

# InfiniteReality™ Video Format Combiner User's Guide

Document Number 007-3279-001

## CONTRIBUTORS

Written by Carolyn Curtis

Illustrated by Carolyn Curtis

Production by Laura Cooper

Engineering contributions by Gregory Eitzmann, David Naegle, Chase Garfinkle,  
Ashok Yerneni, Rob Wheeler, Ben Garlick, Mark Schwenden, and Ed Hutchins

Cover design and illustration by Rob Aguilar, Rikk Carey, Dean Hodgkinson,  
Erik Lindholm, and Kay Maitz

© 1996, Silicon Graphics, Inc.— All Rights Reserved

The contents of this document may not be copied or duplicated in any form, in whole  
or in part, without the prior written permission of Silicon Graphics, Inc.

## RESTRICTED RIGHTS LEGEND

Use, duplication, or disclosure of the technical data contained in this document by  
the Government is subject to restrictions as set forth in subdivision (c) (1) (ii) of the  
Rights in Technical Data and Computer Software clause at DFARS 52.227-7013  
and/or in similar or successor clauses in the FAR, or in the DOD or NASA FAR  
Supplement. Unpublished rights reserved under the Copyright Laws of the United  
States. Contractor / manufacturer is Silicon Graphics, Inc., 2011 N. Shoreline Blvd.,  
Mountain View, CA 94043-1389.

Silicon Graphics, the Silicon Graphics logo, Onyx, and IRIS are registered trademarks  
and IRIX, Sirius Video, POWER Onyx, and InfiniteReality are trademarks of Silicon  
Graphics, Inc. X Window System is a trademark of Massachusetts Institute of  
Technology.

---

# Contents

**List of Figures** vii

**List of Tables** ix

**About This Guide** xi

Audience xi

Structure of This Guide xi

Conventions xii

- 1. InfiniteReality Graphics and the Video Format Combiner Utility** 1
  - Channel Input and Output 2
  - Video Format Combinations 2
  - Video Format Combiner Utility and the Sirius Video Option 2
  - Programmable Querying of Video Format Combinations 3
- 2. Creating and Setting Video Format Combinations** 5
  - Loading the Video Format Combiner GUI 5
    - Bandwidth Used Scale 7
    - Reduced Fill Scale 7
    - Error Indicator 7
  - Downloading a Video Format Combination 8
  - Specifying Channels 8
    - Selecting a Video Format for a Channel 9
    - Resizing and Repositioning a Channel 12
    - Copying a Channel 12
    - Aligning Channels 13
    - Associating a Video Channel With an X Window 13
  - Modifying Channel Attributes 14
    - Field Layout 15
    - Format 16

- Origin and Size 16
- Pixel Format 16
- Dither 17
- Setup 17
- Tri-Level Sync 17
- Cursor Priority 18
- Channel Placement in the Framebuffer 18
- Horizontal and Vertical Phase 19
- Sync 20
- Gamma 20
- Alpha Channel 20
- Gain 21
- Selecting Channels for a Combination 22
- Changing Global Attributes for Combinations 23
  - Setting Pixel Depth 24
  - Setting Sync Source and Format 26
- Saving a Video Format Combination 26
- Using the Target Hardware Facility 27
- 3. Using the Encoder Channel 29**
  - Setting Up the Encoder Channel 30
    - Independent Mode 30
    - Dependent Mode 30
  - Modifying Encoder Channel Attributes 31
  - Specifying the Filter Width 32
- 4. Using the Combiner With Sirius Video 33**
  - Setting Up the Sirius Video Channel 33
    - Independent Mode 33
    - Dependent Mode 34
  - Modifying Sirius Video Channel Attributes 35

- 5. **Combination Considerations** 37
  - Trading Off Format Parameters in a Combination 37
    - Swap Rate 37
    - Transmission Bandwidth 38
    - DAC Output Bandwidth 39
    - Framebuffer Memory and Read/Write Bandwidth 39
  - Emergency Backup Combination 40
- A. **Error Messages** 41
- Index** 69



---

## List of Figures

<b>Figure 2-1</b>	Video Format Combiner Main Window	6
<b>Figure 2-2</b>	Select Format Window	9
<b>Figure 2-3</b>	Video Format Specified for Channel 0	11
<b>Figure 2-4</b>	Channel Attributes Window	14
<b>Figure 2-5</b>	Stacked Field Layout	15
<b>Figure 2-6</b>	Framebuffer	18
<b>Figure 2-7</b>	Run-Time Panning and Tile Boundaries	19
<b>Figure 2-8</b>	Combination Attributes Window	23
<b>Figure 2-9</b>	Framebuffer and Pixel Depth	25
<b>Figure 2-10</b>	Target Hardware Window	27
<b>Figure 3-1</b>	Encoder, Roaming in Channel 2	29
<b>Figure 3-2</b>	Encoder Channel Attributes Window	31
<b>Figure 3-3</b>	Varying the Width of the Filter	32
<b>Figure 4-1</b>	Sirius Video Channel Attributes Window	35
<b>Figure GI-1</b>	Color Burst and Chrominance Signal	51
<b>Figure GI-2</b>	Component Video Signals	53
<b>Figure GI-3</b>	Horizontal Blanking	57
<b>Figure GI-4</b>	Horizontal Blanking Interval	58
<b>Figure GI-5</b>	Waveform Monitor Readings With and Without Setup	62
<b>Figure GI-6</b>	Red or Blue Signal	64
<b>Figure GI-7</b>	Y or Green Plus Sync Signal	64
<b>Figure GI-8</b>	Video Waveform: Composite Video Signal With Setup (Typical NTSC)	65
<b>Figure GI-9</b>	Video Waveform: Composite Video Signal (Typical PAL)	66



---

## List of Tables

<b>Table 2-1</b>	Resizing and Repositioning	12
------------------	----------------------------	----



---

## About This Guide

The Video Format Combiner is a utility for the InfiniteReality™ video display subsystem for the Silicon Graphics® ONYX® or POWER Onyx™ deskmount or rackmount graphics workstation running IRIX™ 6.2 or later. The display subsystem supports programmable pixel timings to allow the system to drive displays with a wide variety of resolutions, refresh rates, and interlace/non-interlace characteristics.

Each InfiniteReality pipe includes either two or eight video output channels. The Video Format Combiner is the utility that you use to select video output formats for the channels and to group the formats of the individual channels into a combination. You also use the Combiner to define the per-channel and global parameters of the combination.

### Audience

This guide is written for the sophisticated graphics and video user who wishes to develop arrangements of video formats for InfiniteReality capabilities.

### Structure of This Guide

This guide contains the following chapters:

- Chapter 1, “InfiniteReality Graphics and the Video Format Combiner Utility,” introduces the main features of InfiniteReality and of the Combiner graphical user interface.
- Chapter 2, “Creating and Setting Video Format Combinations,” explains how to use Combiner features to design video format combinations by selecting channels and specifying attributes.
- Chapter 3, “Using the Encoder Channel,” describes how to use the Combiner Encoder channel, which encodes its own or another channel’s pixels to NTSC or PAL for industrial-quality video.

- Chapter 4, “Using the Combiner With Sirius Video,” explains how to configure the Sirius Video™ channel that interfaces with an installed Sirius Video option.
- Chapter 5, “Combination Considerations,” gives tips for creating viable video format combinations and explains the Emergency Backup Combination and setting up an analog alpha out channel.
- Appendix A, “Error Messages,” presents Video Format Combiner error messages, which appear near the bottom of the Combiner main window, and suggests solutions.

A glossary and an index complete this guide.

## Conventions

In command syntax descriptions and examples, square brackets ( [ ] ) surrounding an argument indicate an optional argument. Variable parameters are in italics. Replace these variables with the appropriate string or value.

In text descriptions, IRIX filenames are in italics. The names of keyboard keys are printed in typewriter font and enclosed in angle brackets, such as <Enter> or <Esc>.

Messages and prompts that appear on-screen are shown in fixed-width type. Entries that are to be typed exactly as shown are in boldface fixed-width type.

## InfiniteReality Graphics and the Video Format Combiner Utility

The InfiniteReality video display subsystem takes rendered images from the raster subsystem digital framebuffer and processes the pixels through digital-to-analog converters (DACs) to generate an analog pixel stream suitable for display on a high-resolution RGB video monitor. The display subsystem supports programmable pixel timings so that the system can drive displays with a wide variety of resolutions, refresh rates, and interlace/non-interlace characteristics.

The InfiniteReality hardware includes either two or eight independent video channels in the standard configuration. The second video channel serves a dual role as a standard RGB video channel or as a composite or S-Video encoder.

For the InfiniteReality video graphics display subsystem, Silicon Graphics provides a configuration utility, the Video Format Combiner, *ircombine(1G)*. The Combiner groups video output formats into a *video format combination*—descriptions of raster sizes and timing to be used on video outputs—to validate that they operate correctly on the InfiniteReality video display subsystem hardware. The Combiner also configures the underlying graphics framebuffer and instructs the video subsystem to convert digital information stored there into a variety of video signals, or channels, ranging from high-resolution to low-resolution outputs. The output can then be displayed on additional monitors or projection devices, or stored on videotape, in any combination. Output can be genlocked to an external reference signal.

This chapter discusses

- channel input and output
- video output combinations
- Video Format Combiner utility and the Sirius Video option
- programmable querying of video format combinations

**Note:** For more information on *ircombine(1G)*, consult its reference page.

## Channel Input and Output

All channels derive their pixels from areas on the raster (framebuffer area). You can use the Combiner to select any area of the screen, or the contents of any window on the screen, as channel input; channels can overlap. A convenience feature allows you to click a specific window on the screen for channel input. You can also copy one channel's format to another.

InfiniteReality's multichannel feature takes the digital, ordered pixel output from the framebuffer and allows you to specify two to eight separate rectangular areas to be sent from the rectangular framebuffer area managed by the X Window System™ to independent component RGB video outputs. Each video channel can have its own video timing. This capability is particularly useful for applications such as visual simulation, virtual reality, or entertainment.

## Video Format Combinations

After you set up the channels, you can save them as a video format combination. You can download a video format combination into the hardware as the current video configuration, store it as the default configuration to be used at system power-on or graphics initialization, or save it in a video format combination file. You can create a combination from scratch or by modifying a current or default combination or a previously saved combination. The Combiner software includes sample combinations.

Video output combinations include "combinations" of a single video output format.

## Video Format Combiner Utility and the Sirius Video Option

The Sirius Video option is available for applications that require live video input. The Sirius Video board provides two inputs that can be blended using incoming alpha-key blending or internal chroma-key generation, or by using a key generated in real time from InfiniteReality graphics.

If the Sirius Video option is installed, you can use an additional, dedicated channel in the Video Format Combiner interface for its output. You can configure this Sirius Video channel to use a high-resolution channel as input, and set it up to pass pixels through or filter them, and set other options as well.

## **Programmable Querying of Video Format Combinations**

You can determine the contents of the currently running combination by using the *XSGlvc* X-server extension library. This library permits query of the individual formats and their origins on the framebuffer.



---

## Creating and Setting Video Format Combinations

This chapter explains how to use the Video Format Combiner GUI to create and set video format combinations:

- loading the Video Format Combiner GUI
- downloading a video format combination
- specifying channels
- modifying channel attributes
- selecting channels for a combination
- changing global attributes for combinations
- saving a video format combination
- using the target hardware facility

### Loading the Video Format Combiner GUI

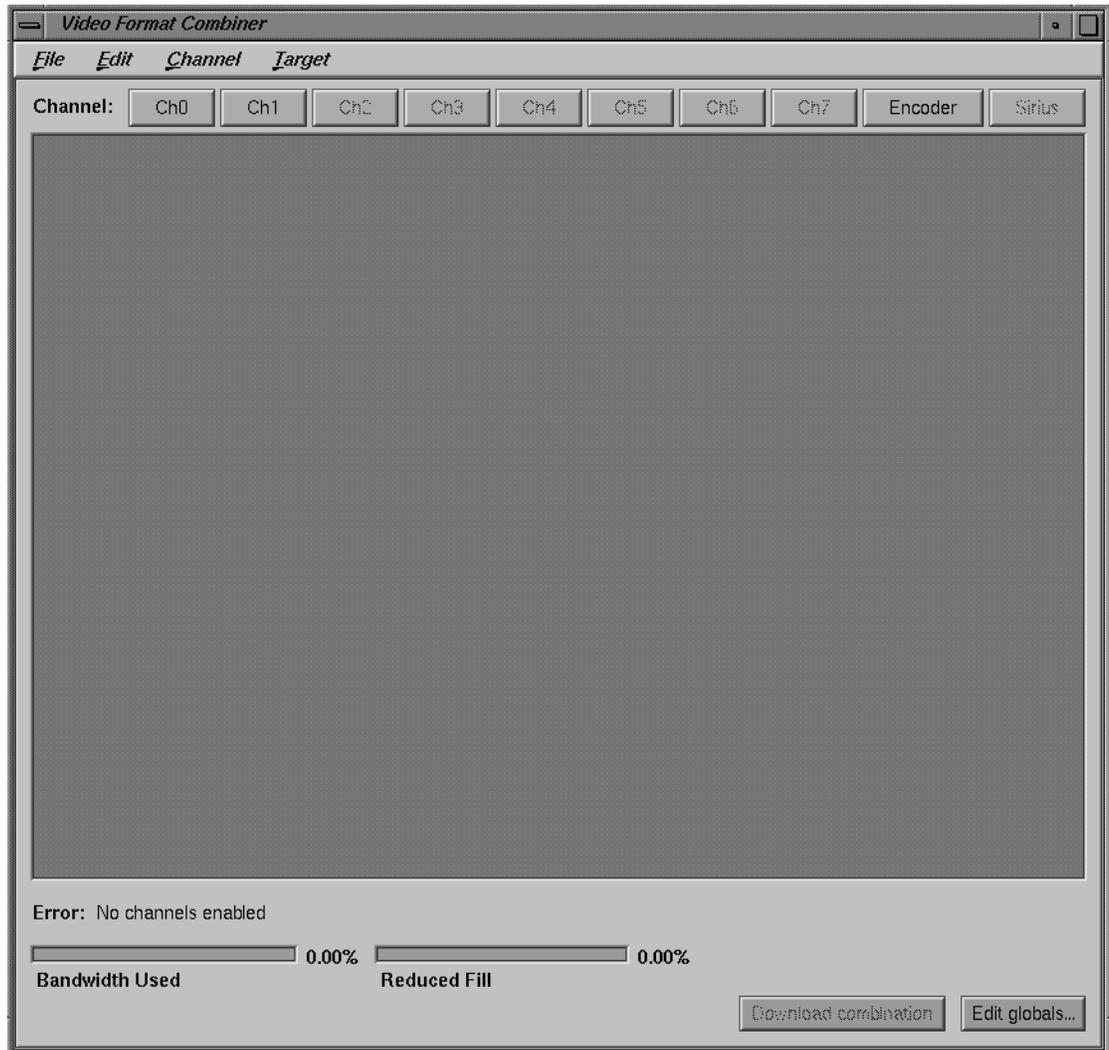
To display the Combiner's graphical user interface, type

```
/usr/gfx/ircombine -gui
```

To specify a different display from the current workstation, such as a different pipe of a rackmount system, a remote workstation, or a specific pipe of a remote rackmount system, as the target on which to display the combination, use

```
/usr/gfx/ircombine -gui -target displayname
```

Figure 2-1 shows the main window.



**Figure 2-1** Video Format Combiner Main Window

The *managed area* is the display surface taking up most of the Combiner main window. It represents the entire raster (framebuffer area); all channels derive their pixels from areas on the raster.

## Bandwidth Used Scale

The Bandwidth Used scale indicates the percentage of the communication bandwidth between the framebuffer and the video subsystem that the combination is using. In general, as long as the bandwidth used is less than 100%, things are fine. However, the bandwidth used is related to the framebuffer memory used to refresh the video channels, which the Reduced Fill scale shows.

Things that can affect the Bandwidth Used scale are

- the size of each channel (its footprint on the framebuffer, not its output resolution, although the two are related)
- the Pixel Output Format selected in the channel attribute window; to save bandwidth, avoid choosing 10-bit components and alpha
- the number of video channels
- setting up the Encoder channel in independent mode; dependent mode does not use any extra bandwidth or fill rate

Adding RM6 boards does not increase the video bandwidth available in the system.

## Reduced Fill Scale

The Reduced Fill scale indicates the percentage of framebuffer memory cycles used for refreshing the video channels of the combination. In general, reduce the pixel-fill specifications of the framebuffer by this percentage. If the application still meets its spec for depth complexity, number of polygons, and so on when the fill rate of the framebuffer is reduced by the percentage indicated by the Reduced Fill scale, things are fine.

Adding RM6 boards amortizes the video-refresh overhead over more boards, so it helps to add RM6 boards if the video refresh overhead is too high for a particular hardware configuration and application. The Reduced Fill scale is affected by the same factors as the Bandwidth Used scale.

## Error Indicator

If system constraints are violated, the Error indicator displays a message; see Appendix A, "Error Messages." The Error indicator should show <none> for a video format combination to be valid.

## Downloading a Video Format Combination

The Combiner software includes sample video format combinations in */usr/gfx/ucode/KONA/dg4/cmb*. To load one of these or any other combination, use one of the following:

- press *Download Combination*
- press **<Ctrl-d>**
- type **setmon -n filename**
- type **ircombine -source file filename -dest active**
- from a program, call **XSGIvcLoadVideoFormatCombination()**

As you build your own combination, you can test it by saving it, downloading it, adjusting, saving and downloading it again, until you are satisfied with it.

## Specifying Channels

When you initialize InfiniteReality graphics, or power on the system with InfiniteReality graphics installed, you can load a previously saved combination, as described above, or use the Combiner to define the video output for the Onyx InfiniteReality workstation, regardless of the number of channels you use.

To initialize graphics, type the following; the parentheses are necessary.

```
(/usr/gfx/stopgfx ; /usr/gfx/startgfx) &
```

This section explains

- selecting a video format for a channel
- resizing and repositioning a channel
- copying a channel
- aligning channels
- redefining a channel
- associating a channel with an X window

## Selecting a Video Format for a Channel

In the Combiner main window, click the button for the channel you want to define; for example, *Ch0* for the first channel. This selection corresponds to **Chan0** on the InfiniteReality I/O panel.

When you click a channel button, the Select Format window appears, as shown in Figure 2-2; select a video format for the channel.

**Note:** For information on the video formats stored in the files, consult */usr/gfx/ucode/KONA/dg4/vfo/README*.



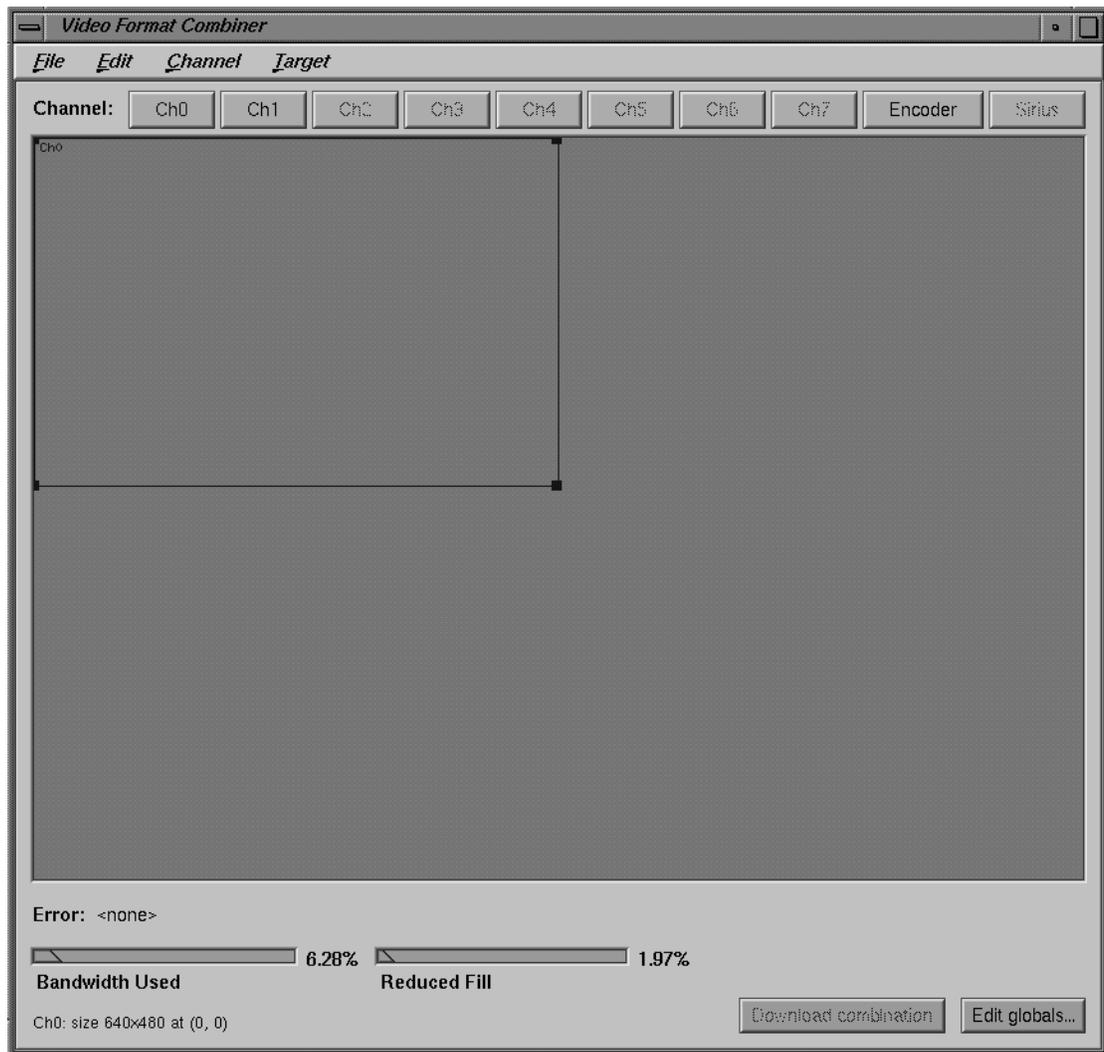
**Figure 2-2** Select Format Window

**Note:** If your system includes channels 2 through 7, assign the formats with the highest bandwidth to Channels 0 and 1, because they have greater capability. For more information, see “DAC Output Bandwidth” on page 39.

You can easily resize the rectangle with the mouse or by editing channel attributes, as explained in “Resizing and Repositioning a Channel” on page 11. The dimensions of the format (for example, 1920 × 1200 for the format file *1920x1200\_75.vfo*) are the output dimensions, but need not be the input dimensions of the channel’s rectangle in the managed area. You can also use the Video Format Compiler to create your own formats.

Figure 2-3 shows the Combiner main window with Channel 0 defined. The channel's rectangle appears in the managed area.

Notice that the Bandwidth Used scale indicates that some of the video subsystem is now in use.



**Figure 2-3** Video Format Specified for Channel 0

The channel rectangle shows the area on the raster from which Channel 0 derives its pixels.

## Resizing and Repositioning a Channel

To change the size and position of the channel in the managed area, use the mouse to move the channel rectangle. Dragging the rectangle changes the channel's origin; dragging one of the handles at a corner resizes the channel. You can also use h, j, k, or l as arrow keys to move the sides of the rectangle, as summarized in Table 2-1.

**Table 2-1** Resizing and Repositioning

Key	Action
h	Move channel rectangle to the left
l	Move channel rectangle to the right
j	Move channel rectangle down
k	Move channel rectangle up
<Ctrl-h>	Decreases channel rectangle's width
<Ctrl-l>	Increases channel rectangle's width
<Ctrl-j>	Increases channel rectangle's height
<Ctrl-k>	Decreases channel rectangle's height

For pixel-accurate positioning and sizing, edit the channel's origin and size, as explained in "Origin and Size" on page 15.

## Copying a Channel

To copy an existing channel format—size, origin, and other characteristics—to a new channel, follow these steps;

1. Click the button of the new channel (for example, *Ch3*); in the Files window, select any video format.
2. Select the rectangle of the channel you want to copy (for example, *Ch0*). In the Edit menu, select "Copy," or press <Ctrl-c>.

3. Select the rectangle of the new channel. Press **<Ctrl-v>**, or select "Paste" in the Edit menu.

To create a new channel that contains all the attributes of an existing channel, just follow steps 2 and 3 above. The next unassigned channel in sequence is assigned; for example, if you are copying Channel 0 and the remaining channels on the system are undefined, the Combiner assigns its format to Channel 1. The new channel appears at the same origin as the copied channel; move it if desired.

### **Aligning Channels**

To align channels exactly, edit their origin and size, as explained in "Origin and Size" on page 15. To use a convenience feature of Combiner to expedite this procedure, follow these steps:

1. In the managed area, move the channel rectangles to the approximate positions where you want them.
2. Select a channel rectangle; select "Snap to neighbor" in the Channel menu. The channel corners align with those of the closest neighboring channel.
3. If necessary, refine the channel's origin and size settings, as explained in "Origin and Size."

### **Associating a Video Channel With an X Window**

The Combiner includes a convenience feature for associating a video channel with an open X window. You can position and size the video channel's rectangle in the managed area to correspond to the X window's position. To use an X-server window as input for the original and size a channel, follow these steps:

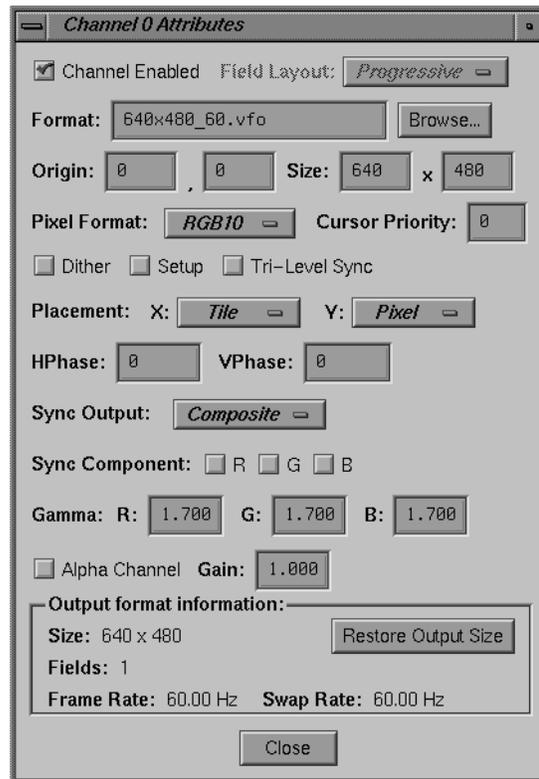
1. Make sure the window you want to use as input for the channel is at the location you want on the screen.
2. In the Combiner main window, select the rectangle of the channel whose input you want to define.
3. Select "Grab window..." in the Channel menu. A crosshatch cursor appears.
4. Move the crosshatch cursor to the window you want to use as input for the channel and click the left or right mouse button.

The Combiner uses that X window's origin and size as the position of the channel.

## Modifying Channel Attributes

Select the channel for which you want to view or modify attributes, and select "Edit attributes..." in the Channel menu. (You can also double-click on the channel's button or rectangle.) The channel window appears; Figure 2-4 shows an example.

**Note:** You can also access the Channel menu by pressing the right mouse button.



**Figure 2-4** Channel Attributes Window

A box at the bottom of this window displays format information stored in the video output file selected for this channel. If you resize the rectangle, click *Restore Output Size*

to restore it to the size specified in the output file. The output pixel format is unrelated to the underlying framebuffer organization, because the InfiniteReality hardware automatically converts to the output pixel format you request.

**Note:** For a combination to be valid, the swap rates of all channels, including the Sirius Video channel, must be the same. The swap rate is shown in near the bottom of the Channel Attributes window. As you add channels to the managed area to create a combination, keep track of the swap rates. For more information on the swap rate, see “Swap Rate” in Chapter 5.

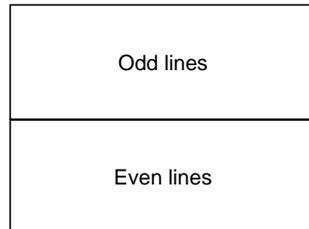
This section explains how to use fields in this window.

### Field Layout

The field layouts are designed to allow you to reduce fill, rendering only the visible lines of each field of an interlaced format. The Combiner attempts to set the correct field layout for each video format automatically.

Use the Field Layout popup menu to select the order in which data is fetched (scanned) from the framebuffer:

- Progressive: specifies sequential data fetching: each field draws from the entire source region of the channel. This setting is the default for formats with one field.
- Interleaved: specifies a two-field format: the lines of the fields are vertically interleaved in the framebuffer.
- Stacked: specifies a two-field format. The lines of each field are contiguous; the lines of the first field occupy the top half of the framebuffer, and the lines of the second field occupy the bottom half, as diagrammed in Figure 2-5. Use this choice to render the even and odd fields separately for a pixel-fill advantage.



**Figure 2-5** Stacked Field Layout

### Format

The text field displays the name of the file containing the video format on this channel. To change the currently selected format for the channel, type a different *.vfo* filename in the text field, or click *Browse* and select a different one in the popup menu.

To change the video format for a channel, follow these steps:

1. Select the channel rectangle.
2. Press the right mouse button; in the menu that appears, select "New channel." The Select Format window appears.
3. In the Select Format scrolling list, choose another video output format.

### Origin and Size

The Origin and Size fields display values for the current video format in the window; these settings define the mapping, or footprint, of the channel in the framebuffer managed area. You can change these values here as well as by manipulating the rectangle with the mouse.

## Pixel Format

Use the choices in the Pixel Format popup menu to select the output pixel format for this channel:

- RGB10: 10-bit RGB with no alpha (this setting is the default); use this setting to save bandwidth when alpha is not required
- RGBA10: 10-bit RGB with alpha

Only the Sirius Video option is capable of outputting this format.

If this format is selected for a channel other than Sirius Video, the channel outputs 10-bit RGB. However, if alpha is enabled (see “Alpha Channel” on page 20), 10-bit alpha is output as blue. This setting is useful for outputting alpha on a dedicated channel.

- RGB12: 12-bit RGB with no alpha
- Z: 24-bit Z component

Each pixel’s z-buffer values are sent instead of its RGB values. This format is useful for “virtual set” applications. A pair of video channels is mapped to the same region of the framebuffer. One channel is set up to get RGB, and the second channel is set up to get Z. The Z component is used as a mask for putting live actors “into” the virtual set.

- FS: color field sequential for special monitors that expect color fields in a particular sequence

## Dither

Check this box to dither the output of this channel by default.

Because the pixel output format might be different from the framebuffer color storage format, the InfiniteReality subsystem must know which method to use to adapt the framebuffer contents to the output format. For example, the framebuffer might contain

12-bit RGB although you have selected the Pixel Format RGB10 to send to the video subsystem. If Dither is not selected, the system truncates the colors. Selecting Dither causes the system to add a fixed pattern of noise to the colors before it truncates them.

### **Setup**

Check this box to enable pedestal (an artificial black level voltage) for this output (disabled by default). Make sure this box is unchecked (setup disabled) for PAL and for Japanese NTSC.

Setup (also called pedestal) is typically used in NTSC and is typically not used in PAL. Setup is the difference between the blanking level and blackest level displayed on the monitor. A black level that is elevated to 7.5 IRE instead of being left at 0.0 IRE is the same as the lowest level for active video. Because the video level is known, this part of the signal is used for black-level clamping circuit operation.

### **Tri-Level Sync**

Check this box to enable trilevel sync for separate RGB ports. This selection is useful for enabling trilevel sync for HDTV monitors that use trilevel sync.

### **Cursor Priority**

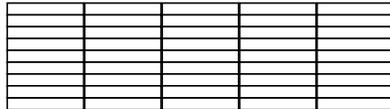
The cursor is visible on only one channel at a time. If your video format combination includes overlapping rectangles, you can set cursor priority to determine which channel displays the cursor when it enters a region where the channels overlap.

To set cursor priority for a channel, enter a numeral between 0 to 255, with lower numbers indicating higher priority.

If you set the same priority number for more than one channel in the combination, the channel that last displayed the cursor retains it. If the cursor did not previously appear in either of the contending channels, the channel with the lower channel number displays of the cursor.

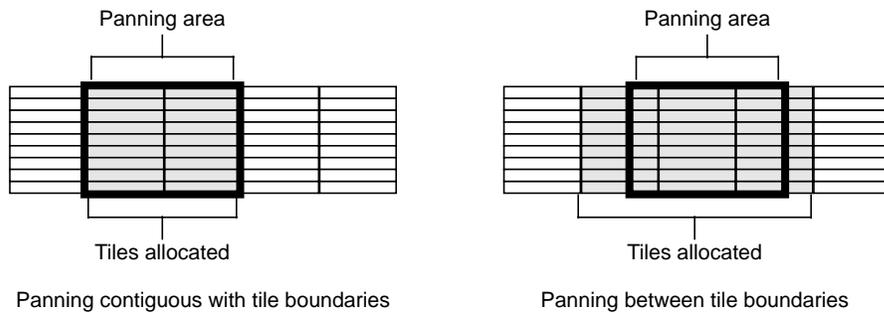
### Channel Placement in the Framebuffer

The framebuffer is divided into individually addressable tiles 160 pixels wide and eight pixels high; see Figure 2-6. With each line, entire tiles in the horizontal direction are transferred, not portions of them.



**Figure 2-6** Framebuffer

If panning is locked to tile boundaries, less bandwidth is used (as shown in the bandwidth scale in the lower left of the Combiner main window). If you choose to pan on a per-pixel basis, the Combiner must account for an extra tile in the horizontal direction. Because more tiles are transferred for every line, the result is some wasted bandwidth; Figure 2-7 diagrams this contrast. During run time, smooth per-pixel panning costs extra bandwidth as well; see the discussion in “Changing Global Attributes for Combinations” on page 22.



**Figure 2-7** Run-Time Panning and Tile Boundaries

The choices in the X and Y popup menus at Placement specify the relationship between the panning area and the number of tiles transferred:

- Locked: prevents the user from panning in this direction during run time
- Pixel: framebuffer use is on a per-pixel basis, allowing run-time panning that is not aligned to tile boundaries (this setting is the default for the Y direction)
- Tile (X direction only): allocates framebuffer use by entire tile and restricts run-time panning to tile boundaries as well (this setting is the default for the X direction)

### Horizontal and Vertical Phase

The horizontal and vertical phase settings apply only if the channel is genlocked to external sync. The settings line up the pixels in two video streams.

- Use the HPhase field to enter a floating point number as the default horizontal phase value for this channel. Positive values delay the genlocked video channel with respect to external sync, and negative values advance the genlocked video channel with respect to sync. The default is 0.
- Use the VPhase field to enter the default vertical phase value for this channel. This value ranges from 0 to the number of lines in the video format minus one. The default value is 0.

### Sync

In addition to composite sync on green, InfiniteReality provides separate horizontal and vertical video signals for compatibility with a wide variety of video equipment such as projectors and recorders.

Make a choice in the Sync Output popup menu to set the output for the alternate sync port:

- Horizontal: put horizontal sync on the **H/C Sync** port
- Composite: put composite sync on the **H/C Sync** port (this setting is the default)

Use the checkboxes at Sync Component to specify whether the R, G, and B components have sync enabled by default.

## Gamma

The brightness characteristic of a monitor's CRT is not linear with respect to voltage; the voltage-to-intensity characteristic is usually close to a power of 2.2. If left uncorrected, the resulting display has too much contrast; detail in black regions is not reproduced. To correct this inconsistency, a gamma value (correction factor) using the 2.2 root of the input signal is included, so that equal steps of brightness or intensity at the input are reproduced with equal steps of intensity at the display.

If you want, change the output video gamma value for this channel for the red, blue, or green components from the factory setting of 1.700.

Although you can assign gamma separately for each channel, the Combiner uses only the gamma map for the last channel loaded. The gamma for all channels derives from the same gamma map, gamma map 0.

## Alpha Channel

Check the checkbox at Alpha Channel to produce the alpha component of the input channel instead of RGB components from the framebuffer. For this setting to be in effect, the RGBA10 pixel format must be selected, as explained in "Pixel Format" on page 16.

When you check this box, gamma correction is disabled for this channel; alpha is never gamma-corrected.

**Note:** For the Sirius Video option, leave the Alpha Channel checkbox unchecked.

Most modern video postproduction equipment, such as Sirius Video option, uses alpha in digital form. For compatibility with older equipment that requires analog alpha, you can use a second channel as the analog alpha output for a channel.

To set up a dedicated channel for alpha out, follow these steps:

1. On the equipment you are using to output analog alpha, hook up to the blue component out; on the alpha channel, the blue component is replaced with 10-bit alpha.
2. Make sure both channels are running the same video format and are positioned and sized identically in the framebuffer managed area. (If you are setting up the channels at this time, you can use the Copy feature as explained in "Copying a Channel" on page 11.)

3. In the Channel Attributes window of the channel you want to use for analog alpha, make sure the Alpha Channel checkbox is checked to enable alpha. Make sure it is unchecked in the other channel's Channel Attributes window.
4. In the Pixel Format popup menu, select the format RGBA10 for each channel.

### Gain

In the text field at Gain, you can change the default output video gain value for this channel from 1.0 to a value between 0.0 to 10.0. The system software forces the value you enter to the closest gain that is physically realizable on the output channel.

Normally, gain is set to a midrange level, such as 6.0. At this value, peak white equals 700 mV.

## Selecting Channels for a Combination

When you save a combination, the video formats on all channels defined in the Combiner main window are included in the combination by default.

To delete a channel from a combination, select its rectangle in the Combiner main window and press <Delete> or <Ctrl-x>.

For test purposes, you can keep the definition of the channel in the combination while disabling the channel output (and thus saving the bandwidth that the channel would consume). To do so, make sure the Channel Enabled checkbox is unchecked (disabled).

For information on setting cursor priority for overlapping channels, see "Cursor Priority" on page 17.

For the combination to be valid, check the following:

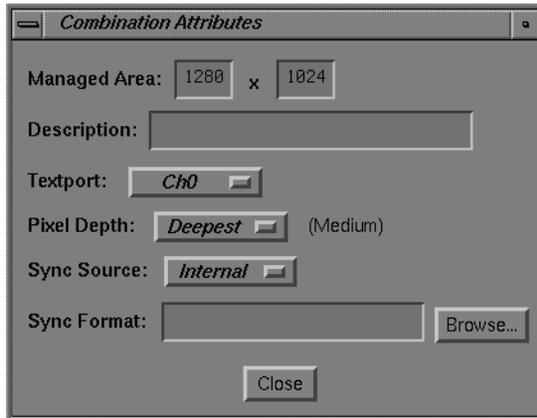
1. Make sure that the swap rates of all channels match, including the Sirius Video channel. The swap rate is shown in each Channel Attributes window. For more information on the swap rate, see "Swap Rate" in Chapter 5.
2. Make sure that the Error indicator in the main Combiner window indicates <none>.

**Note:** The Combiner can write invalid combinations so that they can be loaded and worked on in another session.

You can use the *setmon* utility to download saved combinations, as indicated in “Downloading a Video Format Combination” on page 8; however, *setmon* rejects invalid combinations. See the *setmon* reference page for details.

## Changing Global Attributes for Combinations

To change attributes such as pixel depth, sync source, and sync format for all combinations at once, click *Edit Globals*. Figure 2-8 shows the Combination Attributes window.



**Figure 2-8** Combination Attributes Window

The numerals displayed in the Managed Area fields show the dimensions of the framebuffer, which is managed by the X Windows server. Note that allocation of the framebuffer is on a tile basis; if you allocate the framebuffer in units other than whole tiles (160-by-8-pixel boundaries), pixels are inaccessible.

If the video output format has a width of 1280 pixels, exactly eight tiles ( $160 \times 8 = 1280$ ) are allocated. If you specify a width of 1281 pixels in the Size field or with the mouse, the Combiner still transfers nine tiles, though the ninth tile is barely used.

The Combiner always shows the allocated area to make it clear that memory is being made inaccessible if the managed area size differs from that of the allocated area. When allocation and managed area size differ, a vertical or horizontal red line (or both) appears

in the Combiner window dividing the managed area (left of or above the line) from the unusable portion of the allocated area (right of or below the line).

In general, it is best to put the managed area on the channel boundaries, even if doing so makes part of a tile inaccessible. The X Window system never moves the cursor outside the managed area, so making managed area follow the borders of the video channels makes it harder to lose the cursor by moving it “off-screen.”

When you save the formats as a video output combination, the extra bandwidth that the Combiner allocated (see “Channel Placement in the Framebuffer” on page 18) is accounted for, even though the channel might be placed on a tile boundary at that time. The Combiner accounts in advance for the worst-case bandwidth that can occur at run time when the user smoothly pans the channel.

To add a description of up to 256 characters to the file that stores the combination, enter it in the Description text field.

In the Textport popup menu, select the channel to be used as the textport output. This channel displays boot information when the workstation is restarted.

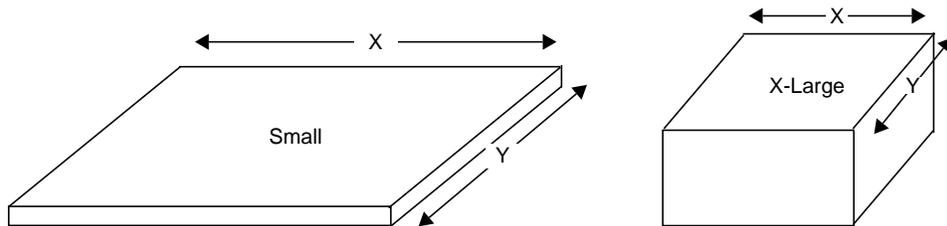
This section explains

- setting pixel depth
- setting sync source and format

## Setting Pixel Depth

Pixel depth specifies the framebuffer pixel depth that this combination allocates when graphics are restarted with this combination (the combination has been saved to the EEPROM), either from a system boot or manually with (*/usr/gfx/stopgfx ; /usr/gfx/startgfx*)&.

Use the Pixel Depth popup menu choices to set the arrangement of pixels in the framebuffer. The amount of memory in the framebuffer is constant, but its arrangement into pixels is flexible. In this context, the framebuffer can be thought of as a mass of clay. The total mass is fixed, but its arrangement can vary: it can be stretched out wide and broad, but shallow, or formed to be narrow, tall, and deep, as diagrammed in Figure 2-9.



**Figure 2-9** Framebuffer and Pixel Depth

- Small, Medium, Large, and X-Large: set pixel depth as indicated by these names and allocate the framebuffer accordingly. Use this choice if pixel depth is more important to your application than the size of the managed area.
- Deepest: instructs the Combiner to select the deepest possible setting for this managed area on this hardware configuration, where X-Large is the deepest, followed by Large, then Medium, and finally Small. Use this choice if the size of the managed area is more important to your application than pixel depth. This setting is the default.

For example, if you set pixel depth to Small, less framebuffer memory is used, but the shallow pixel depth results in fewer bits per pixel. Setting pixel depth to X-Large results in more bits per pixel, at the expense of the X,Y dimensions of the managed area. The X-Large setting provides richer information per pixel, which is useful for multisampling or other applications, and also provides more depth for stencil and z-buffering.

Deeper pixels cost a slight premium in rendering bandwidth and video bandwidth. Although framebuffer pixel depth has less impact on video bandwidth than the output pixel format (RGB10, RGB12, and so on), if speed is a concern, you can gain a slight advantage by selecting Small. In most cases, pixel depth is determined by the rendering features required, such as antialiasing, precision of the Z buffer, stencil buffer, and so on), rather than by video bandwidth considerations.

If you use one of these depth settings and then change the combination without restarting graphics, the depth setting is not guaranteed, because graphics must restart to accommodate the change in depth. In this case, the system software converts the setting to the current pixel depth without notice.

Also, if the system configuration does not support your choice when you restart graphics (for example, not enough RM6 boards), the configuration fails and the system software

uses the Emergency Backup Combination (see “Emergency Backup Combination” on page 40). The actual pixel depth setting appears in parentheses beside the Pixel Depth popup menu; in the example in Figure 2-8, this value is Small.

For more information on setting pixel depth, see “Framebuffer Memory and Read/Write Bandwidth” on page 39.

### Setting Sync Source and Format

At Sync Source, select the InfiniteReality internal sync (Internal) or an external source that is connected to the **Genlock In** port.

In the Sync Format text field, enter the name of the video format file that describes the format to which to lock the video subsystem. If you do not specify an input genlock format, the video format of the combination’s lowest-numbered channel with a valid sync signal is used. Click *Browse* to select a sync format contained in a file. You can create your own sync format file using the Video Format Compiler.

## Saving a Video Format Combination

Using “Save” in the File menu saves the combination. If you have saved the combination previously, use “Save as...” to save the combination under a different filename.

To save a combination as the default file that is loaded the next time you initialize graphics or power on the system, save it to the EEPROM; select “Save to EEPROM...” in the File menu or enter `setmon -x filename`.

The combination saved to the EEPROM does not take effect immediately, but is used the next time graphics is initialized, the system is powered on, or *ircombine* is run with the EEPROM as source and the current configuration as destination.

To add a description (256 characters or less) of the combination to the file, use the Description field in the Combination Attributes window, as explained in “Changing Global Attributes for Combinations” on page 22.

As you build your own combination, you can test it by saving it, downloading it, adjusting, saving and downloading it again, until you are satisfied with it. To “test” it on different equipment, follow instructions in “Using the Target Hardware Facility.”

**Note:** To save the combination in the directory `/usr/gfx/ucode/KONA/dg4/cmb`, you must be running `ircombine` as superuser, or the system administrator must change permissions on that directory to allow writing to it.

### Using the Target Hardware Facility

The Combiner includes a utility that lets you validate the combination using a hardware configuration you design. For example, if an error message appears indicating that the combination is too large for the hardware, you can use this Combiner facility to define a hardware configuration with more RM6 boards for test purposes.

This feature also allows you to run `ircombine` on other Silicon Graphics hardware to select the InfiniteReality hardware configuration most appropriate for a particular application. You can create valid combinations this way, and later download them with `ircombine` or `setmon` onto InfiniteReality hardware configurations that meet or exceed the specifications you selected in the Target Hardware window.

In the Target menu, select “Edit user-defined hardware...” The Target Hardware window appears, as shown in Figure 2-10.



Figure 2-10 Target Hardware Window

You can choose

- number of raster managers (RM6 boards)
- number of channels: 2 or 8
- Option Board: select Sirius Video, if desired

Click *Close* to apply the settings to the combination.



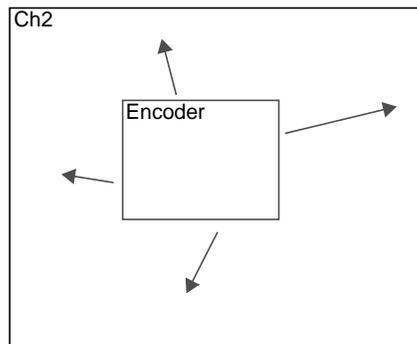
---

## Using the Encoder Channel

The Encoder channel, which can take the place of Channel 1, encodes its own or another channel's pixels to NTSC or PAL television standard for industrial-quality video out via the RCA and BNC connectors for NTSC and PAL or the S-Video connector on the InfiniteReality I/O panel.

The Encoder channel can encompass an entire video channel, allowing whole-screen recording without an external scan converter. For video formats with pixel clock rates above 120 MHz, the Encoder works in pass-through mode, allowing you to record a pannable NTSC- or PAL-sized region of the display.

Alternatively, to save bandwidth, the Encoder channel can take a source channel's pixels and encode them. The Encoder channel samples the source channel's pixels at a point in the channel rectangle that you determine and encodes them to the standard you choose. Figure 3-1 diagrams a roaming Encoder whose source is Channel 2.



**Figure 3-1** Encoder, Roaming in Channel 2

Flexible, built-in video resampling allows the Encoder to process any rectangular area of any video channel, up to and including the full screen. Also, applications can be recorded even if they were not written with NTSC or PAL resolution in mind. The Encoder handles

without distortion video formats that have aspect ratios other than the 3:4 aspect ratio of NTSC or PAL.

This chapter explains

- setting up the Encoder channel
- modifying Encoder channel attributes
- specifying the filter width

## Setting Up the Encoder Channel

The Encoder channel can be set up in independent mode or dependent mode.

### Independent Mode

In the Encoder Attributes window, select None in the Source Channel popup menu to set up the Encoder channel in independent mode. In this mode, pixels coming to the Encoder channel are 10 bits.

Like other channels, the Encoder in independent mode consumes pixel bandwidth. The Combiner validates the selection along with the other channels.

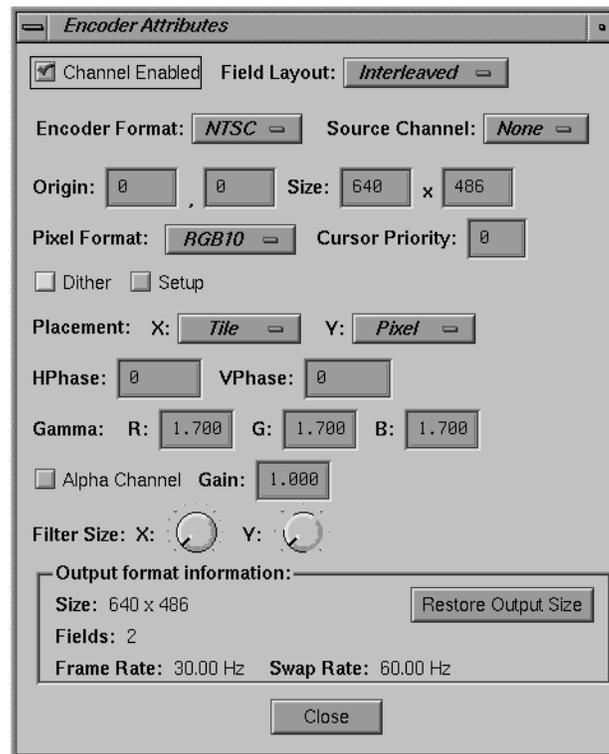
### Dependent Mode

For dependent mode, select a specific channel in the Source Channel popup menu on which the Encoder is to be dependent. In this mode, a portion of the visible surface from one of the other high-resolution channels (as selected in the Source Channel popup menu) is sent to the Encoder for video out.

If the size of the selected region matches the size of the output format, the Encoder uses a precision of 10 bits per component and does not process the pixels (pass-through mode). However, if you have enlarged the input size so that it is larger than the output size of the format, pixels are filtered and processed to fit the output size (reduced mode). Filtered pixels have a precision of 8 bits per component.

## Modifying Encoder Channel Attributes

To view or change attributes of the Encoder channel, select the channel rectangle and select “Edit attributes...” in the Channel menu. The Encoder Attributes window appears, as shown in Figure 3-2.



**Figure 3-2** Encoder Channel Attributes Window

This window shares features of the Channel Attributes window. See “Modifying Channel Attributes” on page 14 for explanations of these features.

Output resolution (Encoder Format in the Encoder Attributes window) is fixed to NTSC or PAL square-pixel formats:

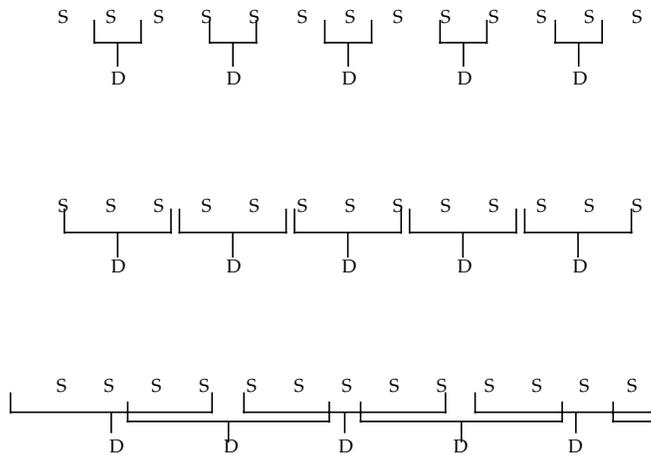
- NTSC: 646 x 486
- PAL: 768 x 576

**Note:** The composite and S-Video Encoder outputs are for industrial or monitoring purposes only, and are not designed for broadcast television use. For broadcast-quality composite video, configure a video channel of the graphics system as RS-170 or Euro 625 resolution and send it through an external broadcast-quality encoder. For CCIR601 or broadcast-quality component video, use the Sirius Video option, as explained in Chapter 4.

### Specifying the Filter Width

In the Encoder Attributes window, use the Filter Size dials to set filter size in the X and Y directions for video resampling. The ticks at the dials correspond to the number of source pixels used to derive the destination pixels. As you adjust the dials, view the changes on an output monitor to determine the desired filter size.

Figure 3-3 diagrams variations of the filter width. In this figure, S = source pixel, and D = destination pixel (after the filter is applied). The wider the filter, the more blurred the output image.



**Figure 3-3** Varying the Width of the Filter

**Note:** Scaling is better done in the Y direction, where any resultant fuzziness is less noticeable, than in the X direction.

## Using the Combiner With Sirius Video

The Combiner has a dedicated channel for use when the Sirius Video option is installed. This chapter explains how to use the Combiner's Sirius Video channel:

- setting up the Sirius Video channel
- modifying Sirius Video channel attributes

When the Sirius Video option is installed, the button labeled *Sirius* on the Combiner main window is accessible.

### Setting Up the Sirius Video Channel

Like the Encoder channel, the Sirius Video channel can be set up in independent mode or dependent mode. Use the Sirius Video Attributes window.

**Note:** For the combination to be valid, the swap rates of all channels, including the Sirius Video channel, must match. For more information on the swap rate, see "Swap Rate" in Chapter 5.

#### Independent Mode

In the Sirius Attributes window, select None in the Source Channel popup menu to set up the Sirius Video channel in independent mode. In this mode, pixels coming to the Sirius Video channel are 10 bits.

Like other channels, the Sirius Video channel in independent mode consumes pixel bandwidth. The Combiner validates the selection along with the other channels.

## Dependent Mode

For dependent mode, select a specific channel in the Source Channel popup menu on which the Sirius Video channel is to be dependent. In this mode, a portion of the visible surface from one of the other high-resolution channels (as selected in the Source Channel popup menu) is sent to the Sirius Video channel for video out. Like the Encoder channel, the Sirius Video channel can roam within its source channel.

If the size of the selected region matches the size of the output format, the Sirius Video option uses a precision of 10 bits per component and does not process the pixels (pass-through mode). However, if you have enlarged the input size so that it is larger than the output size of the format, pixels are filtered and processed to fit the output size (reduced mode). Filtered pixels have a precision of 8 bits per component.

**Note:** On systems with the Sirius Video option, set up the default video combination with the Sirius Video channel dependent on a high-resolution channel. If you want the alpha component, select the RGBA10 pixel format on both the Sirius Video channel and the channel on which it is dependent.

## Modifying Sirius Video Channel Attributes

Double-click the *Sirius* button in the Combiner main window, or double-click on the Sirius Video rectangle to edit Sirius Video attributes; Figure 4-1 shows the window.

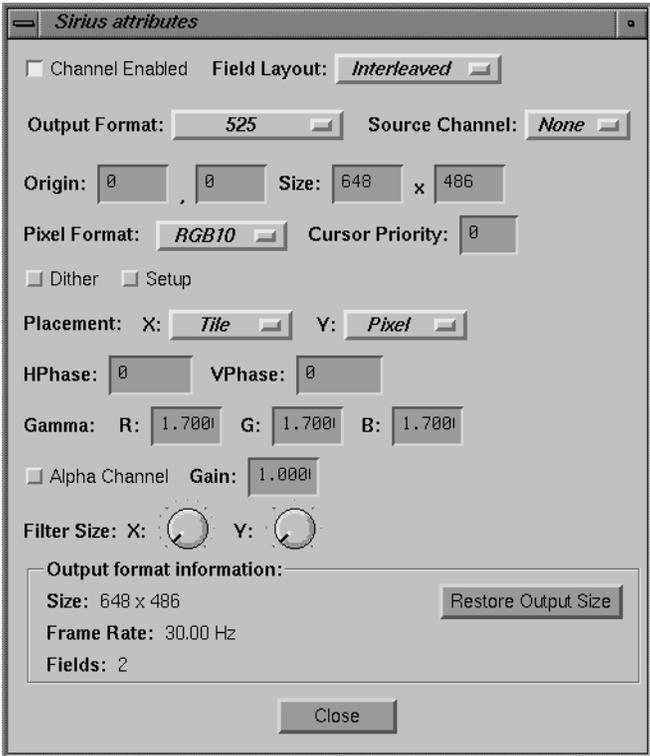


Figure 4-1 Sirius Video Channel Attributes Window

**Note:** Most features of this window are the same as for channel attributes or Encoder attributes; see the explanations in Chapters 2 and 3 for information.

Select the output format for Sirius Video from the Output Format popup menu:

- 525: square pixel 646 x 486 NTSC resolution
- 625: square pixel 768 x 576 PAL resolution
- CCIR601 525: non-square 720 x 486 NTSC resolution
- CCIR601 625: non-square 720 x 576 PAL resolution

At Pixel Format, select RGBA10 for 10-bit RGB with alpha. This selection presents alpha information for the Sirius Video board.

**Note:** Make sure that the Alpha Channel checkbox is unchecked.

To genlock the high-resolution channel to the Sirius Video board, click *Edit globals...* in the Combiner main window, and select a sync source and format, as explained in “Setting Sync Source and Format” in Chapter 2.

---

## Combination Considerations

This chapter explains

- trading off format parameters in a combination
- Emergency Backup Combination

### Trading Off Format Parameters in a Combination

When you use the Combiner to set video combinations for InfiniteReality's multichannel capability, you must take into account

- swap rate
- transmission bandwidth
- DAC output bandwidth
- framebuffer memory and read/write bandwidth

#### Swap Rate

All video formats that run together in a video combination must provide time for housekeeping operations such as updating cursor position and glyph, setting parameters for dynamic pixel resampling, and so on. This updating and resampling might be done at field boundaries, frame boundaries, or in certain cases, across multiple frames. The interval consumed by such housekeeping operations is called the video format's *maximum swap rate*.

The swap rate must be the same for all video formats in a combination, so that the InfiniteReality Video Display subsystem can perform these housekeeping services for all running video formats at the same time. Generally, the 525 formats imply a 60 Hz swap; the 625 formats imply a 50 Hz swap rate.

All of the following formats, which run at 60 Hz, would make up a legal combination:

- 30 Hz interlaced NTSC (swap rate is the 60 Hz field rate for this format)
- 60 Hz noninterlaced 1280 x 1024
- 120 Hz stereo (swap rate is the 60 Hz frame rate)
- 180 Hz color field sequential (swap rate is the 60 Hz frame rate)

Examples of illegal combinations are NTSC (60 Hz swap rate) and PAL (50 Hz swap rate), or the 60 Hz and 72 Hz versions of 1280 x 1024.

**Note:** Although NTSC really runs at 59.94 Hz field rates, it is close enough to be run with 60 Hz 1280 x 1024. In general, if the swap rates are close enough to allow the video formats to be reliably synchronized to one another, they are considered “equal.” This tolerance is usually less than 2%.

The swap rate for the video output format you select for a channel is displayed in its Channel Attributes window. Consult `/usr/gfx/ucode/KONA/dg4/vfo/README` for complete information on each format, including its swap rate.

## Transmission Bandwidth

Digital video data is sent serially from the InfiniteReality framebuffer to the video subsystem. Each video channel uses a portion of the system’s aggregate video transmission bandwidth. The bandwidth each channel requires depends on the resolution of the video output format running on that channel, and on the type of video it receives from the framebuffer. The requirements of all video channels must total less than the aggregate video bandwidth of the system.

Depending on the number and precision of color components requested from the framebuffer by each video channel, the video transmission bandwidth ranges from approximately 290 million pixels per second (for 10-bit RGB or 12-bit color field-sequential video) to 210 million pixels per second (for 10-bit RGBA or 12-bit RGB video). The digital data stream from the framebuffer to a particular video channel must be of a single type and precision, for example, 10-bit RGB. However, different video channels (and their associated digital video streams from the framebuffer) can be mixed: one channel can request 10-bit RGB, while another channel can request 12-bit RGB, and yet another channel can request 10-bit RGBA.

The transmission format for video is not the same as the depth of pixels in the framebuffer. Pixel depth (Small, Medium, Large, X-Large) must be uniform across the entire managed area of the framebuffer. As explained in the preceding paragraph, the format of the colors transmitted as video from the framebuffer to the video display subsystem can vary for each channel to conserve the aggregate video transmission bandwidth.

Framebuffer memory stores lots of nonvideo information about each pixel (like multisample and Z information); the color fields are a subset of the framebuffer pixel storage. Thus, it is important to distinguish between the framebuffer representation of pixels and the format selected for video transmission to the video display subsystem, in order to understand and optimize the performance of both the framebuffer memory and the video display subsystem.

**Note:** The total framebuffer-to-video-subsystem bandwidth is independent of the number of RM6 boards in the system.

### **DAC Output Bandwidth**

The DACs of the first two video channels, 0 and 1, have a bandwidth limit of 220 million pixels per second; the remaining six have a bandwidth limit of 170 million pixels per second. Because of each channel's dynamic resampling capability, the actual video resolution as output by the DAC (and seen by the display monitor connected to the channel) can be higher than the resolution of the rectangular region of the framebuffer assigned to that channel. Therefore, in a video format combination, always assign the video formats with the highest bandwidth video to Channels 0 and 1.

Dynamic resampling can be used to statically resample a region of the framebuffer area, allowing a smaller region of the framebuffer to be enlarged (zoomed) to fit the resolution of the video format of the video channel assigned to that region of the framebuffer. This feature facilitates efficient use of framebuffer memory without requiring nonstandard video formats.

### **Framebuffer Memory and Read/Write Bandwidth**

A video format combination must all be able to tile rectangular subregions of a rectangular framebuffer. Depending on the depth of pixels selected and the number of RM6 boards installed, some combinations can require more than one RM board. When you specify the framebuffer pixel depth, take into account the quality of multisampling,

the precision of the z-buffer, RGB component resolution, and total fill-rate requirement of the application.

Although adding RM6 boards does not increase the total framebuffer-to-video subsystem bandwidth, it does increase the total drawing fill rate available, and amortizes the video refresh overhead over the additional RM6 boards, leaving more fill capacity available for drawing.

Finally, refreshing the video channels entails considerable overhead for the framebuffer. The greater the number and resolution of video channels, the more the pixel fill rate of the framebuffer is reduced. A particular combination of formats might fit into a one- or two-RM6 configuration, but it might reduce the fill rate unacceptably.

If the combination uses too much bandwidth, you can reduce the size of the input image, but what is sent to the monitor is not changed. For example, this technique is useful if you are using a fixed-frequency monitor that supports only certain formats.

## Emergency Backup Combination

The InfiniteReality system software includes an Emergency Backup Combination (EBC). If the system software cannot load a combination during the boot process because the combination saved to the EEPROM is not valid (for example, if it exceeds the capacity of the system), it uses the EBC.

The EBC is a video format combination consisting of the minimum configuration possible: one RM6 board, two channels, no option boards, and one video format, 1280x1024\_60, on Channel 0. This file is editable only by Silicon Graphics engineering personnel.

If the InfiniteReality system is using the EBC, load the EEPROM with a combination you prefer:

1. In the File menu, select "Open..." and specify the filename.
2. In the File menu, select "Save to EEPROM..."

---

## Error Messages

This appendix presents Video Format Combiner error messages, which appear near the bottom of the Combiner main window.

External sync format maximum swap rate does not match other channels

The maximum swap rate in the format specified as the external sync format (that is, the glXSwapBuffers rate that the application could run at in this combination) must match the maximum swap rate of the output channels. Select a sync format that has the same maximum swap rate as the other channels, or vice versa; see also “Swap Rate” on page 37.

External sync source does not have a unique sync signature

The video format you have chosen as the sync format (see “Setting Sync Source and Format” on page 26) cannot be used as a sync input because it does not have a unique signature. Choose a different format for input sync.

External sync source requires a sync format

In the Global Attributes window, set a sync format for the external sync.

Illegal cursor priority on channel <channel>. 0 is top priority

Illegal cursor priority on channel <channel>. 255 is lowest priority

Reset the cursor priority for the channel.

Illegal Pixel Format

This channel is incapable of outputting the pixel format selected for this channel in the Channel Attributes window. Consult the equipment documentation and select an appropriate pixel format.

Illegal Textport channel number (%d)

You have selected a channel to serve as textport that is dependent on another channel.

Managed area must have positive, greater than zero size

Internal Combiner error; contact Silicon Graphics technical support.

Managed Area height must be less than <number> pixels

Managed Area must be less than <number> pixels total

Managed Area width must be less than <number> pixels

In the Global Attributes window, respecify the dimensions of the managed area to be within the values specified.

Managed Area too large, and/or pixel size too big given <number> RM6's

The combination requires more RM6 boards than are available on the system on which you are running the combination. Reduce the size of the managed area or the pixel size; see "Changing Global Attributes for Combinations" on page 23.

Maximum tiles per request less than 1

Internal Combiner error; contact Silicon Graphics technical support.

Maximum swap rate must be greater than 0 swaps per second

Internal Combiner error; contact Silicon Graphics technical support.

Maximum swap rates on all channels must be the same

In the Channel Attributes window for each channel, check the swap rate. Select formats so that swap rates match; see "Swap Rate" on page 37.

Need <number> RM6 boards

This combination requires the specified number of additional RM6 boards.

No channels enabled

You must specify at least one channel to create a combination.

Only the Encoder or Sirius Video channels can be dependent

You have specified an invalid channel as a dependent channel.

Textport channel <channel> invalid

The channel you have selected as a textport channel cannot support this task.

---

too many channels: specified hardware supports 2 channels

The combination requires more than two channels, but the system on which you wish to run it has only two.

Video bandwidth exceeded

Reduce the bandwidth consumed by dropping a channel from the combination or by reducing the size of one or more channels.

<channel> and <channel> channels cannot be enabled together

The two channels specified cannot be used simultaneously because they are incompatible with one another.

<channel> channel cannot request its own data

Internal Combiner error; contact Silicon Graphics technical support.

<channel> channel format too fast for this channel

This channel is incapable of outputting the video format selected for this channel in the Channel Attributes window. See "DAC Output Bandwidth" on page 39 and select an appropriate pixel format.

<channel> channel format must have at least 1 field

You have selected an invalid video format; choose another.

<channel> channel format must have less than <number> fields

You have selected an invalid video format; choose another.

<channel> channel gain must be  $\geq 0$  and  $\leq 10$

The value you selected for the channel's gain exceeds the allowed range.

<channel> channel horizontal phase must be  $\geq 0$  and  $\leq$  <number>

The value you selected for the channel's horizontal phase exceeds the allowed range.

<channel> channel is dependent on invalid channel <channel>

The channel you have selected as the source is unavailable; select another in the Attributes window.

<channel> channel is dependent on non-requesting channel <channel>

The channel you have selected as the source is itself dependent; select a channel that is independent.

<channel> channel must be stereo if other channels are stereo

For any combination that contains a stereo format, the specified channel must also run a stereo format. You can also consider using the specified channel as the stereo channel.

<channel> channel needs to be at least <number> pixels tall

<channel> channel needs to be at least <number> pixels wide

<channel> channel needs to be less than <number> pixels tall

<channel> channel needs to be less than <number> pixels wide

Resize the channel so that it meets the dimension specified.

<channel> channel off bottom edge of managed area

<channel> channel off left edge of managed area or source channel

<channel> channel off right edge of managed area

<channel> channel off top edge of managed area or source channel

Reposition the channel so that it is within the managed area; set its origin in the Channel Attributes window or move it with the mouse.

<channel> channel off bottom edge of source channel <channel>

<channel> channel off right edge of source channel <channel>

Reposition the dependent channel so that it is within or congruent with the boundaries of the channel on which it is dependent. Set its origin in the Channel Attributes window or move it with the mouse.

<channel> channel SCH phase must be  $\geq -1800$  and  $\leq 1800$

The value you selected for the channel's SCH phase exceeds the allowed range.

<channel> channel vertical phase must be 0 to lines-1

The value you selected for the channel's vertical phase exceeds the allowed range. Select a value between zero and the total number of vertical lines in this format minus one.

<channel> channel width must be a multiple of <number>

Set the channel width accordingly.

---

<channel> channel X filter size must be between 1 and 13

The value you selected for the channel's X filter size exceeds the allowed range.

<channel> channel x position must be a multiple of <number>

Move the channel so that it is positioned as indicated.

<channel> channel Y filter size must be between 1 and 7

The value you selected for the channel's Y filter size exceeds the allowed range.



---

## Glossary

### **active video**

The portion of the video signal containing the chrominance or luminance information; all video lines not occurring in the vertical blanking signal containing the chrominance or luminance information. See also *chrominance*, *composite video*, *horizontal blanking*, *luminance*, and *video waveform*.

### **aliasing**

One of several types of digital video artifact appearing as jagged edges. Aliasing results when an image is sampled that contains frequency components above the Nyquist limit for the sampling rate. See also *Nyquist limit*.

### **alpha**

See *alpha value*.

### **alpha blending**

Overlaying one image on another so that some of the underlying image may or may not be visible. See also *key*.

### **alpha value**

The component of a pixel that specifies the pixel's opacity, translucency, or transparency. The alpha component is typically output as a separate component signal.

### **antialiasing**

Filtering or blending lines of video to smooth the appearance of jagged edges in order to reduce the visibility of aliasing.

### **APL**

Average Picture Level, with respect to blanking, during active picture time, expressed as a percentage of the difference between the blanking and reference white levels. See also *blanking level*.

**artifact**

In video systems, an unnatural or artificial effect that occurs when the system reproduces an image; examples are aliasing, pixellation, and contouring.

**aspect ratio**

The ratio of the width to the height of an electronic image. For example, the standard aspect ratio for television is 4:3.

**back porch**

The portion of the horizontal pedestal that follows the horizontal synchronizing pulse. In a composite signal, the color burst is located on the back porch, but is absent on a YUV or GBR signal. See also *blanking level*, *video waveform*.

**black burst**

Active video signal that has only black in it. The black portion of the video signal, containing color burst. See also *color burst*.

**black level**

In the active video portion of the video waveform, the voltage level that defines black. See also *horizontal blanking* and *video waveform*.

**blanking level**

The signal level at the beginning and end of the horizontal and vertical blanking intervals, typically representing zero output (0 IRE). See also *video waveform* and *IRE units*.

**breezeway**

In the horizontal blanking part of the video signal, the portion between the end of the horizontal sync pulse and the beginning of the color burst. See also *horizontal blanking* and *video waveform*.

**broad pulses**

Vertical synchronizing pulses in the center of the vertical interval. These pulses are long enough to be distinguished from other pulses in the signal; they are the part of the signal actually detected by vertical sync separators.

**Bruch blanking**

In PAL signals, a four-field burst blanking sequence used to ensure that burst phase is the same at the end of each vertical interval.

**burst, burst flag**

See *color burst*.

**burst lock**

The ability of the output subcarrier to be locked to input subcarrier, or of output to be genlocked to an input burst.

**burst phase**

In the RS-170A standard, burst phase is at field 1, line 10; in the European PAL standards, it is at field 1, line 1. Both define a continuous burst waveform to be in phase with the leading edge of sync at these points in the video timing. See also *vertical blanking interval* and *video waveform*.

**B-Y (B minus Y) signal**

One of the color difference signals used on the NTSC and PAL systems, obtained by subtracting luminance (Y) from the blue camera signal (B). This signal drives the horizontal axis of a vectorscope. Color mixture is close to blue; phase is 180 degrees opposite of color sync burst; bandwidth is 0.0 to 0.5 MHz. See also *luminance*, *R-Y signal*, *Y signal*, and *Y/R-Y/B-Y*.

**C signal**

Chrominance; the color portion of the signal. For example, the Y/C video format used for S-VHS has separate Y (luminance) and C (chrominance) signals. See also *chrominance*.

**CAV**

Component Analog Video; a generic term for all analog component video formats, which keep luminance and chrominance information separate. D1 is a digital version of this signal. See also *component video*.

**CCIR 601**

The digital interface standard developed by the CCIR (Comite' Consultatif International de Radiodiffusion, International Radio Consultative Committee) based on component color encoding, in which the luminance and chrominance (color difference) sampling frequencies are related in the ratio 4:2:2: four samples of luminance (spread across four pixels), two samples of  $C_R$  color difference, and two samples of  $C_B$  color difference. The standard, which is also referred to as 4:2:2, sets parameters for both 525-line and 625-line systems.

**chroma**

See *chrominance*.

**chroma signal**

A 3.58 MHz (NTSC) or 4.43 MHz (PAL) subcarrier signal for color in television. SECAM uses two frequency-modulated color subcarriers transmitted on alternate horizontal lines;  $SC_R$  is 4.406 MHz and  $SC_B$  is 4.250 MHz.

**chrominance**

In an image reproduction system, a separate signal that contains the color information. Black, white, and all shades of gray have no chrominance and contain only the luminance (brightness) portion of the signal. However, all colors have both chrominance and luminance.

Chrominance is derived from the I and Q signals in the NTSC television system and the U and V signals in the PAL television system. See also *luminance*.

**chrominance signal**

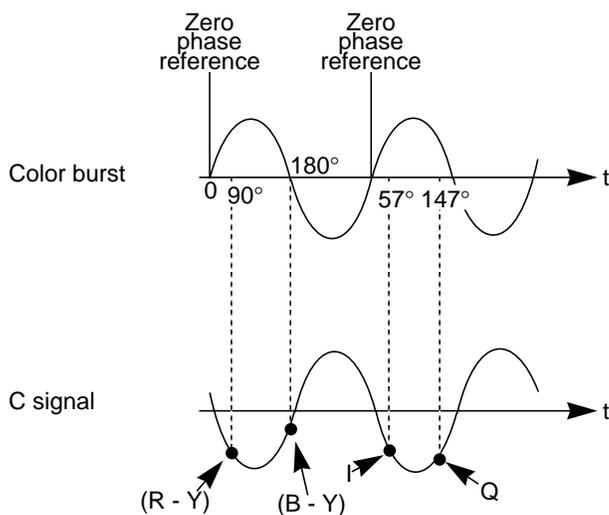
Also called the chroma, or C, signal. The high-frequency portion of the video signal (3.58 MHz for NTSC, 4.43 MHz for PAL) color subcarrier with quadrature modulation by I (R-Y) and Q (B-Y) color video signals. The amplitude of the C signal is saturation; the phase angle is hue. See also *color subcarrier*, *hue*, and *saturation*.

**color burst**

Also called burst and burst flag. The segment of the horizontal blanking portion of the video signal that is used as a reference for decoding color information in the active video part of the signal. The color burst is required for synchronizing the phase of 3.58 MHz oscillator in the television receiver for correct hues in the chrominance signal.

In composite video, the image color is determined by the phase relationship of the color subcarrier to the color burst. The color burst sync is 8 to 11 cycles of 3.58 MHz color subcarrier transmitted on the back porch of every horizontal pulse; the hue of the color sync phase is yellow-green.

Figure G1-1 diagrams the relationship of the color burst and the chrominance signal. See also *color subcarrier* and *video waveform*.



**Figure GI-1** Color Burst and Chrominance Signal

#### color difference signals

Signals used by color television systems to convey color information so that the signals go to zero when the picture contains no color; for example, unmodulated R-Y and B-Y, I and Q, U, and V.

#### color-frame sequence

In NTSC and S-Video, a two-frame sequence that must elapse before the same relationship between line pairs of video and frame sync repeats itself. In PAL, the color-frame sequence consists of four frames.

#### color subcarrier

A portion of the active portion of a composite video signal that carries color information, referenced to the color burst. The color subcarrier's amplitude determines saturation; its phase angle determines hue. Hue and saturation are derived with respect to the color burst. Its frequency is defined as 3.58 MHz in NTSC and 4.43 MHz in PAL. See also *color burst*.

#### colorspace

A color component encoding format defined by three color components, such as R, G, and B or Y, U, and V.

**component video**

A color encoding method for the three color signals—R, G, and B; Y, I, and Q; or Y, U, and V—that make up a color image. See also *RGB*, *YIQ*, and *YUV*.

**component video signals**

A video signal in which luminance and chrominance are sent as separate components, for example:

- RGB (basic signals generated from a camera)
- YIQ (used by the NTSC broadcasting standard)
- Y/R-Y/B-Y (used by Betacam and M-II recording formats and SECAM broadcasting standard)
- YUV (subset of Y/R-Y/B-Y used by the PAL broadcasting standard)

Separating these components yields a signal with a higher color bandwidth than that of composite video.

Figure G1-2 depicts video signals for one horizontal scan of a color-bar test pattern. The RGB signals change in relation to the individual colors in the test pattern. When a secondary color is generated, a combination of the RGB signals occurs. Since only the primary and secondary colors are being displayed at 100% saturation, the R, G, and B waveforms are simply on or off. For more complex patterns of color, the individual R, G, and B signals would be varying amplitudes in the percentages needed to express that particular color.

See also *composite video*, *RGB*, *YUV*, *Y/R-Y/B-Y*, and *YIQ*.

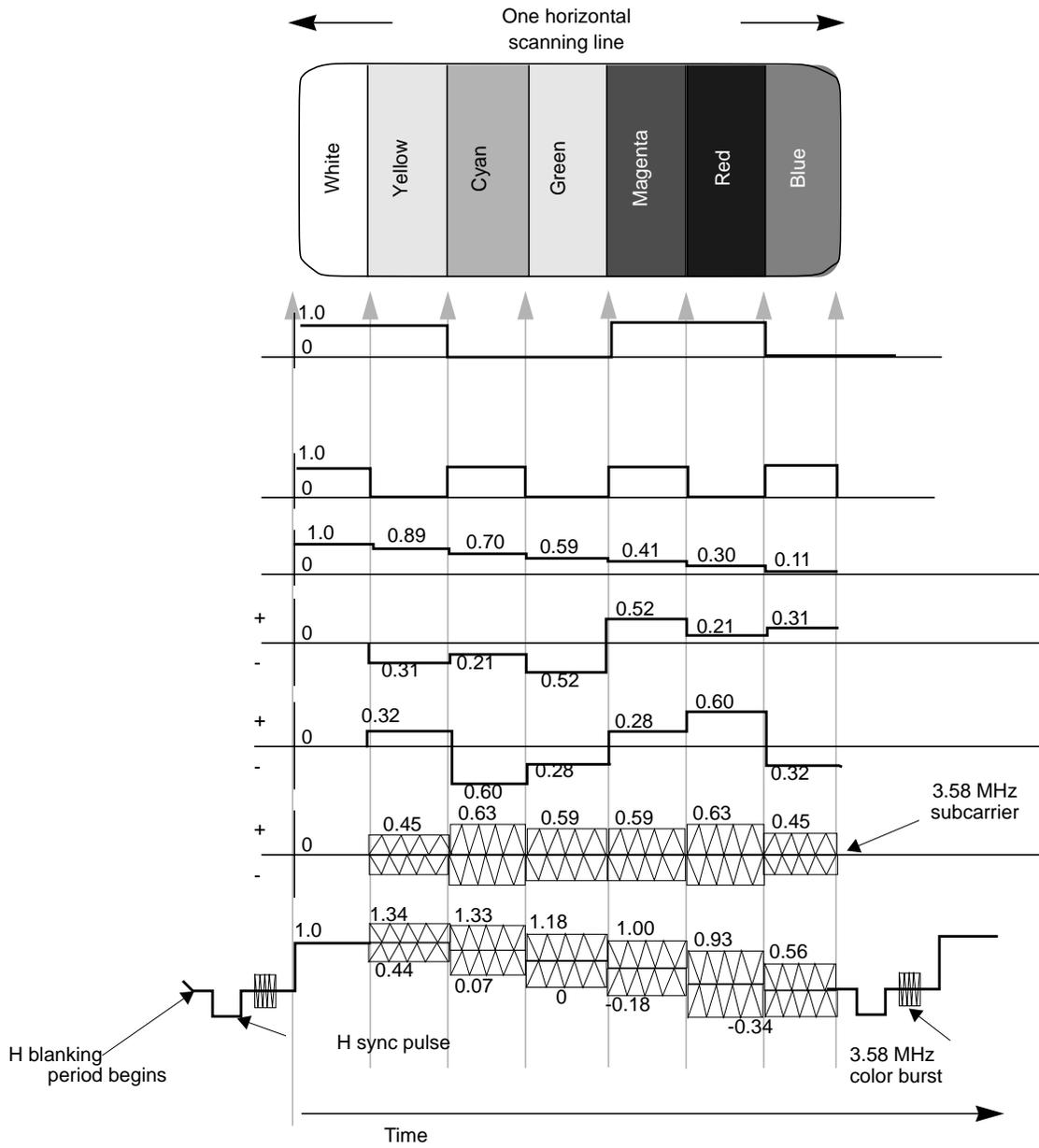


Figure GI-2 Component Video Signals

**composite video**

A color encoding method or a video signal that contains all of the color, brightness, and synchronizing information in one signal. The chief composite television standard signals are NTSC, PAL, and SECAM. See also *NTSC*, *PAL*, and *SECAM*.

**D1**

Digital recording technique for component video; also known as CCIR 601, 4:2:2. D1 is the best choice for high-end production work where many generations of video are needed. D1 can be an 8-bit or 10-bit signal. See also *CCIR 601*.

**D2**

Digital recording technique for composite video. As with analog composite, the luminance and chrominance information is sent as one signal. A D2 VTR offers higher resolution and can make multiple generation copies without noticeable quality loss, because it samples an analog composite video signal at four times the subcarrier (using linear quantization), representing the samples as 8-bit digital words. D2 is not compatible with D1.

**decoder**

Hardware or software that converts, or decodes, a composite video signal into the various components of the signal. For example, to grab a frame of composite video from a VHS tape deck and store it as an RGB file, it would have to be decoded first. Several Silicon Graphics video options have on-board decoders.

**dithering**

Approximating a signal value on a chroma-limited display device by producing a matrix of color values that fool human perception into believing that the signal value is being reproduced accurately. For example, dithering is used to display a true-color image on a display capable of rendering only 256 unique colors, such as IndigoVideo images on a Starter Graphics display.

**encoder**

Device that combines the R, G, and B primary color video signals into hue and saturation for the C portion of a composite signal. Several Silicon Graphics video options have on-board encoders.

**equalizing pulse**

Pulse of one half the width of the horizontal sync pulse, transmitted at twice the rate of the horizontal sync pulse, during the portions of the vertical blanking interval immediately before and after the vertical sync pulse. The equalizing pulse makes the vertical deflection start at the same time in each interval, and also keeps the horizontal sweep circuits in step during the portions of the vertical blanking interval immediately before and after the vertical sync pulse.

**field**

One of two (or more) equal parts of information in which a frame is divided in interlace scanning. A vertical scan of a frame carrying only its odd-numbered or its even-numbered lines. The odd field and even field make up the complete frame. See also *frame* and *interlace*.

**field blanking**

The blanking signals at the end of each field, used to make the vertical retrace invisible. Also called vertical blanking; see *vertical blanking* and *vertical blanking interval*.

**filter**

To process a clip with spatial or frequency domain methods. Each pixel is modified by using information from neighboring (or all) pixels of the image. Filter functions include blur (low-pass) and crisp (high-pass).

*Q change or delete?*

**frame**

The result of a complete scanning of one image. In television, the odd field (all the odd lines of the frame) and the even field (all the even lines of the frame) make up the frame. In motion video, the image is scanned repeatedly, making a series of frames.

**frequency**

Signal cycles per second.

**frequency interlace**

Placing of harmonic frequencies of C signal midway between harmonics of horizontal scanning frequency  $F_h$ . Accomplished by making color subcarrier frequency exactly 3.579545 MHz. This frequency is an odd multiple of  $H/2$ .

**front porch**

The portion of the video signal between the end of active video and the falling edge of sync. See also *back porch*, *horizontal blanking*, and *video waveform*.

**G-Y signal**

Color mixture close to green, with a bandwidth 0.0 MHz to 0.5 MHz. Usually formed by combining B-Y and R-Y video signals.

**gamma correction**

Correction of gray-scale inconsistency. The brightness characteristic of a CRT is not linear with respect to voltage; the voltage-to-intensity characteristic is usually close to a power of 2.2. If left uncorrected, the resulting display has too much contrast and detail in black regions is not reproduced.

To correct this inconsistency, a correction factor using the 2.2 root of the input signal is included, so that equal steps of brightness or intensity at the input are reproduced with equal steps of intensity at the display.

**genlocking**

Synchronizing with another video signal serving as a master timing source. The master timing source can be a composite video signal, a video signal with no active video (only sync information), or, for video studio, a device called house sync. When there is no master sync available, VideoFramer, for example, can be set to “free run” (or “stand-alone”) mode, so that it becomes the master timing device to which other devices sync. See also *line lock*.

**gray-scale**

Monochrome or black-and-white, as in a monitor that does not display color.

**H rate**

Number of complete horizontal lines, including trace and retrace, scanned per second.

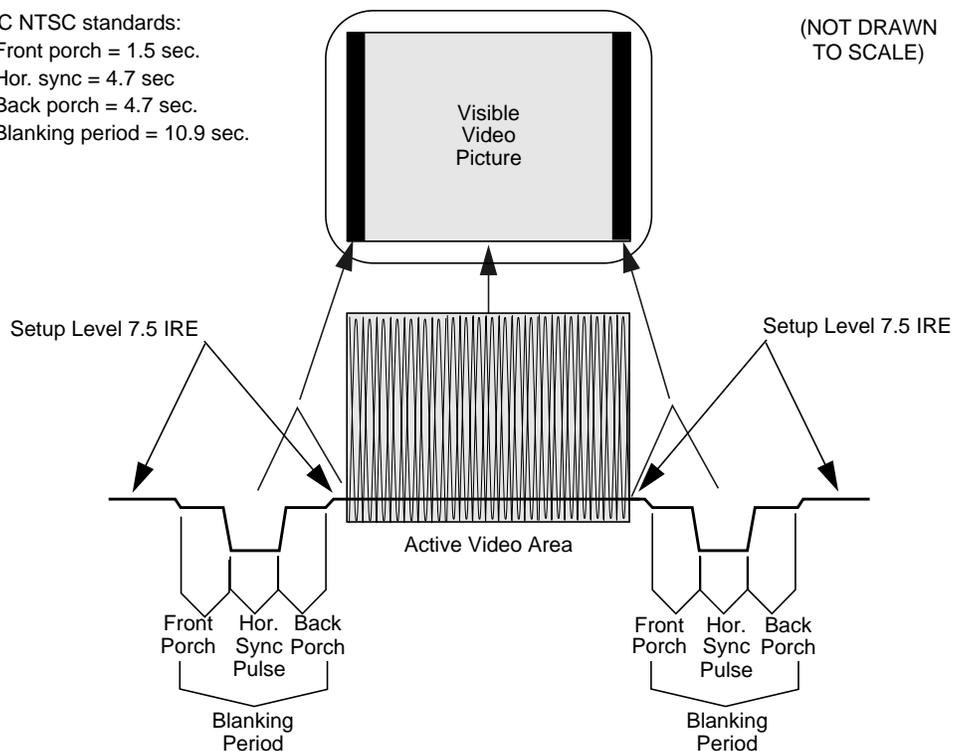
**HDTV**

High-definition television. Though there is more than one proposal for a broadcast standard for HDTV, most currently available equipment is based on the 1125/60 standard, that is, 1125 lines of video, with a refresh rate of 60Hz, 2:1 interlacing (same as NTSC and PAL), and aspect ratio of 16:9 (1920 x 1035 viewable resolution), trilevel sync, and 30 MHz RGB and luminance bandwidth.

### horizontal blanking

The period when the electron beam is turned off, beginning when each scan line finishes its horizontal path (scan) across the screen (see Figure GI-3).

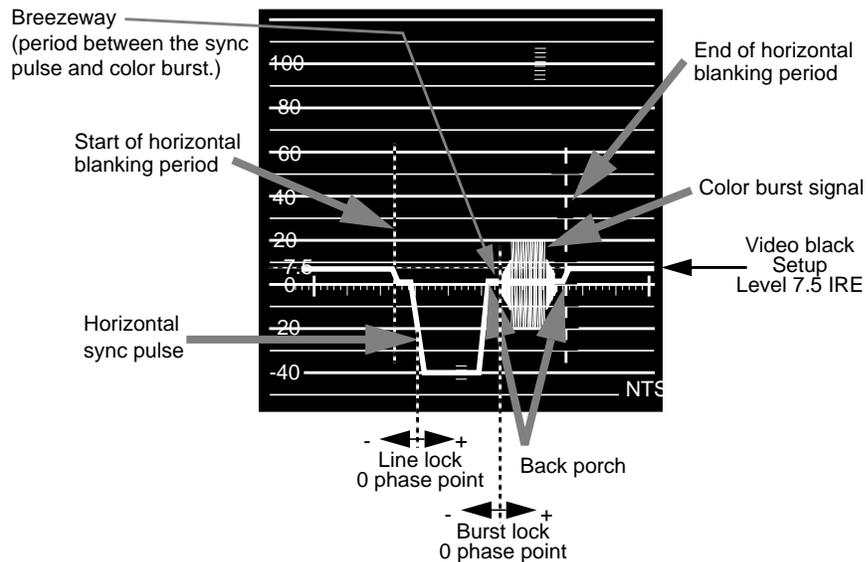
FCC NTSC standards:  
 Front porch = 1.5 sec.  
 Hor. sync = 4.7 sec.  
 Back porch = 4.7 sec.  
 Blanking period = 10.9 sec.



**Figure GI-3** Horizontal Blanking

### horizontal blanking interval

Also known as the horizontal retrace interval, the period when a scanning process is moving from the end of one horizontal line to the start of the next line. This portion of the signal is used to carry information other than video information. See also *video waveform*.



**Figure GI-4** Horizontal Blanking Interval

**horizontal drive**

The portion of the horizontal blanking part of the video signal composed of the sync pulse together with the front porch and breezeway; that is, horizontal blanking minus the color burst. See also *video waveform*.

**horizontal sync**

The lowest portion of the horizontal blanking part of the video signal, it provides a pulse for synchronizing video input with output. Also known as h sync. See also *horizontal blanking* and *video waveform*.

**hue**

The designation of a color in the spectrum, such as cyan, blue, magenta. Sometimes called tint on NTSC television receivers. The varying phase angles in the 3.58 MHz (NTSC) or 4.43 MHz (PAL) C signal indicate the different hues in the picture information.

**I signal**

Color video signal transmitted as amplitude modulation of the 3.58 MHz C signal (NTSC). The hue axis is orange and cyan. This signal is the only color video signal with a bandwidth of 0 to 1.3 MHz.

**interlace**

A technique that uses more than one vertical scan to reproduce a complete image. In television, the 2:1 interlace used yields two vertical scans (fields) per frame: the first field consists of the odd lines of the frame, the other of the even lines. See also *field* and *frame*.

**leading edge of sync**

The portion of the video waveform after active video, between the sync threshold and the sync pulse. See also *video waveform*.

**line frequency**

The number of horizontal scans per second, normally 15,734.26 times per second for NTSC color systems. The line frequency for the PAL 625/50H. system is 15,625 times per second.

**line lock**

Input timing that is derived from the horizontal sync signal, also implying that the system clock (the clock being used to sample the incoming video) is an integer multiple of the horizontal frequency and that it is locked in phase to the horizontal sync signal. See also at *video waveform*.

**live video**

Video being delivered at a nominal frame rate appropriate to the format.

**luma**

See *luminance*.

**luminance**

The video signal that describes the amount of light in each pixel. Luminance is a weighted sum of the R, G, and B signals. See also *chrominance* and *Y signal*.

**NTSC**

A color television standard or timing format encoding all of the color, brightness, and synchronizing information in one signal. Used in North America, most of South America, and most of the Far East, this standard is named after the National Television Systems Committee, the standardizing body that created this system in the U.S. in 1953. NTSC employs a total of 525 horizontal lines per frame, with two fields per frame of 262.5 lines each. Each field refreshes at 60Hz (actually 59.94Hz).

**PAL**

A color television standard or timing format developed in West Germany and used by most other countries in Europe, including the United Kingdom but excluding France, as well as Australia and parts of the Far East. PAL employs a total of 625 horizontal lines per frame, with two fields per frame of 312.5 lines per frame. Each field refreshes at 50Hz. PAL encodes color differently from NTSC. PAL stands for Phase Alternation Line or Phase Alternated by Line, by which this system attempts to correct some of the color inaccuracies in NTSC. See also *NTSC* and *SECAM*.

**pedestal**

See *setup*; see also *video waveform*.

**pixel**

Picture element; the smallest addressable spatial element of the computer graphics screen. A digital image address, or the smallest reproducible element in analog video. A pixel can be monochrome, gray-scale, or color, and can have an alpha component to determine opacity or transparency. Pixels are referred to as having a color component and an alpha component, even if the color component is gray-scale or monochrome.

**pixel map**

A two-dimensional piece of memory, any number of bits deep. See also *bitmap*.

**Q signal**

The color video signal that modulates 3.58 MHz C signal in quadrature with the I signal. Hues are green and magenta. Bandwidth is 0.0 MHz to 0.5 MHz. See also *C signal*, *I signal*, *YC*, and *YIQ*.

**raster**

The scanning pattern for television display; a series of horizontal lines, usually left to right, top to bottom. In NTSC and PAL systems, the first and last lines are half lines.

**resolution**

Number of horizontal lines in a television display standard; the higher the number, the greater a system's ability to reproduce fine detail.

**RGB**

Red, green, blue; the basic component set used by graphics systems and some video cameras, in which a separate signal is used for each primary color.

**RGB format**

The technical specification for NTSC color television. Often (incorrectly) used to refer to an RGB signal that is being sent at NTSC composite timings, for example, a Silicon Graphics computer set to output 640 x 480. The timing would be correct to display on a television, but the signal would still be split into red, green and blue components. This component signal would have to go through an encoder to yield a composite signal (RS-170A format) suitable for display on a television receiver.

**R-Y (R minus Y) signal**

A color difference signal obtained by subtracting the luminance signal from the red camera signal. It is plotted on the 90 to 270 degree axis of a vector diagram. The R-Y signal drives the vertical axis of a vectorscope. The color mixture is close to red. Phase is in quadrature with B-Y; bandwidth is 0.0 MHz to 0.5 MHz. See also *luminance*, *B-Y (B minus Y) signal*, *Y/R-Y/B-Y*, and *vectorscope*.

**sample**

To read the value of a signal at evenly spaced points in time; to convert representational data to sampled data (that is, synthesizing and rendering).

**sampling rate, sample rate**

Number of samples per second.

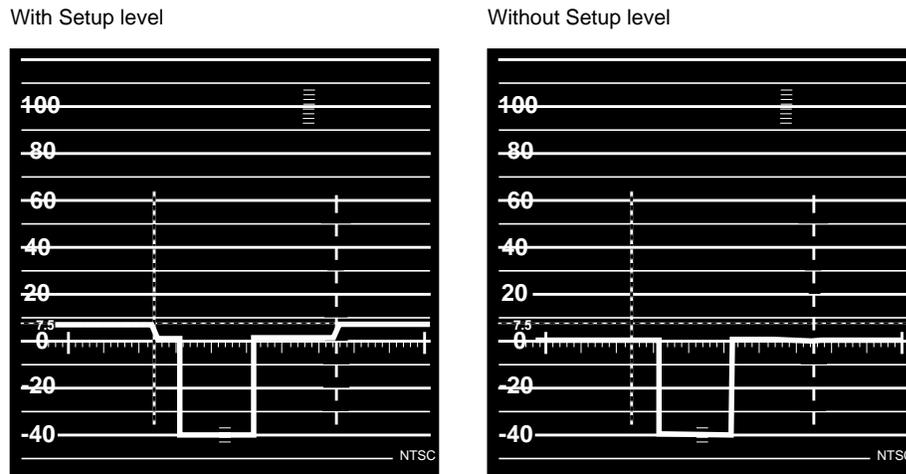
**scaling**

To change the size of an image.

**setup**

The difference between the blackest level displayed on the receiver and the blanking level (see Figure G1-5). A black level that is elevated to 7.5 IRE instead of being left at 0.0 IRE, the same as the lowest level for active video. Because the video level is known, this part of the signal is used for black-level clamping circuit operation. Setup is typically used in the NTSC video format and is typically not used in the PAL video format; it was originally introduced to simplify the design of early television receivers, which had trouble distinguishing between video black levels and horizontal blanking. Also called pedestal.

Figure G1-5 shows waveform displays of a signal with and without setup. See also *video waveform*.



**Figure GI-5** Waveform Monitor Readings With and Without Setup

**subcarrier**

A portion of a video signal that carries a specific signal, such as color. See *color subcarrier*.

**S-Video**

Video format in which the Y (luminance) and C (chrominance) portions of the signal are kept separate. Also known as YC.

**sync information**

The part of the television video signal that ensures that the display scanning is synchronized with the broadcast scanning. See also *video waveform*.

**sync pulse**

A vertical or horizontal pulse (or both) that determines the display timing of a video signal. Composite sync has both horizontal and vertical sync pulses, as well as equalization pulses. The equalization pulses are part of the interlacing process.

**sync tip**

The lowest part of the horizontal blanking interval, used for synchronization. See also *video waveform*.

**synchronize**

To perform time shifting so that things line up.

**threshold**

In a digital circuit, the signal level that is specified as the division point between levels used to represent different digital values; for example, the sync threshold is the level at which the leading edge of sync begins. See also *video waveform*.

**U signal**

One of the chrominance signals of the PAL color television system, along with V. Sometimes referred to as B-Y, but U becomes B-Y only after a weighting factor of 0.493 is applied. The weighting is required to reduce peak modulation in the composite signal.

**V signal**

One of the chrominance signals of the PAL color television system, along with U. Sometimes referred to as R-Y, but V becomes R-Y only after a weighting factor of 0.877 is applied. The weighting is required to reduce peak modulation in the composite signal.

**vertical blanking**

The portion of the video signal that is blanked so that the vertical retrace of the beam is not visible.

**vertical blanking interval**

The blanking portion at the beginning of each field. It contains the equalizing pulses, the vertical sync pulses, and vertical interval test signals (VITS). Also the period when a scanning process is moving from the lowest horizontal line back to the top horizontal line.

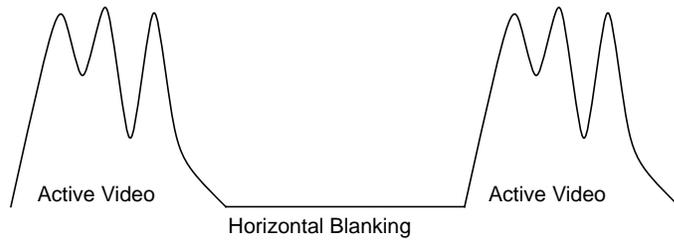
**video signal**

The electrical signal produced by a scanning image sensor.

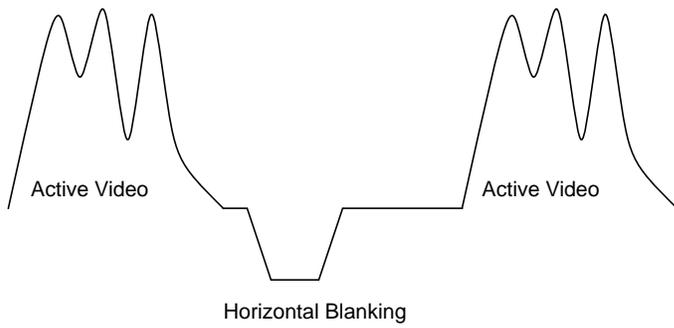
**video waveform**

The main components of the video waveform are the active video portion and the horizontal blanking portion. Certain video waveforms carry information during the horizontal blanking interval.

Figure G1-6 and Figure G1-7 diagram a typical red or blue signal, which carries no information during the horizontal blanking interval, and a typical Y or green-plus-sync signal, which carries a sync pulse.



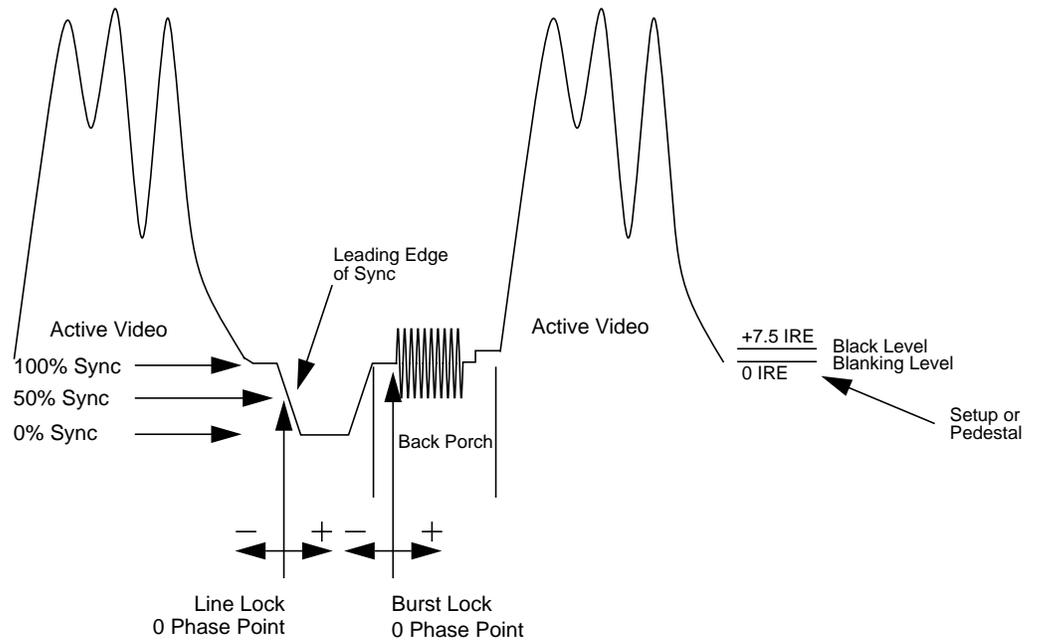
**Figure GI-6** Red or Blue Signal



**Figure GI-7** Y or Green Plus Sync Signal

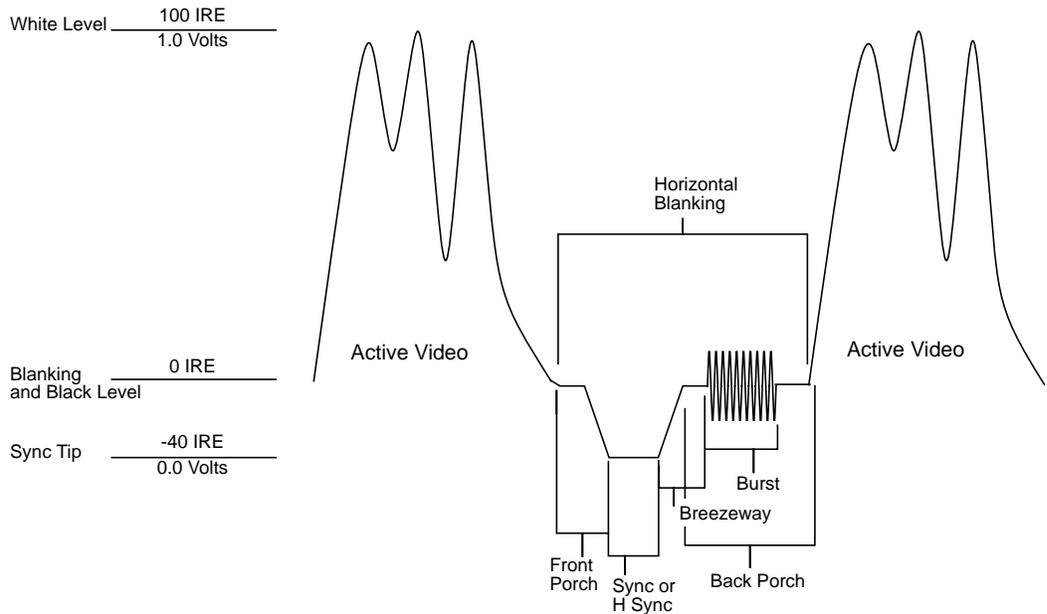
Figure GI-8 and Figure GI-9 show the video waveform and its components for composite video in more detail. The figures show the composite video waveform with and without setup, respectively.

Figure GI-8 shows a composite video signal with setup.



**Figure G1-8** Video Waveform: Composite Video Signal With Setup (Typical NTSC)

Figure G1-9 shows a composite video signal without setup.



**Figure GI-9** Video Waveform: Composite Video Signal (Typical PAL)

**white level**

In the active video portion of the video waveform, the 1.0-volt (100 IRE) level. See also *video waveform*.

**Y signal**

Luminance, corresponding to the brightness of an image. See also *luminance* and *Y/R-Y/B-Y*.

**YC**

A color space (color component encoding format) based on YIQ or YUV. Y is luminance, but the two chroma signals (I and Q or U and V) are combined into a composite chroma called C, resulting in a two-wire signal. C is derived from I and Q as follows:

$$C = I \cos(2 fsc t) + Q \sin(2 fsc t)$$

where fsc is the subcarrier frequency. YC-358 is the NTSC version of this luminance/chrominance format; YC-443 is the PAL version. Both are referred to as S-Video formats.

**YIQ**

A color space (color component encoding format) used in decoding, in which Y is the luminance signal and I and Q are the chrominance signals. The two chrominance signals I and Q (in-phase and quadrature, respectively) are two-phase amplitude-modulated; the I component modulates the subcarrier at an angle of 0 degrees and the Q component modulates it at 90 degrees. The color burst is at 33 degrees relative to the Q signal.

The amplitude of the color subcarrier represents the saturation values of the image; the phase of the color subcarrier represents the hue value of the image.

$$Y = 0.299R + 0.587G + 0.114B$$

$$I = 0.596R - 0.275G - 0.321B$$

$$Q = 0.212R - 0.523G + 0.311B$$

**Y/R-Y/B-Y**

A name for the YUV color component encoding format that summarizes how the chrominance components are derived. Y is the luminance signal and R-Y and B-Y are the chrominance signals. R-Y (red minus Y) and B-Y (blue minus Y) are the color differences or chrominance components. The color difference signals R-Y and B-Y are derived as follows:

$$Y = 0.299R + 0.587G + 0.114B$$

Y/R-Y/B-Y has many variations, just as NTSC and PAL do. All component and composite color encoding formats are derived from RGB without scan standards being changed. The matrix (amount of red, green, and blue) values and scale (amplitude) factors can differ from one component format to another (YUV, Y/R-Y, B-Y, SMPTE Y/R-Y, B-Y).

**YUV**

A color space (color component encoding format) used by the PAL video standard, in which Y is the luminance signal and U and V are the chrominance signals. The two chrominance signals U and V are two-phase amplitude-modulated. The U component modulates the subcarrier at an angle of 0 degree, but the V component modulates it at 90 degrees or 180 degrees on alternate lines. The color burst is also line-alternated at +135 and -135 degrees relative to the U signal. The YUV matrix multiplier derives colors from RGB via the following formula:

$$Y = .299R + .587G + .114B$$

$$C_R = R - Y$$

$$C_B = B - Y$$

In this formula, Y represents luminance; red and blue are derived from it:  $C_R$  denotes red and (V),  $C_B$  denotes blue. V corresponds to  $C_R$ ; U corresponds to  $C_B$ . The U and V signals are carried on the same bandwidth. This system is sometimes referred to as Y/R-Y/B-Y.

The name for this color encoding method is YUV, despite the fact that the order of the signals according to the formula is YVU.

---

# Index

## A

allocation, framebuffer, 23-24  
alpha, 16  
  and Sirius Video, 36  
  channel, 20-21  
  and Sirius Video, 21  
  dedicated analog alpha out, 21

## B

bandwidth, 7, 24, 25  
  and channel assignment, 9  
  and channel boundaries, 18-19  
  and Encoder channel, 29  
  DAC output, 39  
  read/write, 39  
  *See also* allocation  
  transmission, 38-39  
Bandwidth Used scale, 7, 10

## C

channel  
  aligning, 13  
  associating with X window, 13  
  attributes, modifying, 14-21  
  copying, 12-13  
  delete from combination, 22  
  menu, 14  
  rectangle

  moving and resizing, 12, 16  
  restoring origin and size, 14  
  window, 14

Channel Attributes window, 14  
color field sequential setting, 17  
combination. *See* video format combination

Combiner, 1

  GUI  
  loading, 5  
  main window, 6, 11  
  setting display, 5

composite output, Encoder channel, 32

conventions, xii

cursor, 24  
  priority, 18

## D

dither, 17  
*Download Combination*, 8

## E

EEPROM  
  and Emergency Backup Combination, 40  
  saving to, 26

Emergency Backup Combination, 26, 40

Encoder channel, 29-32  
  attributes, modifying, 31-32

- composite output, 32
- dependent mode, 30
- filter, specifying, 32
- independent mode, 30
- output resolution, 31
- setting up, 30
- S-Video output, 32

Error indicator, 7

error messages, 41-45

## F

- field layout, 15
- filter, Encoder channel, 32
- framebuffer
  - allocation, 23-24
  - and channel placement, 18-19
  - and managed area, 6
    - for combination, 23
  - and output pixel format, 14
  - and raster, 2
  - and read/write bandwidth, 39
  - arrangement, 24-25
  - tiles, 18-19
    - for combination, 23

## G

- gain, 21
- gamma, 20
- Grab window*, 13
- graphics, initializing, 8, 24

## H

- horizontal phase, 19

## I

- InfiniteReality graphics, 1
- initializing graphics, 8, 24
- ircombine*, 1, 5, 8
  - on other Silicon Graphics hardware, 27
- IRIX version, xi

## M

- managed area, 6
  - for combination, 23

## P

- panning, 18-19
- pedestal, 17
- pixel depth, 24-26
- pixel format, 16-17
  - Sirius Video channel, 36

## R

- red line, 23
- Reduced Fill scale, 7
- Restore Output Size*, 14
- RGB10, 16
- RGB12, 16
- RGBA10, 16
- RM6 board, 7, 25, 27-28

## S

- setmon*, 8, 22, 26
- setup, 17

Sirius Video, 2, 33-36  
channel  
    attributes, modifying, 35-36  
    output formats, 36  
    pixel format, 36  
    setting up, 33-34  
dependent mode, 34  
independent mode, 33  
in Target Hardware window, 28  
*Snap to neighbor*, 13  
S-Video output, Encoder channel, 32  
swap rate, 15, 22, 37-38  
sync  
    channel attribute, 20  
    video format combination, 26

## T

Target Hardware window, 27-28  
textport, 24  
trilevel sync, 17

## U

*/usr/gfx/ucode/KONA/dg4/vfo/README*, 9

## V

vertical phase, 19  
video bandwidth, 25  
video format  
    changing for channel, 16  
    in Channel Attributes window, 14, 16  
    information, 9  
    origin and size in Channel Attributes window, 16  
    selecting for channel, 9

video format combination, 1, 2, 5-28  
    checking validity, 22  
    creating, 8-28  
    deleting channel, 22  
    description, 24  
    downloading, 8  
    global attributes, 23-26  
    querying, 3  
    samples, 8  
    saving, 26-27  
    testing, 22, 27-28  
    tradeoffs, 37-40  
Video Format Combiner. *See* Combiner

## X

**XSGIvcLoadVideoFormatCombination()**, 8  
XSGIvc X-server extension library, 3  
X window, associating with channel, 13

## Z

Z component, 17



## Tell Us About This Manual

As a user of Silicon Graphics documentation, your comments are important to us. They help us to better understand your needs and to improve the quality of our documentation.

Any information that you provide will be useful. Here is a list of suggested topics to comment on:

- General impression of the document
- Omission of material that you expected to find
- Technical errors
- Relevance of the material to the job you had to do
- Quality of the printing and binding

## Important Note

Please include the title and part number of the document you are commenting on. The part number for this document is 007-3279-001.

Thank you!

## Three Ways to Reach Us



The **postcard** opposite this page has space for your comments. Write your comments on the postage-paid card for your country, then detach and mail it. If your country is not listed, either use the international card and apply the necessary postage or use electronic mail or FAX for your reply.



If **electronic mail** is available to you, write your comments in an e-mail message and mail it to either of these addresses:

- If you are on the Internet, use this address: [techpubs@sgi.com](mailto:techpubs@sgi.com)
- For UUCP mail, use this address through any backbone site:  
*[your\_site]!sgi!techpubs*



You can forward your comments (or annotated copies of pages from the manual) to Technical Publications at this **FAX** number:

415 965-0964