SGI DMediaPro™ DM3 Board User's Guide

007-4647-001

CONTRIBUTORS

Written by Francisco Razo and Nancy Heller

Illustrated by Dan Young

Edited by Susan Wilkening

Production by Glen Traefald

Engineering contributions by Joe Kennedy, Brandon Corey, Jeff Hane, Dennis Yakesh, and David Press

Cover design by Sarah Bolles, Sarah Bolles Design, and Dany Galgani, SGI Technical Publications.

Record of Revision

Version	Description
001	August 2003 Initial version

Contents

Figures

Tables

About This Guide

This guide provides an overview of the DMediaPro DM3 board. It describes how to connect the DMediaPro DM3 board to an SGI video breakout box (VBOB) for high-definition and standard-definition input/output (I/O). It also describes how to synchronize audio with video when using the DMediaPro DM3 board, and how to program the board using the OpenML media library software development kit (ML) and device-specific controls.

The DMediaPro DM3 board transmits and receives uncompressed high-definition video and standard-definition video in real-time. The DMediaPro DM3 board can be installed and used with the SGI Onyx 350 dual-head graphics system and the Silicon Graphics Tezro visual (deskside and rackmountable) workstations. In the future, the DMediaPro DM3 board (referred to as the DM3 board throughout this guide) will also be compatible with other SGI systems.

This guide is provided for the sophisticated video user in a professional or research environment. You should be familiar with video standards, the operation of the applicable SGI system or workstation, and the ML-related information in the *OpenML Media Library Software Development Kit Programmer's Guide*.

Important Information

Only a trained SGI support service engineer can install the DM3 board in an SGI Onyx 350 graphics system, and a rackmountable or deskside Silicon Graphics Tezro visual workstation.

Note: This option board requires IRIX 6.5.21 (with any required patches) or later.

You control and program the board using ML. For a description of ML device-independent calls and controls, see the *OpenML Media Library Software Development Kit Programmer's Guide* (007-4504-00*x*).

Chapter Descriptions

The following topics are covered in this guide:

- Chapter 1, “Features and Capabilities,” describes the main components of the DM3 board, and includes a general overview of product features.
- Chapter 2, “Connecting the DMediaPro DM3 Board to VBOB,” shows you how to set up high-definition and standard-definition input/output (I/O) using the SGI video breakout box (VBOB).
- Chapter 3, “Setting Up Audio Synchronization,” shows you how to synchronize audio with video when using the DM3 board in a typical studio configuration with a tape deck.
- Chapter 4, “Programming the DMediaPro DM3 Board,” shows you how to use ML and device-specific controls to program the board.

Related Publications

The following publications contain additional information that may be helpful:

In addition to this guide, the *OpenML Media Library Software Development Kit Programmer's Guide* (007-4504-00*x*) and the *SGI Video Breakout Box Owner's Guide* (007-4243-00*x),* are shipped with the DM3 board. If you are using an SGI PCI Digital Audio Board, see the *PCI Digital Audio Board Installation Guide* (007-3502-00*x*).

It is also a good idea to have your system owner's guide available.

You can obtain SGI documentation, release notes, or man pages in the following ways:

- See the SGI Technical Publications Library at http://docs.sgi.com. Various formats are available. This library contains the most recent and most comprehensive set of online books, release notes, man pages, and other information.
- If it is installed on your SGI system, you can use InfoSearch, an online tool that provides a more limited set of online books, release notes, and man pages. With an IRIX system, select **Help** from the Toolchest, and then select **InfoSearch**. Or you can type **`infosearch`** on a command line.

- You can also view release notes by typing either **grelnotes** or **relnotes** on a command line.
- You can also view man pages by typing **man** <*title*> on a command line. For example, to display the man page for the Add_disk command, type the following on a command line:

  ```
  man Add_disk
  ```

  References in the documentation to man pages include the name of the command and the section number in which the command is found. For example, "Add_disk(1)" refers to the Add_disk command and indicates that it is found in section 1 of the IRIX reference.

  For additional information about displaying man pages using the man command, see man(1).

  In addition, the apropos command locates man pages based on keywords. For example, to display a list of man pages that describe disks, type the following on a command line:

  ```
  apropos disk
  ```

  For information about setting up and using apropos, see apropos(1) and makewhatis(1M).

Conventions

The following conventions are used throughout this document:

Convention	Meaning
Command	This fixed-space font denotes literal items such as commands, files, routines, path names, signals, messages, and programming language structures.
variable	The italic typeface denotes variable entries and words or concepts being defined. Italic typeface is also used for book titles.
user input	This fixed-space font denotes literal items that the user enters in interactive sessions. Output is shown in nonbold, fixed-space font.
[ ]	Brackets enclose optional portions of a command or directive line.
...	Ellipses indicate that a preceding element can be repeated.

`man page`(*x*)	Man page section identifiers appear in parentheses after man page names.
GUI element	This font denotes the names of graphical user interface (GUI) elements such as windows, screens, dialog boxes, menus, toolbars, icons, buttons, boxes, fields, and lists.

Product Support

SGI provides a comprehensive product support and maintenance program for its products, as follows:

- If you are in North America, contact the Technical Assistance Center at 1 (800) 800 4SGI or contact your authorized service provider.
- If you are outside North America, contact the SGI subsidiary or authorized distributor in your country.

Reader Comments

If you have comments about the technical accuracy, content, or organization of this document, contact SGI. Be sure to include the title and document number of the manual with your comments. (Online, the document number is located in the front matter of the manual. In printed manuals, the document number is located at the bottom of each page.)

You can contact SGI in any of the following ways:

- Send e-mail to the following address:

  techpubs@sgi.com

- Use the Feedback option on the Technical Publications Library website:

  http://docs.sgi.com

- Contact your customer service representative and ask that an incident be filed in the SGI incident tracking system.

- Send mail to the following address:

  Technical Publications
  SGI
  1600 Amphitheatre Pkwy., M/S 535
  Mountain View, California 94043-1351

- Send a fax to the attention of “Technical Publications” at +1 650 932 0801.

SGI values your comments and will respond to them promptly.

Features and Capabilities

This chapter describes the features and capabilities of the DMediaPro DM3 board. The following topics are covered:

Board Components

Figure 1-1 shows the DM3 board with its connectors and LEDs. The board ships with two low-voltage differential signaling (LVDS) cables that interface with the SGI video breakout box (VBOB).

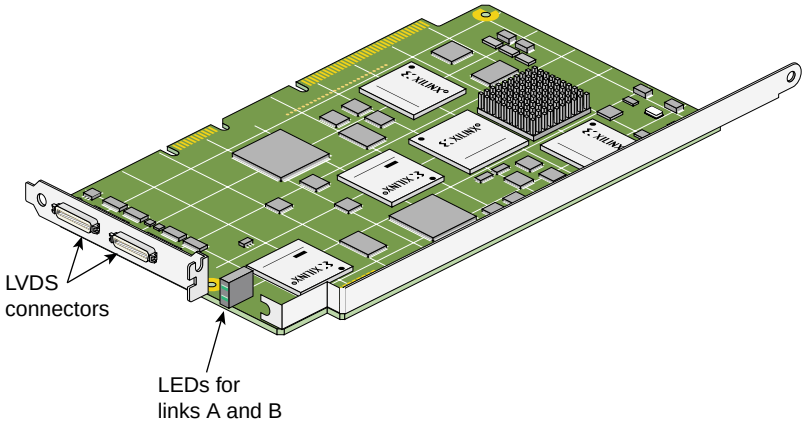

Figure 1-1 DMediaPro DM3 Board and Panel

Input/Output (I/O)

The DMediaPro system uses the VBOB for analog genlock connections and serial digital interface (SDI) high-definition (HD) and standard-definition (SD) video I/O.

Note: For information on I/O and genlock timing features, see Chapters 3 and 4 in this guide. For a complete description of the VBOB, including all the I/O connectors, see the *SGI Video Breakout Box Owner's Guide*.

Supported Video Formats

The DM3 board supports video formats defined by the advanced television standards committee (ATSC), as well as several formats defined for high-definition digital motion pictures and post production. In addition, the board supports several standard-definition formats. These formats can have pixel clock rates of up to 74.25 MHz. These formats include support for:

- 1080p, 1080i, 720p, 576i, and 480i video formats
- 23.98, 24, 25, 29.97, 30, 59.94, and 60 Hz vertical rates
- 16 x 9 and 4 x 3 aspect ratios

Examples of supported formats are SMPTE 274M (interlaced and progressive), SMPTE 296 (progressive), ITU-R BT.601-5 (interlaced), SMPTE 260M, and SMPTE 240M.

Note: ITU-R BT.601-5 is also known as Rec. 601 and CCIR 601.

Table 1-1 lists the DM3 board video timings that are supported for this release of the board.

Table 1-1 Supported Video Timings

Description	Frame Rate	Timing
1920x1080 interlaced	30/1.001 Hz	ML_TIMING_1125_1920x1080_5994i
1280x720 progressive	60/1.001 Hz	ML_TIMING_750_1280x720_5994p
1280x720 progressive	60 Hz	ML_TIMING_750_1280x720_60p
1920x1080 interlaced	25 Hz	ML_TIMING_1125_1920x1080_50i
1920x1080 interlaced	30 Hz	ML_TIMING_1125_1920x1080_60i
1920x1080 progressive	24 Hz	ML_TIMING_1125_1920x1080_24p
1920x1080 progressive	24/1.001 Hz	ML_TIMING_1125_1920x1080_2398p
1920x1080 progressive	25 Hz	ML_TIMING_1125_1920x1080_25p
1920x1080 progressive segmented frame	25 Hz	ML_TIMING_1125_1920x1080_25PsF

Table 1-1 Supported Video Timings **(continued)**

Description	Frame Rate	Timing
1920x1080 progressive segmented frame	24 Hz	ML_TIMING_1125_1920x1080_24PsF
1920x1080 progressive segmented frame	24/1.001 Hz	ML_TIMING_1125_1920x1080_2398PsF
1920x1035 interlaced	30/1.001 Hz	ML_TIMING_1125_1920x1035_5994i
720 x 487 (NTSC)	30/1.001 Hz	ML_TIMING_525
720 x 576 (PAL)	25 Hz	ML_TIMING_625

In progressive segmented frame (PsF) formats, the progressive frame is transmitted as two fields that are from the same progressive scan, while in interlaced formats the two fields are temporally displaced.

All formats are 8-bit or 10-bit. These formats are used for content creation and telecine output, and support serial-parallel conversion.

Other Features

The DM3 board includes the following features:

- Programmable field size (see “Re-sizing Field Height” on page 65).
- YCrCb with 8- or 10-bits per component (4:2:2 or 4:4:4 sampling rates).
- Alpha channel support.
- Video interface support for RGB 8- or 10-bits.
- Support for up to 64 bits/pixel RGB in memory.
- Real-time “transparent” color space conversion in ITU-R Rec. 601, ITU-R Rec. 709, SMPTE 240M, and key scaling.
- User-programmable horizontal and vertical phase adjustment of the output video.
- Unadjusted system time (UST) and media stream count (MSC) support on input and output.
- Gamma correction support through a user-downloadable 13-bit-wide lookup table.

- 3/2 pulldown mode on output.
- Packing modes (see Table 1-2). For VL/ML packing conversions, see "ML_IMAGE_PACKING" on page 51.

Table 1-2 Supported Packing Modes

ML_IMAGE_PACKING	ML_IMAGE_SAMPLING
ML_PACKING_8	ML_SAMPLING_422
ML_PACKING_8_3214	ML_SAMPLING_422
ML_PACKING_10	ML_SAMPLING_422
ML_PACKING_10_3214	ML_SAMPLING_422
ML_PACKING_10in16L	ML_SAMPLING_422
ML_PACKING_10in16L_3214	ML_SAMPLING_422
ML_PACKING_10in16R	ML_SAMPLING_422
ML_PACKING_10in16R_3214	ML_SAMPLING_422
ML_PACKING_10_10_10_2	ML_SAMPLING_4224
ML_PACKING_10_10_10_2_3214	ML_SAMPLING_4224
ML_PACKING_8	ML_SAMPLING_444
ML_PACKING_8_R	ML_SAMPLING_444
ML_PACKING_S12in16L	ML_SAMPLING_444
ML_PACKING_S12in16R	ML_SAMPLING_444
ML_PACKING_8	ML_SAMPLING_4444
ML_PACKING_8_R	ML_SAMPLING_4444
ML_PACKING_10_10_10_2	ML_SAMPLING_4444
ML_PACKING_10_10_10_2_R	ML_SAMPLING_4444

Theory of Operation

The DM3 board is a full-duplex, dual-link interface between the SGI 400-MHz Crosstalk XIO bus and the DMediaPro low-voltage differential signaling (LVDS) links. The LVDS links run at seven times the programmed video rates, from 189 MHz to 519.75 MHz, corresponding to seven times the 27 MHz to 74.25 MHz video link rates.

The DM3 board supports high definition (HD), standard definition (SD), and serial digital transport interface (SDTI) video formats. The board also supports downloadable lookup tables (LUTs) for linear, log, and gamma correction.

The main components of the DM3 board are as follows:

- An XIO-to-graphics interface contains an internal 100-MHz bus that interfaces with the board's video input and output pipes via direct memory access (DMA) engines.

  Note: This 100-MHz bus architecture is common to both the SGI HD I/O board and the SGI InfiniteReality graphics systems.

- A guava interface contains the DMA engines, a local bus controller, and a PIO unit. This interface controls all of the DM3 board’s field programmable gate arrays (FPGAs).
- A video input formatter formats and controls the video input.
- A high-definition output controller formats and controls the video output.
- A control link contains the LVDS channel link control and the status logic.
- Packers, unpackers, and input and output color space converters support various packings and color spaces.

Figure 1-2 is a simplified top-level diagram of the DM3 board.

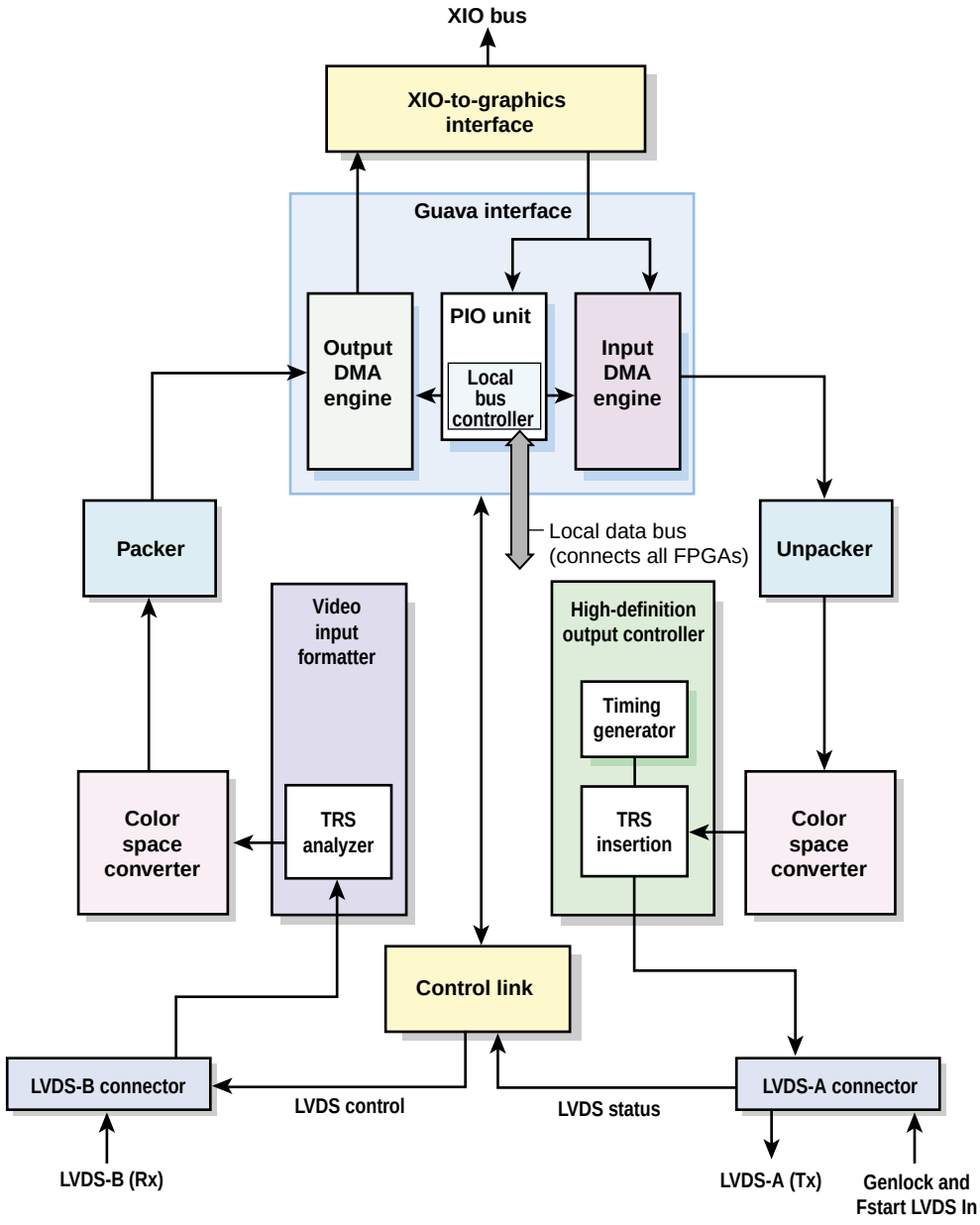

Figure 1-2 DMediaPro DM3 Board Top-level Diagram

Connecting the DMediaPro DM3 Board to VBOB

This chapter shows you how to connect the DM3 board to the SGI video breakout box (VBOB) for all high-definition video, standard-definition video, and genlock input/output (I/O). The DMediaPro/VBOB configuration allows you to transmit and receive uncompressed high-definition (HD) video and standard-definition (SD) video without using two separate connections.

Note: For detailed information on VBOB, see the *SGI Video Breakout Box Owner's Guide*.

The following topics are covered:

Getting Started

Before you connect any cables, follow these steps:

1. Power down the SGI host system.
2. Be sure the VBOB power switch at the back of the chassis is in the **Off (0)** position.
3. Connect one end of the VBOB power cord to the VBOB power plug and the other end of the cord to your AC power source.

Connecting the DMediaPro DM3 Board to VBOB

To connect the DM3 board to VBOB, follow these steps:

1. Connect one end of the LVDS A cable (black cable with a white connector on each end) to the DM3 board's LVDS A connector. Connect the other end of this cable to the VBOB LVDS A connector, as shown in Figure 2-1.
2. Connect one end of the LVDS B cable (black cable with black connector on each end) to the DM3 board's LVDS B connector. Connect the other end of this cable to the VBOB LVDS B connector.

Note: The length of the LVDS cables may affect/restrict hardware positions.

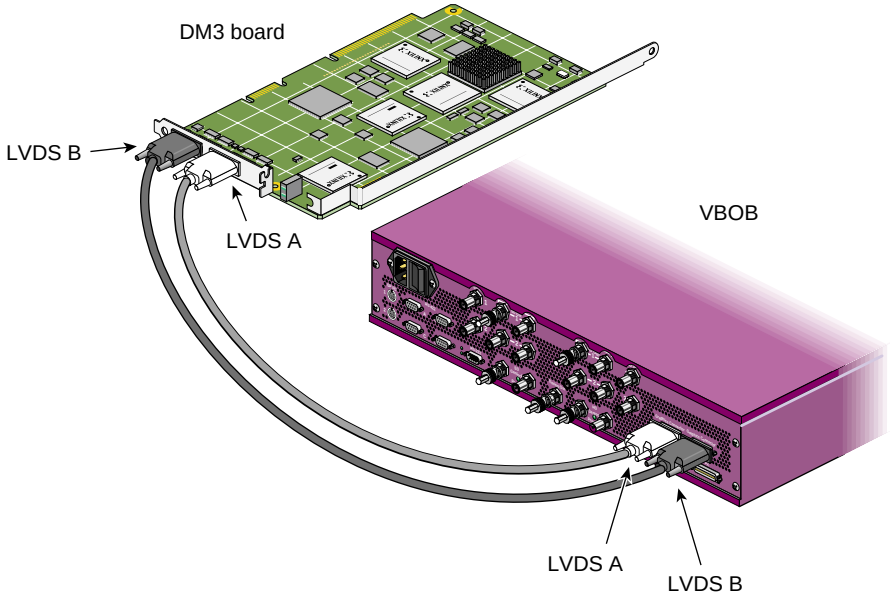

Figure 2-1 Connecting the DMediaPro DM3 Board to VBOB

Setting Up VBOB for High-definition/Standard-definition I/O

After you complete the connections shown in Figure 2-1, you can connect the required cables from the VBOB HD in/out or SD in/out connector(s) to the desired high-definition or standard-definition video device(s). If you are genlocking the serial digital output to an external source, connect an analog sync source to the VBOB HD genlock in or SD genlock in connector (see Figure 2-2).

If you are not using high-definition or standard-definition genlock loop-throughs, attach a 75 ohm terminator to the appropriate connectors.

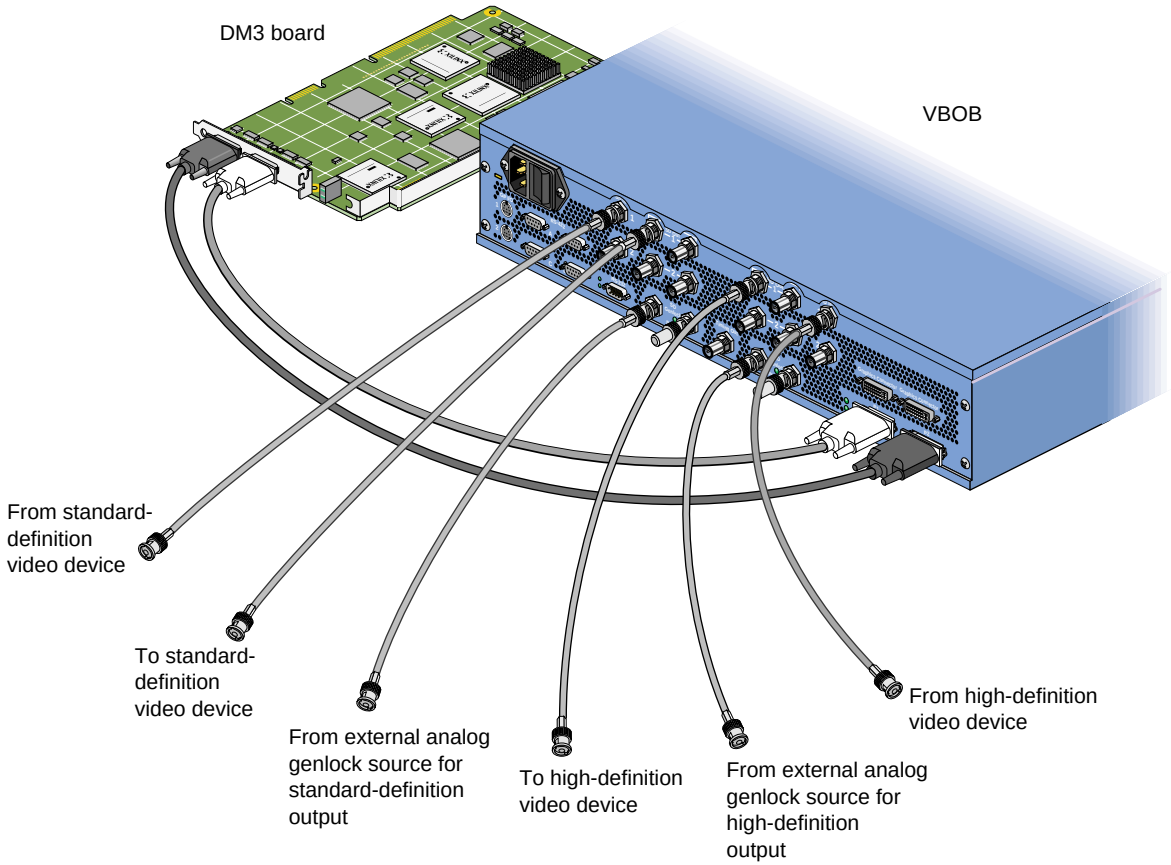

Figure 2-2 Connecting the I/O Cables to VBOB

After you connect the required cables, follow these steps:

1. Turn the VBOB power switch to the **On (1)** position.
2. Power on the SGI host system.
3. Boot the operating system.

Precautions on SGI Onyx and Origin Systems

To ensure proper airflow and cooling of your system, perform the following activities:

- Secure all cables to the sides of the rack; the cables should not hang freely from the VBOB or any modules.
- Keep the rack doors closed during normal operation.

Setting Up Audio Synchronization

This chapter shows you how to synchronize audio with video when you use the DM3 board in a typical studio configuration with a tape deck.

The following topics are covered:

Synchronizing Audio

To synchronize audio on an SGI Onyx 350 graphics system or a Silicon Graphics Tezro visual workstation (rackmountable and deskside), you need to install an SGI PCI digital audio board in your system.

Note: For instructions on installing the SGI PCI digital audio board, see your system user's guide.

Warning: To prevent serious physical damage to the Onyx 350 graphics system and the Silicon Graphics Tezro visual workstation (rackmountable and deskside), you must use the latest revision of the PCI digital audio board (PN 030-1649-001 or later).

To synchronize audio on an SGI Onyx 350 graphics system and a Silicon Graphics Tezro visual workstation (rackmountable and deskside), follow these steps:

1. Connect a house sync generator to a digital tape deck, as shown in Figure 3-1.
2. Connect the breakout cable that is included with the PCI digital audio board to the DB15 connector on the board.
3. Connect the AES 1/2 output on the digital tape deck to the PCI digital audio board breakout cable's AES input connector (red wire).

Note: For additional functionality, you can purchase a second PCI digital audio board for your system.

If you have a second PCI digital audio board installed, proceed as follows:

1. Connect the provided breakout cable to the DB15 connector on the second board.
2. Connect the AES 3/4 output on the digital tape deck to the breakout cable's AES input connector (red wire) on the second board (see Figure 3-1).

   For more information on the PCI digital audio board and breakout cable, see the *PCI Digital Audio Board Installation Guide*.

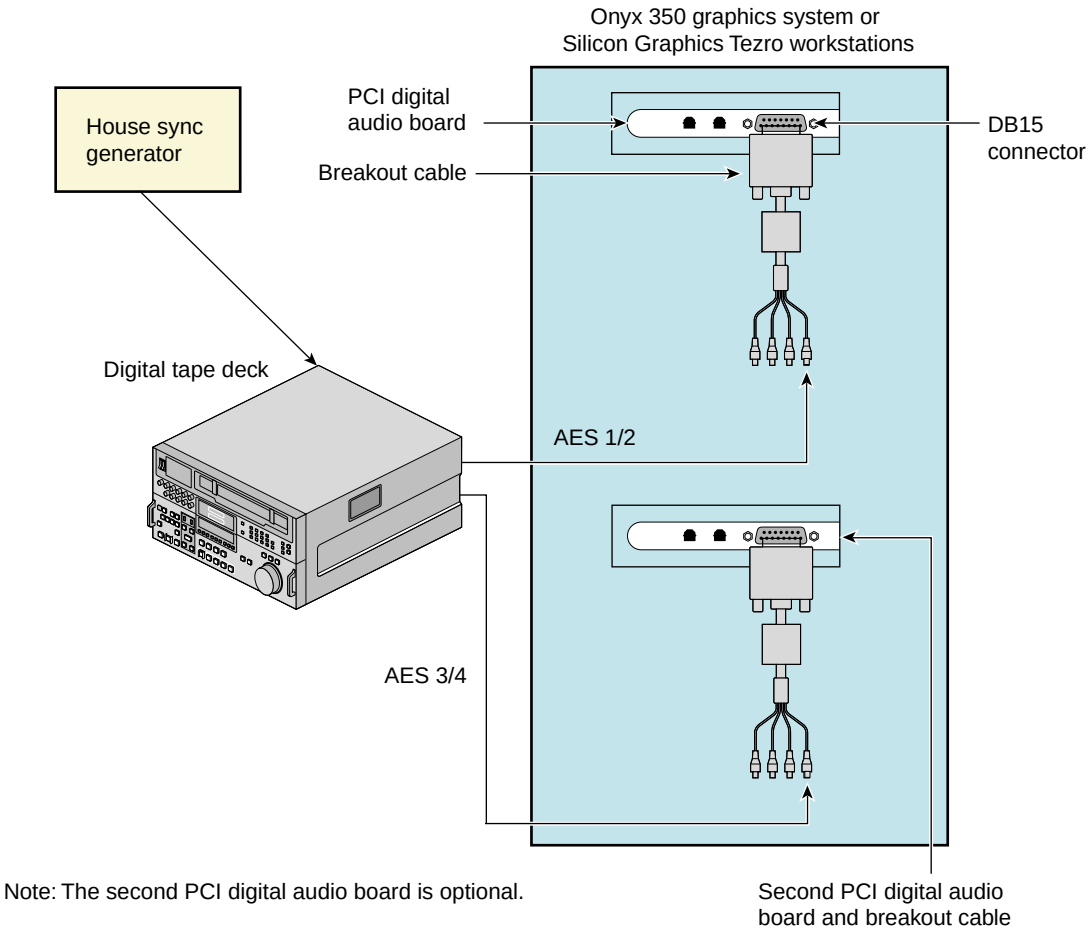

Figure 3-1 Synchronizing Audio

Setting Up Audio Panel

For the configuration shown in Figure 3-1, you can use the **Audio Panel** to perform the following functions:

- Display the AES device controls
- Set the default input and output to AES I/O
- Set the audio signal path to AES I/O
- Set the output sync source

Displaying AES Device Controls

To display the AES device controls, follow these steps:

1. Open the **Audio Panel** by selecting **Desktop > Control Audio** from the Toolchest.
2. Select the desired AES device control panel from the **View** menu.

   The selected device control panel appears, where you can adjust parameters such as sample rate, input source, and output destination. For details, refer to the **Audio Panel Help** menu.

Setting Default Input and Output to AES

The **Audio Panel** window displays device controls for ADAT In, ADAT Out, AES In, and AES Out for each PCI digital audio board installed in your system. You can designate any of these devices as a default, which means your system uses the device as the internal audio device unless otherwise specified by another application. To set the default input and output, simply open the **Audio Panel** and select a default input and output device in the **Default** menu.

Setting Input and Output Signal Path

You can select either electrical or optical signal paths for the AES input and output devices. To set the input source with the **Audio Panel**, follow these steps:

1. Position the cursor over the input device panel (if the device panel you need is not displayed, use the **View** menu), then hold down the right mouse button to display the menu for the desired input device.
2. Select **Preferences** from the input device menu.

   The **Preferences** menu appears for the respective input device.
3. Choose the input source (for example, AES In) that you want the input device to use. “AES In” is the electrical input.

To set the output destination with the **Audio Panel**, follow these steps:

1. Position the cursor over the output device panel (if the device panel you need is not displayed, use the **View** menu), then hold down the right mouse button to display the menu for the desired output device.
2. Select **Preferences** from the output device menu, as shown in Figure 3-2.

   The **Preferences** menu appears for the respective output device, as shown in Figure 3-3.
3. Choose the output destination (for example, “AES Out”) that you want the output device to use. “AES Out” is the coaxial digital electrical output.

   Selecting the optical input or output for AES disables the ADAT input or output, because the two devices share the same optical connector. Similarly, selecting the optical input or output for ADAT automatically switches the AES input or output to the coaxial digital electrical connector.

   If more than one PCI digital audio board is installed, the system names each board using an incremental prefix naming scheme. For this type of configuration, one AES input is displayed as “RAD 1.AES In” and the other as “RAD 2.AES In.” Also, one AES output is displayed as “RAD 1.AES Out” and the other as “RAD 2.AES Out (see Figure 3-2).

Setting Output Sync Source

To set the output sync source, follow these steps:

1. Position the cursor over the output device panel (if the device panel you need is not displayed, use the **View** menu), then hold down the right mouse button to display the menu for the desired output device.
2. Select **Preferences** from the output device menu, as shown in Figure 3-2.

   The **Preferences** menu appears for the respective output device, as shown in Figure 3-3.

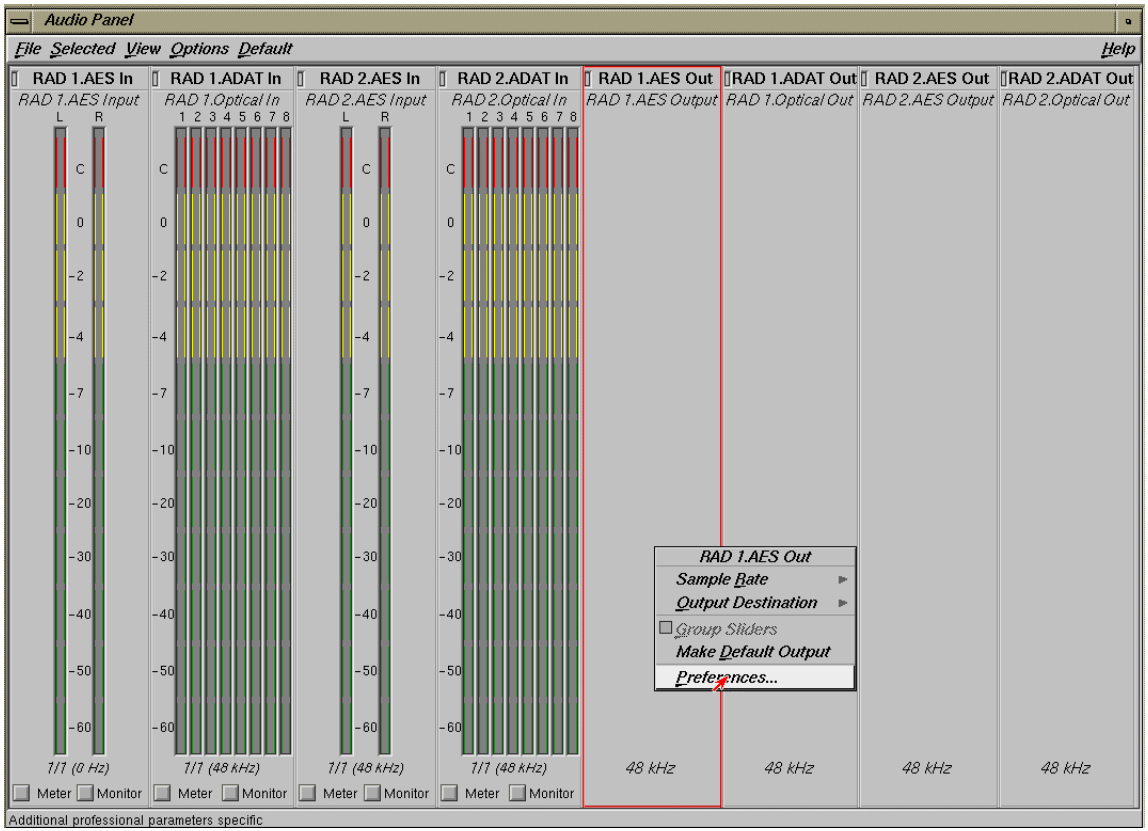

Figure 3-2 Selecting Preferences for Output Device

3. Choose the sync source that you want the output device to use.

The example in Figure 3-3 shows how to select a sync source for a system that contains one PCI digital audio board.

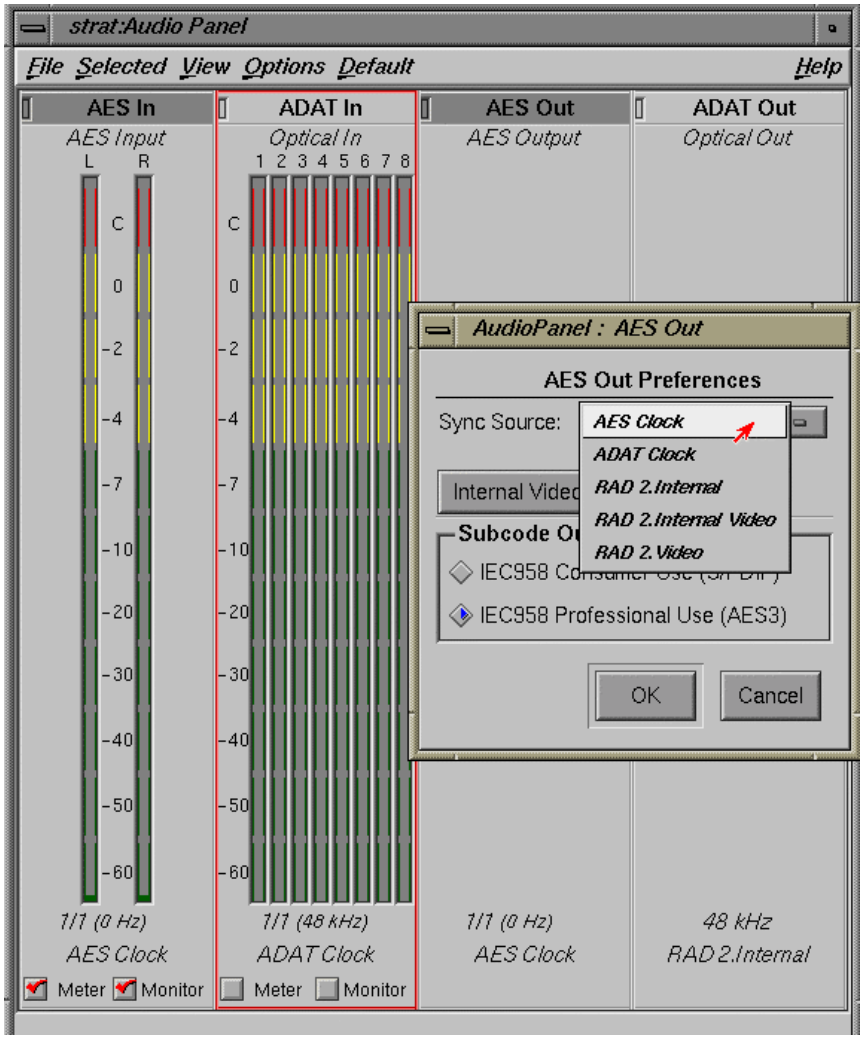

Figure 3-3 Synchronizing One PCI Digital Audio Board

Figure 3-4 and Figure 3-5 show how to select a sync source for a system that contains two PCI digital audio boards. Figure 3-4 shows the selection using the **RAD 1.AES Out Preferences** window, which contains the preferences for the first PCI digital audio board.

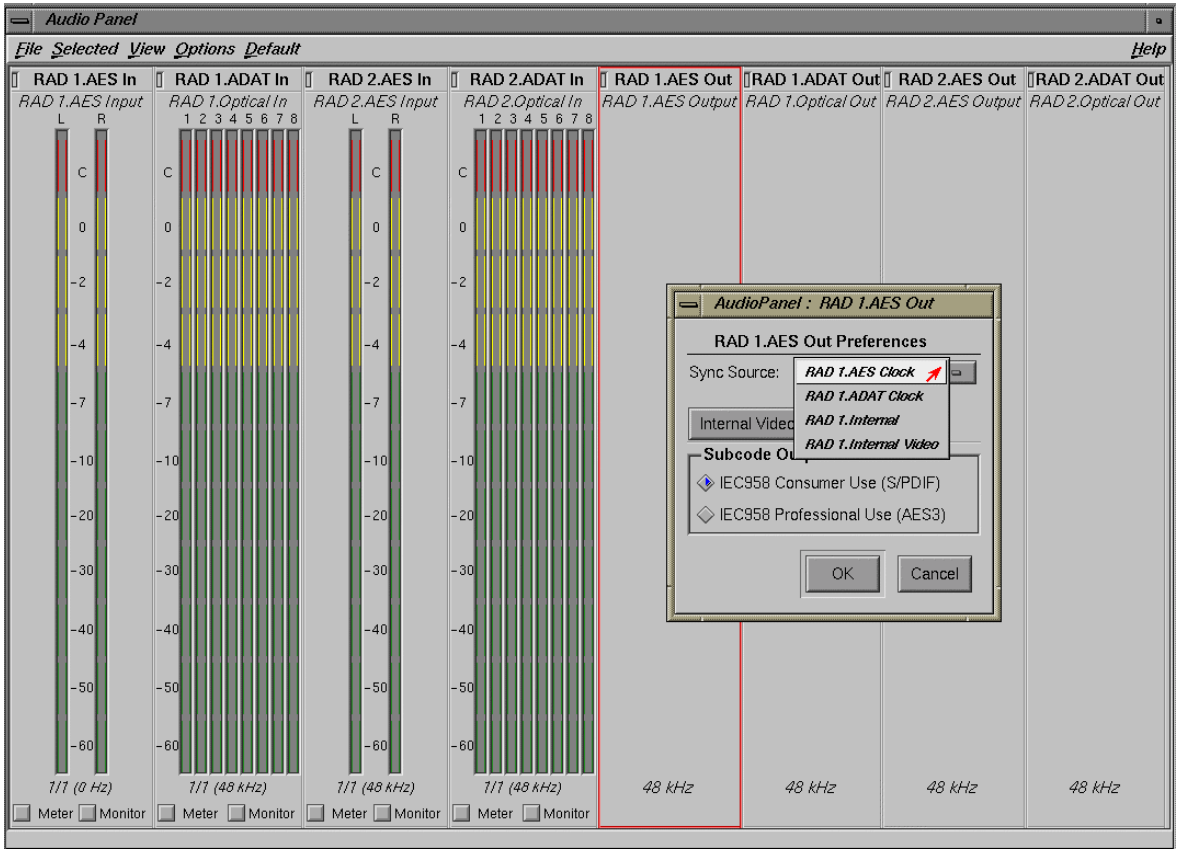

Figure 3-4 Synchronizing Two PCI Digital Audio Boards (RAD 1.AES Out Preferences)

Figure 3-5 shows the selection using the **RAD 2.AES Out Preferences** window, which contains the preferences for the second PCI digital audio board.

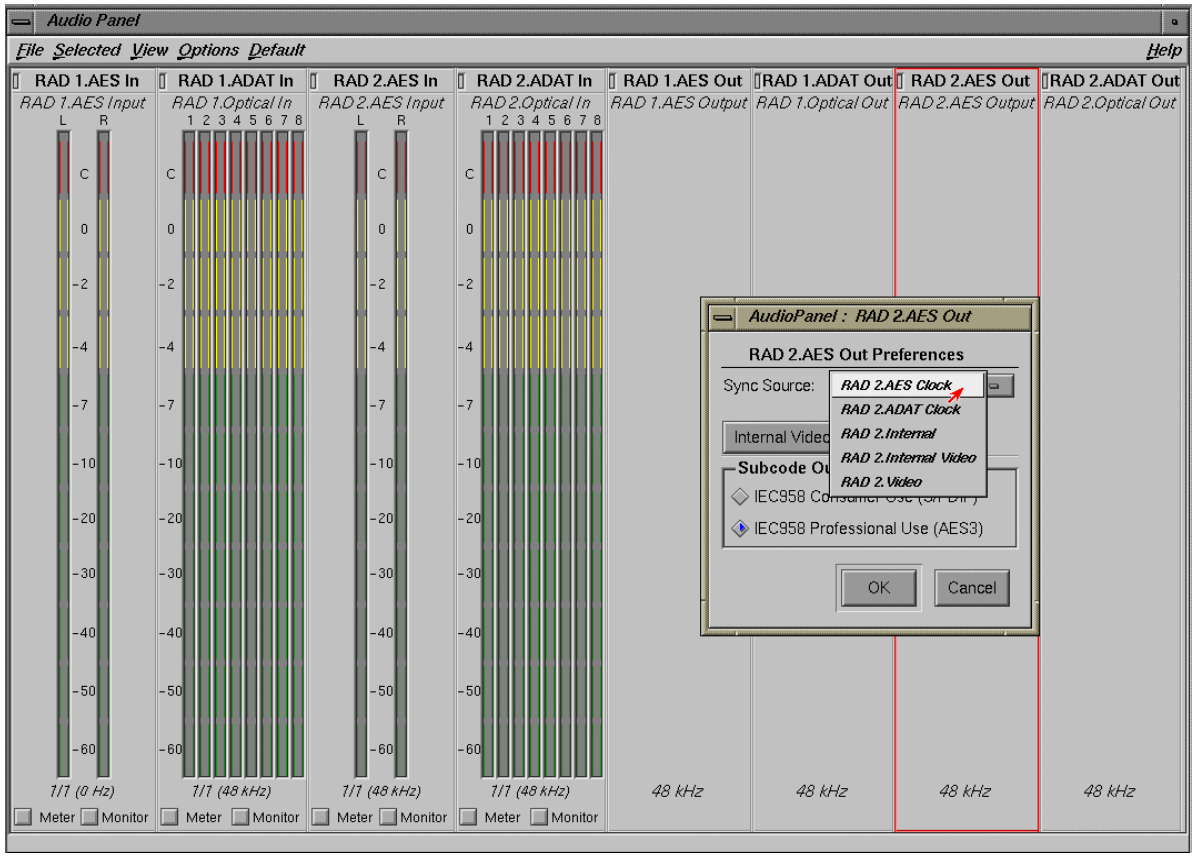

Figure 3-5 Synchronizing Two PCI Digital Audio Boards (RAD 2.AES Out Preferences)

Other Configurations

The DM3 board does not support audio synchronization using Internal Video as a sync source in the Audio Panel AES or ADAT Out Preferences, or the Analog In/Out Preferences.

In addition to the examples described earlier in this chapter, there are other optional configurations. For example, you can synchronize the PCI digital audio board's audio input to a house sync generator as follows:

1. Connect the house sync generator to one of the breakout cable's video reference loop connectors (green or black wires).

   If you are not using the other video reference loop connector to loop the video reference signal through another device, place a 75-ohm terminator on the unused connector.

2. From the **Audio Panel**, hold down the right mouse button to display the menu for the desired output device.
3. Select **Preferences** in the output device menu**.**
4. Select **Video** as the sync source.

Note: To synchronize a digital audio (AES or ADAT) input to a video reference, the input must be synchronized externally because it is externally sampled. The examples described earlier in this chapter (using a tape deck) reflect this requirement.

Programming the DMediaPro DM3 Board

This chapter provides an overview for programming a DM3 board. The DM3 board supports the OpenML media library software development kit (ML). This application programmatic interface (API) is described in the *OpenML Media Library Software Development Kit Programmer's Guide*.

The following topics are covered:

Programming Basics for the DMediaPro DM3 Board

After installing the DMediaPro software and the DM3 board, if you want to build programs that run under ML, follow these steps:

1. In your source code, include the following header files:
   - `ML/ml.h`
   - `ML/mlu.h`
   - `ML/ml_xtdigvid.h`
2. Link with `libML` and `libMLU`

   You can find several useful ML programming examples on the DMediaPro CD in the `/usr/share/src/ml/video/xtdigvid/examples` directory. You can also find some programming examples later in this chapter.

DMediaPro Controls

This section covers the following topics:

Path Controls and Jack Controls

The DM3 board supports two types of controls: path controls and jack controls.

Path Controls

With path controls, you can set up the following types of data transfer paths:

- Memory-to-video path
- Video-to-memory path

Using these path controls, you can transfer data from memory to an SD/HD video jack or transfer data from an SD/HD video jack to memory (see Figure 4-1).

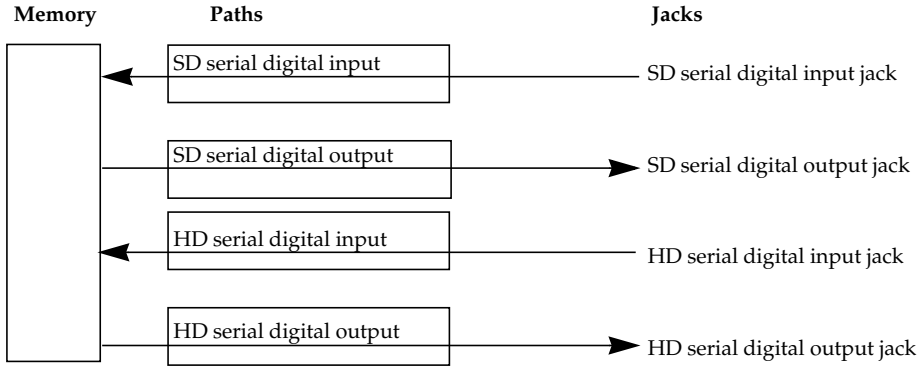

Figure 4-1 Setting Up Paths

Jack Controls

With jack controls you can adjust controls on a jack without setting up a data transfer path.

Using some jack controls, you can adjust certain parameters while a data transfer is in progress. However, you can only use this type of jack control to adjust parameters that do not affect memory-to-video and video-to-memory data transfers. For example, you can adjust the EE mode (XTDIGVID_EE_MODE_INT32) during a transfer (see Figure 4-2), but you cannot use jack controls to adjust the colorspace or memory packing order parameters.

Note: When a jack control is adjustable during a transfer, a special access mode is automatically enabled. This access mode overrides the default setting which normally disallows the changing of controls during a transfer.

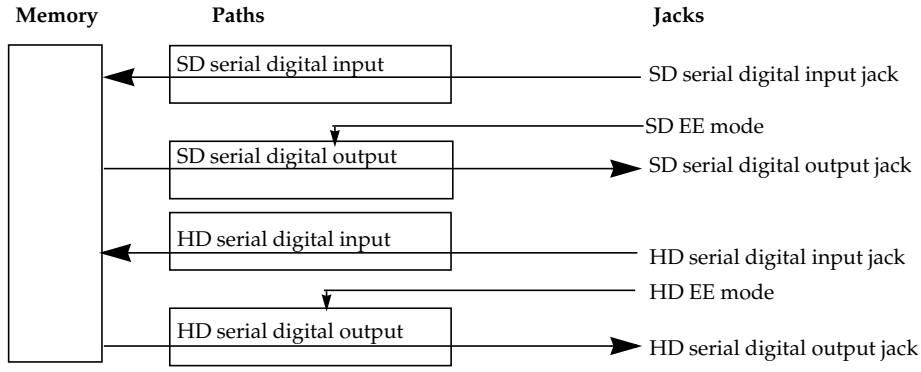

Figure 4-2 Using Jack Control to Change EE Mode During Transfer

Following are the DM3 board jack controls for the HD jacks and the SD jacks.

HD Serial Digital Input Jack

```
XTDIGVID_LOOPBACK_INT32
```

HD Serial Digital Output Jack

```
ML_VIDEO_GENLOCK_SOURCE_TIMING_INT32
ML_VIDEO_GENLOCK_TYPE_INT32
XTDIGVID_EE_MODE_INT32
ML_VIDEO_H_PHASE_INT32
ML_VIDEO_V_PHASE_INT32
```

SD Serial Digital Input Jack

```
XTDIGVID_LOOPBACK_INT32
```

SD Serial Digital Output Jack

```
ML_VIDEO_GENLOCK_SOURCE_TIMING_INT32
ML_VIDEO_GENLOCK_TYPE_INT32
XTDIGVID_EE_MODE_INT32
ML_VIDEO_H_PHASE_INT32
ML_VIDEO_V_PHASE_INT32
```

Note: For an example on setting up a jack control, go to "Setting Controls on a Jack" on page 69.

DMediaPro Control Summary

Table 4-1 shows the input/output paths for each of the DM3 board controls.

Table 4-1 HD/SD Input and Output Paths

Control Parameters	Input Video Path	Input Image Memory Buffer	Output Video Path	Output Image Memory Buffer
ML_IMAGE_BUFFER_POINTER		X		X
ML_IMAGE_WIDTH_INT32		X		X
ML_IMAGE_HEIGHT_1_INT32		X		X
ML_IMAGE_HEIGHT_2_INT32		X		X
ML_IMAGE_PACKING_INT32		X		X
ML_IMAGE_SAMPLING_INT32		X		X
ML_IMAGE_COMPRESSION_INT32		X		X
ML_IMAGE_COLORSPACE_INT32		X		X
ML_IMAGE_ROW_BYTES_INT32		X		X
ML_IMAGE_SKIP_PIXELS_INT32		X		X
ML_IMAGE_SKIP_ROWS_INT32		X		X
ML_IMAGE_ORIENTATION_INT32		X		X
ML_IMAGE_TEMPORAL_SAMPLING_INT32		X		X
ML_IMAGE_INTERLEAVE_MODE_INT32		X		X
ML_IMAGE_DOMINANCE_INT32		X		X
ML_IMAGE_BUFFER_SIZE_INT32		X (Read only)		X (Read only)
ML_VIDEO_UST_INT64	X (Read only)		X (Read only)	
ML_VIDEO_MSC_INT64	X (Read only)		X (Read only)	
ML_VIDEO_TIMING_INT32	X		X	
ML_VIDEO_COLORSPACE_INT32	X		X	
ML_VIDEO_PRECISION_INT32	X		X	

Table 4-1 HD/SD Input and Output Paths **(continued)**

Control Parameters	Input Video Path	Input Image Memory Buffer	Output Video Path	Output Image Memory Buffer
ML_VIDEO_SAMPLING_INT32	X		X	
ML_VIDEO_START_X_INT32	X		X	
ML_VIDEO_START_Y_F1_INT32	X		X	
ML_VIDEO_START_Y_F2_INT32	X		X	
ML_VIDEO_WIDTH_INT32	X		X	
ML_VIDEO_HEIGHT_F1_INT32	X		X	
ML_VIDEO_HEIGHT_F2_INT32	X		X	
XTDIGVID_LUT_YG_INT32_ARRAY	X		X	
XTDIGVID_LUT_UB_INT32_ARRAY	X		X	
XTDIGVID_LUT_VR_INT32_ARRAY	X		X	
ML_DEVICE_STATE_INT32	X		X	
ML_DEVICE_EVENTS_INT32_ARRAY	X		X	
ML_QUEUE_SEND_COUNT_INT32	X		X	
ML_QUEUE_RECEIVE_COUNT_INT32	X		X	
ML_QUEUE_SEND_WAITABLE_INT32	X		X	
ML_QUEUE_RECEIVE_WAITABLE_INT32	X		X	
ML_VIDEO_GENLOCK_TYPE_INT32			X	
ML_VIDEO_GENLOCK_SOURCE_TIMING_INT32			X	
ML_VIDEO_H_PHASE_INT32			X	
ML_VIDEO_V_PHASE_INT32			X	
ML_VIDEO_OUTPUT_REPEAT_INT32			X	
XTDIGVID_EE_MODE_INT32			X	
XTDIGVID_FF_MODE_INT32			X	

Table 4-1 HD/SD Input and Output Paths **(continued)**

Control Parameters	Input Video Path	Input Image Memory Buffer	Output Video Path	Output Image Memory Buffer
XTDIGVID_LOOPBACK_INT32	X (Serial input only; not applicable for graphics input)			
ML_VIDEO_GENLOCK_SIGNAL_ PRESENT_INT32			X (Read only)	
ML_VIDEO_SIGNAL_PRESENT_INT32	X (Read only)			
XTDIGVID_GENLOCK_STATE_INT32			X (Read only)	
XTDIGVID_GENLOCK_ERROR_STATUS_INT32			X (Read only)	
ML_OPEN_MODE_INT32	X		X	
ML_OPEN_SEND_QUEUE_COUNT_INT32	X		X	
ML_OPEN_RECEIVE_QUEUE_COUNT_INT32	X		X	
ML_OPEN_MESSAGE_PAYLOAD_SIZE_INT32	X		X	
ML_OPEN_EVENT_PAYLOAD_COUNT_INT32	X		X	
ML_OPEN_SEND_SIGNAL_COUNT_INT32	X		X	

Table 4-2 defines the value(s) and use for each of the DM3 board controls. For a detailed description of the ML_ controls, see the *OpenML Media Library Software Development Kit Programmer's Guide*. You can find more information on the device-specific (XTDIGVID_) controls later in this guide.

Table 4-2 DM3 Board Control Parameters, Value(s), and Use

Control Parameters	Range or Value(s)	Use
ML_IMAGE_BUFFER_POINTER	N/A	Points to the first byte of an image buffer in memory. The buffer address must comply with the alignment constraints for buffers on the specific path or transcoder to which it is being sent.
ML_IMAGE_WIDTH_INT32	Defined by selected timing.	Defines image width. Must be the same as ML_VIDEO_WIDTH_INT32.
ML_IMAGE_HEIGHT_1_INT32	Number of lines in the frame for progressive or interleaved images. Number of lines in the first field for non-interleaved images.	Defines the height of F1 field in lines. Must be set to full image height for progressive formats or when ML_INTERLEAVE_MODE_INT32 is set to ML_INTERLEAVE_MODE_INTERLEAVED.
ML_IMAGE_HEIGHT_2_INT32	Number of lines in the second field of a non-interleaved image (must be set to zero for progressive and interleaved modes).	Defines the height of F2 field in lines. Set to zero for progressive formats or when ML_INTERLEAVE_MODE_INT32 is set to ML_INTERLEAVE_MODE_INTERLEAVED.

Table 4-2 DM3 Board Control Parameters, Value(s), and Use **(continued)**

Control Parameters	Range or Value(s)	Use
ML_IMAGE_PACKING_ INT32	ML_PACKING_8 ML_PACKING_8_R ML_PACKING_8_3214 ML_PACKING_10 ML_PACKING_10_3214 ML_PACKING_10in16L ML_PACKING_10in16L_3214 ML_PACKING_10in16R ML_PACKING_10in16R_3214 ML_PACKING_S12in16L ML_PACKING_S12in16R ML_PACKING_10_10_10_2 ML_PACKING_10_10_10_2_R ML_PACKING_10_10_10_2_3214	Describes the packing format of data in memory as follows: ML_PACKING_*type_size_order* *type* is the base type of each component. Leave blank for an unsigned integer. Use S for a signed integer (in a future release, ML may also support R for real numbers). *size* defines the number of bits per component. The size may refer to simple, padded, or complex packings. *order* is the order of the components in memory. Leave blank for natural ordering (1,2,3,4). Use R for reversed ordering (4,3,2,1). For all other orderings, specify the component order explicitly. See the *OpenML Media Library Software Development Kit Programmer's Guide* for more detailed information.
ML_IMAGE_ SAMPLING_INT32	ML_SAMPLING_422 (CbYCr only) ML_SAMPLING_4224 (CbYCr only) ML_SAMPLING_444 (CbYCr and RGB) ML_SAMPLING_4444 (CbYCr and RGB)	Specifies how often each component is sampled for each pixel.
ML_IMAGE_ COMPRESSION_INT32	ML_COMPRESSION_UNCOMPRESSED	Describes the desired compression factor. The size of the uncompressed buffer depends on image width, height, packing, and sampling. The default value is implementation-dependent, but should represent a reasonable trade-off between compression time, quality, and bandwidth.

Table 4-2 DM3 Board Control Parameters, Value(s), and Use **(continued)**

Control Parameters	Range or Value(s)	Use
ML_IMAGE_ COLORSPACE_INT32	ML_COLORSPACE_RGB_601_FULL ML_COLORSPACE_CbYCr_601_HEAD ML_COLORSPACE_CbYCr_601_FULL ML_COLORSPACE_RGB_601_HEAD ML_COLORSPACE_CbYCr_709_HEAD ML_COLORSPACE_CbYCr_709_FULL ML_COLORSPACE_RGB_709_HEAD ML_COLORSPACE_RGB_709_FULL ML_COLORSPACE_CbYCr_240M_HEAD ML_COLORSPACE_CbYCr_240M_FULL ML_COLORSPACE_RGB_240M_HEAD ML_COLORSPACE_RGB_240M_FULL	Specifies the colorspace of image data in memory as follows: ML_COLORSPACE_*representation_standard_ range* *representation* controls how to interpret each component. *standard* indicates how to interpret particular values as actual colors. *range* is either ML_RANGE_FULL, where the smallest and largest values are limited only by the available packing size, or ML_RANGE_HEAD, where the smallest and largest values are somewhat less than the theoretical minimum/maximum values to allow some "headroom."
ML_IMAGE_ROW_ BYTES_INT32	[0 - 0]	Specifies the number of bytes along one row of the image memory buffer. If this value is 0, each row is exactly IMAGE_WIDTH * pixels wide. Default is 0. For more information, see the *OpenML Media Library Software Development Kit Programmer's Guide*.
ML_IMAGE_SKIP_ PIXELS_INT32	[0 - 0]	Specifies the number of pixels to skip at the start of each row in the image memory buffer. Default is 0. Must be 0 if ROW_BYTES is 0.
ML_IMAGE_SKIP_ ROWS_INT32	[0 - 0]	Specifies the number of rows to skip at the start of each image memory buffer. Default is 0.
ML_IMAGE_ ORIENTATION_INT32	ML_ORIENTATION_TOP_TO_BOTTOM ML_ORIENTATION_BOTTOM_TO_TOP	Sets the orientation of the image in either natural video order or natural graphics order. ML_ORIENTATION_TOP_TO_ BOTTOM sets the image in natural video order – pixel [0,0] is at the top left of the image. ML_ORIENTATION_BOTTOM_TO_ TOP sets the image in natural graphics order – pixel [0,0] is at the bottom left of the image.

Table 4-2 DM3 Board Control Parameters, Value(s), and Use **(continued)**

Control Parameters	Range or Value(s)	Use
ML_IMAGE_ TEMPORAL_ SAMPLING_INT32	ML_TEMPORAL_SAMPLING_FIELD_BASED ML_TEMPORAL_SAMPLING_PROGRESSIVE	Specifies whether the image temporal sampling is progressive or interlaced.
ML_IMAGE_ INTERLEAVE_MODE_ INT32	ML_INTERLEAVE_MODE_INTERLEAVED ML_INTERLEAVE_MODE_SINGLE_FIELD	Specifies whether the two fields are interleaved into a single image (and reside in a single buffer) or are stored in two separate fields (therefore, in two separate buffers). This parameter is used only for interlaced images and is ignored for signals with progressive timing.
ML_IMAGE_ DOMINANCE_INT32	ML_DOMINANCE_F1 (default) ML_DOMINANCE_F2	Identifies frame boundaries in a field sequence (interlaced formats).
ML_IMAGE_BUFFER_ SIZE_INT32	The size of the image buffer in bytes.	Obtains the size of the image buffer in bytes. This read-only parameter is computed using the current path control settings. This value represents the worst-case buffer size.
ML_VIDEO_UST_INT64	Value of unadjusted system time (UST) at the time the data begins to move through the jack.	Timestamps when the data begins to move through the jack. For video, this corresponds to the beginning of a vertical sync. This parameter can be sent to a video path along with a buffer, using the dmSendBuffers call.
ML_VIDEO_MSC_ INT64	Value of media stream count (MSC) at the time the data begins to move through the jack	Timestamps when the data begins to move through the jack. For video, this corresponds to the beginning of a vertical sync. This parameter can be sent to a video path along with a buffer, using the dmSendBuffers call.

Table 4-2 DM3 Board Control Parameters, Value(s), and Use **(continued)**

Control Parameters	Range or Value(s)	Use
ML_VIDEO_TIMING_INT32	ML_TIMING_1125_1920x1080_60i ML_TIMING_1125_1920x1080_5994i ML_TIMING_750_1280x720_60p ML_TIMING_750_1280x720_5994p ML_TIMING_1125_1920x1080_50i ML_TIMING_1125_1920x1080_25p ML_TIMING_1125_1920x1080_24p ML_TIMING_1125_1920x1080_2398p ML_TIMING_1125_1920x1035_5994i ML_TIMING_1125_1920x1080_24PsF ML_TIMING_1125_1920x1080_2398PsF ML_TIMING_1125_1920x1080_25PsF ML_TIMING_525 ML_TIMING_625	Sets the timing on an input or output video path. Not all timings are supported on all devices. On devices that can auto-detect, the timing can be read-only on input.
ML_VIDEO_COLORSPACE_INT32	ML_COLORSPACE_CbYCr_601_HEAD ML_COLORSPACE_RGB_601_HEAD ML_COLORSPACE_CbYCr_709_HEAD ML_COLORSPACE_RGBr_709_HEAD ML_COLORSPACE_CbYCr_240M_HEAD ML_COLORSPACE_RGB_240M_HEAD	Specifies the colorspace of the video data on the video jack. For input paths, this is the colorspace you expect to receive at the jack. For output paths, it is the colorspace you desire at the jack.
ML_VIDEO_PRECISION_INT32	[8 - 8], [10 - 10]	Specifies whether the external video interface is 8 bits or 10 bits wide. This is the precision (number of bits of resolution) of the signal at the jack. This is an integer. A precision value of 10 specifies a 10-bit signal. A value of 8 specifies an 8-bit signal.
ML_VIDEO_SAMPLING_INT32	ML_SAMPLING_422 ML_SAMPLING_4224 ML_SAMPLING_444 ML_SAMPLING_4444	Specifies the sampling of video data on the video jack.
ML_VIDEO_START_X_INT32	[1 - 1]	Sets the start horizontal location on each line of the video signal.

Table 4-2 DM3 Board Control Parameters, Value(s), and Use **(continued)**

Control Parameters	Range or Value(s)	Use
ML_VIDEO_START_Y_F1_INT32	Defined by selected timing.	Sets the start vertical location on F1 fields of the video signal. For progressive signals, this parameter specifies the start of every frame.
ML_VIDEO_START_Y_F2_INT32	Defined by selected timing.	Sets the start vertical location on F2 fields of the video signal. This parameter is ignored for progressive timing signals.
ML_VIDEO_WIDTH_INT32	Defined by selected timing.	Defines video width. This must be the same as ML_IMAGE_WIDTH_INT32. Sets the horizontal width of the clipping region on each line of the video signal.
ML_VIDEO_HEIGHT_F1_INT32	Number of lines in the frame for progressive formats. Number of lines in the first field for interlaced or progressive segmented frames (PsF) formats.	Defines the height of F1 field in lines. Must be set to full image height for progressive formats.
ML_VIDEO_HEIGHT_F2_INT32	Number of lines in the second field for interlaced or PsF formats.	Defines the height of F2 field in lines. For progressive signals, this value must be 0.
XTDIGVID_LUT_YG_INT32_ARRAY	Integer for YG LUT values.	Defines the LUT values for the YG LUT.
XTDIGVID_LUT_UB_INT32_ARRAY	Integer for UB LUT values.	Defines the LUT values for the UB LUT.
XTDIGVID_LUT_VR_INT32_ARRAY	Integer for VR LUT values.	Defines the LUT values for the VR LUT.
ML_DEVICE_STATE_INT32	ML_DEVICE_STATE_TRANSFERRING ML_DEVICE_STATE_WAITING ML_DEVICE_STATE_ABORTING ML_DEVICE_STATE_FINISHING ML_DEVICE_STATE_READY	Defines how the device transitions through well-known states. The device state can be changed with dmBeginTransfer() and dmEndTransfer().
ML_DEVICE_EVENTS_INT32_ARRAY	ML_EVENT_VIDEO_SEQUENCE_LOST ML_EVENT_VIDEO_SYNC_LOST ML_EVENT_VIDEO_SYNC_GAINED	Processes exceptional events, which require that the device send a message back to the application. The application must explicitly ask for such events.
ML_QUEUE_SEND_COUNT_INT32	Number of messages in the queue.	Returns the number of items in the queue from the application to the device.

Table 4-2 DM3 Board Control Parameters, Value(s), and Use **(continued)**

Control Parameters	Range or Value(s)	Use
ML_QUEUE_RECEIVE_ COUNT_INT32	Number of messages in the queue.	Returns the number of items (both replies and events) in the queue between the device and the application.
ML_QUEUE_SEND_ WAITABLE_INT32	Number of messages in the queue.	Returns an ML waitable (a file descriptor in UNIX implementations). This occurs when more than N slots are free in the queue between the application and the device. This is used when an application needs to enqueue messages in chunks. The N setting is specified at open time via ML_OPEN_SEND_SIGNAL_COUNT.
ML_QUEUE_RECEIVE_ WAITABLE_INT32	Number of messages in the queue.	Returns an ML waitable (a file descriptor in UNIX implementations). This occurs when an unread message from the device to the application is in the queue. Applications can wait on this, rather than polling for replies from the device.
ML_VIDEO_ GENLOCK_TYPE_INT32	XTDIGVID_GENLOCK_SRC_TYPE_INTERNAL XTDIGVID_GENLOCK_SRC_TYPE_ANALOG XTDIGVID_GENLOCK_SRC_TYPE_DIGITAL_ JACK_A XTDIGVID_GENLOCK_SRC_TYPE_DIGITAL_ JACK_B	Sets the desired type of genlock sync source. Describes the genlock signal type. Only accepted on output paths. Each genlock type is specified as either a 32-bit resource ID or ML_VIDEO_GENLOCK_TYPE_INTERNAL.

Table 4-2 DM3 Board Control Parameters, Value(s), and Use **(continued)**

Control Parameters	Range or Value(s)	Use
ML_VIDEO_ GENLOCK_SOURCE_ TIMING_INT32	XTDIGVID_GENLOCK_TIMING_ AUTODETECT ML_TIMING_525 ML_TIMING_625 ML_TIMING_1125_1920x1080_60i ML_TIMING_1125_1920x1080_5994i ML_TIMING_750_1280x720_60p ML_TIMING_750_1280x720_5994p ML_TIMING_1125_1920x1080_50i ML_TIMING_1125_1920x1080_25p ML_TIMING_1125_1920x1080_24p ML_TIMING_1125_1920x1080_2398p ML_TIMING_1125_1920x1035_5994i ML_TIMING_1125_1920x1080_24PsF ML_TIMING_1125_1920x1080_2398PsF ML_TIMING_1125_1920x1080_25PsF	Describes the genlock source timing. Only accepted on output paths. Each genlock source is specified as an output timing on the path and corresponds to the same timings that are provided for ML_VIDEO_TIMING_INT32.
ML_VIDEO_H_PHASE_INT32	Defined by selected timing.	Sets or gets the video signal horizontal phase relative to the genlock offset.
ML_VIDEO_V_PHASE_INT32	Defined by selected timing.	Sets or gets the video signal vertical phase relative to the genlock offset.

Table 4-2 DM3 Board Control Parameters, Value(s), and Use **(continued)**

Control Parameters	Range or Value(s)	Use
ML_VIDEO_OUTPUT_REPEAT_INT32	ML_VIDEO_REPEAT_NONE ML_VIDEO_REPEAT_FIELD ML_VIDEO_REPEAT_FRAME	Controls whether the system repeats DMbuffers when the output is underflowing. If the application is performing output, and fails to provide buffers fast enough (the queue to the device under-flows), then this control determines the device behavior. ML_VIDEO_REPEAT_NONE The device does nothing, usually resulting in black output. ML_VIDEO_REPEAT_FIELD The device repeats the last field for interlaced non-interleaved transfers. For progressive signals or interleaved formats, this is the same as ML_VIDEO_REPEAT_FRAME. ML_VIDEO_REPEAT_FRAME The device repeats the last two fields. This output capability is device dependent. The allowable settings should be queried via the get capabilities of ML_VIDEO_OUTPUT_REPEAT_INT32. On input, any signal outside the clipping region is simply ignored. On output, specific parameters control the generated signal. For more information on these parameters, see the *OpenML Media Library Software Development Kit Programmer's Guide*.
XTDIGVID_EE_MODE_INT32	XTDIGVID_EE_MODE_DISABLE XTDIGVID_EE_MODE_ENABLE	Loops-through serial input to serial output.
XTDIGVID_FF_MODE_INT32	XTDIGVID_FF_MODE_DISABLE XTDIGVID_FF_MODE_ENABLE	Assists an application in performing 3/2 pulldown on output. Field/frame mode.
XTDIGVID_LOOPBACK_INT32	XTDIGVID_LOOPBACK _DISABLE XTDIGVID_LOOPBACK _ENABLE	Sends the VBOB LVDS input (DMediaPro output) to the VBOB LVDS output.

Table 4-2 DM3 Board Control Parameters, Value(s), and Use **(continued)**

Control Parameters	Range or Value(s)	Use
ML_VIDEO_ GENLOCK_SIGNAL_ PRESENT_INT32	ML_TIMING_NONE ML_TIMING_UNKNOWN ML_TIMING_525 ML_TIMING_625 ML_TIMING_1125_1920x1080_60i ML_TIMING_1125_1920x1080_5994i ML_TIMING_750_1280x720_60p ML_TIMING_750_1280x720_5994p ML_TIMING_1125_1920x1080_50i ML_TIMING_1125_1920x1080_25p ML_TIMING_1125_1920x1080_24p ML_TIMING_1125_1920x1080_2398p ML_TIMING_1125_1920x1035_5994i ML_TIMING_1125_1920x1080_24PsF ML_TIMING_1125_1920x1080_2398PsF	Provides read-only controls on the video output path. getControls returns the timing on the currently selected genlock jack. This control only functions after opening a path.
ML_VIDEO_SIGNAL_ PRESENT_INT32	ML_TIMING_NONE ML_TIMING_UNKNOWN ML_TIMING_525 ML_TIMING_625 ML_TIMING_1125_1920x1080_60i ML_TIMING_1125_1920x1080_5994i ML_TIMING_750_1280x720_60p ML_TIMING_750_1280x720_5994p ML_TIMING_1125_1920x1080_50i ML_TIMING_1125_1920x1080_25p ML_TIMING_1125_1920x1080_24p ML_TIMING_1125_1920x1080_2398p ML_TIMING_1125_1920x1035_5994i ML_TIMING_1125_1920x1080_24PsF ML_TIMING_1125_1920x1080_2398PsF	Provides read-only controls on the video input path. getControls returns the detected timing on the currently selected input jack. This control only functions after opening a path.
XTDIGVID_GENLOCK_ STATE_INT32	XTDIGVID_GENLOCK_STATE_IS_UNLOCKED XTDIGVID_GENLOCK_STATE_IS_LOCKED XTDIGVID_GENLOCK_STATE_IS_UNKNOWN	Provides read-only controls on the video output paths. getControls returns the current genlock state of an open path. If GENLOCK_TYPE is INTERNAL, this control returns *_UNLOCKED.

Table 4-2 DM3 Board Control Parameters, Value(s), and Use **(continued)**

Control Parameters	Range or Value(s)	Use
XTDIGVID_GENLOCK_ERROR_STATUS_INT32	XTDIGVID_GENLOCK_ERROR_STATUS_NONE XTDIGVID_GENLOCK_ERROR_STATUS_NO_SIGNAL XTDIGVID_GENLOCK_ERROR_STATUS_UNKNOWN_SIGNAL XTDIGVID_GENLOCK_ERROR_STATUS_ILLEGAL_COMBINATION XTDIGVID_GENLOCK_ERROR_STATUS_TIMING_MISMATCH	Provides read-only controls on the video output paths. This control can be used to query for the cause of a SYNC_LOST event. It can also be used to query for the cause of a GENLOCK_STATE_IS_UNLOCKED return when the expected return is GENLOCK_STATE_IS_LOCKED.
ML_OPEN_MODE_INT32	ML_MODE_RWE	Specifies that the board can only be opened in the RWE mode. The application's intended use for the device.
ML_OPEN_SEND_QUEUE_COUNT_INT32	[2 - 1024]	Specifies the application's preferred size (number of messages) in the send queue. This affects the amount of memory allocated for this queue when the device is opened. The default is device-dependent.
ML_OPEN_RECEIVE_QUEUE_COUNT_INT32	[4 - 1024]	Specifies the application's preferred size (number of messages) in the receive queue. This affects the amount of memory allocated for this queue when the device is opened. The default is device-dependent. A null value indicates that the application does not expect to receive any events from a jack.
ML_OPEN_MESSAGE_PAYLOAD_SIZE_INT32	[1024 - 134217728]	Specifies the application's preferred size (in bytes) in the queue message payload area. This payload area holds messages in both the send and receive queues. The default is device-dependent.

Table 4-2 DM3 Board Control Parameters, Value(s), and Use **(continued)**

Control Parameters	Range or Value(s)	Use
ML_OPEN_EVENT_ PAYLOAD_COUNT_ INT32	[2 - 1024]	Specifies the application's preferred size (number of messages) in the queue event payload area. This payload area holds the contents of event messages in the receive queue. The default is device-dependent. A null value indicates that the application does not expect to receive any events from a jack.
ML_OPEN_SEND_ SIGNAL_COUNT_INT32	[1 - 1024]	Specifies the application's preferred low-water level (number of empty message slots) in the send queue. When the device de-queues a message and causes the number of empty slots to exceed this level, then the device signals the send queue event. The default is device-dependent.

DMediaPro Default Path Controls

This section provides the default values for the HD and SD input path controls and HD and SD output path controls.

HD Default Input Path Controls

The following default values are for the HD input path controls:

```
ML_VIDEO_TIMING_INT32 = ML_TIMING_1125_1920x1080_5994i
ML_VIDEO_PRECISION_INT32 = 8
ML_VIDEO_COLORSPACE_INT32 = ML_COLORSPACE_CbYCr_709_HEAD
ML_IMAGE_TEMPORAL_SAMPLING_INT32 = ML_TEMPORAL_SAMPLING_FIELD_BASED
ML_VIDEO_SAMPLING_INT32 = ML_SAMPLING_422
ML_VIDEO_START_Y_F1_INT32 = 21
ML_VIDEO_START_Y_F2_INT32 = 584
ML_VIDEO_HEIGHT_F1_INT32 = 540
ML_VIDEO_HEIGHT_F2_INT32 = 540
ML_VIDEO_WIDTH_INT32 = 1920
ML_IMAGE_WIDTH_INT32 = 1920
ML_IMAGE_HEIGHT_1_INT32 = 1080
ML_IMAGE_HEIGHT_2_INT32 = 0
ML_IMAGE_PACKING_INT32 = ML_PACKING_8
ML_IMAGE_SAMPLING_INT32 = ML_SAMPLING_444
ML_IMAGE_COLORSPACE_INT32 = ML_COLORSPACE_RGB_709_FULL
ML_IMAGE_INTERLEAVE_MODE_INT32 = ML_INTERLEAVE_MODE_INTERLEAVED
ML_IMAGE_DOMINANCE_INT32 = ML_DOMINANCE_F1
ML_IMAGE_ORIENTATION_INT32 = ML_ORIENTATION_TOP_TO_BOTTOM
XTDIGVID_LOOPBACK_INT32 = XTDIGVID_LOOPBACK_DISABLE
ML_IMAGE_COMPRESSION_INT32 = ML_COMPRESSION_UNCOMPRESSED
ML_IMAGE_SKIP_PIXELS_INT32 = 0
ML_IMAGE_ROW_BYTES_INT32 = 0
ML_IMAGE_SKIP_ROWS_INT32 = 0
ML_VIDEO_START_X_INT32 = 1
```

SD Default Input Path Controls

The following default values are for the SD input path controls:

```
ML_VIDEO_TIMING_INT32 = ML_TIMING_525
ML_VIDEO_PRECISION_INT32 = 8
ML_VIDEO_COLORSPACE_INT32 = ML_COLORSPACE_CbYCr_601_HEAD
ML_IMAGE_TEMPORAL_SAMPLING_INT32 = ML_TEMPORAL_SAMPLING_FIELD_BASED
ML_VIDEO_SAMPLING_INT32 = ML_SAMPLING_422
ML_VIDEO_START_Y_F1_INT32 = 20
ML_VIDEO_START_Y_F2_INT32 = 283
ML_VIDEO_HEIGHT_F1_INT32 = 244
ML_VIDEO_HEIGHT_F2_INT32 = 243
ML_VIDEO_WIDTH_INT32 = 720
ML_IMAGE_WIDTH_INT32 = 720
ML_IMAGE_HEIGHT_1_INT32 = 487
ML_IMAGE_HEIGHT_2_INT32 = 0
ML_IMAGE_PACKING_INT32 = ML_PACKING_8
ML_IMAGE_SAMPLING_INT32 = ML_SAMPLING_444
ML_IMAGE_COLORSPACE_INT32 = ML_COLORSPACE_RGB_601_FULL
ML_IMAGE_INTERLEAVE_MODE_INT32 = ML_INTERLEAVE_MODE_INTERLEAVED
ML_IMAGE_DOMINANCE_INT32 = ML_DOMINANCE_F1
ML_IMAGE_ORIENTATION_INT32 = ML_ORIENTATION_TOP_TO_BOTTOM
XTDIGVID_LOOPBACK_INT32 = XTDIGVID_LOOPBACK_DISABLE
ML_IMAGE_COMPRESSION_INT32 = ML_COMPRESSION_UNCOMPRESSED
ML_IMAGE_SKIP_PIXELS_INT32 = 0
ML_IMAGE_ROW_BYTES_INT32 = 0
ML_IMAGE_SKIP_ROWS_INT32 = 0
ML_VIDEO_START_X_INT32 = 1
```

HD Default Output Path Controls

The following default values are for the HD output path controls:

```
ML_VIDEO_TIMING_INT32 = ML_TIMING_1125_1920x1080_5994i
ML_VIDEO_PRECISION_INT32 = 8
ML_VIDEO_COLORSPACE_INT32 = ML_COLORSPACE_CbYCr_709_HEAD
ML_IMAGE_TEMPORAL_SAMPLING_INT32 = ML_TEMPORAL_SAMPLING_FIELD_BASED
ML_VIDEO_SAMPLING_INT32 = ML_SAMPLING_422
ML_VIDEO_START_Y_F1_INT32 = 21
ML_VIDEO_START_Y_F2_INT32 = 584
ML_VIDEO_HEIGHT_F1_INT32 = 540
ML_VIDEO_HEIGHT_F2_INT32 = 540
ML_VIDEO_WIDTH_INT32 = 1920
ML_IMAGE_WIDTH_INT32 = 1920
ML_IMAGE_HEIGHT_1_INT32 = 1080
ML_IMAGE_HEIGHT_2_INT32 = 0
ML_IMAGE_PACKING_INT32 = ML_PACKING_8
ML_IMAGE_SAMPLING_INT32 = ML_SAMPLING_444
ML_IMAGE_COLORSPACE_INT32 = ML_COLORSPACE_RGB_709_FULL
ML_IMAGE_INTERLEAVE_MODE_INT32 = ML_INTERLEAVE_MODE_INTERLEAVED
ML_IMAGE_DOMINANCE_INT32 = ML_DOMINANCE_F1
ML_IMAGE_ORIENTATION_INT32 = ML_ORIENTATION_TOP_TO_BOTTOM
ML_VIDEO_GENLOCK_TYPE_INT32 = XTDIGVID_GENLOCK_SRC_TYPE_INTERNAL
ML_VIDEO_GENLOCK_SOURCE_TIMING_INT32 = ML_TIMING_1125_1920x1080_5994i
ML_VIDEO_OUTPUT_REPEAT_INT32 = ML_VIDEO_REPEAT_NONE
XTDIGVID_EE_MODE_INT32 = XTDIGVID_EE_MODE_DISABLE
XTDIGVID_FF_MODE_INT32 = XTDIGVID_FF_MODE_DISABLE
ML_IMAGE_COMPRESSION_INT32 = ML_COMPRESSION_UNCOMPRESSED
ML_IMAGE_SKIP_PIXELS_INT32 = 0
ML_IMAGE_ROW_BYTES_INT32 = 0
ML_IMAGE_SKIP_ROWS_INT32 = 0
ML_VIDEO_START_X_INT32 = 1
```

SD Default Output Path Controls

The following default values are for the SD output path controls:

```
ML_VIDEO_TIMING_INT32 = ML_TIMING_525
ML_VIDEO_PRECISION_INT32 = 8
ML_VIDEO_COLORSPACE_INT32 = ML_COLORSPACE_CbYCr_601_HEAD
ML_IMAGE_TEMPORAL_SAMPLING_INT32 = ML_TEMPORAL_SAMPLING_FIELD_BASED
ML_VIDEO_SAMPLING_INT32 = ML_SAMPLING_422
ML_VIDEO_START_Y_F1_INT32 = 20
ML_VIDEO_START_Y_F2_INT32 = 283
ML_VIDEO_HEIGHT_F1_INT32 = 244
ML_VIDEO_HEIGHT_F2_INT32 = 243
ML_VIDEO_WIDTH_INT32 = 720
ML_IMAGE_WIDTH_INT32 = 720
ML_IMAGE_HEIGHT_1_INT32 = 487
ML_IMAGE_HEIGHT_2_INT32 = 0
ML_IMAGE_PACKING_INT32 = ML_PACKING_8
ML_IMAGE_SAMPLING_INT32 = ML_SAMPLING_444
ML_IMAGE_COLORSPACE_INT32 = ML_COLORSPACE_RGB_601_FULL
ML_IMAGE_INTERLEAVE_MODE_INT32 = ML_INTERLEAVE_MODE_INTERLEAVED
ML_IMAGE_DOMINANCE_INT32 = ML_DOMINANCE_F1
ML_IMAGE_ORIENTATION_INT32 = ML_ORIENTATION_TOP_TO_BOTTOM
ML_VIDEO_GENLOCK_TYPE_INT32 = XTDIGVID_GENLOCK_SRC_TYPE_INTERNAL
ML_VIDEO_GENLOCK_SOURCE_TIMING_INT32 = ML_TIMING_525
ML_VIDEO_OUTPUT_REPEAT_INT32 = ML_VIDEO_REPEAT_NONE
XTDIGVID_EE_MODE_INT32 = XTDIGVID_EE_MODE_DISABLE
XTDIGVID_FF_MODE_INT32 = XTDIGVID_FF_MODE_DISABLE
ML_IMAGE_COMPRESSION_INT32 = ML_COMPRESSION_UNCOMPRESSED
ML_IMAGE_SKIP_PIXELS_INT32 = 0
ML_IMAGE_ROW_BYTES_INT32 = 0
ML_IMAGE_SKIP_ROWS_INT32 = 0
ML_VIDEO_START_X_INT32 = 1
```

ML_TIMING

The ML_TIMING control sets the timing type, which expresses the timing of video presented to an input or an output.

Each value for ML_TIMING indicates the raster configuration of a particular SMPTE specification, such as SMPTE 274M-1995. The values are named according to the raster format:

- The first field is the number of total lines, such as 1125, 750, 525, or 625.
- The second field is the size of the active region, in pixels by lines.
- The third field is the vertical refresh rate and the scanning format; the scanning format is as follows:
  - i: interlaced
  - p: progressive (noninterlaced)
  - PsF (progressive segmented frame)

    In PsF formats, the frame is transmitted as two fields that are of the same time instant; in interlaced formats, the two fields are temporally displaced.

For example, ML_TIMING_1125_1920x1080_5994i specifies 1125 total lines, an active region of 1920 pixels by 1080 lines, 59.94 fields per second, and 2:1 interlacing.

Note: If you change the ML_TIMING from SD to HD or HD to SD at the beginning of a data transfer, several seconds may elapse before the change takes effect.

Input Timing Auto Detect

The input timing auto detect function for the DM3 board determines whether the detected signal matches the user requested signal. If the detected signal does not match the requested signal, there are two possible results:

- If the application is registered for SYNC_LOST events on the input path, the application receives a SYNC_LOST event.
- If the device is not looking for a SYNC_LOST event, an ML_DEVICE error is sent to the application. If an error is detected before a transfer is started, the dmmodule does not allow the transfer to begin.

For more information on input timing-related events and device errors, see "DMediaPro Events" on page 60.

Genlock

Genlock enables the SGI VBOB to receive an external sync signal, which locks the timing of the output video picture. This allows you to maintain a common timing across multiple video devices. With the DMediaPro/VBOB system, you can set up genlock as follows:

- SD output
  - ML_TIMING_525 locked to an NTSC reference connected to the VBOB SD video input or SD genlock input
  - ML_TIMING_625 locked to a PAL reference connected to the VBOB SD video input or SD genlock input
- HD output
  - Supported genlock input timings (see Table 4-3) referenced by the VBOB HD video input or HD genlock input and locked to a compatible video output timing. The only exceptions are ML_TIMING_ 525 and ML_TIMING_625, which can be used only as a reference on the VBOB HD genlock input, but not on the VBOB HD video input.

Table 4-3 lists each of the supported genlock input timings and their compatible HD video output timings. The prefix, ML_TIMING_ should appear before each timing, but was omitted to avoid redundancy.

Table 4-3 Supported HD Input/Output Timings

Genlock Input Timing	Video Output Timing
525	1125_1920x1080_5994i 750_1280x720_5994p 1125_1920x1080_2398p 1125_1920x1080_2398PsF 1125_1920x1035_5994i
625	1125_1920x1080_50i 1125_1920x1080_25p 1125_1920x1080_25PsF
750_1280x720_60p	750_1280x720_60p 1125_1920x1080_24p 1125_1920x1080_24PsF 1125_1920x1080_60i
750_1280x720_5994p	750_1280x720_5994p 1125_1920x1035_5994i 1125_1920x1080_5994i 1125_1920x1080_2398p 1125_1920x1080_2398PsF
1125_1920x1080_5994i	750_1280x720_5994p 1125_1920x1035_5994i 1125_1920x1080_5994i 1125_1920x1080_2398p 1125_1920x1080_2398PsF
1125_1920x1080_2398p	1125_1920x1080_2398p 1125_1920x1080_2398PsF
1125_1920x1080_2398PsF	1125_1920x1080_2398PsF 1125_1920x1080_2398p
1125_1920x1080_24p	1125_1920x1080_24p 1125_1920x1080_24PsF

Table 4-3 Supported HD Input/Output Timings **(continued)**

Genlock Input Timing	Video Output Timing
1125_1920x1080_24PsF	1125_1920x1080_24PsF 1125_1920x1080_24p
1125_1920x1080_50i	1125_1920x1080_50i 1125_1920x1080_25p 1125_1920x1080_25PsF
1125_1920x1080_60i	1125_1920x1080_60i 750_1280x720_60p 1125_1920x1080_24p 1125_1920x1080_24PsF
1125_1920x1035_5994i	1125_1920x1035_5994i 750_1280x720_5994p 1125_1920x1080_2398p 1125_1920x1080_2398PsF 1125_1920x1080_5994i
1125_1920x1080_25p	1125_1920x1080_25p 1125_1920x1080_25PsF 1125_1920x1080_50i
1125_1920x1080_25 PsF	1125_1920x1080_25p 1125_1920x1080_25PsF 1125_1920x1080_50i

Genlock Auto Detect

The genlock auto detect function for the DM3 board automatically identifies the genlock input signal (genlock source timing), and determines whether the signal is compatible with the video output signal. If the DM3 board determines that the two signals are compatible, the board automatically locks the genlock source timing with the video output timing.

To enable this function, set the ML_VIDEO_GENLOCK_SOURCE_TIMING_INT32 control parameter to the XTDIGVID_GENLOCK_TIMING_AUTODETECT value (see Table 4-2 on page 31).

For more information on genlock-related events and device errors, see "DMediaPro Events" on page 60.

ML_IMAGE_PACKING

In ML, the combination of ML_IMAGE_PACKING and ML_IMAGE_SAMPLING is the equivalent to VL_PACKING in the video library (VL) environment. Table 4-4 shows how the ML combinations correspond to VL_PACKING. If you are unfamiliar with VL_PACKING, see the *HD I/O Board Owner's Guide* for a detailed description.

Table 4-4 VL/ML Packing Conversions

VL_PACKING	ML_IMAGE_PACKING	ML_IMAGE_SAMPLING
VL_PACKING_R242_8	ML_PACKING_8	ML_SAMPLING_422
VL_PACKING_242_8	ML_PACKING_8_3214	ML_SAMPLING_422
VL_PACKING_R242_10	ML_PACKING_10	ML_SAMPLING_422
VL_PACKING_242_10	ML_PACKING_10_3214	ML_SAMPLING_422
VL_PACKING_R242_10_in_16_L	ML_PACKING_10in16L	ML_SAMPLING_422
VL_PACKING_242_10_in_16_L	ML_PACKING_10in16L_3214	ML_SAMPLING_422
VL_PACKING_R242_10_in_16_R	ML_PACKING_10in16R	ML_SAMPLING_422
VL_PACKING_242_10_in_16_R	ML_PACKING_10in16R_3214	ML_SAMPLING_422
VL_PACKING_R2424_10_10_10_2Z	ML_PACKING_10_10_10_2	ML_SAMPLING_4224

Table 4-4 VL/ML Packing Conversions **(continued)**

VL_PACKING	ML_IMAGE_PACKING	ML_IMAGE_SAMPLING
VL_PACKING_2424_10_10_10_2Z	ML_PACKING_10_10_10_2_3214	ML_SAMPLING_4224
VL_PACKING_444_8	ML_PACKING_8	ML_SAMPLING_444
VL_PACKING_R444_8	ML_PACKING_8_R	ML_SAMPLING_444
VL_PACKING_444_12_in_16L	ML_PACKING_S12in16L	ML_SAMPLING_444
VL_PACKING_444_12_in_16_R	ML_PACKING_S12in16R	ML_SAMPLING_444
VL_PACKING_4444_8	ML_PACKING_8	ML_SAMPLING_4444
VL_PACKING_R4444_8	ML_PACKING_8_R	ML_SAMPLING_4444
VL_PACKING_4444_10_10_10_2	ML_PACKING_10_10_10_2	ML_SAMPLING_4444
VL_PACKING_R4444_10_10_10_2	ML_PACKING_10_10_10_2_R	ML_SAMPLING_4444

Note: If you change packings at the beginning of a data transfer, several seconds may elapse before the change takes effect.

ML_COLORSPACE

The ML_COLORSPACE control specifies the color space of video data in memory or for input and output. A color space is a color component encoding format, for example, RGB and YUV. Because video equipment uses more than one color space, the DMediaPro video paths, in addition to the image memory buffers, support the ML_COLORSPACE control.

Each component of an image has:

- A color that it represents
- A canonical minimum value
- A canonical maximum value

Normally, a component stays within the minimum and maximum values. For example, for a luma signal such as Y, you can think of these limits as the black level and the peak white level, respectively. For an unsigned component with *n* bits, you can determine the full range minimum value and/or maximum value as follows:

$[0, (2^n)-1]$

This provides the maximum resolution for each component.

Color Spaces and Color Models

Various HDTV specifications define color models differently from those defined in Recommendation 601 (ITU-R BT.601-5), which is used by most standard-definition digital video equipment. For HDTV, the DM3 board supports the following three color models:

- SMPTE 240M
- Recommendation 709 (ITU-R BT.709-2)
- Recommendation 601 (ITU-R BT.601-5)

Within each color model, four different color spaces exist:

- YCrCb: headroom range

  Headroom range means that black is at, for example, code 64 rather than 0, and white is at, for example, code 940 rather than 1023. Headroom-range color spaces can accommodate overshoot (superwhite) and undershoot (superblack) colors. Full-range color spaces clamp these out-of-range colors to black and white.
- YUV: full range
- RGB_H: headroom range
- RGB_F: full range

For image memory buffers, these four color spaces are defined for each of three color models, resulting in 12 color spaces. Note that all 12 are supported on image memory buffers, but only YCrCb and RGB_H color spaces are supported on video paths.

Color space conversion is performed within a color model if the color spaces are different on the image memory buffer and video paths. Conversion between the color models is not supported.

Note: If you change this control at the beginning of a data transfer, several seconds may elapse before the change takes effect.

Typically, two sets of colors are used together, RGB (RGBA) and YCrCb/YUV (VYUA). YCrCb (YUV), the most common representation of color from the video world, represents each color by a luma component called Y and two components of chroma, called Cr (or V), and Cb (or U). The luma component is loosely related to brightness or luminance, and the chroma components make up a quantity loosely related to hue. These components are defined in ITU-R BT.601-5 (also known as Rec. 601 and CCIR 601), ITU-R BT.709-2, and SMPTE 240M.

The alpha channel is not a real color. For that channel, the minimum value specifies completely transparent, and the maximum value specifies completely opaque.

For more information about color spaces, see *A Technical Introduction to Digital Video*, by Charles A. Poynton (New York: Wiley, 1996).

ML_COLORSPACE Control of Blanking

Along with image memory buffer color space, ML_COLORSPACE determines the color-conversion matrix values. In addition, this control affects the type of blanking output by the board during horizontal and vertical blanking, and during an active video timeframe when data is not being transferred. On a video output path, ML_COLORSPACE affects the type of blanking that the board outputs, in accordance with SMPTE 274M:

- YCrCb: blanking is Y = 64, Cr/Cb = 512, A = 64
- RGB_H: blanking is R = 64, G = 64, B = 64, A = 64

ML_COLORSPACE and Lookup Tables

The DM3 board supports lookup tables (LUTs) on input and output for gamma correction or decorrection. To successfully run an application with linear components, you can use LUTs to convert between linear and nonlinear spaces.

The DM3 board hardware has three LUTs, one LUT for each RGB color component. Each LUT has 8,192 entries; each entry stores 13 bits. The application programs the entries in each table. The LUTs produce offsets, if they are required by the memory storage format.

The LUTs perform rounding as follows:

- If the LUT is not explicitly programmed by the application, the output LUT is in pass-through mode, all rounding is performed in the color space converter, and the input LUT performs both rounding and offset.
- If the LUT is programmed explicitly by the application, the application can control rounding as part of the lookup table function. The packer (hardware that reads the LUT and formats data for the host memory; see Figure 4-3) performs a final conversion from 13-bit LUT format to host memory format.

An application can also use the LUT to convert between video path RGB_H and image memory buffer RGB_F. Because each component is independent of the others for this conversion, a matrix multiplication is not needed (pass-through mode). The required component scaling and rounding can be placed into each LUT.

ML_COLORSPACE Example

Figure 4-3 shows an example color space conversion. In the example, RGB are values in linear space and R’G’B’ are values in nonlinear space after the opto-electric transfer function is applied as specified in ITU-R BT.709. You can use the LUTs to apply this function or its inverse to convert between RGB and R’G’B’.

This example also shows a typical video capture path. The input jack is YCrCb 4:2:2 and the desired result in system memory is RGB. First, an appropriate filter interpolates YCrCb 4:2:2 to YCrCb 4:4:4 to fill in the missing CrCb samples. Then a 3x3 matrix multiplier with appropriate offsets and coefficients obtains RGB values for each pixel. At this point, you can use the LUT option to convert gamma pre-corrected RGB values to linear RGB values. Finally, the packer swizzles the bits into the desired memory packing format and DMA places the result in system memory.

Figure 4-3 Color Space Conversion Example

Field Dominance

Field dominance identifies the frame boundaries in a field sequence; that is, it specifies which pair of fields in a field sequence constitutes a frame. You can use ML_IMAGE_DOMINANCE_INT32 to specify where an edit occurs, as follows:

- ML_DOMINANCE_F1: the edit occurs on the nominal video field boundary (field 1 or F1).
- ML_DOMINANCE_F2: the edit occurs on the intervening field boundary (field 2 or F2).

You can determine whether a field is field 1 or field 2 by setting bit 9, the F bit, in the XYZ word of the EAV and SAV sequences, as follows:

- For field 1 (also called the odd field), set the F bit to 0.
- For field 2 (also called the even field), set the F bit to 1.

Note: Field dominance has no effect on progressive timings.

Figure 4-4 shows fields and frames as defined for digital 1080-line formats for the DM3 board.

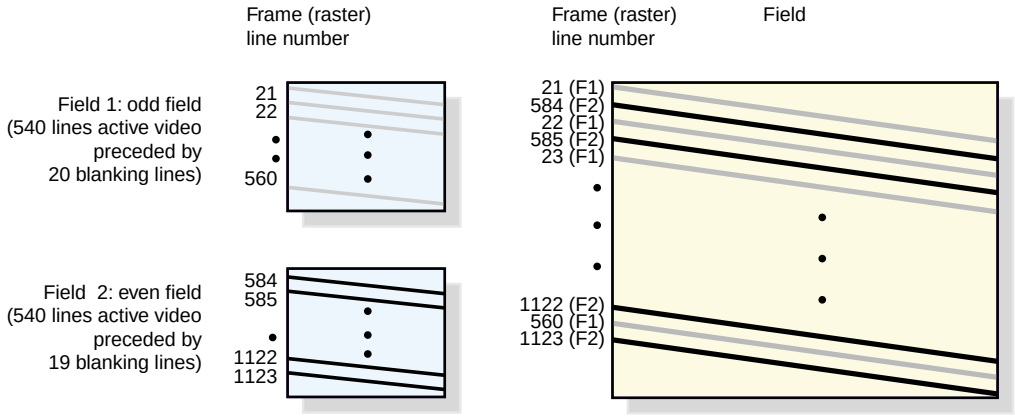

Figure 4-4 Fields and Frames for SMPTE 274M

Editing is usually on field 1 boundaries, where field 1 is defined as the first field in the video standard's two-field output sequence. However, you may want to edit on F2 boundaries, which fall on the field between the video standard's frame boundary. To do so, use this control, then program your deck to select F2 edits.

A set of frames for output must be de-interlaced into fields differently, depending on the specified output field dominance. For SMPTE 274M, the top line is in F1, as shown in Figure 4-4. For SMPTE 240M, the top line is in F2. For example, when F1 dominance is selected, the field with the topmost line must be the first field to be transferred; when F2 dominance is selected, the field with the topmost line must be the second field to be transferred.

EE Mode

The DM3 board supports an EE mode (XTDIGVID_EE_MODE_INT32). In this mode, the serial input is looped-through directly to the serial output. EE mode can only function correctly if the LVDS output is genlocked to the same source as the device feeding the LVDS input. This genlock mode is commonly referred to as "reclocking," which is used in DAs, D to As, and data serializers. Reclocking ensures that the re-transmitted signals have sufficient jitter attenuation applied to reject jitter from the digital inputs.

When using EE mode, you must consider the following issues:

- The dmmodule does not enforce the genlock requirement. You can enable EE mode, but the output display may be unstable.
- You can enable EE mode while an output transfer is running. For example, if an SD output transfer is running and SD EE mode is enabled on the output path, EE mode "hijacks" the serial output jack.

Automatically Correcting for Output Underflow

If the application is not sending buffers fast enough for the receiving equipment's video frame rate, you can set ML_VIDEO_OUTPUT_REPEAT_INT32 to repeat MLbuffers automatically. The values for this control vary, depending on whether the transfer is progressive or interlaced.

- ML_VIDEO_REPEAT_NONE

  Repeats nothing, usually resulting in black output. This is the most useful for debugging, because underflow is then quite visible on output.

- ML_VIDEO_REPEAT_FIELD

  Repeats the last field (non-interleaved) or the last frame (interleaved or progressive). This setting is spatially imperfect, but does not cause flicker.

- ML_REPEAT_FRAME (the default)

  Repeats the last two fields (non-interleaved) or the last frame (interleaved or progressive). This setting is spatially better than ML_VIDEO_REPEAT_FIELD, but causes flicker.

Capturing Graphics to Video

To capture graphics to video, you can use OpenGL to read pixels into memory. However, the coordinate system differs between video and OpenGL; under OpenGL, the origin is at the lower left corner and, in video, the origin is in the upper left corner. To adjust for this difference, set the ML_IMAGE_ORIENTATION_INT32 parameter to ML_BOTTOM_TO_TOP. For more information, see Table 4-4 on page 51 in this guide and the *OpenML Media Library Software Development Kit Programmer's Guide*.

Note: If you have an HD GVO board (DG5-2/TVO) with an SGI video breakout box installed in an SGI Onyx compatible system, it is easier to capture graphics to video using the HD GVO system. For more information, see the *SGI HD-GVO Owner's Guide*.

DMediaPro Events

In some cases, an exceptional event occurs, which requires the device to send a message back to the application. For this type of event message, you must initiate a request. When the application requests an event, it must read its receive queue often enough to prevent the device from running out of the required message space for the specific enqueue request. If the queue begins to fill up, the device enqueues an event message, which terminates the exceptional event.

The device does not have to allocate space in the data area for reply messages. It automatically stops sending notifications of events when the receive queue begins to fill up. Space is reserved in the receive queue for a reply to every message that the application enqueues. When there is insufficient space, any attempt to send new messages fails.

The DM3 board currently supports the following ML exceptional events:

```
ML_EVENT_VIDEO_SEQUENCE_LOST
ML_EVENT_VIDEO_SYNC_LOST
ML_EVENT_VIDEO_SYNC_GAINED
```

Table 4-5 summarizes these events.

Table 4-5 ML Exceptional Events

Event	Use
ML_EVENT_VIDEO_SEQUENCE_LOST	A field/frame was dropped.
ML_EVENT_VIDEO_SYNC_LOST	Genlock was lost or the input signal was lost.
ML_EVENT_VIDEO_SYNC_GAINED	Genlock sync lock occurred or a valid signal was found on the input.

Note: Other events, for example, ML_BUFFERS_COMPLETE, are automatically sent to the application. For more information, see the *OpenML Media Library Software Development Kit Programmer's Guide.*

The following text lists the ML controls and event records:

```
event == ML_EVENT_VIDEO_SEQUENCE_LOST

       eventRecord[0].param = ML_VIDEO_UST_INT64;
       eventRecord[0].value.int64 = ust;
       eventRecord[0].length = 1;
       eventRecord[1].param = ML_VIDEO_MSC_INT64;
       eventRecord[1].value.int64 = msc;
       eventRecord[1].length = 1;
       eventRecord[2].param = ML_END;
```

`event == ML_EVENT_VIDEO_SYNC_GAINED, ML_EVENT_VIDEO_SYNC_LOST` (output path)

```
       eventRecord[0].param = ML_VIDEO_GENLOCK_SIGNAL_PRESENT_INT32;
       eventRecord[0].value.int32 = <detectedTiming>
```
(see Table 4-2)
```
       eventRecord[0].length = 1;
       eventRecord[1].param = XTDIGVID_GENLOCK_ERROR_STATUS_INT32;
       eventRecord[1].value.int32 = <syncLostReason>
```
(see Table 4-6)
```
       eventRecord[1].length = 1;
       eventRecord[2].param = ML_END;
```

`event == ML_EVENT_VIDEO_SYNC_GAINED, ML_EVENT_VIDEO_SYNC_LOST` (input path)

```
       eventRecord[0].param = ML_VIDEO_SIGNAL_PRESENT_INT32;
       eventRecord[0].value.int32 = <detectedTiming>
```
(see Table 4-2)
```
       eventRecord[0].length = 1;
       eventRecord[1].param = ML_END;
```

Table 4-6 describes the XTDIGVID_GENLOCK_ERROR_STATUS_INT32 values for ML_EVENT_VIDEO_SYNC_LOST on the output path. It also lists the corresponding values for ML_VIDEO_GENLOCK_SIGNAL_PRESENT_INT32.

Table 4-6 Error Status Values for ML_EVENT_VIDEO_SYNC_LOST (Output Path)

Error Status Values	ML_VIDEO_GENLOCK_SIGNAL_PRESENT_INT32
XTDIGVID_GENLOCK_ERROR_STATUS_NO_SIGNAL (No signal detected on specified genlock jack)	ML_TIMING_NONE
XTDIGVID_GENLOCK_ERROR_STATUS_UNKNOWN_SIGNAL (Unknown signal detected or unable to identify the signal on a specified genlock jack)	ML_TIMING_UNKNOWN
XTDIGVID_GENLOCK_ERROR_STATUS_ILLEGAL_COMBINATION (Detected signal on the genlock jack is an illegal or unsupported combination for the specified output timing)	ID of the timing detected on the genlock jack
XTDIGVID_GENLOCK_ERROR_STATUS_TIMING_MISMATCH (Detected signal does not match the user's request using the ML_VIDEO_GENLOCK_SOURCE_TIMING_INT32 parameter)	ID of the timing detected on the genlock jack
XTDIGVID_GENLOCK_ERROR_STATUS_NONE (No error status)	ID of the timing detected on the genlock jack

Examples

This section provides the following examples:

Capturing 487 Line 525

The following ML control settings capture 487 line 525:

```
ML_VIDEO_TIMING_INT32 = ML_TIMING_525
ML_VIDEO_COLORSPACE_INT32 = ML_COLORSPACE_CbYCr_601_HEAD
ML_VIDEO_PRECISION_INT32 = 8
ML_VIDEO_START_Y_F1_INT32 = 20
ML_VIDEO_START_Y_F2_INT32 = 283
ML_VIDEO_HEIGHT_F1_INT32 = 244
ML_VIDEO_HEIGHT_F2_INT32 = 243
ML_IMAGE_TEMPORAL_SAMPLING_INT32 = ML_TEMPORAL_SAMPLING_FIELD_BASED
ML_VIDEO_SAMPLING_INT32 = ML_SAMPLING_422
ML_VIDEO_WIDTH_INT32 = 720
ML_IMAGE_WIDTH_INT32 = 720
ML_IMAGE_HEIGHT_1_INT32 = 487
ML_IMAGE_HEIGHT_2_INT32 = 0
ML_IMAGE_SAMPLING_INT32 = ML_SAMPLING_444
ML_IMAGE_COLORSPACE_INT32 = ML_COLORSPACE_RGB_601_FULL
ML_IMAGE_PACKING_INT32 = ML_PACKING_8
ML_IMAGE_INTERLEAVE_MODE_INT32 = ML_INTERLEAVE_MODE_INTERLEAVED
ML_IMAGE_DOMINANCE_INT32 = ML_DOMINANCE_F1
ML_IMAGE_ORIENTATION_INT32 = ML_ORIENTATION_TOP_TO_BOTTOM
XTDIGVID_LOOPBACK_INT32 = XTDIGVID_LOOPBACK_DISABLE
ML_IMAGE_COMPRESSION_INT32 = ML_COMPRESSION_UNCOMPRESSED
ML_IMAGE_ROW_BYTES_INT32 = 0
ML_IMAGE_SKIP_PIXELS_INT32 = 0
ML_IMAGE_SKIP_ROWS_INT32 = 0
ML_VIDEO_START_X_INT32 = 1
```

Playback from Memory to Video

The following ML control settings perform a memory-to-video transfer in HD 720p format:

```
ML_VIDEO_TIMING_INT32 = ML_TIMING_750_1280x720_5994p
ML_VIDEO_PRECISION_INT32 = 8
ML_VIDEO_COLORSPACE_INT32 = ML_COLORSPACE_CbYCr_709_HEAD
ML_IMAGE_TEMPORAL_SAMPLING_INT32 = ML_TEMPORAL_SAMPLING_PROGRESSIVE
ML_VIDEO_SAMPLING_INT32 = ML_SAMPLING_422
ML_VIDEO_START_Y_F1_INT32 = 26
ML_VIDEO_START_Y_F2_INT32 = 0
ML_VIDEO_HEIGHT_F1_INT32 = 720
ML_VIDEO_HEIGHT_F2_INT32 = 0
ML_VIDEO_WIDTH_INT32 = 1280
ML_IMAGE_WIDTH_INT32 = 1280
ML_IMAGE_HEIGHT_1_INT32 = 720
ML_IMAGE_HEIGHT_2_INT32 = 0
ML_IMAGE_PACKING_INT32 = ML_PACKING_8
ML_IMAGE_SAMPLING_INT32 = ML_SAMPLING_444
ML_IMAGE_COLORSPACE_INT32 = ML_COLORSPACE_RGB_709_FULL
ML_IMAGE_INTERLEAVE_MODE_INT32 = ML_INTERLEAVE_MODE_INTERLEAVED
ML_IMAGE_DOMINANCE_INT32 = ML_DOMINANCE_F1
ML_IMAGE_ORIENTATION_INT32 = ML_ORIENTATION_TOP_TO_BOTTOM
ML_VIDEO_GENLOCK_TYPE_INT32 = XTDIGVID_GENLOCK_SRC_TYPE_INTERNAL
ML_VIDEO_GENLOCK_SOURCE_TIMING_INT32 = ML_TIMING_525
XTDIGVID_FF_MODE_INT32 = XTDIGVID_FF_MODE_DISABLE
ML_VIDEO_OUTPUT_REPEAT_INT32 = ML_VIDEO_REPEAT_NONE
ML_IMAGE_COMPRESSION_INT32 = ML_COMPRESSION_UNCOMPRESSED
ML_IMAGE_ROW_BYTES_INT32 = 0
ML_IMAGE_SKIP_PIXELS_INT32 = 0
ML_IMAGE_SKIP_ROWS_INT32 = 0
ML_VIDEO_START_X_INT32 = 1
```

Re-sizing Field Height

The DM3 board uses the standard NTSC field height of 487 lines for standard-definition video. However, some SGI products (for example, DIVO and DIVO DVC), use a 486 line NTSC field height. The following example provides ML code that you can use to retrieve the default sizing parameters for any given timing. Then you can reset these parameters from 487 lines to 486 lines, which provides backward compatibility with SGI standard-definition products, such as DIVO and DIVO DVC.

Note: This example is the accepted method for retrieving these values, so you do not have to perform any calculations.

```
// timing = ML_TIMING_525

short progressive;

DMpv videoSizeDefaults[] = {
   ML_IMAGE_TEMPORAL_SAMPLING_INT32, 0, 0, 0,
   ML_VIDEO_START_Y_F1_INT32, 0, 0, 0,
   ML_VIDEO_START_Y_F2_INT32, 0, 0, 0,
   ML_VIDEO_HEIGHT_F1_INT32, 0, 0, 0,
   ML_VIDEO_HEIGHT_F2_INT32, 0, 0, 0,
   ML_VIDEO_WIDTH_INT32, 0, 0, 0,
   ML_END, 0, 0, 0
};

if (starty1 != NULL) {
   *tempSampling = videoSizeDefaults[0].value.int32;
   progressive = (*tempSampling == ML_TEMPORAL_SAMPLING_PROGRESSIVE);
   *starty1 = videoSizeDefaults[1].value.int32;
   /* set starty2 to 0 if progressive, func returns -1 */
   *starty2 = (progressive ? 0 :videoSizeDefaults[2].value.int32);
   *height1 = videoSizeDefaults[3].value.int32;
   /* set F2 height to 0 if progressive, func returns -1 */
   *height2 = (progressive ? 0 : videoSizeDefaults[4].value.int32);
   *width = videoSizeDefaults[5].value.int32;

switch (timing) {
   case ML_TIMING_525:
      switch (rasterSize) {
      case NTSC_486: //
            *starty1 = 21;
```

```
                *starty2 = 283;
                *height1 = 243;
                *height2 = 243;
                break;
        }
        break;
}
```

The following text lists the ML control settings:

```
ML_VIDEO_TIMING_INT32 = ML_TIMING_525
ML_VIDEO_COLORSPACE_INT32 = ML_COLORSPACE_CbYCr_601_HEAD
ML_VIDEO_PRECISION_INT32 = 8
ML_VIDEO_START_Y_F1_INT32 = 21
ML_VIDEO_START_Y_F2_INT32 = 283
ML_VIDEO_HEIGHT_F1_INT32 = 243
ML_VIDEO_HEIGHT_F2_INT32 = 243
ML_IMAGE_TEMPORAL_SAMPLING_INT32 = ML_TEMPORAL_SAMPLING_FIELD_BASED
ML_VIDEO_SAMPLING_INT32 = ML_SAMPLING_422
ML_VIDEO_WIDTH_INT32 = 720
ML_IMAGE_WIDTH_INT32 = 720
ML_IMAGE_HEIGHT_1_INT32 = 486
ML_IMAGE_HEIGHT_2_INT32 = 0
ML_IMAGE_SAMPLING_INT32 = ML_SAMPLING_444
ML_IMAGE_COLORSPACE_INT32 = ML_COLORSPACE_RGB_601_FULL
ML_IMAGE_PACKING_INT32 = ML_PACKING_8
ML_IMAGE_INTERLEAVE_MODE_INT32 = ML_INTERLEAVE_MODE_INTERLEAVED
ML_IMAGE_DOMINANCE_INT32 = ML_DOMINANCE_F1
ML_IMAGE_ORIENTATION_INT32 = ML_ORIENTATION_TOP_TO_BOTTOM
XTDIGVID_LOOPBACK_INT32 = XTDIGVID_LOOPBACK_DISABLE
ML_IMAGE_COMPRESSION_INT32 = ML_COMPRESSION_UNCOMPRESSED
ML_IMAGE_ROW_BYTES_INT32 = 0
ML_IMAGE_SKIP_PIXELS_INT32 = 0
ML_IMAGE_SKIP_ROWS_INT32 = 0
ML_VIDEO_START_X_INT32 = 1
```

Programmable Lookup Tables (LUTs)

This device-specific example shows you how to use the LUTs to invert video. You can perform this example with 8-bit packings only (load with an inverse ramp). Following are the DMediaPro control settings:

```
for (i=1; i<=NUM_DEFINED_LUT_ENTRIES;i++)
lutentries[NUM_DEFINED_LUT_ENTRIES-i] = i;

pv->param = XTDIGVID_LUT_YG_INT32_ARRAY;
pv->value.pInt32 = lutentries;
pv->length=NUM_DEFINED_LUT_ENTRIES;
pv->maxLength=sizeof(lutentries)/sizeof(DMint32);
pv++;

pv->param = XTDIGVID_LUT_UB_INT32_ARRAY;
pv->value.pInt32 = lutentries;
pv->length=NUM_DEFINED_LUT_ENTRIES;
pv->maxLength=sizeof(lutentries)/sizeof(DMint32);
pv++;

pv->param = XTDIGVID_LUT_VR_INT32_ARRAY;
pv->value.pInt32 = lutentries;
pv->length=NUM_DEFINED_LUT_ENTRIES;
pv->maxLength=sizeof(lutentries)/sizeof(DMint32);
pv++;
```

FF Mode

For output transfers, you can use field/frame mode (FF mode) (XTDIGVID_FF_MODE_INT32) to assist an application in performing 3/2 pulldown. You can only use this mode when there are 1080p 23.97 frames in memory and you want to output 1080i 59.94 fields. By default FF mode is disabled. To enable FF mode, follow these steps:

1. Set XTDIGVID_FF_MODE_INT32 to the value XTDIGVID_FF_MODE_ENABLE.
2. Set the ML_IMAGE_INTERLEAVE_MODE_INT32 to ML_INTERLEAVE_MODE_INTERLEAVED.

In field/frame mode, you can send the DM3 board an entire frame, but the board only extracts a single field. For example, if the 1080p frames in memory are labeled A,B,C..., and FF mode is enabled, you can send the board AAABBCCCDD, and it will output field 1 from A, field 2 from A, field 1 from A, field 2 from B, and field 1 from B. Each buffer you send is treated as an interleaved frame, but only a single field is extracted from it. As a result, your application does not have to manually extract fields from the frames in memory.

UST (unadjusted system time) and MSC (media stream count) function exactly as they do in the one-field-per buffer case; MSC increases by one for each buffer. To specify whether the first field is an F1 or an F2, use the ML_IMAGE_DOMINANCE_INT32 control.

The following device-specific example shows you how to allocate buffers for a fixed set of images and how to place the images in the buffer to achieve the desired results.

```
// buffer allocation
if (ffmode)
   bufferCount = imageCount*5/2;
else
   bufferCount = (imageCount > 1 ? imageCount : maxBuffers);

bufArray = (void *) malloc(bufferCount * sizeof(void *));
if (bufArray == NULL) {
   fprintf(stderr, "Cannot allocate buffer array\n");
   exit(-1);
}
else
   bzero(bufArray, (bufferCount * sizeof(void *)));
```

```
// filling the buffers in ffmode
if (ffmode) {
       /*
```

In field/frame mode, you must send each frame 2.5 times (on average), so you must duplicate entries in the buffer array. Begin with buffer array entries as follows:

```
ABCD......
```

And finish with:

```
AAABBCCCDD

   */

   for(fnum = bufferCount-1; fnum>0; fnum--)
      bufArray[fnum] = bufArray[(int)(fnum*2/5)];
```

Setting Controls on a Jack

This section provides example code for setting up a jack control. The following example uses a jack control to enable a loopback on the HD input jack:

```
// enable loopback on HD input jack

#include <stdio.h>
#include <ML/ml.h>
#include <ML/mlu.h>
#include <ML/ml_xtdigvid.h>

int main( int argc, char **argv )
{
   DMstatus status;
   DMopenid jack;

   // open the HD input jack on the xt-digvid device
   {
      DMint64 sysId = ML_SYSTEM_LOCALHOST;
      DMint64 devId;
      DMint64 jackId;
```

```
            char *jackName = "HDSerialDigitalInputJack";

        if( status = dmuFindDeviceByName( sysId, "xt-digvid", &devId )) {
        fprintf( stderr, "xt-digvid: %s\n", dmStatusName( status ));
        return( 1 );
        }

        if( status = dmuFindJackByName( devId, jackName, &jackId )) {
        fprintf( stderr, "%s: %s\n", jackName, dmStatusName( status ));
        return( 1 );
        }

        if( status = dmOpen( jackId, NULL, &jack ) ) {
        fprintf( stderr, "open %s: %s\n", jackName, dmStatusName( status
        ));
        return( 1 );
        }
    }

    // set the loopback control
    {
        DMpv ctrls[] = {
        XTDIGVID_LOOPBACK_INT32, 0, 0, 0,
        ML_END, 0, 0, 0
        };

        ctrls[ 0 ].value.int32 = XTDIGVID_LOOPBACK_ENABLE;

        if( status = dmSetControls( jack, ctrls )) {
        fprintf( stderr, "dmSetControls: %s\n",dmStatusName(status));
        return( 1 );
        }
    }

    dmClose( jack );
    return 0;
}
```

Synchronizing Data Streams and Signals

You can use UST (unadjusted system time) and MSC (media stream count) signals to synchronize data streams. These are special signals that are recognized or generated by the DM3 board. For more information, see the *OpenML Media Library Software Development Kit Programmer's Guide*.

Restrictions and Important Notes

Consider the following information when programming the DM3 board:

- You can only open one input path and one output path at the same time.
- Before you configure a path with SetControls, your image controls and your video controls must be compatible. Because the DM3 board validates the path configuration at SetControl time, set all video controls and image controls at the same time. If this is inconvenient, start from a valid configuration and change "blocks" of controls. This alternative method also results in a valid path configuration.
- The image width and height must correspond to the video width and height as follows:
  - Progressive formats

    ML_IMAGE_WIDTH = ML_VIDEO_WIDTH

    ML_IMAGE_HEIGHT_1 = ML_VIDEO_HEIGHT_F1

    ML_IMAGE_HEIGHT_2 = 0
  - Interlaced formats with ML_INTERLEAVE_MODE_INT32 set to ML_INTERLEAVE_MODE_INTERLEAVED

    ML_IMAGE_WIDTH = ML_VIDEO_WIDTH

    ML_IMAGE_HEIGHT_1 = ML_VIDEO_HEIGHT_F1 + ML_VIDEO_HEIGHT_F2

    ML_IMAGE_HEIGHT_2 = 0

  - Interlaced formats with ML_INTERLEAVE_MODE_INT32 set to ML_INTERLEAVE_MODE_SINGLE_FIELD

    ML_IMAGE_WIDTH = ML_VIDEO_WIDTH

    ML_IMAGE_HEIGHT_1 = ML_VIDEO_HEIGHT_F1

    ML_IMAGE_HEIGHT_2 = ML_VIDEO_HEIGHT_F2

- The VBOB does not distinguish between 25PsF and 50i timings. There are three possible results:
  - If the input timing is 25PsF, the detected input signal is 50i.
  - If the output timing is 25PsF and the genlock source is 25PsF, the detected genlock signal is 50i.
  - If the output timing is 50i and the genlock source is 25PsF, the detected genlock signal is 50i.

Index

F

G

H

I

Q

R

S

T

U

V

W

X

Y