

Multi-Channel Option Programmer's Guide

Document Number 007-1812-030

CONTRIBUTORS

Written by Carolyn Curtis

Illustrated by Dan Young and Carolyn Curtis

Edited by Cynthia Kleinfeld and Nan Schweiger

Production by Laura Cooper

Engineering contributions by Jeff Milo, Kirk Law, Gregory Eitzmann, Ed

Miskiewicz, Sam Gupta, Paul Spencer, John Hallesy, and Scott Pritchett

© Copyright 1994, Silicon Graphics, Inc.— All Rights Reserved

This document contains proprietary and confidential information of Silicon Graphics, Inc. The contents of this document may not be disclosed to third parties, 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 Unpublished rights reserved under the Copyright Laws of the United States.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 94039-7311.

Silicon Graphics and IRIS are registered trademarks and IRIX, RealityEngine, RealityEngine², SkyWriter, VTX, Crimson, Onyx, and POWER Series are trademarks of Silicon Graphics, Inc. Truevision is a registered trademark and VIDI/O BOX is a trademark of Truevision, Inc.

Contents

Introduction xi

- 1. Using the Multi-Channel Option** 1
 - The Multi-Channel Option and the Graphics Subsystem 2
 - Video Output Formats and Raster Memory Boards 4

- 2. Using *setmon*** 9
 - setmon Syntax 9
 - Setting Genlock 10
 - Cursor Position 11
 - Cursor Position Under IRIX 4.0.5H and 5.0.1 11
 - Cursor Position Under IRIX 5.1 and 5.2 11
 - Raster Memory Boards and Formats 12
 - Video Output Formats 13
 - Display Surface Dimensions 14
 - 2@1280x1024_60 16
 - 2@640x480_60+1@1280x1024_60 17
 - 2@1025x768_60 18
 - 3@1025x768_60 19
 - 3@960x680_60 21
 - 2@640x480_60 22
 - 4@640x480_60 23
 - 6@640x480_60 24
 - 2@640x486_30i 25
 - 4@640x486_30i 26
 - 6@640x486_30i 27
 - Programming Example 28

- 3. Using the Multi-Channel Option Panel 31**
 - Setting Channel-specific Controls 34
 - Setting Sync 34
 - Setting Setup 34
 - Setting Gamma 34
 - Setting Gain 35
 - Setting System-wide Controls 35
 - Genlock 35
 - Horizontal Phase 35
 - Group Controls 36
 - Saving Settings 36
 - Restoring Settings 36

- A. Hardware Specifications 39**
 - Electrical Levels 39
 - Sync Pair Outputs 40
 - Genlock Input 40
 - Format Timings 41
 - I/O Door Panel for Version with No Breakout Box 42
 - Breakout Box Connectors 43

- B. Multi-Channel Option Error Messages 45**
 - Messages Printed by IRIX Command Programs 45
 - Messages Printed during Boot or Found in SYSLOG 47

- C. Setting VME Addresses 55**
 - Setting VME Addresses for Crimson, POWER Series, or SkyWriter With IRIX 4.0.5H 56
 - Setting VME Addresses for Crimson, POWER Series, or SkyWriter With IRIX 5.1 58
 - Setting VME Addresses for Onyx With IRIX 5.0.1 or 5.1 61
 - Testing VME Addresses 62

D. Cabling the VIDI/O BOX 65

Glossary 67

Index 83

Figures

Figure 1-1	Multi-Channel Option and Graphics Subsystem	3
Figure 2-1	Display Surface for 2@1280x1024_60	16
Figure 2-2	Display Surface for 2@640x480_60+1@1280x1024_60	17
Figure 2-3	Display Surface for 2@1025x768_60	18
Figure 2-4	Display Surface for 3@1025x768_60	19
Figure 2-5	Display Surface for 3@960x680_60	21
Figure 2-6	Display surface for 2@640x480_60	22
Figure 2-7	Display Surface for 4@640x480_60	23
Figure 2-8	Display Surface for 6@640x480_60	24
Figure 2-9	Display Surface for 2@640x486_30i	25
Figure 2-10	Display Surface for 4@640x486_30i	26
Figure 2-11	Display Surface for 6@640x486_30i	27
Figure 3-1	Multi-Channel Option Panel	33
Figure A-1	I/O Door Panel for Rackmount Chassis (No Breakout Box)	42
Figure A-2	Multi-Channel Option Breakout Box	43
Figure D-1	Cabling the VIDI/O BOX	66

Tables

Table 1-1	Multi-Channel Option configurations	2
Table 1-2	Raster Memory and Pixel Depth	4
Table 1-3	Example Multi-Channel Option Video Formats	6
Table 1-4	Example Resolutions for a 12-bit RGBA Feature Set (12 Bits per RGBA (48 Bits), Double-buffered (+48 Bits), WIDs, etc. (+64 Bits) = 160 Bits)	7
Table 1-5	Example Resolutions for an 8-bit RGB Feature Set (8 Bits per RGB (24 Bits), Double-buffered (+24 Bits), WIDs, etc. (+64 Bits) = 112 Bits)	8
Table 2-1	Example Multi-Channel Option Format Specifications	13
Table 3-1	Factory Settings for Panel	37
Table A-1	Electrical Levels	39
Table A-2	Timings for Video Output Formats	41
Table A-3	Multi-Channel Option Breakout Box Connectors	43
Table C-1	Default VME Addresses for Crimson, POWER Series, or SkyWriter Running IRIX 4.0.5H	56
Table C-2	Default VME Addresses for Crimson, POWER Series, or SkyWriter Running IRIX 5.1	58
Table C-3	Default VME Addresses and Adapter Numbers for Onyx Deskside or Rack Running IRIX 5.0.1 or 5.1	61

Introduction

The Multi-Channel Option is designed for users who need multiple viewports from within an application, such as visual simulation, virtual reality, or entertainment applications.

Use the following information as a quick reference for Multi-Channel Option.

- Available video output formats are listed in

```
/usr/gfx/ucode/RE/vs2/vof/{1rm,2rm,4rm}
```

where `1rm`, `2rm`, and `4rm` correspond to the number of Raster Memory (RM) boards in your system.

- To start Multi-Channel Option, follow these steps:

1. Run `setmon`:

```
setmon -S[g][t] format
```

where `format` is the name of the desired video format. For the Multi-Channel Option, the flag `-S` is required; note that it is uppercase. The flag `-g` genlocks to an external source; `-t` genlocks to an external source at TTL levels.

2. Switch to multi-channel mode:

```
/usr/gfx/stopgfx  
/usr/gfx/startgfx
```

When the Multi-Channel Option is running, RealityEngine™ graphics does not output to the standard output display.

- For multipipe systems, set the environment variable `DISPLAY` to the pipe number being configured. For example, to set pipe 0 to `2@1280x1024_60`, pipe 1 to `6@640x480_60`, and pipe 2 to `3@1024x768_60`:

```
$ setenv DISPLAY :0.0  
$ setmon -S 2@1280x1024_60
```

```
$ setenv DISPLAY :0.1
$ setmon -S 6@640x480_60
$ setenv DISPLAY :0.2
```

```
$ setmon -S 3@1025x768_60
$ /usr/gfx/stopgfx
$ /usr/gfx/startgfx
```

- For information on cursor control, see “Cursor Position” in Chapter 2, “Using setmon.”
- For instructions on using the Multi-Channel Option panel, including information on multipipe systems, see Chapter 3, “Using the Multi-Channel Option Panel.”

The Multi-Channel Option runs on these compute platforms:

- Crimson™ equipped with a RealityEngine graphics subsystem
- POWER Series™ equipped with RealityEngine graphics
- SkyWriter™ equipped with RealityEngine graphics
- Onyx™ systems equipped with a VTX™ or RealityEngine²™ graphics subsystem

Requirements are:

- IRIX™ version:
 - 4.0.5H, 5.1, or later for Crimson, POWER Series, or SkyWriter
 - 5.0.1, 5.1, or later for Onyx systems
- on any compute platform, at least one RM board, revision level 00101 or greater; 00102 or greater for field sequential formats

The option consists of a board that fits into a VME slot in the chassis, plus paddleboards, cables, I/O panels, and software.

The *Multi-Channel Option Owner's Guide* includes the following chapters and appendices:

- Chapter 1 "Using Multi-Channel Option" explains features of the option, including formats.
- Chapter 2 "Using *setmon*" explains how to use the *setmon* command to set output format.
- Chapter 3 "Using the Multi-Channel Option Panel" explains how to use the panel to fine-tune video outputs.
- Appendix A "Hardware Specifications" summarizes Multi-Channel Option technical information.
- Appendix B "Multi-Channel Option Error Messages" lists error messages with explanations and solutions.
- Appendix C "Setting VME Addresses" explains how to reset the Multi-Channel Option board address.
- Appendix D "Cabling the VIDI/O BOX" explains how to cable the VIDI/O BOX™ available from Truevision® (Indianapolis, IN) so that it works with the Multi-Channel Option.

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 IRIS® keyboard keys are printed in boldface typewriter font and enclosed in angle brackets, such as <Enter> or <Esc>.

Messages and prompts that appear on-screen are shown in typewriter font. Entries that are to be typed exactly as shown are in boldface typewriter font.

Using the Multi-Channel Option

The Multi-Channel Option is designed for users who need multiple viewports from within an application, such as visual simulation, virtual reality, or entertainment applications. Tightly coupled to the Silicon Graphics® graphics subsystem, the Multi-Channel Option converts digital information stored in the graphics framebuffer into a variety of multiple analog video signals (or channels), ranging from multiple high-resolution (1280 x 1024) to low-resolution outputs, such as VGA and RS-170. The output can then be displayed on additional monitors, projection devices, or both. Output can be genlocked to an external reference signal.

The Multi-Channel Option requires RealityEngine, VTX, or RealityEngine² graphics.

Each Multi-Channel Option can drive up to six independent channels from a single workstation, eliminating the need to purchase multiple systems and genlock the outputs together. A system with more than one RealityEngine or RealityEngine² graphics subsystem can support multiple Multi-Channel Option boards. The Multi-Channel Option can display the same format on all outputs or, depending on the system configuration and the format chosen, display two different formats simultaneously.

Each I/O panel mounted on the chassis door provides two channels of output. The Multi-Channel Option is available in the configurations summarized in Table 1-1.

Table 1-1 Multi-Channel Option configurations

Platform	Number of Channels	
	No Breakout Box	Breakout Box
Crimson or POWER Series	Up to 4	Up to 6
POWER Series rack system	Up to 6	Up to 12
SkyWriter rack system with 2 Multi-Channel Option boards	Up to 6	Up to 12
Onyx deskside system	Up to 4	Up to 6
Onyx rack system with 3 Multi-Channel Option boards	Up to 14	Up to 18

For each configuration, a separate I/O plate provides genlock loopthrough functionality. For installations not using multisync monitors, horizontal and vertical sync are provided on a separate I/O plate for each pair of channels.

This chapter explains

- the Multi-Channel Option and the graphics subsystem
- video output formats and Raster Memory boards

The Multi-Channel Option and the Graphics Subsystem

Designed to be an integral part of the graphics subsystem, the Multi-Channel Option requires no special programming interface (library) for operation. You use the standard system command *setmon* to specify the output format for the Multi-Channel Option to use when the graphics subsystem is restarted. Silicon Graphics supplies several combinations of formats with the Multi-Channel Option.

Because the Multi-Channel Option is treated as a full-fledged member of the graphics subsystem, lengthy transport delays do not occur; additive transport delay is less than 0.5 ms. Figure 1-1 shows how the Multi-Channel Option is integrated into system graphics.

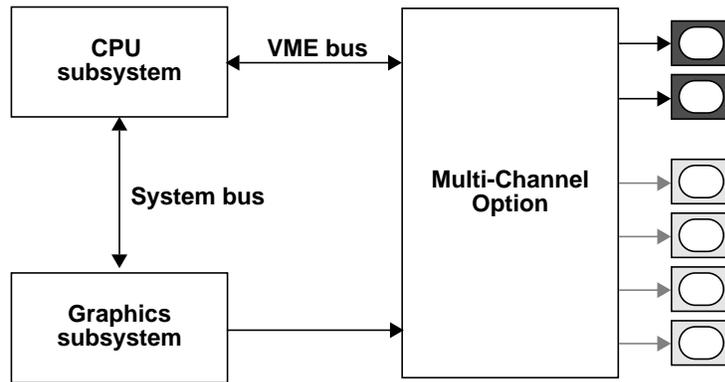


Figure 1-1 Multi-Channel Option and Graphics Subsystem

The Multi-Channel Option requires no adjustment other than specifying output format. For applications requiring sensitive customization, the Multi-Channel Option includes a graphical interface for changing video horizontal phase calibration, enabling video pedestal, making monitor-specific gamma corrections, and enabling or disabling sync and genlock. This interface is explained in Chapter 3, "Using the Multi-Channel Option Panel," later in this guide.

The number of screens into which the Multi-Channel Option divides the display surface is determined by several factors, including the Multi-Channel Option configuration that is present and the video and graphics bandwidth of the IRIS system. *Pixel depth*, that is, the number of bits per pixel, and antialiasing vary with the number of Raster Memory boards (RMs) installed.

Note: When the Multi-Channel Option is running, RealityEngine graphics does not output to the standard output display. ♦

Video Output Formats and Raster Memory Boards

The video output formats you can use are determined by the number of RM boards in the system, the video bandwidth of the system, and graphics drawing features. Graphics drawing features are related to pixel depth. Graphics drawing features include

- 12 bits per component
- double buffering
- stereo or quad buffering
- z-buffer use

Note: The default format for any configuration is the highest resolution format. ♦

Table 1-2 shows the relationship between pixel depth selected and the number of available RMs in the system.

Table 1-2 Raster Memory and Pixel Depth

Total RMs	Pixel Depth (Bits per Pixel)		
	Small Pixels (256 b/p)	Medium Pixels (512 b/p)	Large Pixels (1024 b/p)
1	1,310,000	655,000	328,000
2	2,621,000	1,310,000	655,000
4	5,243,000	2,621,000	1,310,000

Video transfer rates and architectural issues affect video formats. For example, a typical 12-bit RGBA feature set is 32-bit z, 12 bits each for R, G, B, and A (48 bits), and double buffering (48 bits), for a total of 128 bits. A typical 8-bit RGB feature set is 24-bit z, 8 bits each for R, G, and B (24 bits), and double buffering (24 bits), for a total of 72 bits. Video formats are those available from the RealityEngine graphics's standard video output (DG2).

The combination of availability of small, medium, or large pixel depth for a particular raster size and the desired features provides for the resulting capabilities of the RealityEngine graphics. The relationship of multisamples to small, medium, and large pixels is better understood from the perspective of noting how many bits it takes per feature. For example, given the feature set

- 12 bits per RGBA (+48 bits)
- double buffering (+48 bits)
- window IDs (WIDs), etc. (+64 bits)
- 4 multisamples of RGBA (+192 bits)
- 4 multisamples of 32-bit z-buffer (+128 bits)
- 4 multisamples of 8-bit stencil (+32 bits)

the total is 512 bits/pixel. Therefore, Table 1-2 shows that medium-sized pixels are needed (512 bits/pixel) since small pixels (256 bits/pixel) do not have enough depth to support the desired options. Display size resolutions are traded off with the pixel depth and, for a given pixel depth, more resolution is obtained with more RM boards. So, if a display resolution of 1280 x 1024 is desired with these features (pixel depth 1.31 mp), two RMs are necessary. If 960 x 680 is desired, this pixel depth (0.655 mp) can be achieved with only one RM.

The Multi-Channel Option can output video for up to six channels in one or two different formats. The output video channels are divided so that one video timebase governs two of the six Multi-Channel Option outputs and the other video timebase governs the other four outputs. Table 1-3 gives example formats for each timebase.

Note: The Multi-Channel Option can run at frame rates other than 60 Hz, but the video formats would be different. Currently available formats are listed in `/usr/gfx/ucode/RE/vs2/vof/{1rm,2rm,4rm}`, where *1rm*, *2rm*, or *4rm* denotes the number of RMs in the system. Chapter 2, "Using setmon," later in this guide explains the relationship between video output formats and RMs. ♦

Table 1-3 Example Multi-Channel Option Video Formats

Timebase 1 (2 ideo outputs)	Timebase 2 (4 video outputs)	Total Outputs	Total Pixels (millions)
2@1280 x 1024 60 Hz	N/A	2	2.621
2@640 x 480 60 Hz	1@1280 x 1024 60 Hz	3	1.925
2@1025 x 768 60 Hz	N/A	2	1.573
2@1025 x 768 60 Hz	1@1025 x 768 60 Hz	3	2.359
2@960 x 680 60 Hz	1@960 x 680 60 Hz	3	1.958
2@640 x 480 60 Hz	N/A	2	0.614
2@640 x 480 60 Hz	2@640 x 480 60 Hz	4	1.229
2@640 x 480 60 Hz	4@640 x 480 60 Hz	6	1.843
2@640 x 486 30 Hz	N/A	2	0.622
2@640 x 486 30 Hz	2@640 x 486 30 Hz	4	1.244
2@640 x 486 30 Hz	4@640 x 486 30 Hz	6	1.866

Table 1-4 shows the relationship of Multi-Channel Option example video resolutions to the number of RMs for a typical 12-bit RGBA feature set (32-bit z, 12 bits per RGBA (48 bits), double buffering (48 bits), for a total of 160 bits). Table 1-5 shows the relationship of Multi-Channel Option example video resolutions to the number of RMs for a typical 8-bit RGB feature set (24-bit z, 8 bits per RGB (24 bits), double buffering (24 bits), for a total of 112 bits).

Note: Currently available formats are listed in `/usr/gfx/ucode/RE/vs2/vof/{1rm,2rm,4rm}`.



Table 1-4 Example Resolutions for a 12-bit RGBA Feature Set (12 Bits per RGBA (48 Bits), Double-buffered (+48 Bits), WIDs, etc. (+64 Bits) = 160 Bits)

Antialiasing or Multisample Size	1 RM	2 RMs	4 RMs
0 multisamples: 160 + 32 (z) + 8 (stencil) = 200 b/p, or small pixels		2@1280x1024 2@640x480_1@1280x1024 2@1025x768 N/A 3@960x680	2@1280x1024 2@640x480_1@1280x1024 2@1025x768 3@1025x768 3@960x680
	2,4@640x480 2,4@640x486(int)	2,4,6@640x480 2,4,6@640x486 (int) ^a	2,4,6@640x480 2,4,6@640x486 (int)
4 multisamples: 160 + 192 (multicolor) + 128 (multi-z) + 32 (multistencil) = 512 b/p, or medium pixels	N/A		2@1280x1024 2@640x480_1@1280x1024 2@1025x768 3@960x680
		2,4,@640x480 2,4,@640x486 (int)	2,4,6@640x480 2,4,6@640x486 (int)
8 multisamples: 160 + 384 (multicolor) + 256 (multi-z) + 64 (multistencil) = 864 b/p, or large pixels	N/A	2@640x480 2@640x486 (int)	2,4@640x480 2,4@640x486 (int)
16 multisamples: exceeds maximum pixel depth	N/A	N/A	N/A

a. *int* stands for *interlaced*.

Table 1-5 Example Resolutions for an 8-bit RGB Feature Set (8 Bits per RGB (24 Bits), Double-buffered (+24 Bits), WIDs, etc. (+64 Bits) = 112 Bits)

Antialiasing or Multisample Size	1 RM	2 RMs	4 RMs
0 multisamples: 112 + 24 (z) + 8 (stencil) = 144 b/p, or small pixels		2@1280x1024	2@1280x1024
		2@640x480_1@1280x1024	2@640x480_1@1280x1024
		2@1025x768	2@1025x768
		N/A	3@1025x768
		3@960x680	3@960x680
	2,4@640x480	2,4,6@640x480	2,4,6@640x480
	2,4@640x486(int)	2,4,6@640x486 (int)	2,4,6@640x486 (int)
4 multisamples: 112 + 96 (multicolor) +96 (multi-z/stencil) = 304 b/p, or small pixels	N/A		2@1280x1024
			2@640x480_1@1280x1024
			2@1025x768
			3@960x680
			2,4,6@640x480
		2,4,6@640x486 (int)	
8 multisamples: 112 + 192 (multicolor) + 192 (multi-z/stencil) = 496 b/p, or medium pixels	N/A		2@1280x1024
			2@640x480_1@1280x1024
			2@1025x768
			3@1025x768 ^a
			3@960x680
		2,4@640x480	
		2,4,6@640x480	
		2,4@640x486 (int)	
		2,4,6@640x486 (int)	
16 multisamples: 112 + 384 (multicolor) + 384 (multi-z/stencil) = 880 b/p, or large pixels	N/A	2@640x480	2,4@640x480
		2@640x486(int)	2,4@640x486 (int)

a. Requires IRIX version 5.1 or later.

Using *setmon*

This chapter explains how to set the video formats that the Multi-Channel Option outputs. It describes

- *setmon* syntax
- setting genlock
- cursor position
- Raster Memory boards and formats
- video output formats
- display surface dimensions

This chapter concludes with a programming example.

***setmon* Syntax**

As an integral part of the graphics system, the Multi-Channel Option's video formats are set by the same method used to select the formats of RealityEngine graphics, namely, the *setmon* command.

Note: It is recommended that you use a dumb terminal or remotely log in over the network to switch video modes between standard and multi-channel.

To start the Multi-Channel Option, follow these steps:

1. Run *setmon*:

```
setmon -S[g][t] format
```

where *format* is the name of the desired video format, as explained later in this chapter. The flag *-S* is required; note that it is uppercase.

2. Switch to multi-channel mode:

```
/usr/gfx/stopgfx  
/usr/gfx/startgfx
```

Caution: When the Multi-Channel Option is running, RealityEngine graphics does not output to the standard output display.

After a power-off, system reset, or reboot, the system remains in multi-channel mode. To return to graphics mode, use:

```
setmon -x format  
/usr/gfx/stopgfx  
/usr/gfx/startgfx
```

Note: Occasional screen anomalies, such as shifting to right or left (tile shift) or bands of incorrectly placed or black information, may appear after graphics initialization. In such cases, restart graphics with the `/usr/gfx/stopgfx` and `/usr/gfx/startgfx` commands. ♦

Setting Genlock

The optional `setmon` flag `-g` sets genlock. By default, the master video output channel on the Multi-Channel Option uses its own horizontal frequency timebase; the option genlocks the master channel to an external sync source.

Genlock provides horizontal, vertical, and frame lock to the external signal. Because only timebase 1 is genlockable, `setmon -Sg` and `setmon -Sgt` genlock only channels 0 and 1 to the external source, regardless of the number of channels you are using. The phases of channels 0 and 1 are identical; their phase relationship to the external signal cannot be adjusted using the `setmon` command. You can, however, use the panel (see Chapter 3, “Using the Multi-Channel Option Panel,” in this guide) to adjust the relationship of their phase to the external genlock signal.

All the channels are vertically locked together, even if timebase 1 is not being genlocked to an external source.

Use the optional flag `-t` to specify that the external sync source is a TTL-level signal.

For more information on *setmon*, see its reference page.

Cursor Position

Cursor control varies, depending on the version of IRIX you are running.

Cursor Position Under IRIX 4.0.5H and 5.0.1

Use the mouse to set cursor position. Cursor position can be set in channel 0 only. To move the MCO panel, the toolchest, the console icon, and the console window into channel 0, follow these steps:

1. Copy `/usr/lib/X11/xdm/Xsession` as `.Xsession` to the home directory you use to log in.

2. To move the opened console window, change

```
/usr/sbin/startconsole -iconic
```

to

```
/usr/sbin/startconsole -iconic -p 0,0
```

3. To move the toolchest and console icons, edit the `.Xdefaults` file. Add:

```
ToolChest.geometry: +10-300
```

```
4Dwm*IconPlacement: left bottom right
```

Cursor Position Under IRIX 5.1 and 5.2

Cursor position can be set in any channel, except for field sequential and interlaced formats (for example, `6@640x486i`), where cursor control is not supported. Use the mouse to set cursor position.

Raster Memory Boards and Formats

Choose the video format for *setmon* based on the video formats available to your system, as determined by the number of channels your Multi-Channel Option supports and the number of RM boards on your system.

For example, to set two 1280 x 1024 displays at 60 Hz, which requires at least two RMs, the command is:

```
setmon -S 2@1280x1024_60
/usr/gfx/stopgfx
/usr/gfx/startgfx
```

Note: The RM boards must be at least revision level 00101 or greater for all formats except field sequential formats, which require 00102 or greater. ♦

The Multi-Channel Option video formats are listed in these directories:

```
/usr/gfx/ucode/RE/vs2/vof/{1rm,2rm,4rm}
```

where *1rm*, *2rm*, and *4rm* correspond to the number of RMs in your system.

For a description of your system, become superuser by using **su** and type

```
/usr/gfx/gfxinfo -v
```

This program tells you how many RM boards you have, their revision level, and whether a Multi-Channel Option board is being driven. If the MCO is being driven, the origins of each viewport are provided as well as the channel they drive. The program also provides other useful information about your graphics system. For example:

```
Graphics board 0 is "RE" graphics.
Managed (":0.0") 1280x2048
MCO Display 0 1280x1024 @ 60Hz, origin (0, 0)
MCO Display 1 1280x1024 @ 60Hz, origin (0, 1024)
8 GE(s) (GE rev. 0x7)
2 RM boards (rev. 00102/00102)
Small pixel depth
10-bit RGBA pixels
Driving Multi-Channel Option
```

Video Output Formats

Some video output formats are specified as shown in Table 2-1. Formats ending in *i* are interlaced.

Table 2-1 Example Multi-Channel Option Format Specifications

Format Specification	Active Pixel Area
2@1280x1024_60	Two screens, each 1280 x 1024, at 60 Hz
2@640x480_60+1@1280x1024_60	Two screens, each 640 x 480, at 60 Hz, and one screen 1280 x 1024
2@1025x768_60	Two screens, each 1024 x 768, at 60 Hz
3@1025x768_60	Three screens, each 1024 x 768, at 60 Hz
3@960x680_60	Three screens, each 960 x 680, at 60 Hz
2@640x480_60	Two screens, each 640 x 480, at 60 Hz (VGA)
4@640x480_60	Four screens, each 640 x 480, at 60 Hz (VGA)
6@640x480_60	Six screens, each 640 x 480, at 60 Hz (VGA)
2@640x486_30i	Two screens, each 640 x 486, at 30MHz, interlaced (RS-170 component)
4@640x486_30i	Four screens, each 640 x 486, at 30MHz, interlaced (RS-170 component)
6@640x486_30i	Six screens, each 640 x 486, at 30MHz, interlaced (RS-170 component)

Note: All formats available for each number of RM boards in a system are listed in the directories */usr/gfx/ucode/RE/vs2/vof/{1rm,2rm,4rm}*. ♦

Display Surface Dimensions

The X display surface for Multi-Channel Option video formats is contained in an area no larger than 2560 x 2048 pixels, that is, a total of 5.243 million pixels. The Multi-Channel Option places the origins of all viewports within this display surface so that the available pixels are utilized with the greatest possible efficiency. Architectural issues, such as the number of RMs present, data transfer bandwidths, and antialiasing or multisample possibilities, are taken into account.

Depending on the requirements of your application, you can draw data into the display surface for output from the Multi-Channel Option two ways:

- one large application window, which consumes the entire display surface for the output format chosen, and draws to the different origins described in this section to create the various viewports

You control the bounds of each viewport; all viewports are swapped at the same time.

- separate windows for each viewport

Your application controls the bounds of each viewport; each window, or viewport, is swapped independently, under the control of the application owning that window process.

This section gives examples of the screen origins for various output formats:

- 2@1280x1024_60
- 2@640x480_60+1@1280x1024_60
- 2@1025x768_60
- 3@1025x768_60
- 3@960x680_60
- 2@640x480_60
- 4@640x480_60
- 6@640x480_60
- 2@640x486_30i

- 4@640x486_30i
- 6@640x486_30i

Note: For more information on origins and display surface sizes, use the *gfxinfo* command (see “Raster Memory Boards and Formats,” earlier in this chapter). A programmatic interface that interprets the *gfxinfo* report is shown in the example program that concludes this chapter. Consult this report before using any formats. Silicon Graphics reserves the right to change origins as needed. ♦

For timing information, see Appendix A, “Hardware Specifications,” later in this guide.

2@1280x1024_60

Figure 2-1 shows the maximum single display surface (shaded) and origins for the viewports (unshaded) for the output format 2@1280x1024_60.

Note: The actual display surface is the largest rectangular area that encloses the viewports (shown in the figure as unshaded). Its size is 1280 x 2048. ♦

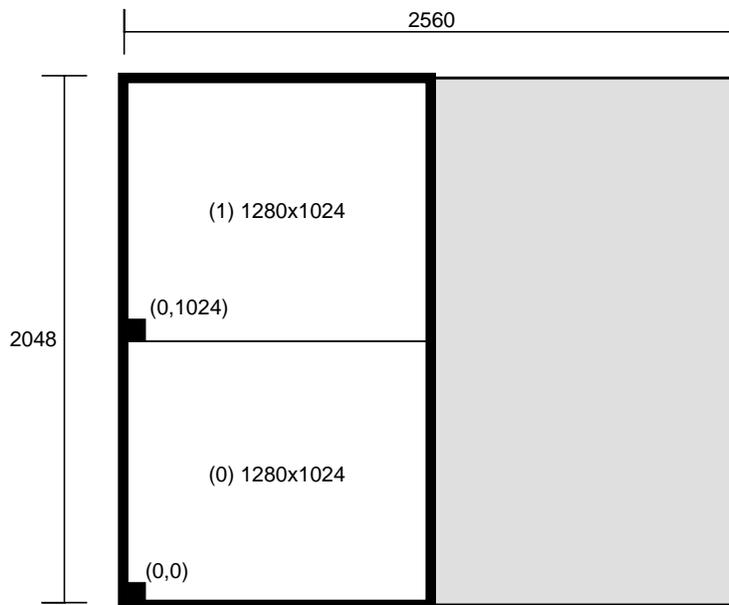


Figure 2-1 Display Surface for 2@1280x1024_60

Output of `gfxinfo -v` for this format is:

```
Managed 1280x2048
MCO Display 0 1280x1024 @ 60Hz, origin (0, 0)
MCO Display 1 1280x1024 @ 60Hz, origin (0, 1024)
8 GE (GE8 rev. 0x7/101)
2 RM4 boards (rev. 00102/00102)
Small pixel depth
10-bit RGBA pixels
Driving Multi-Channel Option
```

2@640x480_60+1@1280x1024_60

Figure 2-2 shows the maximum single display surface (shaded) and origins for the viewports (unshaded) for the output format 2@640x480_60+1@1280x1024_60.

Note: The actual display surface is the largest rectangular area that encloses the viewports (shown in the figure as unshaded). Its size is 1280 x 1504. ♦



Figure 2-2 Display Surface for 2@640x480_60+1@1280x1024_60

Output of `gfxinfo -v` for this format is:

```
Managed 1280x1504
MCO Display 0 640x480 @ 60Hz, origin (0, 0)
MCO Display 1 640x480 @ 60Hz, origin (640, 0)
MCO Display 2 1280x1024 @ 60Hz, origin (0, 480)
8 GE (GE8 rev. 0x7/101)
2 RM4 boards (rev. 00102/00102)
Small pixel depth
10-bit RGBA pixels
Driving Multi-Channel Option
```

2@1025x768_60

Figure 2-3 shows the maximum single display surface (shaded) and origins for the viewports (unshaded) for the output format 2@1025x768_60.

Note: The actual display surface is the largest rectangular area that encloses the viewports (shown in the figure as unshaded). Its size is 1025 x 1536. ♦

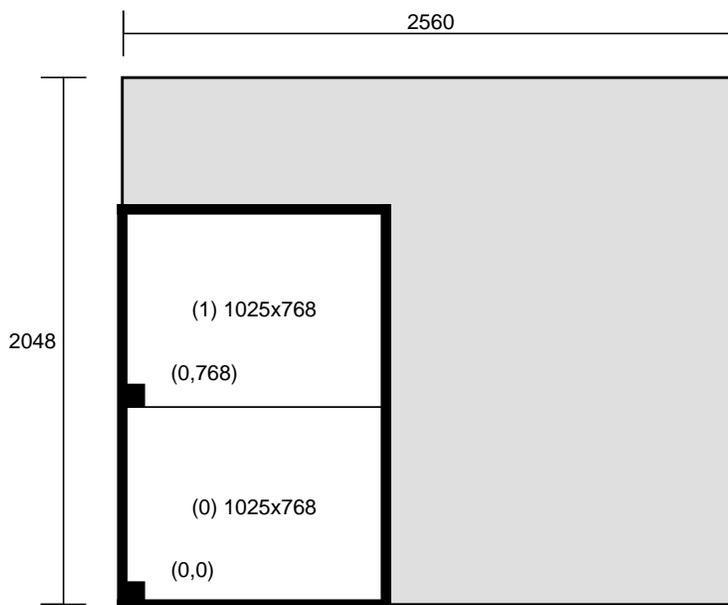


Figure 2-3 Display Surface for 2@1025x768_60

Output of `gxfinfo -v` for this format is:

```
Managed 1024x1536
MCO Display 0 1025x768 @ 60Hz, origin (0, 0)
MCO Display 1 1025x768 @ 60Hz, origin (0, 768)
8 GE (GE8 rev. 0x7/101)
4 RM4 boards (rev. 00102/00102/00102/00102)
Small pixel depth
10-bit RGBA pixels
Driving Multi-Channel Option
```

3@1025x768_60

Figure 2-4 shows the maximum single display surface (shaded) and origins for the viewports (unshaded) for the output format 3@1025x768_60.

The third viewport (labeled (2) in Figure 2-4) has its origin (offset) at 1030,0 because it must start on a ten-pixel boundary. This requirement results in an unused five-pixel area between the third viewport and the other two viewports.

Note: The actual display surface is the largest rectangular area that encloses the viewports (shown in the figure as unshaded). Its size is 2055 x 1536. †

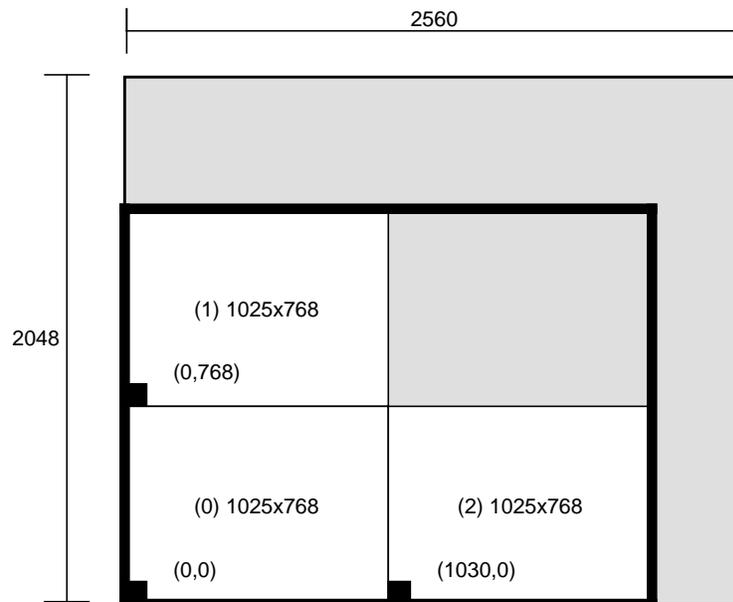


Figure 2-4 Display Surface for 3@1025x768_60

Output of `gxfxinfo -v` for this format is:

```
Managed 2055x1536
MCO Display 0 1025x768 @ 60Hz, origin (0, 0)
MCO Display 1 1025x768 @ 60Hz, origin (0, 768)
MCO Display 2 1025x768 @ 60Hz, origin (1030, 0)
8 GE (GE8 rev. 0x7/101)
```

```
4 RM4 boards (rev. 00102/00102/00102/00102)
Small pixel depth
10-bit RGBA pixels
Driving Multi-Channel Option
```

IRIX versions 5.0.1, 5.1, 5.1.1, and 5.1.1.2, you must manually reconfigure RealityEngine graphics organization to support 10-bit RGB without alpha. To do so, edit the file `/usr/bin/X11/X`. The file starts out with:

```
#!/bin/sh
#
# /usr/bin/X11/X
# initialize graphics device and start X Window System
# $Revision: 1.8 $
#
glGammaFile=/etc/config/system.glGammaVal
glGammaDefault="1.7"

if /etc/chkconfig windowssystem; then
    if [ -x /usr/gfx/gfxinit ]; then
        /usr/gfx/gfxinit
        if [ -r $glGammaFile -a -s $glGammaFile ] ; then
```

Modify this file to include these statements between lines 11 and 12:

```
PIXDENS=PIX32_RGB10
export PIXDENS
```

The beginning of the file should look like:

```
#!/bin/sh
#
# /usr/bin/X11/X
# initialize graphics device and start X Window System
# $Revision: 1.8 $
#
glGammaFile=/etc/config/system.glGammaVal
glGammaDefault="1.7"

if /etc/chkconfig windowssystem; then
    if [ -x /usr/gfx/gfxinit ]; then
        PIXDENS=PIX32_RGB10
        export PIXDENS
        /usr/gfx/gfxinit
        if [ $? -eq 0 -a -x /usr/bin/X11/Xsgi ]; then
```

3@960x680_60

Figure 2-5 shows the maximum single display surface (shaded) and origins for the viewports (unshaded) for the output format 3@960x680_60.

Note: The actual display surface is the largest rectangular area that encloses the viewports (shown in the figure as unshaded). Its size is 960 x 2040. †

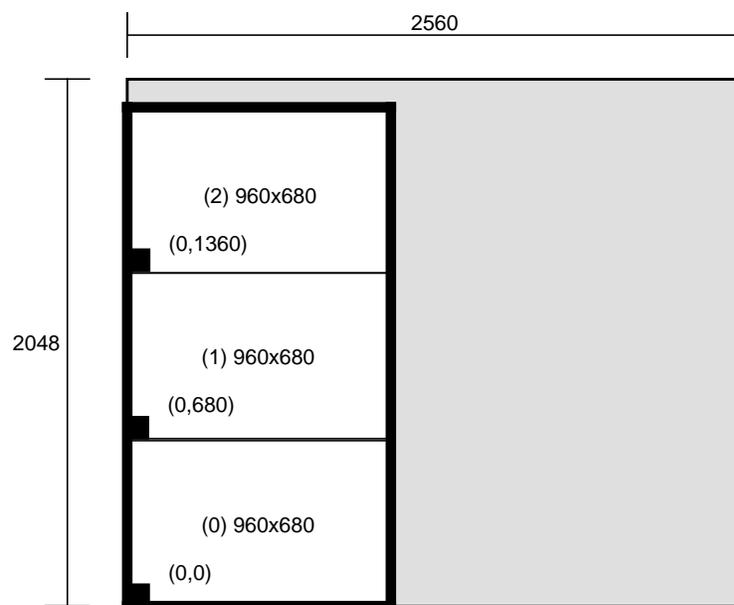


Figure 2-5 Display Surface for 3@960x680_60

Output of `gxfinfo -v` for this format is:

```
Managed 960x2040
MCO Display 0 960x680 @ 60Hz, origin (0, 0)
MCO Display 1 960x680 @ 60Hz, origin (0, 680)
MCO Display 2 960x680 @ 60Hz, origin (0, 1360)
8 GE (GE8 rev. 0x7/101)
2 RM4 boards (rev. 00102/00102)
Small pixel depth
10-bit RGBA pixels
Driving Multi-Channel Option
```

2@640x480_60

Figure 2-5 shows the maximum single display surface (shaded) and origins for the viewports (unshaded) for the output format 2@640x480_60.

Note: The actual display surface is the largest rectangular area that encloses the viewports (shown in the figure as unshaded). Its size is 640 x 960. ♦

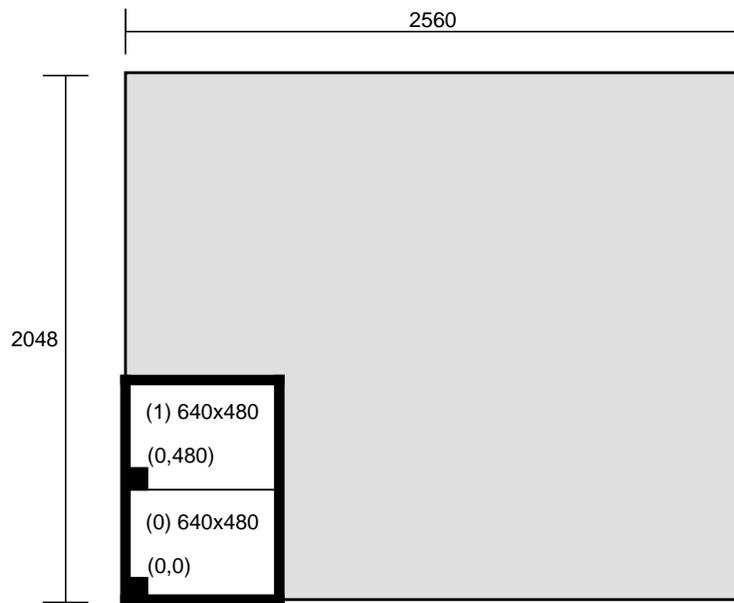


Figure 2-6 Display surface for 2@640x480_60

Output of `gfxinfo -v` for this format is:

```
Managed 640x960
MCO Display 0 640x480 @ 60Hz, origin (0, 0)
MCO Display 1 640x480 @ 60Hz, origin (0, 480)
8 GE (GE8 rev. 0x7/101)
2 RM4 boards (rev. 00102/00112)
Large pixel depth
10-bit RGBA pixels
Driving Multi-Channel Option
```

4@640x480_60

Figure 2-7 shows the maximum single display surface (shaded) and origins for the viewports (unshaded) for the output format 4@640x480_60.

Note: The actual display surface is the largest rectangular area that encloses the viewports (shown in the figure as unshaded). Its size is 1280 x 960. ♦

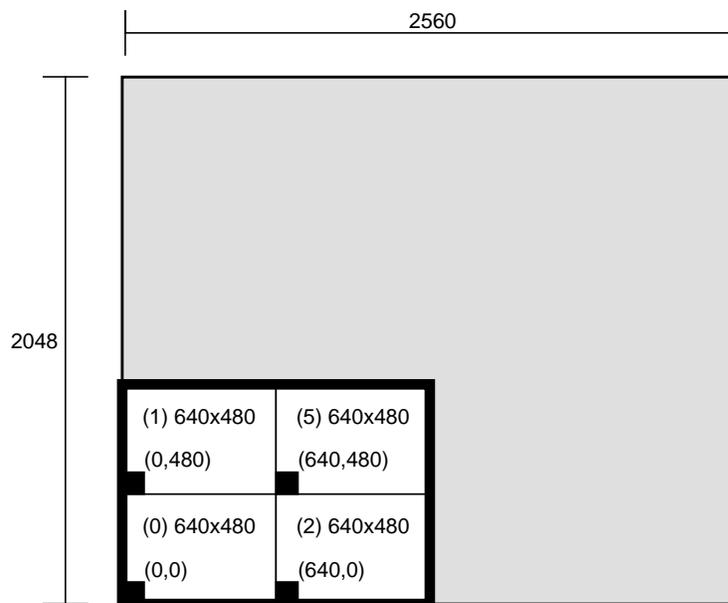


Figure 2-7 Display Surface for 4@640x480_60

Output of `gfxinfo -v` for this format is:

```
Managed 1280x960
MCO Display 0 640x480 @ 60Hz, origin (0, 0)
MCO Display 1 640x480 @ 60Hz, origin (0, 480)
MCO Display 2 640x480 @ 60Hz, origin (640, 0)
MCO Display 5 640x480 @ 60Hz, origin (640, 480)
8 GE (GE8 rev. 0x7/101)
2 RM4 boards (rev. 00102/00102)
Large pixel depth
10-bit RGBA pixels
Driving Multi-Channel Option
```

6@640x480_60

Figure 2-8 shows the maximum single display surface (shaded) and origins for the viewports (unshaded) for the output format 6@640x480_60.

Note: The actual display surface is the largest rectangular area that encloses the viewports (shown in the figure as unshaded). Its size is 1920 x 960. ♦

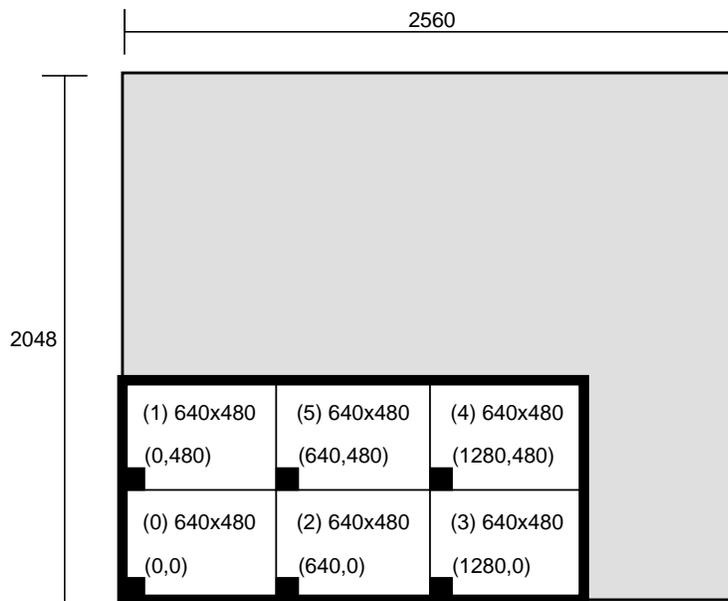


Figure 2-8 Display Surface for 6@640x480_60

Output of `gfbxinfo -v` for this format is:

```
Managed 1920x960
MCO Display 0 640x480 @ 60Hz, origin (0, 0)
MCO Display 1 640x480 @ 60Hz, origin (0, 480)
MCO Display 2 640x480 @ 60Hz, origin (640, 0)
MCO Display 3 640x480 @ 60Hz, origin (1280, 0)
MCO Display 4 640x480 @ 60Hz, origin (1280, 480)
MCO Display 5 640x480 @ 60Hz, origin (640, 480)
8 GE (GE8 rev. 0x7/101)
4 RM4 boards (rev. 00102/00112)
Large pixel depth
```

10-bit RGBA pixels
Driving Multi-Channel Option

2@640x486_30i

Figure 2-9 shows the maximum single display surface (shaded) and origins for the viewports (unshaded) for the output format 2@640x486_30i (RS-170).

Note: The actual display surface is the largest rectangular area that encloses the viewports (shown in the figure as unshaded). Its size is 640 x 972. †

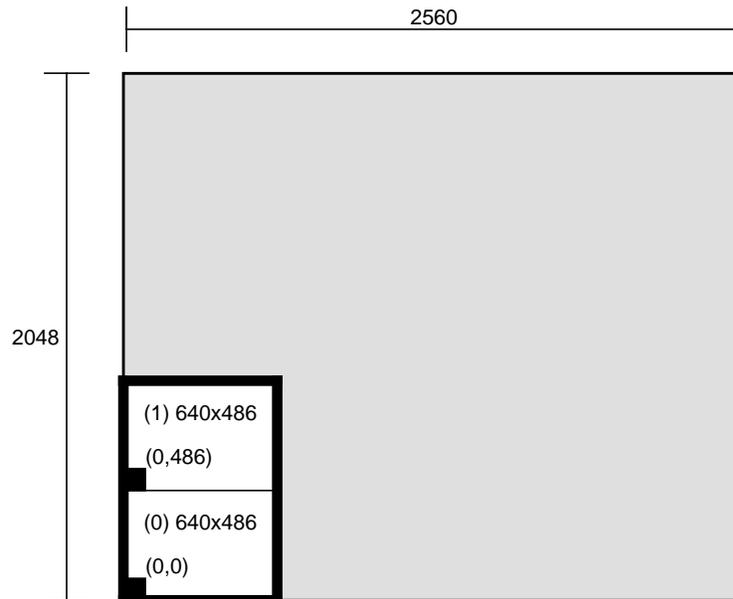


Figure 2-9 Display Surface for 2@640x486_30i

Output of `gfxinfo -v` for this format is:

```
Managed 640x972
MCO Display 0 640x486 @ 60Hz, origin (0, 0)
MCO Display 1 640x486 @ 60Hz, origin (0, 486)
8 GE (GE8 rev. 0x7/101)
```

2 RM4 boards (rev. 00102/00112)
Large pixel depth
10-bit RGBA pixels
Driving Multi-Channel Option

4@640x486_30i

Figure 2-10 shows the maximum single display surface (shaded) and origins for the viewports (unshaded) for the output format 4@640x486_30i (RS-170).

Note: The actual display surface is the largest rectangular area that encloses the viewports (shown in the figure as unshaded). Its size is 1280 x 972. ♦

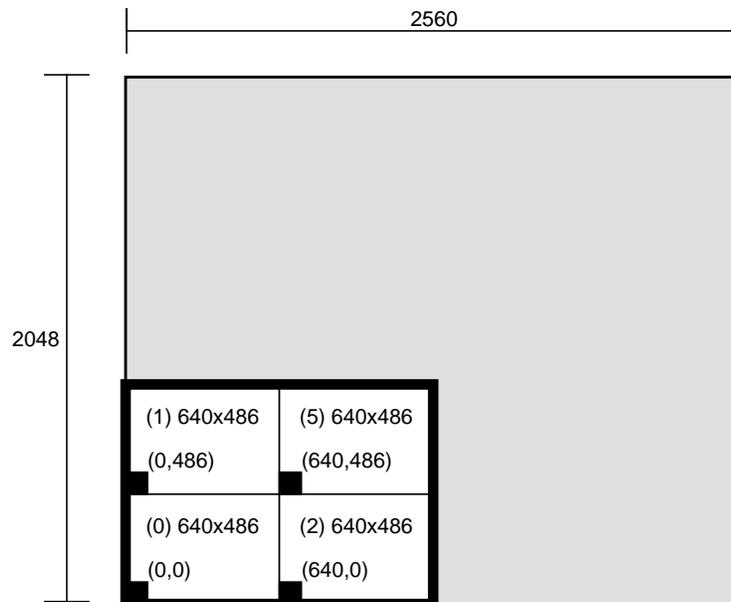


Figure 2-10 Display Surface for 4@640x486_30i

Output of `gfxinfo -v` for this format is:

```
Managed 1280x972
MCO Display 0 640x486 @ 60Hz, origin (0, 0)
MCO Display 1 640x486 @ 60Hz, origin (0, 486)
MCO Display 2 640x486 @ 60Hz, origin (640, 0)
```

```
MCO Display 5 640x486 @ 60Hz, origin (640, 486)
8 GE (GE8 rev. 0x7/101)
2 RM4 boards (rev. 00102/00112)
Large pixel depth
10-bit RGBA pixels
Driving Multi-Channel Option
```

6@640x486_30i

Figure 2-11 shows the maximum single display surface (shaded) and origins for the viewports (unshaded) for the output format 6@640x486_30i (RS-170).

Note: The actual display surface is the largest rectangular area that encloses the viewports (shown in the figure as unshaded). Its size is 1920 x 972. ♦

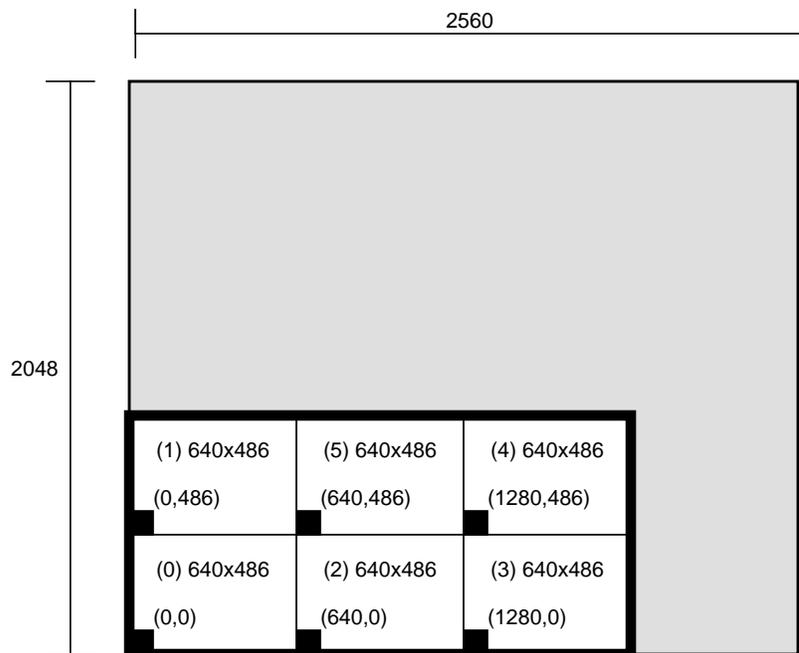


Figure 2-11 Display Surface for 6@640x486_30i

Output of `gfxinfo -v` for this format is:

```
Managed 1920x972
MCO Display 0 640x486 @ 60Hz, origin (0, 0)
MCO Display 1 640x486 @ 60Hz, origin (0, 486)
MCO Display 2 640x486 @ 60Hz, origin (640, 0)
MCO Display 3 640x486 @ 60Hz, origin (1280, 0)
MCO Display 4 640x486 @ 60Hz, origin (1280, 486)
MCO Display 5 640x486 @ 60Hz, origin (640, 486)
8 GE (GE8 rev. 0x7/101)
2 RM4 boards (rev. 00102/00112)
Large pixel depth
10-bit RGBA pixels
Driving Multi-Channel Option
```

Programming Example

This section contains an example program.

```
#include <stdio.h>
#include <device.h>
#include <gl.h>

/* This program will draw into all Displays allowed by the MCO */
/* First it gets Display number, x-y sizes, */
/* and x-y origins from gfxinfo. */
/* gfxinfo returns information in the following, example, format */
/* <blah-blah> */
/* Managed 1920x1024 */
/* MCO Display 0 1280x1024 @ 60Hz, origin (0, 0) */
/* MCO Display 2 1280x1024 @ 60Hz, origin (0, 1024) */
/* 8 GE(s) (GE rev. 0x7) */
/* 2 RM board(s) */
/* <blah-blah> */
/* Then it draws screen size and origin information into each MCO */
/* Display area */
main()
{
    FILE          *fp;
    char          hz_str[20];
    char          buf[100];
    int           num, sx, sy, xo, yo;
```

```
prefposition(0, getgdesc(GD_XPMAX) - 1, 0, getgdesc(GD_YPMAX) - 1);
noborder();
winopen("MCO test");
gconfig();
color(6);
clear();

fp = popen("/usr/gfx/gfxinfo", "r");

if (!fp) {
printf("Could not execute /usr/gfx/gfxinfo\n");
exit(0);
}
/* first search for 'MCO' followed by 'Display', */
/* then scan out the screen number, size and origin. */

while ((fscanf(fp, "%s", buf)) != EOF)
if (strcmp(buf, "MCO") == 0) {
fscanf(fp, "%s", buf);
if (strcmp(buf, "Display") == 0) {
fscanf(fp, "%d %dx%d @ %s origin (%d,%d)",
&num, &sx, &sy, hz_str, &xo, &yo);

/* draw into this screen area */

label_display(num, sx, sy, xo, yo);
}
}
pclose(fp);

while (!getbutton(ESCKEY))
sginap(1);

}

/* This routine prints information into the MCO screens as they */
/* are scanned from the gfxinfo text */
label_display(int num, int sx, int sy, int xo, int yo)
{
char tmp[200];

pushviewport();
viewport(xo, xo + sx, yo, yo + sy);
ortho2(0., 1.0 * sx, 0., 1.0 * sy);
color(num % 8);
```

```
clear();

sprintf(tmp, "Display %d size %d x %d at (%d,%d)", num, sx, sy, xo, yo);
color((num + 2) % 8);
cmov2i((sx - strlen(tmp) * 9) / 2, sy / 2);
charstr(tmp);
popviewport();
}
```

Using the Multi-Channel Option Panel

The Multi-Channel Option panel is a graphical user interface that enables you to set video parameters for individual output channels. The panel is a variation of the panel for RealityEngine graphics. If the system detects the presence of a Multi-Channel Option board, the panel appropriate for the Multi-Channel Option is displayed.

If your system has more than one Multi-Channel Option board, run the panel on each pipe; make settings for the appropriate channels in the panel running on that pipe.

Note: Because field sequential and interlaced formats (for example, 6@640x486_30i) under IRIX 5.1 do not have cursor control, and because display sync can be lost if an incorrect parameter is selected, it is recommended that you run the Multi-Channel Option panel on another Silicon Graphics workstation on your network. ♦

Call up the panel with

```
/usr/sbin/vout
```

Note: This command calls the Multi-Channel Option panel only if the workstation is in Multi-Channel Option mode; otherwise, the standard RealityEngine panel is displayed. ♦

If the system is running 4.0.5H, 5.0.1, or 5.1 and has more than one Multi-Channel Option, for example, on pipes 0 through 2, type

```
setenv DISPLAY :0.0
/usr/sbin/vout
setenv DISPLAY :0.1
/usr/sbin/vout
setenv DISPLAY :0.2
/usr/sbin/vout
```

If the system is running a release greater than 5.1, you can specify a pipe number. In this case, the `DISPLAY` variable only determines where the panel will be displayed and the pipe variable determines which pipe will be controlled. If no pipe variable is specified, the default is pipe 0. For example, for Multi-Channel Options on pipes 0 through 2, type

```
setenv DISPLAY :0.2
/usr/sbin/vout -p 0
setenv DISPLAY :0.2
/usr/sbin/vout -p 1
setenv DISPLAY :0.2
/usr/sbin/vout -p 2
```

In this example, all three `vout` panels would be displayed on pipe 2, but would each control a different pipe.

It is recommended that you display the panel on a remote system. For example, to control pipe 2 from a remote system named `remote_name`, log in remotely to the multipipe system and type

```
setenv DISPLAY <remote_name>:0
usr/sbin/vout -p 2
```

Figure 3-1 shows the panel.

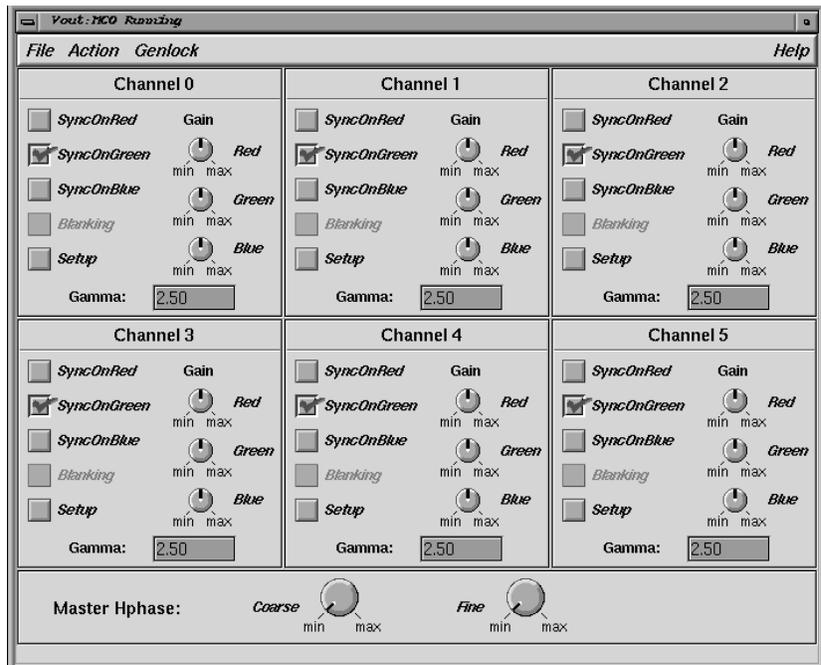


Figure 3-1 Multi-Channel Option Panel

Note: Settings in this figure may not match factory settings. Setting blanking is not implemented in this version. ♦

The segments of the panel allow you to set video options for each channel separately. A menu option allows you to apply settings from one segment to all active channels. Changes are reflected in video output as you make them.

This chapter explains

- setting channel-specific controls
- setting system-wide controls
- saving settings
- restoring settings

Note: Factory settings are listed in the section on restoring settings later in this chapter. ♦

Setting Channel-specific Controls

This section explains how to set channel-specific controls:

- sync
- setup
- gamma
- gain

Note: You can calibrate all channels at once by using the “Group Controls” option in the “Action” menu, as explained in “Setting System-wide Controls,” later in this chapter. ♦

Setting Sync

Select any or all of the three settings (“Sync on Red”, “Sync on Green”, and “Sync on Blue”).

Setting Setup

Click this box to enable setup, also known as pedestal. Setup is the difference between the blackest level displayed on the receiver and the blanking level; it is also defined as a black level that is elevated to 7.5 IRE instead of being left at 0.0 IRE, that is, 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.

Setting Gamma

Enter a gamma value in the text box to correct for inconsistency in the CRT.

The CRT is not linear with respect to voltage; the voltage-to-intensity characteristic is usually close to a power of 1.7. 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 1.7

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.

Setting Gain

Use the dials to set gain (brightness) for red, green, and blue output.

Setting System-wide Controls

The system-wide controls are:

- the Genlock menu item
- horizontal phase
- the “Group Controls” option

Genlock

Use two options in the Genlock menu item to set genlock on or off:

- Set “Genlock on” for a 1V (video) signal; this option is on by default.
- Set “TTL on” for a 4V (TTL) signal; this option is off by default.

Horizontal Phase

Use the Master Hphase dials at the bottom of the panel to set coarse and fine horizontal phase.

Note: Setting coarse horizontal phase to maximum is not recommended; loss of lock can occur. †

Group Controls

Use the “Group Controls” option in the Action menu to apply settings made for one channel to all outputs.

Saving Settings

Use the “Save Changes” option in the File menu to save your settings. Settings made in this panel are stored in an EEPROM on the Multi-Channel Option board.

When you start the Multi-Channel Option, the values that are last saved are automatically used. Thus, you must save settings in order for them to be in effect the next time you use the Multi-Channel Option.

Restoring Settings

Both factory settings and the user-defined settings last saved are stored on the Multi-Channel Option board’s EEPROM.

- To restore settings to the factory values, select “Restore Factory Values” in the File menu.
- To restore settings to those last saved, select “Restore User Values” in the File menu.

Factory settings are given in Table 3-1.

Table 3-1 Factory Settings for Panel

Setting	Value
Sync on red	Disabled
Sync on green	Enabled
Sync on blue	Disabled
Setup	Enabled
Gamma value	1.7
Gain for red, green, and blue	93%
Coarse horizontal phase	Minimum ^a
Fine horizontal phase	Minimum
Genlock	Disabled
TTL sync	Disabled
Group controls	Disabled

a. Setting horizontal phase to maximum is not recommended; loss of lock can occur.

Hardware Specifications

This appendix summarizes technical specifications for the Multi-Channel Option hardware:

- electrical levels
- sync pair outputs
- genlock input
- format timings
- I/O door panel for version with no breakout box
- breakout box connectors

Electrical Levels

Composite sync amplitude is rated into a 75-ohm load. Because calibration is not possible, levels can vary. The electrical levels at the RGB outputs are given in Table A-1.

Table A-1 Electrical Levels

Specification	Value
Video amplitude	Into 75 ohms: 1 V peak-to-peak with sync 0.714 V peak-to-peak without sync
Composite sync	TTL level into 75 ohms
Separate H and V sync	TTL level into 75 ohms

Sync Pair Outputs

Separate composite, horizontal, and vertical sync pair outputs are available for each MCO channel. Use the panel (see Chapter 3, “Using the Multi-Channel Option Panel,”) to turn sync on or off for each video output (for example, sync on green).

Genlock Input

The genlock input is a high-impedance input unless a 75-ohm terminator is attached to the second genlock BNC (**GEN OUT**). A 75-ohm terminator must be attached at the end of the line.

The genlock circuit of the Multi-Channel Option locks to the leading edge of the incoming sync. The Multi-Channel Option can lock to a composite video source or to a composite sync source.

The genlock input level is 1 V peak-to-peak or 2 V to 4 V peak-to-peak. The input signal must conform to the timing and level specifications of the Multi-Channel Option video output format specified using the *setmon* command, as explained in Chapter 2, “Using setmon,” earlier in this guide.

Format Timings

Table A-2 lists the timings for some Multi-Channel Option video output formats.

Table A-2 Timings for Video Output Formats

Format	Active Width and Height	Total Width and Height	Pixel Clock	Horizontal Frequency	Vertical Frequency and Format Type
2@1280x1024_60	1280 x 1024	1680 x 1065	107.352000 MHz	63.900 KHz	60 Hz noninterlaced
2@640x480_60+	640x480	800x532	25.536000 MHz	31.920 KHz	60 Hz noninterlaced
1@1280x1024_60	1280x1024	1680x1064	107.251200 MHz	63.840 KHz	60 Hz noninterlaced
2,3@1025x768_60	1025 x 768	1320 x 813	64.389600 MHz	48.780 KHz	60 Hz noninterlaced
3@960x680_60	960 x 680	1260 x 720	54.432000 MHz	43.200 KHz	60 Hz noninterlaced
2,4,6@640x480_60	640 x 480	800 x 525	25.200000 MHz	31.500 KHz	60 Hz noninterlaced
2,4,6@640x486_30i	640 x 486	780 x 525	12.285000 MHz	15.750 KHz	30 Hz interlaced (2 fields)

For information on display surface dimensions, see Chapter 2, “Using setmon,” earlier in this guide.

Breakout Box Connectors

Figure A-2 shows the Multi-Channel Option breakout box.

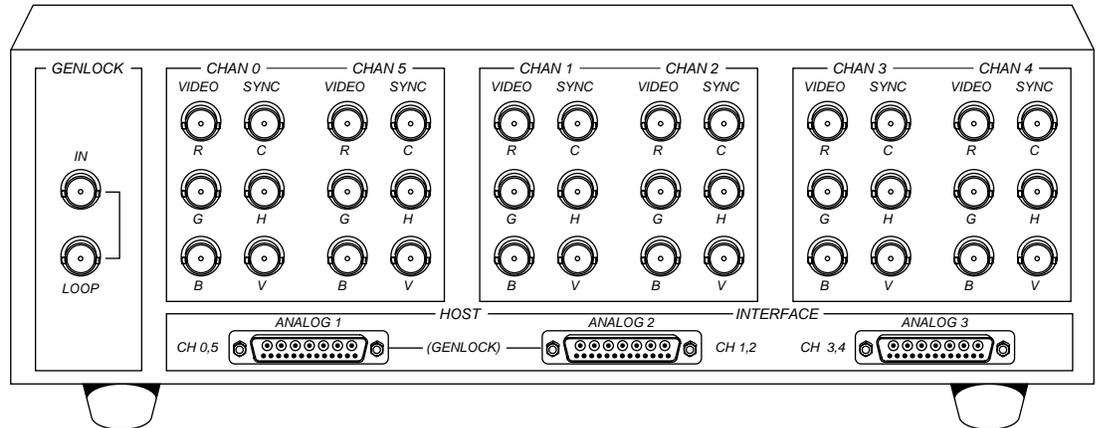


Figure A-2 Multi-Channel Option Breakout Box

Table A-3 summarizes breakout box connectors.

Table A-3 Multi-Channel Option Breakout Box Connectors

Section	Connector	Description
GENLOCK	IN	External sync source (house sync)
	LOOP	Loophrough for genlock input with buffered signal to workstation
CHAN 0, 5, 1, 2, 3, 4	VIDEO	Output to external video
	R	Red
	G	Green
	B	Blue
	SYNC	Output sync
	C	Composite sync
	H	Horizontal sync
	V	Vertical sync
HOST INTERFACE	ANALOG 1, 2, 3	Interface for Multi-Channel Option board via cables from I/O door panels

Multi-Channel Option Error Messages

This appendix lists error messages and their explanations and solutions. The error messages fall into two groups:

- messages printed by IRIX command programs
- messages printed during boot or found in *SYSLOG*

Messages Printed by IRIX Command Programs

Listed here are messages associated with programs that use or operate the MCO. Typically, only *setmon* and SSE diagnostics cause these errors.

ERROR: couldn't get number of boards

Explanation: The operating system prevented the software from manipulating the RealityEngine graphics board.

Solution: Check software revisions.

ERROR: couldn't open /dev/graphics

Explanation: The operating system prevented the software from manipulating the RealityEngine graphics board.

Solution: Check software revisions.

setmon: can't open vof file, filename

Explanation: The specified MCO format file is corrupt or is from an older revision of system software. For the 5.0.1 software release only, a CD containing the MCO video output formats (VOFs) was provided. As of release 5.1, the VOFs are incorporated into the standard system software subsystem package *ee1.sw.unix*. This error message appears if the older 5.0.1 VOFs are loaded on top of the 5.1 VOFs.

Solution: Identify and repair the file. If necessary, reinstall the *ee1.sw.unix* software.

setmon: cannot detect MCO in appropriate address. Is software installed properly (EEPROM written)?

Explanation: Some hardware responded at the specified VME address, but the EEPROM at that address did not contain the MCO identification sequence.

Solution: Ensure that the MCO has its EEPROM correctly loaded by running the `/usr/gfx/ucode/RE/vs2/vs2prom` program (or `/usr/gfx/ucode/REC/vs2/vs2prom` or `/usr/gfx/ucode/REV/vs2/vs2prom`, depending on whether you are running RealityEngine, Crimson, or Onyx).

setmon: internal error in filename

Explanation: The specified MCO format file was corrupt.

Solution: Identify and repair the file. If necessary, install the Multi-Channel Option software again.

setmon: no MCO board is present -- rerun setmon without -S

Explanation: Nothing responded in the address specified in the DG2 EEPROM (where the address of the MCO board is specified).

Solution: Check the address of the MCO as set on MCO board DIP switches for correctness. Use the `/usr/gfx/ucode/RE/vs2/vs2addrset` program (or `/usr/gfx/ucode/REC/vs2/vs2addrset` or `/usr/gfx/ucode/REV/vs2/vs2addrset`, depending on whether you are running RealityEngine, Crimson, or Onyx) to set and validate the address of the MCO in the DG2 EEPROM.

setmon: unexpected EOF in filename

Explanation: The specified MCO format file was not readable.

Solution: Identify and repair the file. If necessary, install the Multi-Channel Option software again.

The Multi-Channel Option hardware is present, but the EEPROM seems to be outdated or corrupt. Run setmon -S to repair it.

Explanation: Software has detected one of the following problems with the Multi-Channel Option EEPROM:

- the stored revision number does not match that which the software expects
- the size of the stored data does not match that which the software expects
- the checksum stored in the EEPROM is not internally consistent

Solution: Run *setmon -S* to rewrite the EEPROM.

VOF is not for VS2 (*magic_number*)

Explanation: The specified MCO format file did not contain the proper magic number. The magic number found in the file is shown.

Solution: Identify and repair the file. If necessary, install the Multi-Channel Option software again.

Messages Printed during Boot or Found in SYSLOG

Listed here are messages that can be printed during boot or that appear in the file */usr/adm/SYSLOG*. In the latter case, each message has the *gfxinit:* prefix.

Couldn't load DG2's special MCO VOF (*filename*): *reason*

Explanation: The special DG2 format file used when running MCO could not be loaded. The name of the file is printed, as is the reason the file could not be loaded.

Solution: Identify and repair the file. If necessary, install the Multi-Channel Option software again.

Could not open MCO VDRC file *filename*: *reason*

Explanation: The initialization software could not load a needed MCO file. The reason is specified.

Solution: Identify and repair the file. If necessary, install the Multi-Channel Option software again.

Error opening /dev/graphics in routine *routine-name* - *reason*

Explanation: The operating system prevented initialization from determining information about or initializing the graphics system.

Solution: Ensure that graphics have stopped before you attempt whatever caused this error. Check software revisions.

Failure in GFX_ATTACH_BOARD ioctl in *routine-name* - *reason*

Explanation: The operating system prevented initialization from determining information about or initializing the graphics system.

Solution: Ensure that graphics have stopped before you attempt whatever caused this error. Check software revisions.

Failure in GFX_MAPALL ioctl in *routine-name*

Explanation: The operating system prevented initialization from determining information about or initializing the graphics system.

Solution: Ensure that graphics have stopped before you attempt whatever caused this error. Check software revisions.

Failure in VENICE_VS2_SET_SHADOW ioctl in *routine-name* - *reason*

Explanation: An internal software error has caused a problem in the specified routine.

Solution: Report this problem to technical support.

graphics getboardinfo() failed during MCO initialization: *reason*

Explanation: The operating system prevented initialization from determining information about the graphics system.

Solution: Ensure that graphics have stopped before you attempt whatever caused this error. Check software revisions.

Invalid pipe number during MCO initialization

Explanation: A pipe number was specified that was out of range of available pipes.

Solution: Report this error to technical support.

MCO: bad base address *address*, using default

Explanation: The DG2 EEPROM contains an address for MCO that is invalid because of range or alignment.

Solution: Rewrite the address in the DG2 EEPROM with the `/usr/gfx/ucode/RE/vs2/vs2prom` program (or `/usr/gfx/ucode/REC/vs2/vs2prom` or `/usr/gfx/ucode/REV/vs2/vs2prom`, depending on whether you are running RealityEngine, Crimson, or Onyx).

MCO: board found at *address*

Explanation: The MCO board was successfully located and identified.

Solution: No resolution is necessary. Consider this a happy message.

MCO: board on pipe <pipe-number> found at <address>, VME adapter <Adapter>, but not properly connected to graphics via DVI cable

Explanation: An MCO was detected at the specified location. However, a check of the DVI cable between MCO and DG2 detected a problem.

Solution: Check connection of DVI cable between MCO and DG2. If connection appears correct, you can use the MCO Data Path diagnostic and then process of elimination to determine if the problem lies in the DG2 board, MCO board, paddleboard, or cabling.

MCO: can't map VME adapter number <address> (10 slot <slot> adapter <adapter>)

Explanation: The RealityEngine device driver was unable to map the specified adapter number on which it believed the MCO was located. This indicates an error in system software.

Solution: Contact technical support.

MCO hardware is present, but unusable because the number of RM boards has changed from *oldcount* to *newcount*. Rerun *setmon*.

Explanation: Each MCO format configuration is unique to the graphics hardware configuration. When *setmon* was run previously, the MCO's EEPROM was loaded with a configuration specified for the number of RM (Raster Memory) boards in place at that time. This message is printed when initialization detects that the number of RM boards has changed. Therefore, the format configuration in the MCO EEPROM is out of date.

Solution: Run *setmon* again, loading the MCO EEPROM properly.

MCO: unable to read signature from <address> (bad read from MCO eeprom)

Explanation: The address specified in the message is a valid address for a Multi-Channel Option, so the RealityEngine device driver attempted to read the MCO signature from that location. The MCO signature was not detected.

Solution: You will see this message whenever a device other than MCO is found in a potential address used by MCO; if another device does exist in that location, you may ignore this message.

If an MCO exists at that location, the on-board EEPROM is not responding as an MCO. Initialize the MCO board's EEPROM with the *vs2prom* program.

MCO: Using MCO on pipe <pipe-number> - address <address>, VME adapter <adapter>

Explanation: The RealityEngine device driver has successfully detected an MCO in the specified location.

Solution: None necessary. This behavior is correct.

Performing MCO initialization on hardware not a RealityEngine

Explanation: Initialization is being performed on a system with incorrect graphics installed; the MCO does not work.

Solution: Install RealityEngine graphics.

RE pipe <pipe-number>: can't read dg2 eeprom

Explanation: The RealityEngine device driver was unable to read the DG2 EEPROM to determine whether MCO should be enabled for output. Because the status of MCO could not be determined, output will be directed to DG2 until *setmon* is run.

Solution: Run *setmon* to reset the DG2 EEPROM.

The address of the MCO responds, but not as a MCO. Check EEPROM validity.

Explanation: A check was made to determine whether the MCO was present. The MCO identification sequence should be stored in the MCO's EEPROM. This MCO's EEPROM did not contain the identification sequence.

Solution: Initialize the MCO board's EEPROM with the *vs2prom* program. Check that the address of the MCO as specified in the DG2 EEPROM is valid; correct it, if necessary, with the */usr/gfx/ucode/RE/vs2/vs2addrset* program (or */usr/gfx/ucode/REC/vs2/vs2addrset* or */usr/gfx/ucode/REV/vs2/vs2addrset*, depending on whether you are running RealityEngine, Crimson, or Onyx).

The MCO hardware is present, but unusable because DG2 hardware is down-rev

Explanation: The MCO requires a DG2 of at least a certain revision level. The installed DG2 board revision is not sufficiently recent.

Solution: Install a DG2 board of recent revision.

The Multi-Channel Option hardware is present, but the EEPROM seems to be outdated or corrupt. Run setmon -S to repair it.

Explanation: Software has detected one of the following problems with the Multi-Channel Option EEPROM:

- the stored revision number does not match that which the software expects
- the size of the stored data does not match that which the software expects
- the checksum stored in the EEPROM is not internally consistent

Solution: Run *setmon -S* to rewrite the EEPROM.

The Multi-Channel Option VME address (<address>) as specified by RealityEngine device driver is unusable.

Explanation: The RealityEngine device driver specified a bad address for MCO.

Solution: Contact technical support.

Unable to find number of graphics boards during MCO initialization: reason

Explanation: The operating system prevented initialization from determining the number of graphics boards.

Solution: Check software revisions.

Unable to load filename in venice_get_vs2_xilinx_config() during MCO initialization: reason

Explanation: The specified format file could not be loaded for the reason detailed.

Solution: Identify and repair the file. If necessary, install the Multi-Channel Option software again.

Unable to map MCO during board probe (reason), assuming it is not present

Explanation: The initialization software was unable to map the MCO board for the specified reason.

Solution: Correct the specified problem. Check whether the file */usr/sysgen/master.d* still contains the entries describing the MCO address (shipped by default).

unable to malloc() for getboardinfo() during MCO initialization

Explanation: An internal error has kept initialization from proceeding.

Solution: Report this error to technical support.

Unable to open /dev/graphics during MCO initialization: reason

Explanation: The operating system prevented the software from manipulating the RealityEngine graphics board.

Solution: Check software revisions.

Unable to open device during initialization of MCO - reason

Explanation: The specified device is the means by which MCO initialization software accesses the MCO hardware. The device could not be opened for the reason specified.

Solution: Correct the problem as specified in the reason. If you cannot correct the situation, report this to technical support.

Unable to set diagnostic flag in routine-name - reason

Explanation: The operating system prevented initialization from determining information about or initializing the graphics system.

Solution: Ensure that graphics have stopped before you attempt whatever caused this error. Check software revisions.

venice_get_vs2_xilinx_config(): Error reading xilinx config file filename

Explanation: The initialization software could not load the specified microcode file.

Solution: Identify and repair the file. If necessary, install the Multi-Channel Option software again.

venice_get_vs2_xilinx_config: unknown xilinx type type

Explanation: The initialization software found a microcode file of the wrong type.

Solution: Identify and repair the file. If necessary, install the Multi-Channel Option software again.

vs2_loaducode(): can't open vof file, filename

Explanation: The specified format file could not be opened.

Solution: Identify and repair the file. If necessary, install the Multi-Channel Option software again.

vs2_loaducode(): internal error in filename

Explanation: The specified MCO format file was corrupt.

Solution: Identify and repair the file. If necessary, install the Multi-Channel Option software again.

vs2_loaducode(): unexpected EOF in filename

Explanation: The specified MCO format file was not readable.

Solution: Identify and repair the file. If necessary, install the Multi-Channel Option software again.

Setting VME Addresses

This appendix contains instructions on resetting the VME address for your Multi-Channel Option board, for example, if you install new operating system software. Resetting the board's VME address is required because system software replaces the memory mapping file with a default file. The instructions are dependent on the type of system you have (Crimson, POWER Series, SkyWriter, or Onyx), as well as the software release on your system (4.0.5H, 5.0.1, or 5.1).

Note: IRIX version 5.2 and later do not require setting VME addresses in software; the *gfxinit* program finds the address of the MCO board automatically.

This appendix explains:

- setting VME addresses for Crimson, POWER Series, or SkyWriter with IRIX 4.0.5H
- setting VME addresses for Crimson, POWER Series, or SkyWriter with IRIX 5.1
- setting VME addresses for Onyx with IRIX 5.0.1 or 5.1
- testing VME addresses

Setting VME Addresses for Crimson, POWER Series, or SkyWriter With IRIX 4.0.5H

Table C-1 shows the default VME base addresses for the Multi-Channel Option board in a in a Crimson deskside (pipe 0 only), POWER Series deskside or rack (pipe 0 only), or two-pipe SkyWriter rack running 4.0.5H software.

Table C-1 Default VME Addresses for Crimson, POWER Series, or SkyWriter Running IRIX 4.0.5H

Pipe	Value	Default VME Base Address
0	VIDEO_VS2_A24_NP_BASE_PIPE0	0xb2800000
1	VIDEO_VS2_A24_NP_BASE_PIPE1	0xb2810000

To set the address, follow these steps:

Note: Before installing new system software, determine the VME address at which the board should be set.

1. As root, for each pipe, type `/usr/gfx/setmon -x 60` to bring the DG2 EEPROM up to date. For example:

```
setenv DISPLAY :0.0
/usr/gfx/setmon -x 60
setenv DISPLAY :0.1
/usr/gfx/setmon -x 60
setenv DISPLAY :0.2
/usr/gfx/setmon -x 60
```

2. Set the MCO board address pointer(s) by writing DG2 EEPROM(s):

- On a deskside system, type as root

```
/usr/gfx/ucode/RE/vs2/vs2addrset -d -v
```

- On a multipipe system, type as root

```
/usr/gfx/ucode/RE/vs2/vs2addrset -p pipe# -a address
-v verbose
```

for each pipe.

The `pipe#` is the pipe number (0 or 1) and `address` is the address given in Table C-1. For example, on a two-pipe SkyWriter, type

```
/usr/gfx/ucode/RE/vs2/vs2addrset -p 0 -a 0xb2800000 -v
```

```
/usr/gfx/ucode/RE/vs2/vs2addrset -p 1 -a 0xb2810000 -v
```

3. If you are not using the default addresses given in Table A-1, you must edit `/usr/sysgen/master.d/mem`. The file looks like:

```
#include <sys/cpu.h>
#include <sys/videoaddr.h>

/*
 * This array defines the device addresses
 * that may be mapped by /dev/mmem.
 * Note: addresses MUST BE kernel virtual addresses,
 * not physical addresses.
 */
struct map mmap_addr[] = {
#if IP4 || IP5 || IP17
    { NBPP, SOUND_DAT_ADDR },          /* sound chip and friends */
#endif
#if IP12 || IP20
    { 0x20000, PHYS_TO_K1(HPC1MEMORY) }, /* DSP, etc. */
    /* HPC registers, in particular HPC1MISCSR */
    { NBPP, PHYS_TO_K1(HPC_0_ID_ADDR ) },
#endif
#if IP20 || IP22
    { NBPP, (PHYS_TO_K1(RPSS_CTR)&~0xfff) },
#endif
#if IP5 || IP17
    { 0x80000, LOGRAM_ADDR },          /* ECC logging RAM */
    { NBPP, IOTIMER_ADDR }, /* free running timer on IO3 */
    { 0x10000, VIDEO_VS2_A24_NP_BASE_PIPE0 }, /* Video Splitter 2 Pipe
0 */
    { 0x10000, VIDEO_VS2_A24_NP_BASE_PIPE1 }, /* Video Splitter 2 Pipe
1 */
    { 0x10000, VIDEO_VS2_A24_NP_BASE_PIPE2 }, /* Video Splitter 2 Pipe
2 */
#endif
#if IP5 || IP9 || IP17
    { 0x800000, 0xda000000 },          /* VME Vigra space for 2 boards */
#endif
    { 0, 0 } , /* NULL ENTRY TERMINATES THIS LIST! */
};
```

In `/usr/sysgen/master.d/mem`, change

```
VIDEO_VS2_A24_NP_BASE_PIPE0
```

or

VIDEO_VS2_A24_NP_BASE_PIPE1

to the virtual VME address you have set for the appropriate pipe.

4. Reconfigure the kernel; as root, type

```
autoconfig -f
```
5. Reboot the system to use the newly configured kernel; as root, type

```
init 6
```

Setting VME Addresses for Crimson, POWER Series, or SkyWriter With IRIX 5.1

Table C-2 shows the default VME base addresses for the Multi-Channel Option board in a Crimson deskside (pipe 0 only), in a POWER Series deskside or rack (pipe 0 only), or in a two-pipe SkyWriter rack running 5.1 software.

Table C-2 Default VME Addresses for Crimson, POWER Series, or SkyWriter Running IRIX 5.1

Pipe	Value	Default VME Base Address
0	VIDEO_VS2_A24_NP_BASE_PIPE0	0x800000
1	VIDEO_VS2_A24_NP_BASE_PIPE1	0x810000

To set the address, follow these steps:

Note: Before installing new system software, determine the VME address to which the board should be set.

1. As root, for each pipe, type `/usr/gfx/setmon -x 60` to ensure the DG2 EEPROM is up to date. For example:

```
setenv DISPLAY :0.0
/usr/gfx/setmon -x 60
setenv DISPLAY :0.1
/usr/gfx/setmon -x 60
setenv DISPLAY :0.2
/usr/gfx/setmon -x 60
```

2. Set the MCO board address pointer(s) by writing DG2 EEPROM(s):

- On a deskside system, type as root

```
/usr/gfx/ucode/RE/vs2/vs2addrset -d -v
```

- On a multipipe system, type as root:

```
/usr/gfx/ucode/RE/vs2/vs2addrset -p pipe# -a address  
-v verbose
```

for each pipe.

The pipe# is the pipe number (0 or 1) and address is the address given in the previous table. For example, on a two-pipe SkyWriter, type:

```
/usr/gfx/ucode/RE/vs2/vs2addrset -p 0 -a 0x800000 -v  
/usr/gfx/ucode/RE/vs2/vs2addrset -p 1 -a 0x810000 -v
```

3. If you are not using the default addresses given in Table C-2, you must edit `/var/sysgen/master.d/mem`. The file looks like:

```
*#ident "@(#)kern-port:master.d/mem 10.2"  
*#ident "$Revision: 3.21 $"  
*  
* MEM  
*  
*FLAG PREFIX SOFT #DEV DEPENDENCIES  
orcs mm 1  
  
$$$  
  
#include <sys/cpu.h>  
  
#if IP5 || IP17  
#include <sys/videoaddr.h>  
#endif  
  
/*  
 * This array defines the device addresses  
 * that may be mapped by /dev/mmem.  
 * Note: addresses MUST BE kernel virtual addresses,  
 * not physical addresses.  
 */  
struct map mmmmap_addrs[] = {  
#if IP4 || IP5 || IP17  
 { NBPP, SOUND_DAT_ADDR, }, /* sound chip and friends */  
#endif
```

```

#if IP12 || IP20
  { 0x20000, PHYS_TO_K1(HPC1MEMORY), }, /* DSP, etc. */
  /* HPC registers, in particular HPC1MISCSR */
  { NBPP, PHYS_TO_K1(HPC_0_ID_ADDR ), },
#endif
#if IP20 || IP22
  { NBPP, (PHYS_TO_K1(RPSS_CTR)&~0xffff) },
#endif
#if IP5 || IP17
  { 0x80000, LOGRAM_ADDR, }, /* ECC logging RAM */
  { NBPP, IOTIMER_ADDR, }, /* free running timer on IO3 */

/* Sirius Video */
  { 0x1000, 0xb7c03000 },

/* Video Splitter 2 (aka Multi Channel Option) Pipe 0 */
  { 0x10000, VIDEO_MP_VME_A24NPBASE +
VIDEO_VS2_A24_NP_BASE_PIPE0 },
/* Video Splitter 2 (aka Multi Channel Option) Pipe 1 */
  { 0x10000, VIDEO_MP_VME_A24NPBASE +
VIDEO_VS2_A24_NP_BASE_PIPE1 },
/* Video Splitter 2 (aka Multi Channel Option) Pipe 2 */
  { 0x10000, VIDEO_MP_VME_A24NPBASE +
VIDEO_VS2_A24_NP_BASE_PIPE2 },
#endif
#if IP19
  { NBPP, EV_RTC, }, /* free running timer on Everest */
#endif
#if DEBUG
  { 0x1000, 0xa0000000 }, /* Allow user to map page for HW
trigger */
#endif
#endif
  { 0, 0 } , /* NULL ENTRY TERMINATES THIS LIST! */
};

```

In */var/sysgen/master.d/mem*, change:

```
VIDEO_VS2_A24_NP_BASE_PIPE0
```

or

```
VIDEO_VS2_A24_NP_BASE_PIPE1
```

to the physical VME address you have set for the appropriate pipe.

4. Reconfigure the kernel; as root, type
`autoconfig -f`
5. Reboot the system to use the newly configured kernel; as root, type
`init 6`

Setting VME Addresses for Onyx With IRIX 5.0.1 or 5.1

Table C-3 shows the default VME base addresses for the Multi-Channel Option board in a three-pipe Onyx rack or in a deskside Onyx system (pipe 0 only) running 5.0.1 software, 5.1 software.

Table C-3 Default VME Addresses and Adapter Numbers for Onyx Deskside or Rack Running IRIX 5.0.1 or 5.1

Pipe	Default VME Base Address	Default VME Bus Adapter Number
0	0x800000	0
1	0x800000	36
2	0x800000	37

Before installing new system software, determine the VME address to which the board should be set. Run `hinvt` to display the bus adapters needed for each MCO. For example, a single-pipe system requires adapter 0, a two-pipe system requires adapters 0 and 36, and a three-pipe system requires adapters 0, 36, and 37.

To set the address, follow these steps:

Caution: Do not attempt to operate an MCO in a system that does not report the correct adapter number for the pipe it is associated with (for example, `pipe_0` to `adapter_0`, `pipe_1` to `adapter_36`, and `pipe_2` to `adapter_37`).

1. As root, type
`cd /dev`
`MAKEDEV usrvme`
2. As root, for each pipe, type `/usr/gfx/setmon -x 60` to bring the DG2 EEPROM up to date. For example:

```
setenv DISPLAY :0.0
/usr/gfx/setmon -x 60
setenv DISPLAY :0.1
/usr/gfx/setmon -x 60
setenv DISPLAY :0.2
/usr/gfx/setmon -x 60
```

3. Set the MCO board address pointer(s) by writing DG2 EEPROM(s):

- On a deskside system, type as root
`/usr/gfx/ucode/RE/vs2/vs2addrset -d -v`
- On a multipipe system, type as root
`/usr/gfx/ucode/RE/vs2/vs2addrset -p pipe# -a address
-b bus_adapter# -v verbose`
for each pipe.

The `pipe#` is the pipe number (0, 1, or 2) and `address` is the address given in Table C-3. For example, on a three-pipe Onyx rack, type

```
/usr/gfx/ucode/RE/vs2/vs2addrset -p 0 -a 0x800000 -b 0 -v
/usr/gfx/ucode/RE/vs2/vs2addrset -p 1 -a 0x800000 -b 36 -v
/usr/gfx/ucode/RE/vs2/vs2addrset -p 2 -a 0x800000 -b 37 -v
```

4. Reconfigure the kernel; as root, type

```
autoconfig -f
```

5. Reboot the system to use the newly configured kernel; as root, type

```
init 6
```

Testing VME Addresses

After you finish setting addresses, running `hinv` should report

```
Multi-Channel Option board installed
```

for each MCO installed.

Caution: Do not attempt to run `setmon -S` on a pipe where the above command does not report properly.

Running `vs2prom -p pipe# -r /dev/null` on each pipe indicates the health of each MCO VME connection by attempting to read the contents of the MCO EEPROM. If the command is successful, a prompt with no message appears. Test for each pipe with an installed MCO.

For example, as root, type

```
/usr/gfx/ucode/RE/vs2/vs2prom -p 0 -r /dev/null  
/usr/gfx/ucode/RE/vs2/vs2prom -p 1 -r /dev/null  
/usr/gfx/ucode/RE/vs2/vs2prom -p 2 -r /dev/null
```

Caution: Do not attempt to run `setmon -s` on a pipe where the above command does not succeed without complaint.

Cabling the VIDI/O BOX

This appendix contains instructions for cabling the VIDI/O BOX available from Truevision (Indianapolis, IN). This option converts RGB output to NTSC output using the *n@640x486_30i* video output format; one VIDI/O BOX is required for each output channel.

To cable this option for use with the Multi-Channel Option, follow these steps:

1. Daisy-chain the MCO green output (sync on green) to the encoder green input, the decoder video input, and the sync input.
2. Set the termination switch for the sync input to **75**.
3. Connect red and blue normally. Set the termination switches to **75**. No connection to the MCO genlock is required.

Figure D-1 diagrams these connections.

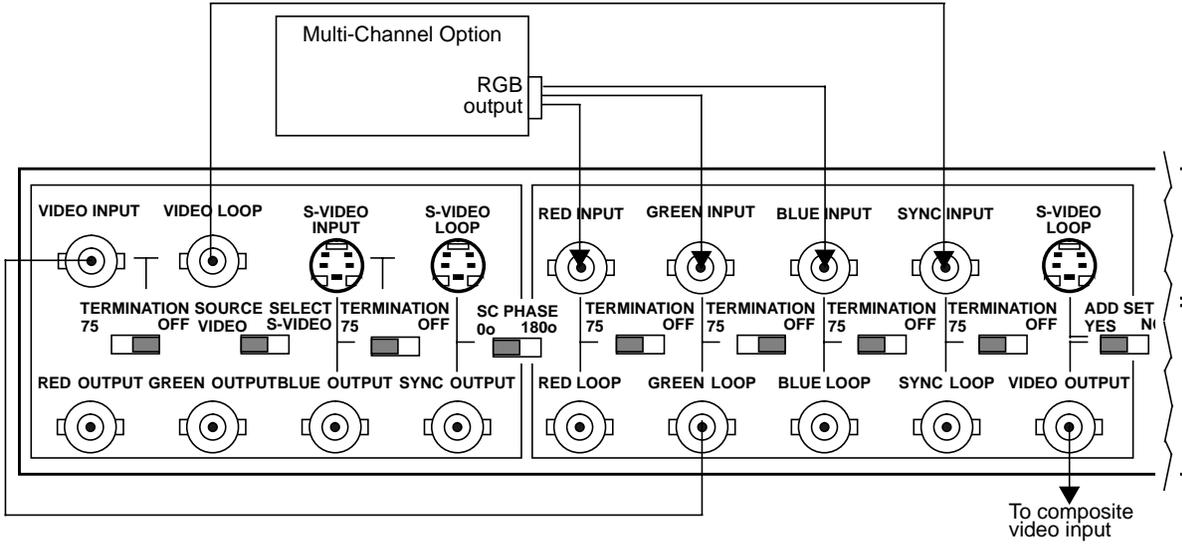


Figure D-1 Cabling the VIDI/O BOX

Glossary

active video

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

aliasing

A rendering technique that assigns to pixels the color of the primitive being rendered, regardless of whether that primitive covers all of the pixel's area or only a portion of the pixel's area. This results in jagged edges, or jaggies. In video systems, aliasing results when an image is sampled that contains frequency components above the Nyquist limit for the sampling rate. *See also* Nyquist limit.

alpha, alpha value

The fourth color component of a pixel, specifying its opacity, translucency, or transparency. The alpha component is typically used to control color blending, and in video systems, is treated as a separate signal output. By convention, OpenGL alpha corresponds to the notion of opacity rather than transparency, meaning that an alpha value of 1.0 implies complete opacity, and an alpha value of 0.0 complete transparency.

alpha plane

A bank of memory that stores alpha values; the values are 8 bits per pixel.

alpha register

A register that stores an alpha value.

alpha value

See alpha.

antialiasing

A rendering technique that assigns pixel colors based on the fraction of the pixel's area that's covered by the primitive being rendered. Antialiased rendering reduces or eliminates the jaggies that result from aliased rendering.

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 image. For example, the standard aspect ratio for television is 4:3. Maintaining the original aspect ratio of an image prevents it from being distorted.

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.

bitblt

Bit block transfer; logical operations on bitmaps.

bit map

A region of memory that contains the pixels representing an image. The pixels are arranged in the sequence in which they are normally scanned to display the image.

bit plane

A rectangular array of bits mapped one-to-one with pixels. The framebuffer is a stack of bit planes.

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 interval 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.

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 (R minus 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.

CCIR 601

The digital interface standard developed by the CCIR (Comité Consultatif International de Radiodiffusion, or 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, also referred to as 4:2:2, sets parameters for both 525-line and 625-line systems.

chroma

See chrominance.

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.

color burst

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 the 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.

Also called burst and burst flag. *See also* color subcarrier and video waveform.

color space

A space defined by three color components, such as R, G, and B.

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. The color subcarrier's frequency is defined as 3.58 MHz in NTSC and 4.43 MHz in PAL. *See also* color burst.

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.

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.

field

One of two (or more) equal parts of information into 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 averaging

A filter that corrects flicker by averaging pixel values across successive fields. *See also* flicker.

flicker

The effect caused by a one-pixel-deep line in a high-resolution graphics frame that is output to a low-resolution monitor, because the line is in only one of the alternating fields that make up the frame. This effect can be filtered out by field averaging. *See also* field and frame.

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.

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 interval, and video waveform.

gamma correction

A function applied to colors stored in the framebuffer to correct for the nonlinear response of the eye (and sometimes of the monitor) to linear changes in color-intensity values.

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 no master sync is available, VideoFramer, for example, can be set to “free run” (or standalone) mode, so that it becomes the master timing device to which other devices sync. *See also* line lock.

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.

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 interval and video waveform.

HSI

See hue-saturation-intensity.

HSV

Hue-saturation-value. *See* hue-saturation-intensity.

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.

hue-saturation-intensity

A tristimulus color system based on the parameters of hue, saturation, and intensity (luminance). Also referred to as HSI or HSV.

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 0.0 to 1.3 MHz.

image plane

See bit plane.

image processing

Manipulating an image by changing its color, brightness, shape, or size.

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, the even lines. *See also* field and frame.

IRE units

A scale for measuring analog video signal levels, normally starting at the bottom of the horizontal sync pulse and extending to the top of peak white. Blanking level is 0 IRE units and peak white level is 100 IRE units (700 mV). An IRE unit equals 7.14 mV (+100 IRE to -40 IRE = 1 V). IRE stands for Institute of Radio Engineers, a forerunner of the IEEE.

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.

level

Signal amplitude.

line

The result of a single pass of the sensor from left to right across the image.

line lock

Input timing derived from the horizontal sync signal and 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 with the horizontal sync signal. *See also* video waveform.

linear matrix transformation

The process of combining a group of signals through addition or subtraction; for example, RGB signals into luminance and chrominance signals.

luma

See luminance.

luminance

The perceived brightness of a surface. Typically refers to a weighted average of red, green, and blue color values that gives the perceived brightness of the combination. For video systems, luminance is the video signal that describes the amount of light in each pixel. *See also* chrominance and Y signal.

multiburst

A test pattern consisting of sets of vertical lines with closer and closer spacing; used for testing horizontal resolution of a video system.

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 uses a total of 525

horizontal lines per frame, with two fields per frame of 262.5 lines each. Each field refreshes at 60 Hz (actually 59.94 Hz).

Nyquist limit

The highest frequency of input signal that can be correctly sampled without aliasing. The Nyquist limit is equal to half of the sampling frequency.

overscan

To scan a little beyond the display raster area of the monitor so that the edges of the raster are not visible. Television is overscanned; computer displays are underscanned.

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 uses a total of 625 horizontal lines per frame, with two fields per frame of 312.5 lines per frame. Each field refreshes at 50 Hz. 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 and video waveform.

pixel

Picture element. Either the smallest addressable spatial element of the computer screen, or the smallest reproducible element in analog video. A pixel can have red, blue, and green color values, an alpha component, and other information associated with it. (Pixels are referred to as having a color component even if they're gray-scale or monochrome.) The bits at location (x, y) of all the bit planes in the framebuffer constitute the single pixel (x, y) . In OpenGL window coordinates, each pixel corresponds to a 1.0×1.0 screen area. The coordinates of the lower left corner of the pixel named x, y are (x, y) , and of the upper right corner are $(x+1, y+1)$. *See also* alpha value and component video.

pixel map

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

Q signal

The color video signal that modulates the 3.58 MHz C signal in quadrature with the I signal. Hues are green and magenta. Bandwidth is 0.0MHz 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.

raster operation, raster op

A logical or arithmetic operation on a pixel value.

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.

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, and Y/R-Y/B-Y.

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

The number of times per second (measured in kHz, where 1 kHz = 1000 times per second) the system reads the file when outputting audio. The greater the sampling rate, the larger the file and the better the quality of the audio output.

saturation

Color intensity; zero saturation is white (no color) and maximum saturation is the deepest or most intense color possible for that hue. In signal terms, saturation is determined by the ratio between luminance level and chrominance amplitude. *See also* hue.

scaling

Changing the size of an image.

scan

To convert an image to an electrical signal by moving a sensing point across the image, usually left to right, top to bottom.

SECAM

Sequentiel Couleur avec Memoire, the color television system developed in France and used there as well as in eastern Europe, the Near East and Mideast, and parts of Africa and the Caribbean.

setup

The difference between the blackest level displayed on the receiver and the blanking level. 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. 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. *See also* video waveform.

SMPTE time code

A signal specified by the Society of Motion Picture and Television Engineers for facilitating videotape editing; this signal uniquely identifies each frame of the video signal. Program originators use vertical blanking interval lines

12 through 14 to store a code identifying program material, time, frame number, and other production information.

subcarrier

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

subpixel

A unit derived from a pixel by using a filter for sizing and positioning.

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 tip

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

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.

time-base errors

Analog artifacts caused by nonuniform motion of videotape or of the tape head drum. Time-base errors usually cause horizontal display problems, such as horizontal jitter.

time code

See SMPTE time code.

time-delay equalization

Frame-by-frame alignment of all video inputs to one sync pulse, so that all frames start at the same time. This alignment is necessary because cable length differences cause unequal delays. *See* time-base errors.

transducer

A microphone, video camera, or other device that can convert sounds or images to electrical signals.

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.

underscan

To scan a television screen so that the edges of the raster are visible. *See also* overscan.

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 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 level

Video signal amplitude.

video signal

The signal from a video device, such as a camera, VCR, or other 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.

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, and 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:

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

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.

Note: 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

Numbers

10-bit RGB without alpha, 3@1025x768_60, 20
2@1025x768_60, display surface, 18
2@1280x1024_60, display surface, 16
2@640x480_60, display surface, 21
2@640x480_60+1@1280x1024, display surface, 17
2@640x480_60, display surface, 22
2@640x486_30i, display surface, 25
3@1025x768_60, 10-bit RGB without alpha, 20
3@1025x768_60, display surface, 19
3@960x680_60, display surface, 21
4@640x480_60, display surface, 23
4@640x486_30i, display surface, 26
6@640x480_60, display surface, 24
6@640x486_30i, display surface, 27

A

Action menu, in panel, 36
antialiasing, 3, 6 through 8, 14

B

breakout box, 43

C

cabling VIDI/O BOX, 65 through 66
conventions, xiii
cursor position, setting, 11

D

DG2, 4
directories for video output formats, 5, 6, 12
display surface, 14 through 27

E

electrical levels of RGB outputs, 39

F

factory settings, 37
feature set
 12-bit, 5 through 7
 8-bit, 6 through 8
File menu, in panel, 36
format, *see* video output format

G

- g flag, 10
- gain, in panel, 35
 - factory setting, 37
- gamma, in panel, 34 through 35
 - factory setting, 37
- genlock
 - in panel, factory setting, 37
 - input specifications, 40
 - setmon* command line, 10
 - setting in panel, 35
- Genlock menu item, in panel, 35
- gfxinfo*, 12
- graphics subsystem, and Multi-channel Option, 2 through 3
- Group Controls option, in panel, 35, 36
 - factory setting, 37
 - see also* system-wide controls

H

- hardware specifications, 39 through 43
- horizontal phase, in panel, 35
 - factory setting, 37

I

- I/O door panel with Multi-channel Option
 - installed, 42
- IRIX version required, xii

M

- mem*, 57 through 58, 59 through 60
- Multi-Channel Option

- and graphics subsystem, 2 through 3
- configuration, 1 through 2
- factory settings, 37
- I/O door panel, 42
- IRIX version required, xii
- more than one installed, 31
- panel, 31 through 37
 - calling up, 31
 - saving settings, 36
 - setting all channels at once, 36
- multisampling, 5, 7 through 8, 14

P

- panel, 31 through 37
 - calling up, 31
 - saving settings, 36
 - setting all channels at once, 36
- pixel
 - depth, 3, 4 through 5
 - in *gfxinfo*, 12
 - small, medium, large, 5

R

- Raster Memory, 3, 4 through 8
 - and video output format, 12
 - revision level required, 12
 - see also* RM
- RealityEngine, 1
- RealityEngine2, 1
- RM, 3, 4 through 8, 14
 - and video output format, 12
 - in *gfxinfo*, 12
 - see also* Raster Memory
- RS-170, 1, 25, 26, 27

S

-S flag, 9
setmon, 9 through 27
 -g flag, 10
 -S flag, 10
 syntax, 9 through 11
setup, in panel, 34
 factory setting, 37
sync pair outputs, hardware specifications, 40
sync, in panel, 34
 factory setting, 37
system-wide controls, in panel, 35
 see also Group Controls option

T

timebase, *see* video timebase
timings, 41

V

VGA, 1, 13
video output format, 13
 and Raster Memory boards, 4 through 8
 directories listing, 5, 6, 12
 display surface, 14 through 27
 in *setmon* command line, 9
 RS-170, 1
 timings, 41
 VGA, 1
video timebase, 5 through 6
VIDI/O BOX, cabling, 65 through 66
viewport, 14 through 27
VME address
 default
 4.0.5H release, 56, 58

 Onyx with 5.0.1 or 5.1 release, 61
 setting, 55 through 63
 4.0.5H release, 56 through 58
 5.1 release, 58 through 61
 Onyx with 5.0.1 or 5.1 release, 61 through 62
 testing, 62 through 63
vout, *see* panel
vs2addrset, 56 through 57, 59, 62
VTX, 1

W

window, and viewports, 14

X

X display surface, 14
 size, 14

We'd Like to Hear From You

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

Please include the title and part number of the document you are commenting on. The part number for this document is 007-2266-001.

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
- For UUCP mail, use this address through any backbone site:
[your_site]!sgi!techpubs



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

415 965-0964