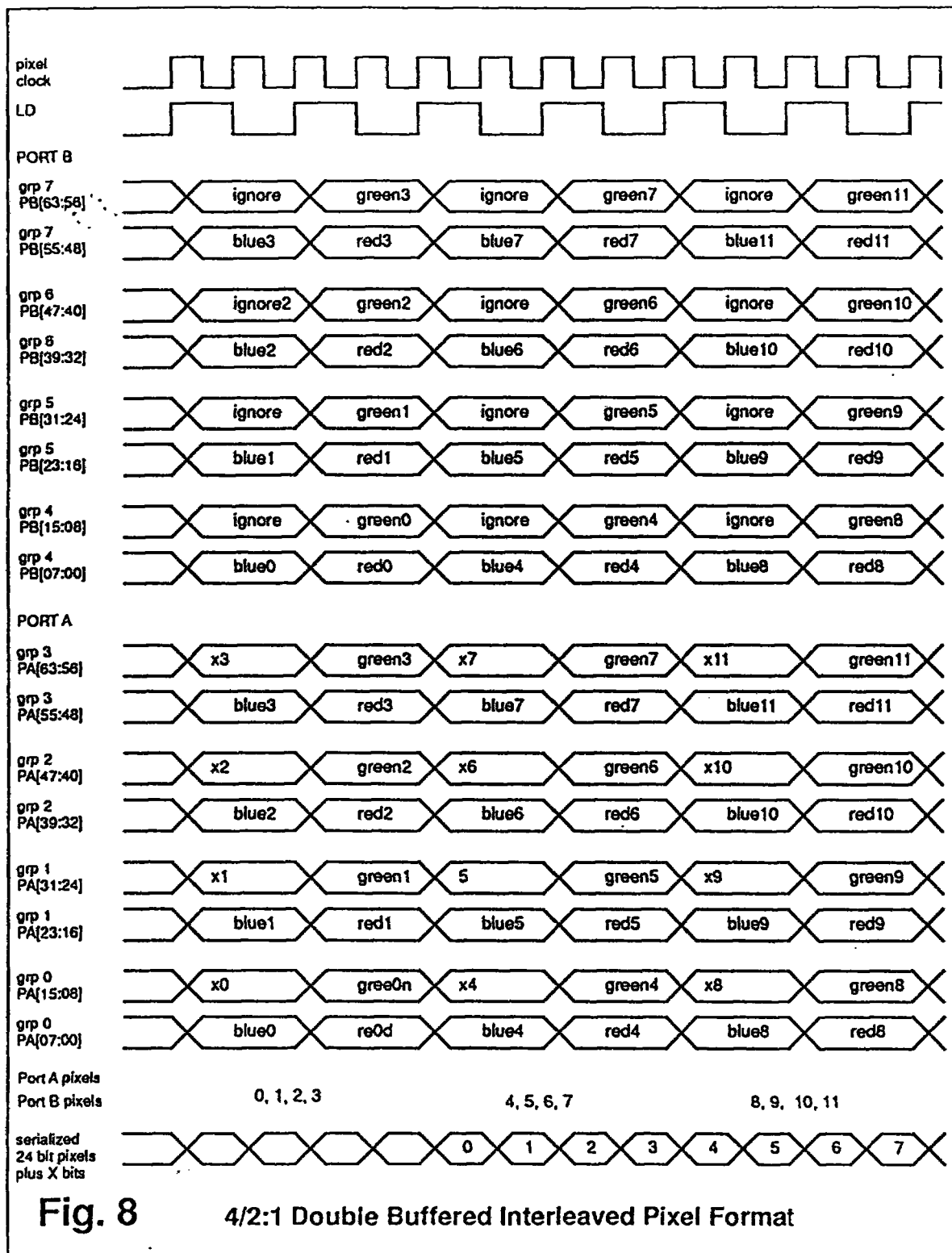


Fig. 8

4/2:1 Double Buffered Interleaved Pixel Format

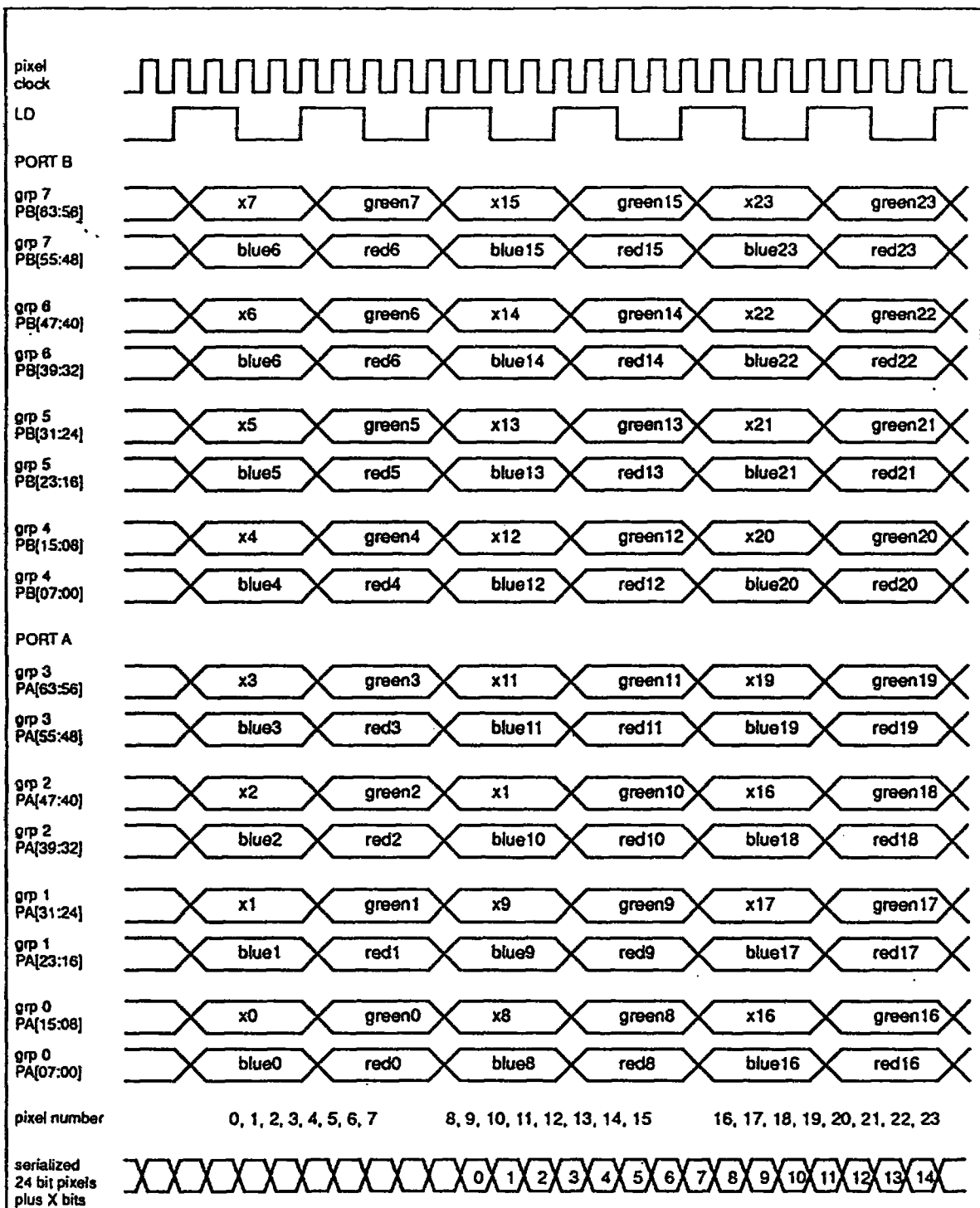
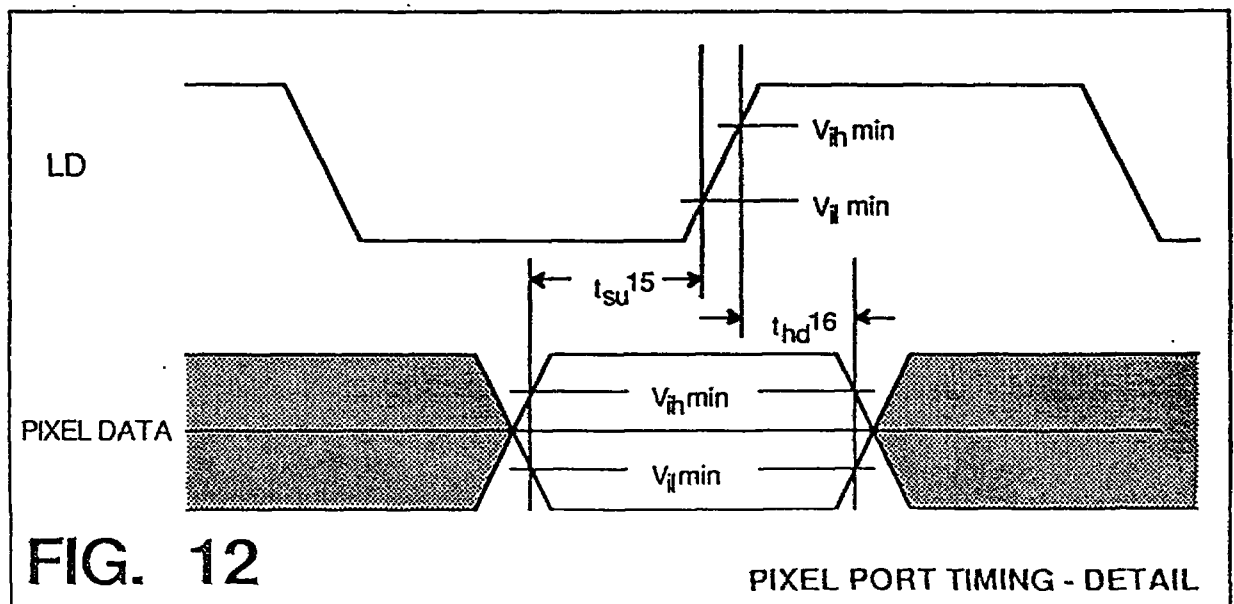
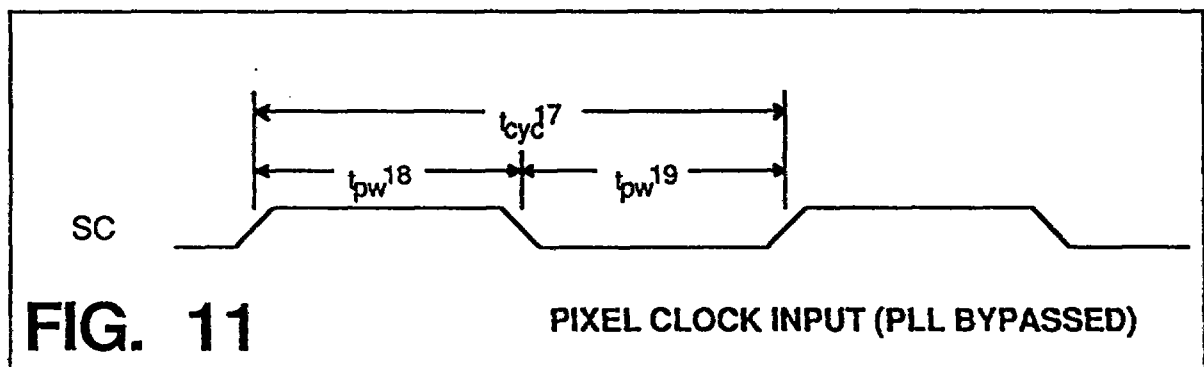
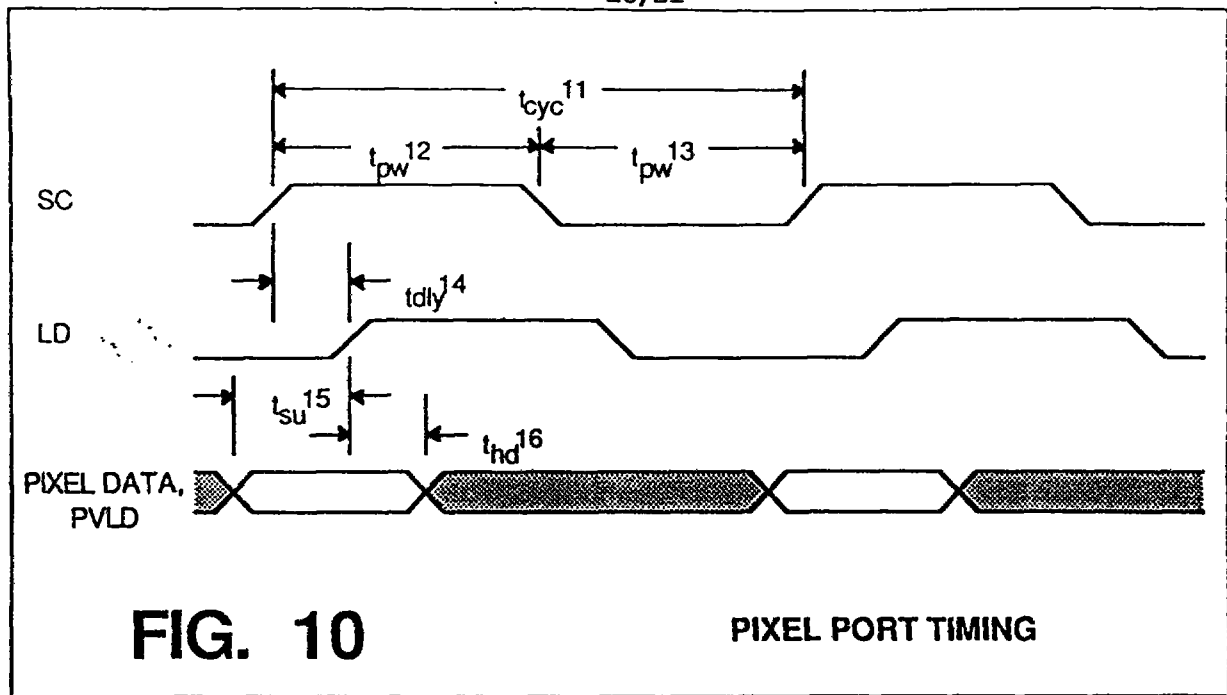
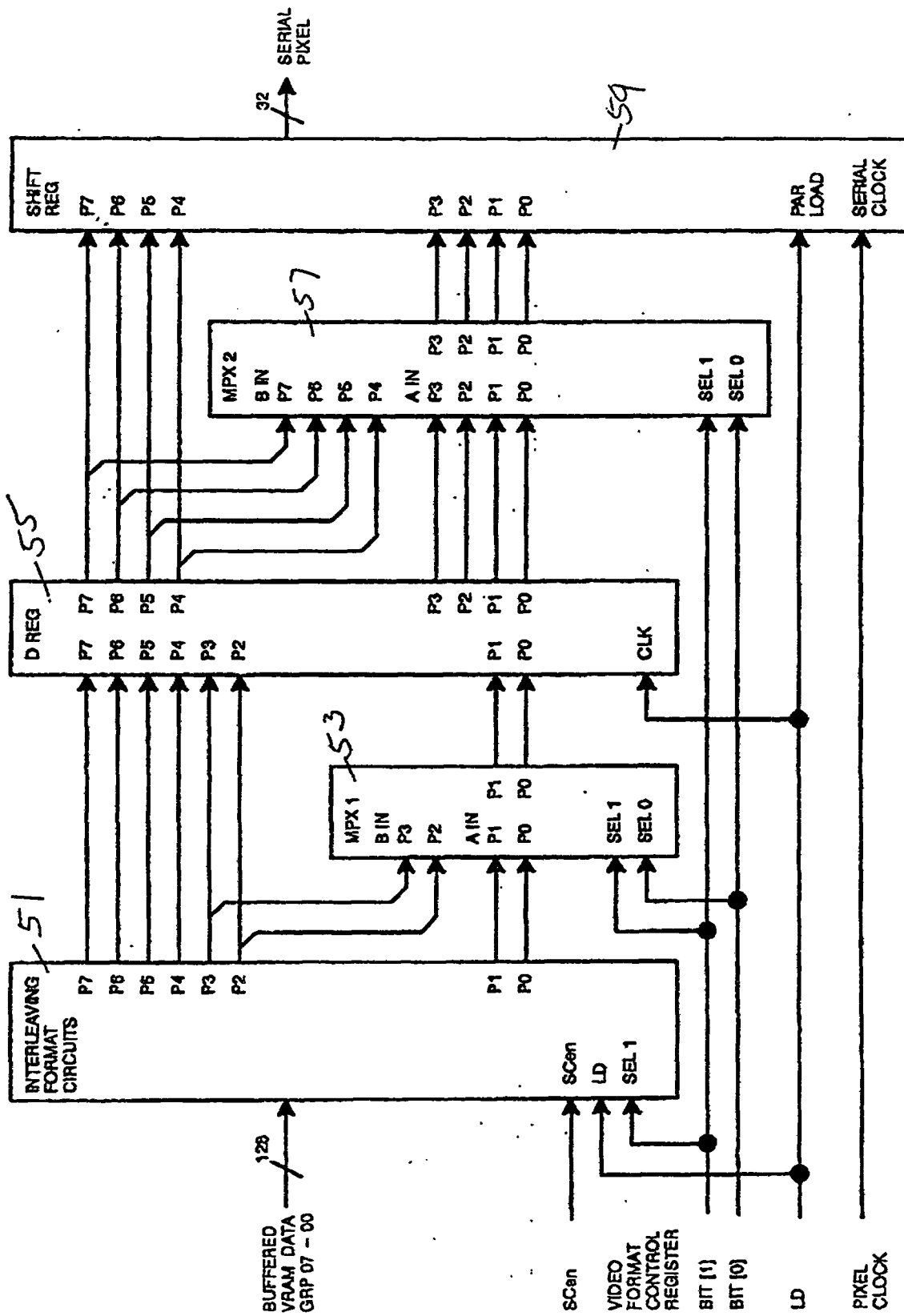


Fig. 9 $8/2:1$ Single Buffered Interleaved Pixel Format

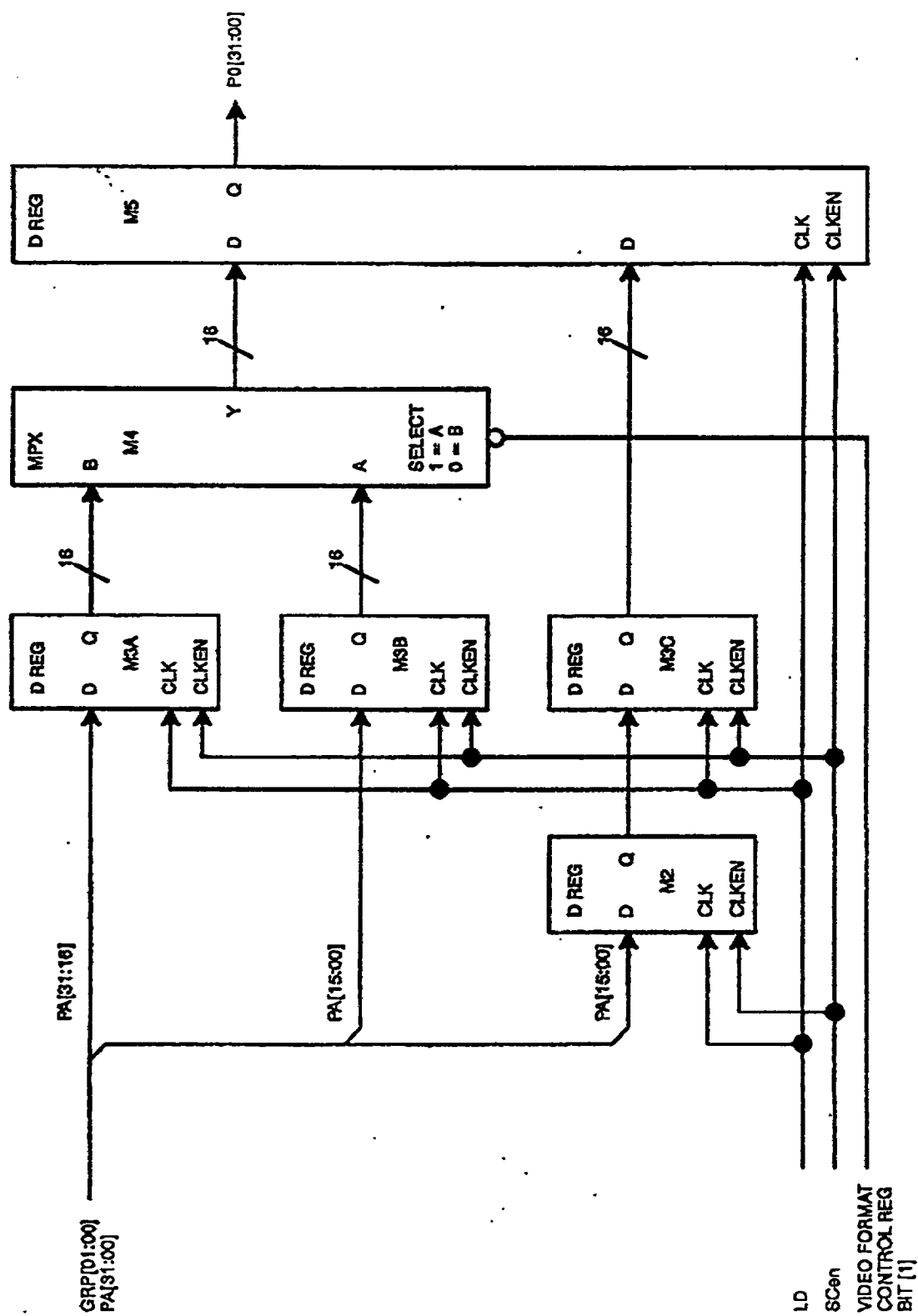




PIXEL PORT - INPUT REGISTERS AND SERIALIZATION

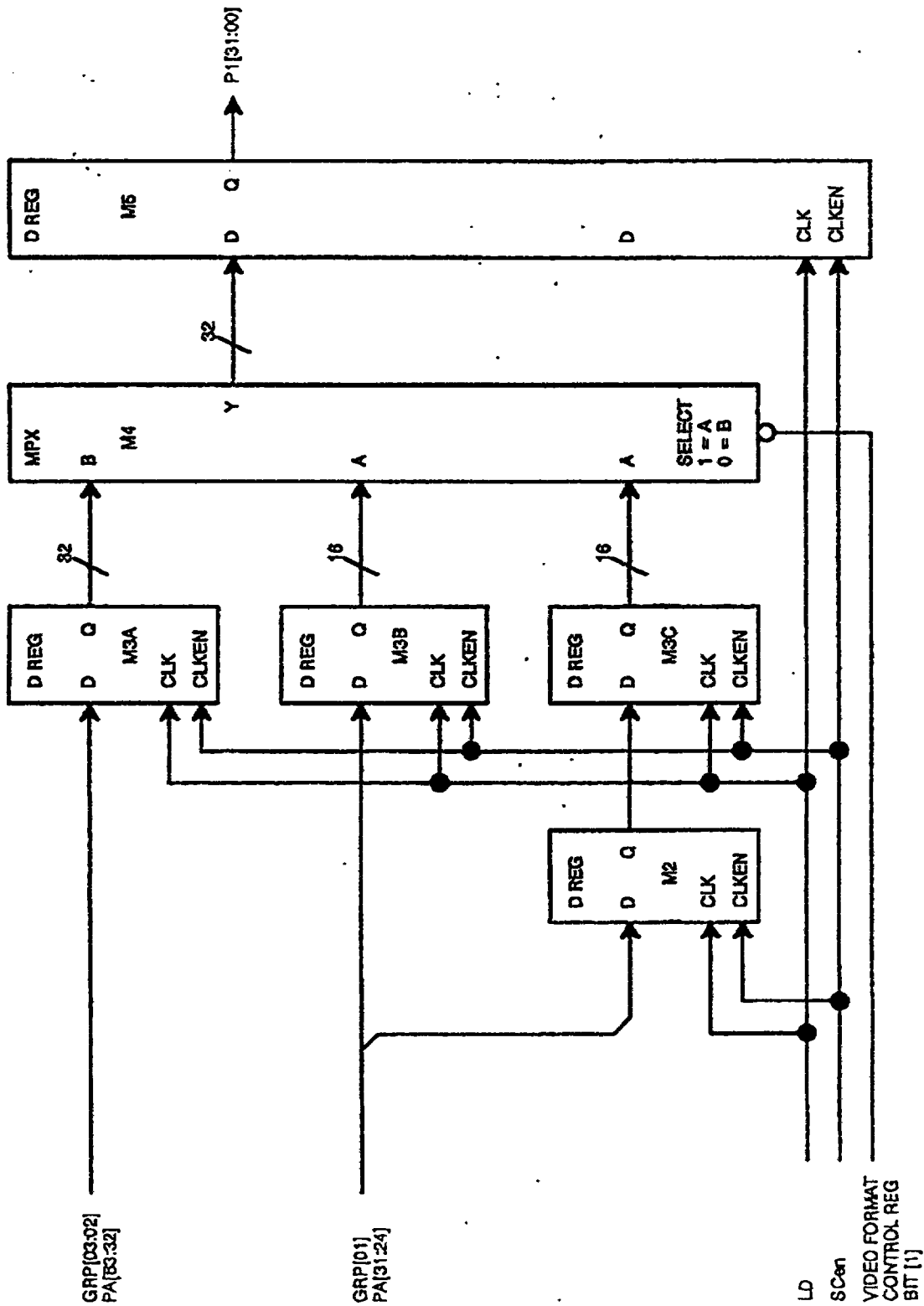
FIG. 13

12/21



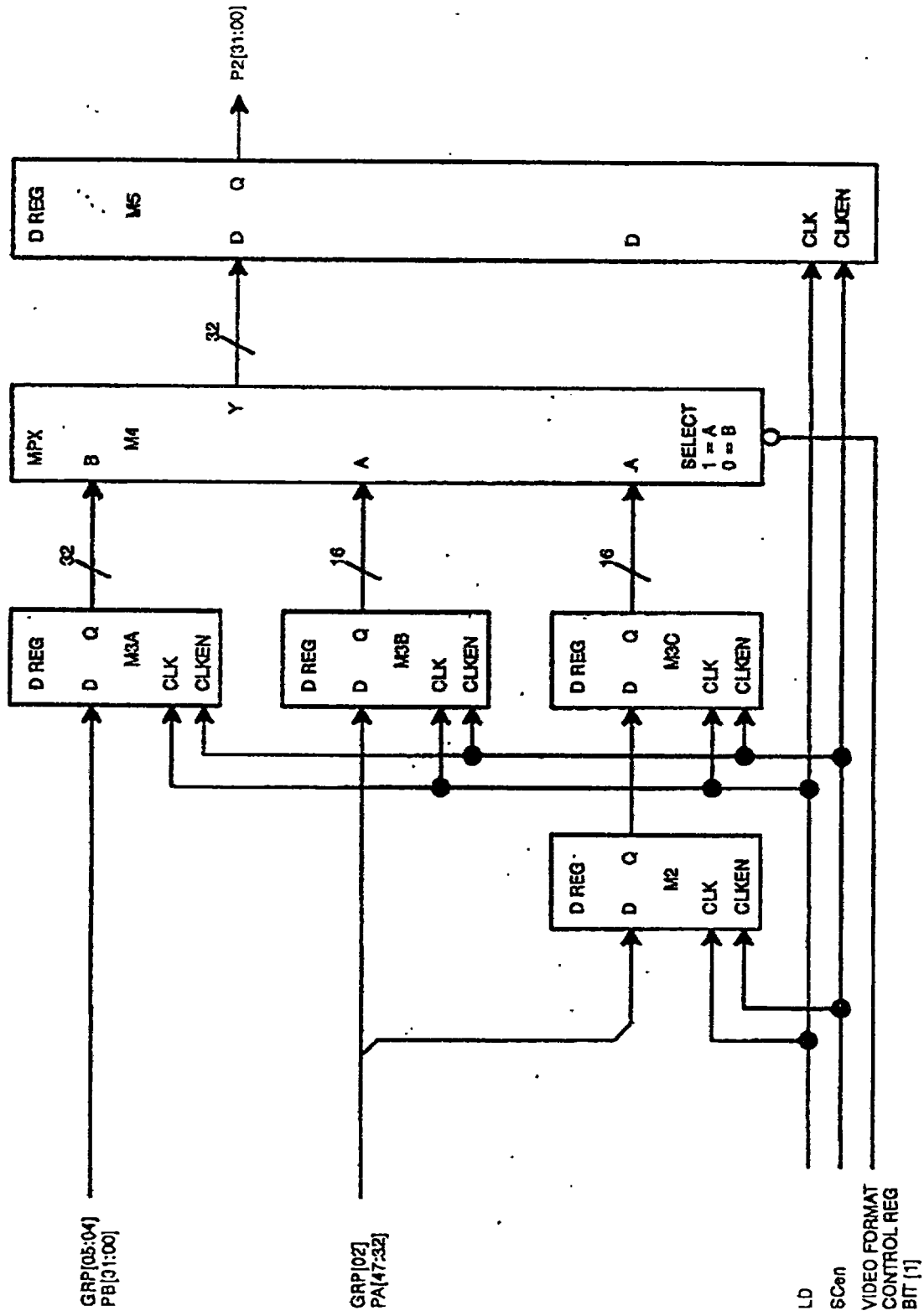
PIXEL PORT -- PIXEL 0 INTERLEAVING FORMAT CIRCUIT

FIG. 14



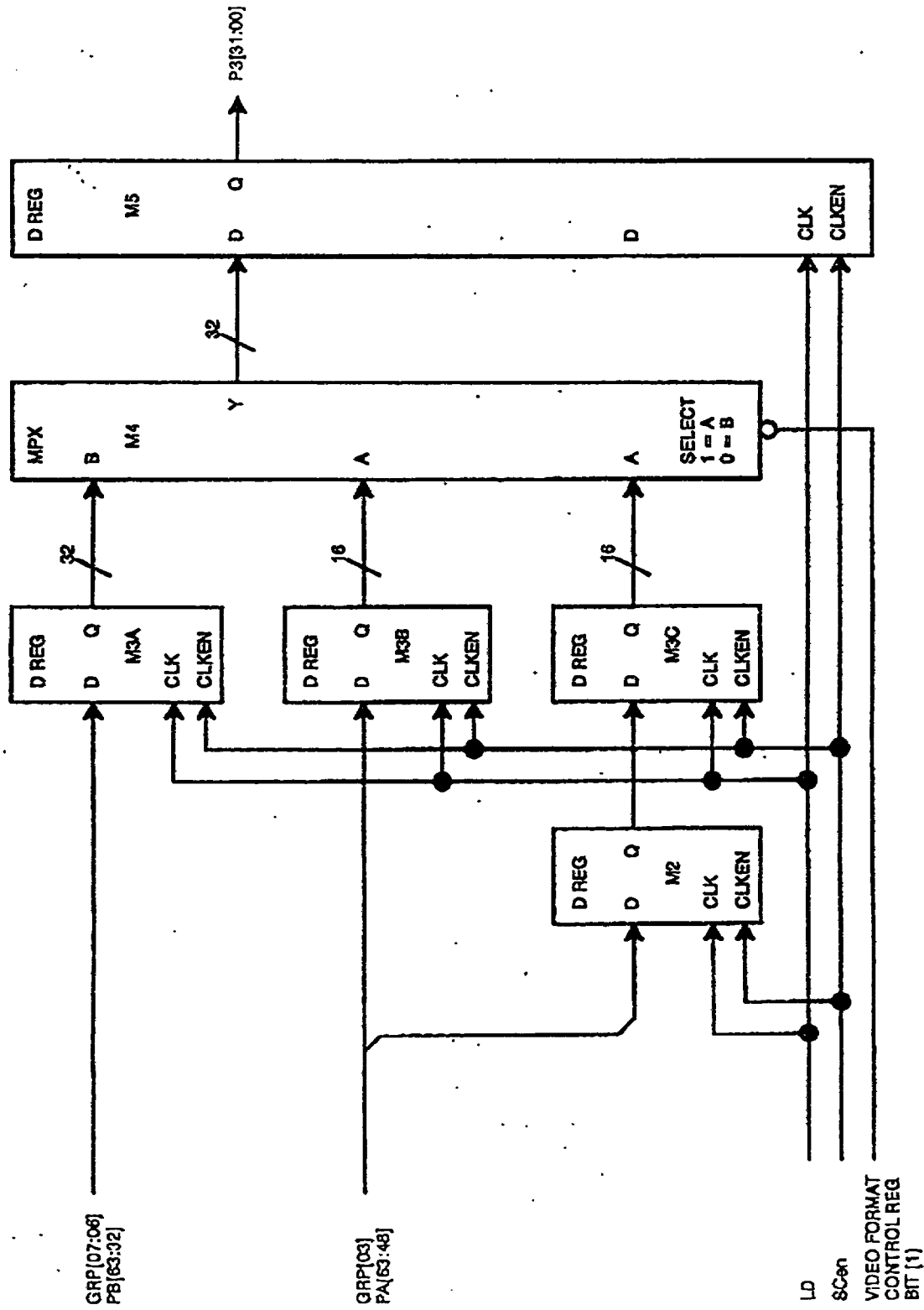
PIXEL PORT - PIXEL 1 INTERLEAVING FORMAT CIRCUIT

FIG. 15



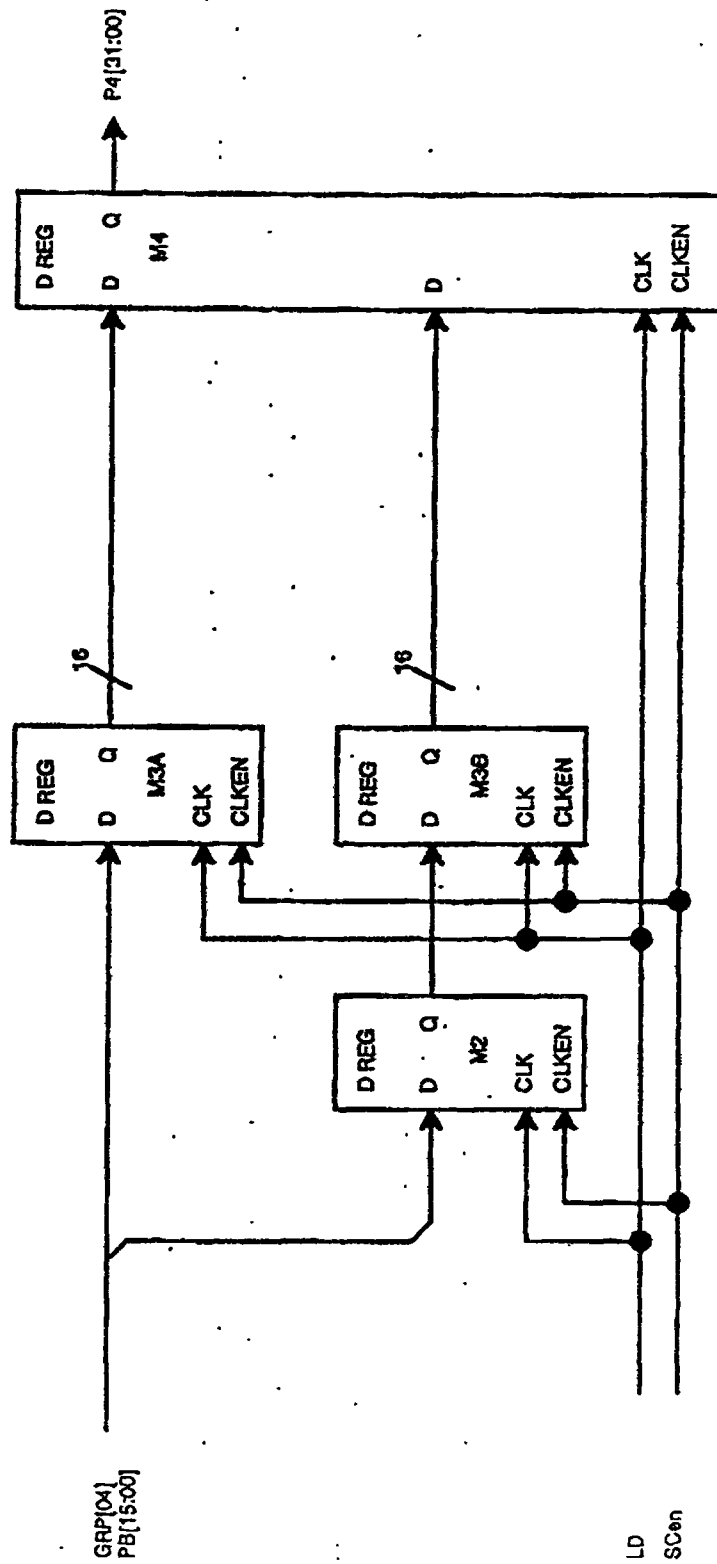
PIXEL PORT - PIXEL 2 INTERLEAVING FORMAT CIRCUIT

FIG. 16



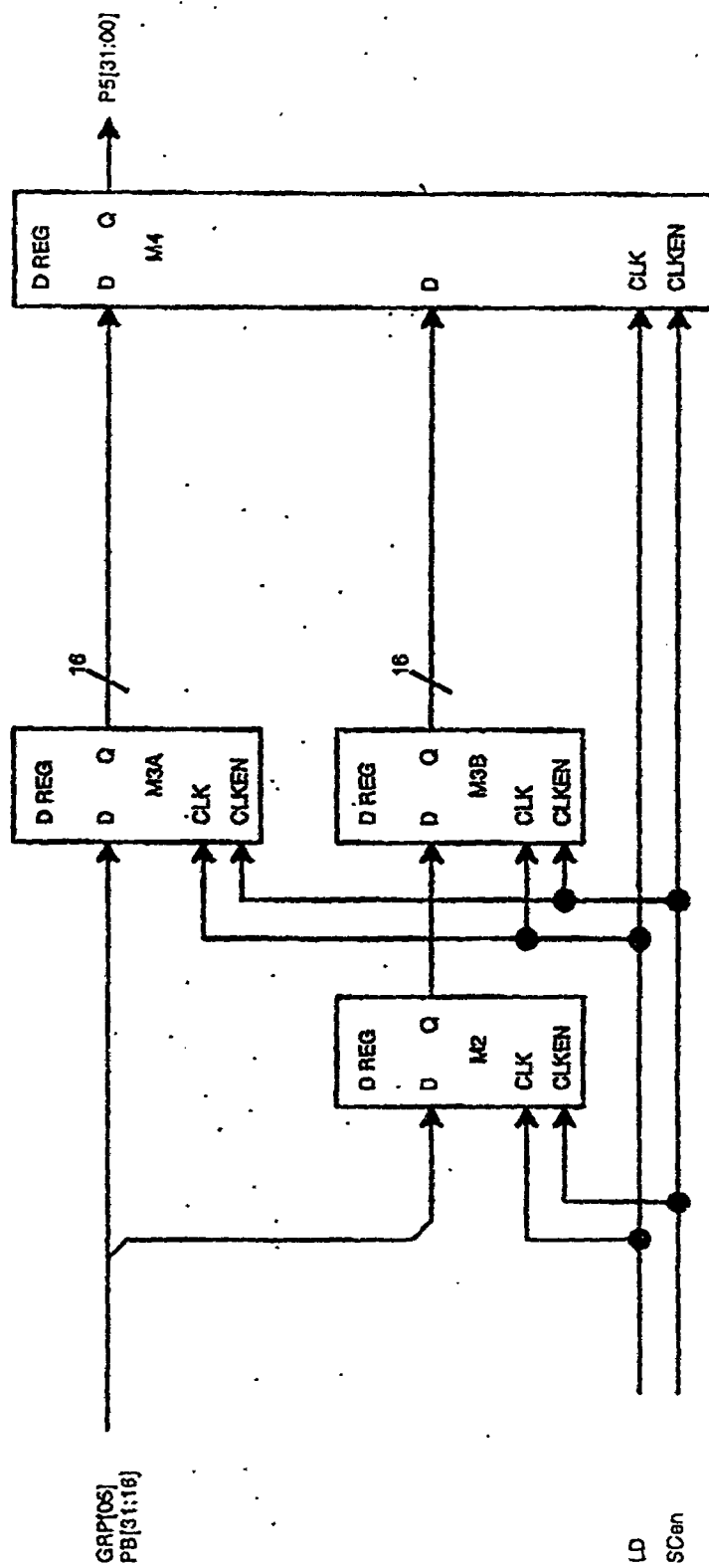
PIXEL PORT - PIXEL 3 INTERLEAVING FORMAT CIRCUIT

Fig. 19



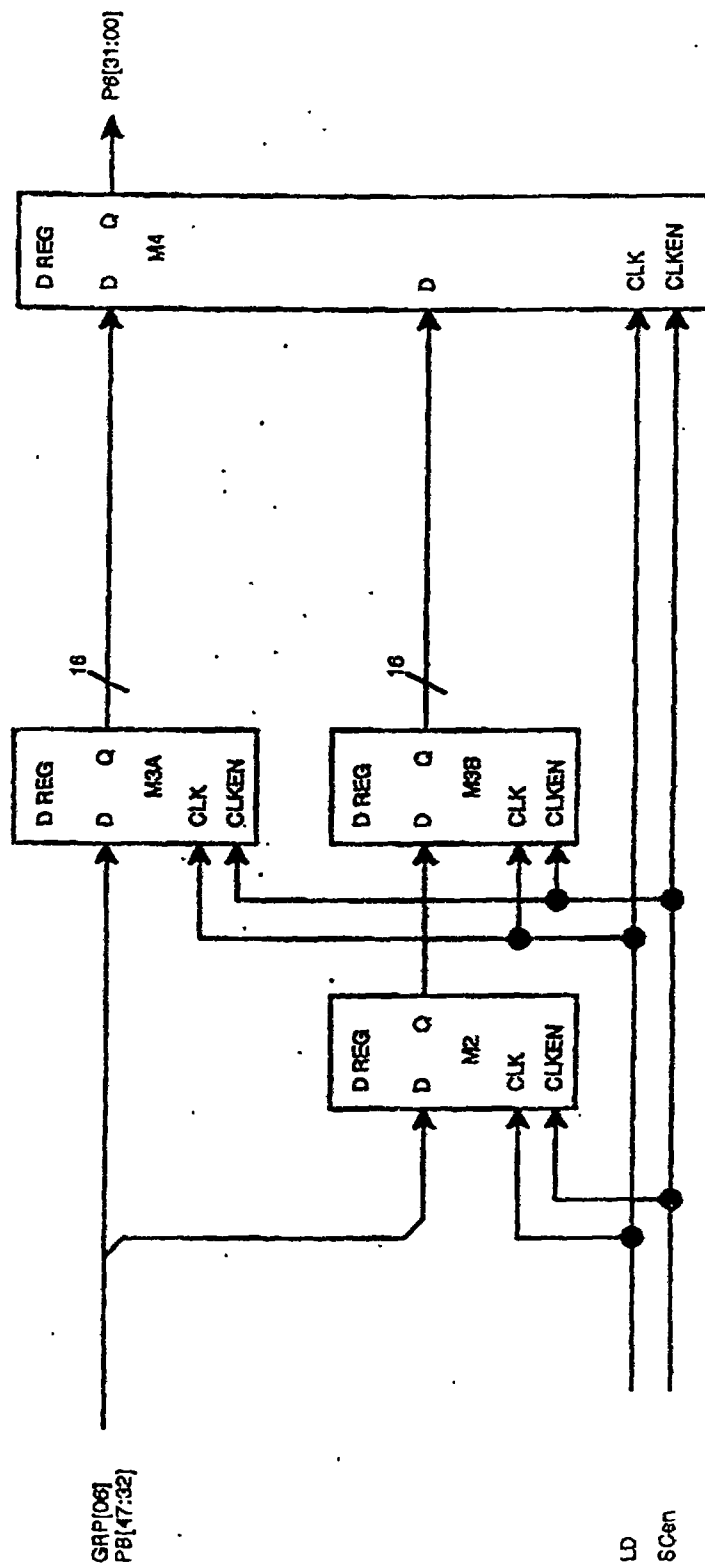
PIXEL PORT -- PIXEL 4 INTERLEAVING FORMAT CIRCUIT

FIG. 18



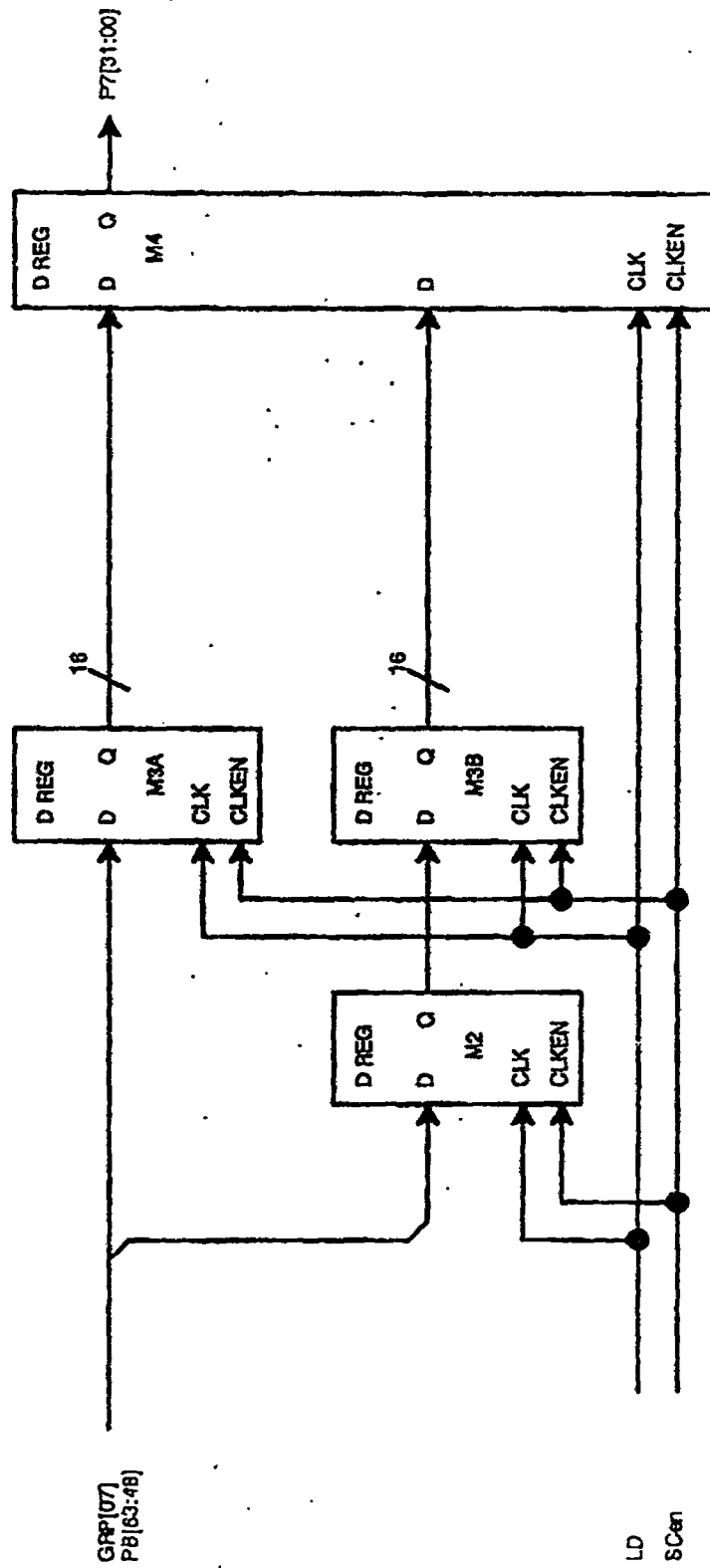
PIXEL PORT - PIXEL 5 INTERLEAVING FORMAT CIRCUIT

FIG. 19



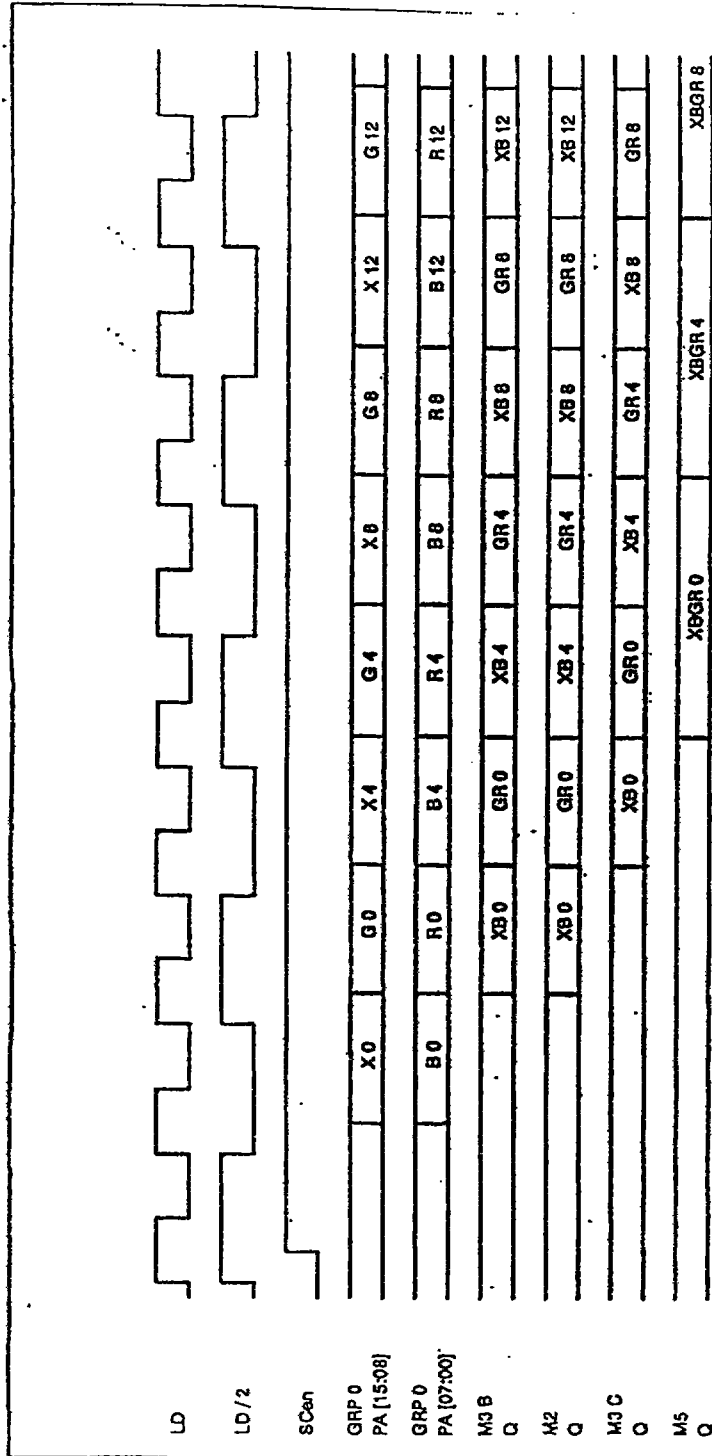
PIXEL PORT - PIXEL 6 INTERLEAVING FORMAT CIRCUIT

F16. 20



PIXEL PORT. - PIXEL 7 INTERLEAVING FORMAT CIRCUIT

FIG. 24



PIXEL PORT - PIXEL 0 FORMAT CIRCUIT TIMING

FIG. 22

OUTPUT PIXEL	2:1 SINGLE BUFFERED	2:1 DOUBLE BUFFERED	4:1 SINGLE BUFFERED	4:2:1 DOUBLE BUFFERED	4:2:1 SINGLE BUFFERED	8:2:1 SINGLE BUFFERED
P7						GRP 7
P6						GRP 6
P5						GRP 5
P4						GRP 4
P3 BUFFER B				GRP 7		
P3 BUFFER A			GRP 7 and 6	GRP 3	GRP 3	GRP 3
P2 BUFFER B				GRP 6		
P2 BUFFER A			GRP 5 and 4	GRP 2	GRP 2	GRP 2
P1 BUFFER B		GRP 7 and 6		GRP 5		
P1 BUFFER A	GRP 3 and 2	GRP 3 and 2	GRP 3 and 2	GRP 1	GRP 1	GRP 1
P0 BUFFER B		GRP 5 and 4		GRP 4		
P0 BUFFER A	GRP 1 and 0	GRP 1 and 0	GRP 1 and 0	GRP 0	GRP 0	GRP 0

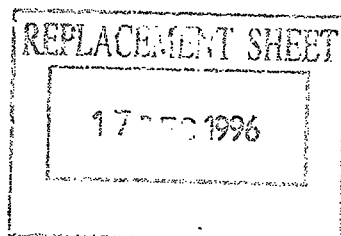
PIXEL PORT - INTERLEAVING FORMAT CIRCUITS ROUTING TABLE

FIG. 23

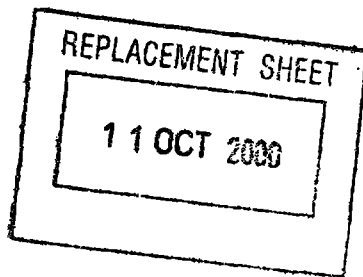
7

several functional blocks as follows: CPU port, interface logic, address pointers and data registers 31, pixel port, pixel input registers and serialization 33, shadow and RAM look-up tables, transfer control and overlay/underlay logic 35, color model selection 37, cursor logic serialization 39, monitor serial port 41, diagnostic registers and control logic 43, digital-analog converters (DAC) 45a-45c and PLL clock synthesizer, pixel clock divider and video timing generator 49. The invention lies mainly in an implementation of the pixel port, pixel input registers and serialization 33 component of the RAMDAC. Therefore, the following description will be limited to the pixel port, pixel input registers and serialization, with information pertaining to the other components of the RAMDAC provided only as needed for an understanding of the present invention. Although the other components shown in Figure 3 may vary between RAMDACs of different manufacturers, persons skilled in the relevant art will recognize these various components and know how they or their equivalents may be implemented

The pixel port is a synchronous input port which accepts interleaved pixel data. Several interleaving formats are required. Selection among these utilizes register programming and is done as part of a boot time configuration process. RAMDAC 21 has two pixel ports, labeled A and B, with a programmable interleaving factor. This configuration accommodates double buffered operation for animation. Here, as in all other cases, the interleaving selection is made during configuration. The selection of port A or port B is



1-00000000



19



CLAIMS

1. An apparatus for multiplexing pixel data from a frame buffer for use by a RAMDAC for display on a display device comprising:

a) interleaving format circuits for coupling to a frame buffer and receiving interleaved pixel data having the form $n/m:1$, where n is a number of whole pixels being transmitted and m is a fraction of the data which forms a whole pixel which is transmitted during a single clock cycle, where $n > m > 1$;

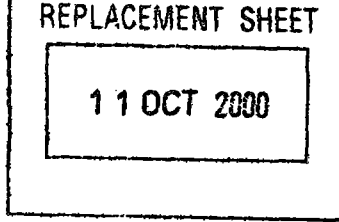
b) logic circuits coupled to the interleaving format circuits for processing predetermined portions of the received interleaved pixel data to produce serialized pixel data for processing by said RAMDAC,

wherein said interleaving format circuits include circuits which undo the interleaving performed by said frame buffer and assemble said pixel data as received from said frame buffer into corresponding sets of pixel data.

2. The apparatus defined by claim 1 wherein said undoing and assembling circuits comprises sets of registers and multiplexors which are coupled together to form a plurality of sub-blocks which cooperatively operate so that each produces a complete pixel formed by a predetermined number of bits.

REPLACEMENT SHEETS	50
2. To replace page 1 to page 1	
✓ 1. To replace page 11 to page 27	
3. To replace page 1 to page 1	
4. To replace page 1 to page 1	
DATE OF RECEIPT	11/10/2000

9606589-1
11/10/2000
AMENDED SHEET



3. The apparatus defined by claim 2 wherein predetermined ones of said sub-blocks receive signals which define the interleaving format performed by said frame buffer.

5

4. The apparatus defined by claim 1 wherein the interleaved pixel data is in one of:

4/2:1 single buffered interleaving format;

4/2:1 double buffered interleaving format;

10

8/2:1 single buffered interleaving format.

C

15

5. The apparatus defined by claim 2 wherein said received pixel data comprises 128 pixel bits divided into two sets containing 64 bits each, the bits in said first set being designated PA(63:00), the bits in said second set being designated PB(63:00), wherein one of said sub-blocks comprises:

a first register which receives bits PA(15:00) of said pixel data;

20

a second register which receives bits PA(31:16) of said pixel data;

a third register which receives bits PA(15:00) of said pixel data;

a fourth register coupled to said first register;

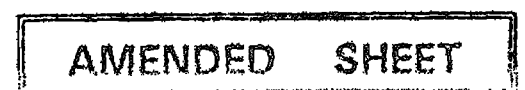
25

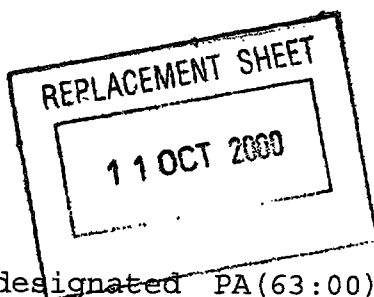
a multiplexor coupled to said second and third registers;

a fifth register coupled to said fourth register and said multiplexor.

30

6. The apparatus defined by claim 2 wherein said received pixel data comprises 128 pixel bits divided into two sets containing 64 bits each, the bits in said first set





21

being designated PA(63:00), the bits in said second set being designated PB(63:00), wherein one of said sub-blocks comprises:

5 a first register which receives bits PA(31:24) of said pixel data;

a second register which receives bits PA(63:32) of said pixel data;

a third register which receives bits PA(31:24) of said pixel data;

10 a fourth register coupled to said first register;

a multiplexor coupled to said second, third and fourth registers;

a fifth register coupled to said multiplexor.

15 7. The apparatus defined by claim 2 wherein said received pixel data comprises 128 pixel bits divided into two sets containing 64 bits each, the bits in said first set being designated PA(63:00), the bits in said second set being designated PB(63:00), wherein one of said sub-blocks
20 comprises:

a first register which receives bits PA(47:32) of said pixel data;

a second register which receives bits PB(31:00) of said pixel data;

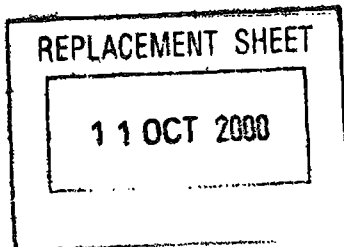
25 a third register which receives bits PA(47:32) of said pixel data;

a fourth register coupled to said first register;

a multiplexor coupled to said second, third and fourth registers;

30 a fifth register coupled to said multiplexor.

AMENDED SHEET



8. The apparatus defined by claim 2 wherein said received pixel data comprises 128 pixel bits divided into two sets containing 64 bits each, the bits in said first set being designated PA(63:00), the bits in said second set being designated PB(63:00), wherein one of said sub-blocks comprises:

a first register which receives bits PA(63:48) of said pixel data;

a second register which receives bits PB(63:32) of said pixel data;

a third register which receives bits PA(63:48) of said pixel data;

a fourth register coupled to said first register;

a multiplexor coupled to said second, third and fourth registers;

a fifth register coupled to said multiplexor.

9. The apparatus defined by claim 2 wherein said received pixel data comprises 128 pixel bits divided into two sets containing 64 bits each, the bits in said first set being designated PA(63:00), the bits in said second set being designated PB(63:00), wherein one of said sub-blocks comprises:

a first register which receives bits PB(15:00) of said pixel data;

a second register which receives bits PB(15:00) of said pixel data;

a third register coupled to said first register;

a fourth register coupled to said second register and said third register.



10. The apparatus defined by claim 2 wherein said received pixel data comprises 128 pixel bits divided into two sets containing 64 bits each, the bits in said first set being designated PA(63:00), the bits in said second set being designated PB(63:00), wherein one of said sub-blocks comprises:

a first register which receives bits PB(31:16) of said pixel data;

a second register which receives bits PB(31:16) of said pixel data;

a third register coupled to said first register;

a fourth register coupled to said second register and said third register.

11. The apparatus defined by claim 2 wherein said received pixel data comprises 128 pixel bits divided into two sets containing 64 bits each, the bits in said first set being designated PA(63:00), the bits in said second set being designated PB(63:00), wherein one of said sub-blocks comprises:

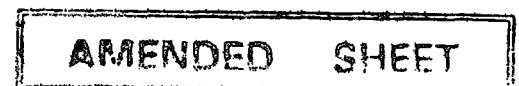
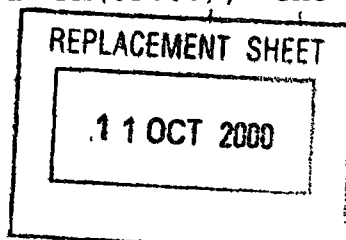
a first register which receives bits PB(47:32) of said pixel data;

a second register which receives bits PB(47:32) of said pixel data;

a third register coupled to said first register;

a fourth register coupled to said second register and said third register.

12. The apparatus defined by claim 2 wherein said received pixel data comprises 128 pixel bits divided into two sets containing 64 bits each, the bits in said first set being designated PA(63:00), the bits in said second set





being designated PB(63:00), wherein one of said sub-blocks comprises:

a first register which receives bits PB(63:48) of said pixel data;

5 a second register which receives bits PB(63:48) of said pixel data;

a third register coupled to said first register;

a fourth register coupled to said second register and said third register.

10

13. The apparatus defined by claim 1 wherein said processing logic circuits comprise:

a first multiplexor coupled to said interleaving format circuits;

15 a pipeline register coupled to said first multiplexor and said interleaving format circuits;

a second multiplexor coupled to pipeline register;

a shift register coupled to said pipeline register and said second multiplexor. register is coupled to pipeline register so that fifth through eighth sets of pixel data input to said shift register are fifth through eighth sets of pixel data output from said pipeline register.

20

25

14. The apparatus defined by claim 13 wherein said interleaving format circuits assemble said pixel data as received from said frame buffer into corresponding sets of pixel data and said first multiplexor is coupled to said interleaving format circuits so that a first set of said pixel data is a first selectable input to said first multiplexor; a second set of said pixel data is a second selectable input to said first multiplexor; a third set of said pixel data is a third selectable input to said first

30

REPLACEMENT SHEET

11 OCT 2000

AMENDED SHEET



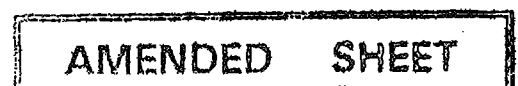
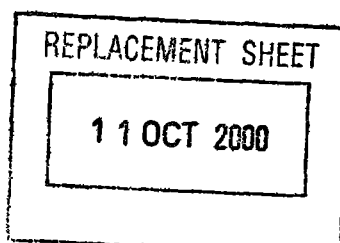
multiplexor; and a fourth set of said pixel data is a fourth selectable input to said first multiplexor.

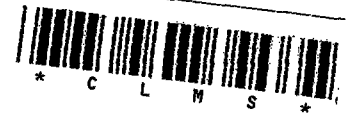
15. The apparatus defined by claim 14 wherein said
5 pipeline register is coupled to said first multiplexor to receive first and second selected outputs from said first multiplexor as first and second inputs and is coupled to said interleaving format circuits to receive:

10 as a third input, said third set of pixel data;
as a fourth input, said fourth set of pixel data;
as a fifth input, a fifth set of pixel data;
as a sixth input, a sixth set of pixel data;
as a seventh input, a seventh set of pixel data;
as an eighth input, an eighth set of pixel data.

15 16. The apparatus defined by claim 15 wherein said second multiplexor is coupled to said pipeline register to receive as a first selectable input, said first through fourth sets of pixel data output from said pipeline
20 register, and as a second selectable input, said fifth through eighth sets of pixel data output from said pipeline register.

25 17. The apparatus defined by claim 16 wherein said shift register is coupled to said multiplexor so that first through fourth sets of pixel data input to said shift register are a selected output of said second multiplexor, and said shift register is coupled to pipeline register so that fifth through eighth sets of pixel data input to said
30 shift register are fifth through eighth sets of pixel data output from said pipeline register.





18. A method for multiplexing pixel data from a frame buffer for use by a RAMDAC for display on a display device comprising the steps of:

5 a) undoing the interleaving performed by said frame buffer and assembling said pixel data as received from said frame buffer into corresponding sets of pixel data, said pixel data having the form $n/m:1$, where n is a number of whole pixels being transmitted and m is a fraction of the data which forms a whole pixel which is transmitted during a
10 single clock cycle, where $n > m$;

b) processing predetermined portions of the received interleaved pixel data to produce serialized pixel data for processing by said RAMDAC,

15 wherein said undoing and assembling produces a plurality of complete pixel sets, each formed by a predetermined number of bits.

19. The method defined by claim 18 wherein said undoing
20 and assembling step produces first through eighth sets of complete pixel data.

20. The method defined by claim 19 wherein said processing step comprises the steps of:

25 first selecting first and second sets of pixel data from among said first through fourth sets of pixel data;

passing said first and second sets of selected pixel data and said third through eighth sets of pixel data;

30 second selecting first through fourth sets of said pixel data from among said first through fourth sets of said pixel data on the one hand and from among said fifth through eighth sets of said pixel data on the other hand;

REPLACEMENT SHEET

11 OCT 2000

AMENDED SHEET



passing said second selected sets of pixel data as first through fourth sets of pixel data and said fifth through eighth sets of pixel data.

