C-CUBE MICROSYSTEMS

100

C-CUBE MPEG VIDEO LAB USER'S GUIDE





C-CUBE MICROSYSTEMS

C-CUBE MPEG VIDEO LAB USER'S GUIDE





This document is preliminary. The specifications contained herein are derived from functional specifications and performance estimates, and have not been verified against production parts.

C-Cube Microsystems reserves the right to change any products described herein at any time and without notice. C-Cube Microsystems assumes no responsibility or liability arising from the use of the products described herein, except as expressly agreed to in writing by C-Cube Microsystems. The use and purchase of this product does not convey a licence under any patent rights, copyrights, trademark rights, or any other intellectual property rights of C-Cube Microsystems.

Trademark Acknowledgment:

The C-Cube corporate logo is a registered trademark of and C-Cube MPEG Video Lab is a trademark of C-Cube Microsystems.

Vectra is a trademark of Hewlett-Packard Company.

Maynard and MaynStream are registered trademarks of Maynard Electronics, Inc.

Microsoft, MS-DOS, and Windows are registered trademarks of Microsoft Corporation.

Bogart image © 1992 Bogart, Inc. Used under license authorized by Curtis Management Group, Indianapolis, Indiana, USA

© C-Cube Microsystems 1992 All rights reserved

> C-Cube Microsystems 1778 McCarthy Boulevard Milpitas, CA 95035 Telephone (408) 944-6300 Fax (408) 944-6314

Preface

This guide is the primary source for installation and operation information for the C-Cube MPEG Video Lab.

This guide describes Revision B of the MPEG Video Decoder Board and Version 1.2 of the Playback software.

This guide is intended for:

- System designers and managers who are evaluating the MPEG
 Video Lab for use as a development vehicle for MPEG applications
- Programmers and software engineers who are writing application programs that interact with the MPEG Video Decoder Board used in the MPEG Video Lab

This guide is divided into these chapters:

- □ Chapter 1, Introduction, describes the hardware and software that comprise the MPEG Video Lab.
- □ Chapter 2, MPEG Overview, presents an overview of MPEG decoding.

Applicability

Audience

Organization

- Chapter 3, Installation, tells how to install and power up the MPEG Video Lab.
- Chapter 4, Using the Playback Software, describes the use of the playback application supplied with the MPEG Video Lab. It includes a tutorial that can be worked in about 15 minutes.
- Chapter 5, The MPEG Video Decoder Board, gives a functional description of the board. It also contains programming information about the board-level registers.
- Appendix A, Encoding Guidelines, gives information for creating coded bitstreams.
- □ Appendix B, MPEG Video Lab Files, describes the files supplied with the MPEG Video Lab.

Related Publications

The MPEG standard is described in detail in *Coded Representation of Picture, Audio and Multimedia/Hypermedia Information,* ISO/IEC JTC 1/SC 29, December 6, 1991.

Conventions

Please note the following conventions that are used in this manual:

- Hexadecimal numbers are indicated by the prefix 0x, for example, 0xFF. Binary numbers are indicated by a subscript, for example, 10₂. Otherwise, all numbers used in this guide are decimal numbers.
- □ The names of fields within a register or data word are set in italics, for example, *IPID*.

Contents

1	Intr	oductio	on	
	1.1	Video L	ab Components	1-1
	1.2	Sample	Bitstreams	1-3
	1.3	Video L	ab Documentation Set	1-3
2	MP	EG Ove	erview	
	2.1	MPEG	Stream Structure	2-2
		2.1.1	MPEG Stream Structure	2-2
		2.1.2	General Decoding Process	2-2
		2.1.3	Video Stream Data Hierarchy	2-3
	2.2	Inter-pi	icture Coding	2-6
		2.2.1	Picture Types	2-6
		2.2.2	Video Stream Composition	2-7
		2.2.3	Motion Compensation	2-8
	2.3	Intra-pi	icture (Transform) Coding	2-10
	2.4	Synchr	onization	2-11
		2.4.1	System Clock References	2-11
		2.4.2	Presentation Time Stamps	2-12
3	Inst	allatio	n	
	3.1	Conten	ts	3-1
	3.2	Installa	ition Procedure	3-2

4	Usi	ng the	e Playback Software	
	4.1	Tutor	ial	4-1
		4.1.1	Starting the Playback Software	4-2
		4.1.2	Playing Back a Video Sequence	4-4
		4.1.3	Pausing, Resuming, and Single-frame Advancing	4-8
		4.1.4	Fast-Forward and Slow-Motion	4-9
		4.1.5	Ending a Playback Session	4-10
	4.2	Refer	rence Guide	4-11
		4.2.1	The Development Window	4-11
		4.2.2	The Player Window	4-12
		4.2.3	The Remote Window	4-14
		4.2.4	The Bitstreams Selection Window	4-16
	4.3	Using	y Scripts	4-17
		4.3.1	Creating a Script	4-17
		4.3.2	Running a Script	4-18
5	The	MPE	G Video Decoder Board	
	5.1	Overv	view	5-1
	5.2	Funct	tional Description	5-2
		5.2.1	MPEG Processor	5-3
		5.2.2	EISA Interface	5-3
		5.2.3	Coded Data FIFO	5-3
		5.2.4	DRAM	5-3
		5.2.5	Color Space Converter	5-4
		5.2.6	Video DAC	5-4
		5.2.7	Raster Timing Generator	5-4
		5.2.8	Line Store	5-4
	5.3	Opera	ation Overview	5-4
	5.4	Progr	rammer's Model	5-6
		5.4.1	EISA Product ID Register	5-7
		5.4.2	Configuration/Status Register	5-7
		5.4.3	Base Address Register	5-9
		5.4.4	Decoder RREG and GREG Registers	5-9
		5.4.5	DRAM Address Register	5-10
		5.4.6	DRAM Data Register	5-10
		5.4.7	Coded Data FIFO Reset Register	5-11
		5.4.8	Coded Data FIFO	5-11
	5.5	Other	System Applications	5-12
		5.5.1	System Design Considerations	5-12
		5.5.2	Installation Guidelines	5-12
	5.6	Spec	ifications	5-14
		5.6.1	General Specifications	5-14

	5.6.2 Interface Specifications	5-14
	5.6.3 Video Display Formats	5-15
Appei	ndix A Encoding Guidelines	
Ā.1	MPEG Draft Standard Compliance	A-1
A.2	Bitstream Naming Conventions	A-2
Apper	ndix B MPEG Video Lab Software Files	
B.1	Software Files	B-1
B.2	704 Files From Previous Versions	B-3

viii C-Cube Microsystems

Figures

1-1	Video Lab Components	1-2
2-1	General MPEG Decoding System	2-3
2-2	MPEG Data Hierarchy	2-3
2-3	Location of Luminance and Chrominance Values	2-4
2-4	Macroblock Composition	2-5
2-5	Forward Prediction	2-6
2-6	Bidirectional Prediction	2-7
2-7	Typical Display Order of Picture Types	2-8
2-8	Video Stream versus Display Ordering	2-8
2-9	Transform Coding Operations	2-11
2-10	SCR Flow in MPEG System	2-12
3-1	Cabling Diagram	3-2
4-1	Playback Software Initial Screen	4-2
4-2	Player and Remote Windows	4-3
4-3	Bitstreams Selection Window	4-4
4-4	Play Button Active	4-5
4-5	Remote Window During Play Sequence	4-6
4-6	Remote Window at End of Video Sequence	4-7
4-7	Remote Window During Pause	4-9
4-8	File Menu	4-10

4-9	Exit Windows Dialog Box	4-11
4-10	Development Window Operations	4-12
4-11	The Player Window	4-12
4-12	The Remote Window	4-14
4-13	The Bitstreams Selection Window	4-16
5-1	MPEG Video Decoder Board Block Diagram	5-2
5-2	MPEG Video Decoder Board Address Map	5-6
5-3	Analog Video Output Connector	5-15
B-1	Directory Structure Showing MPEG Video Lab Files	B-2

Tables

3-1	MPEG Video Lab Shipping Boxes	3-1	
5-1	VF Field Values	5-8	
5-2	CF Field Values	5-8	
5-3	Video Display Formats	5-15	
B-1	Bitstream Files Supplied with MPEG Video Lab	B-3	

xii C-Cube Microsystems

Chapter 1 Introduction

The MPEG Video Lab provides the tools needed to decode and preview MPEG bitstreams at bit rates up to 8 Mbits per second. It supports these video formats:

□ NTSC: from 352 x 240 to 704 x 480 at 30 Hz

□ PAL: from 352 x 288 to 704 x 576 at 25 Hz

The MPEG Video Lab also supports RGB video output format.

This chapter describes the components of Video Lab and discusses the documentation and sample bitstreams provided with the MPEG Video Lab.

The basic MPEG Video Lab system shown in Figure 1-1 is based on the HP486/33T Vectra[™] personal computer, which uses the EISA bus. The standard configuration includes the following components (supplier is Hewlett-Packard except where noted otherwise in parentheses):

1.1 Video Lab Components

Introduction 1-1





Video Lab Components

- Eight Mbytes RAM
- Super VGA monitor
- □ Extended keyboard with 101 keys
- Two-button mouse
- □ Floppy disk drive: 3.5-inch, 1.44 Mbyte capacity
- □ Hard disk: 670 Mbytes unformatted (Micropolis 1624)
- DAT tape drive: 2 Gbytes (Maynard® MaynStream®)
- Super VGA graphics card
- □ Disk controller card (Adaptec AHA 1740)
- MPEG Video Decoder Board (C-Cube® Microsystems)
- □ Ethernet card (3Com W/RG58)
- □ MS-DOS® 5.0 and Windows® 3.0 (Microsoft®), with Windows running in 386 extended mode
- □ PC-NFS networking software (Sun Microsystems)

- □ Board driver software (C-Cube Microsystems)
- Sample bitstreams
- Complete documentation

Options are an RGB video monitor and the C-Cube MPEG encoder software.

C-Cube Microsystems supplies a variety of sample bitstreams with Video Lab to demonstrate the basic operation of the system for a range of resolutions. Appendix B summarizes the bitstreams provided. You can use these bitstreams for demonstrations and product development.

Your system may also include some bitstreams not listed in Appendix B. These extra bitstreams may have restrictions on their use. Contact C-Cube Microsystems for more information.

This guide is the primary source of the information needed to operate Video Lab. However, you may need to refer to manuals that provide more detailed information about some of the MPEG Video Lab components. C-Cube Microsystems supplies with the MPEG Video Lab a complete set of documentation for all the major components in the system. The manuals supplied with the MPEG Video Lab are listed below.

- D Setting Up Your HP Vectra 486/33T (Hewlett-Packard)
- D1182 Video Graphics Color Display Installation Guide (Hewlett-Packard)
- □ Super VGA Board User's Manual (Hewlett-Packard)
- Dealer Configuration File Creation Guide (Hewlett-Packard)
- □ MaynStream DOS User's Manual (Maynard)
- □ ASPI Manager Installation Guide (Adaptec)
- □ ASPI MS-DOS Manager Software Manual (Adaptec)
- □ AHA-174x Family Configuration & Download Utilities Software Manual (Adaptec)
- Dia MS-DOS 5.0 User's Guide & Reference (Hewlett-Packard)
- □ MS-DOS 5.0 Getting Started (Hewlett-Packard)
- D Windows User's Guide (Microsoft)

1.2 Sample Bitstreams

1.3 Video Lab Documentation Set

Windows Questions & Answers (Microsoft)
 QuickC for Windows (Microsoft)

Chapter 2 MPEG Overview

This chapter presents an overview of the Moving Picture Experts Group (MPEG) standard that is implemented by the MPEG Video Lab.The standard is officially known as ISO/IEC Draft Standard "Coded representation of picture, audio and multimedia/hypermedia information," CD 11172, December 6, 1991. It is more commonly referred to as the MPEG standard.

MPEG addresses the compression and decompression of video and audio signals and the synchronization of audio and video signals during playback of decompressed MPEG data. The MPEG video algorithm can compress video signals to about 1/2 to 1 bit per coded pixel. At a compressed data rate of 1.2 Mbits per second, a coded resolution of 352 x 240 at 30 Hz is often used, and the resulting video quality is comparable to VHS.

2.1 MPEG Stream Structure

This section explains the general structure of an MPEG stream and introduces some basic concepts used in the rest of the chapter.

2.1.1 MPEG Stream Structure

In its most general form, an MPEG stream is made up of two layers:

- □ The system layer contains timing and other information needed to demultiplex the audio and video streams and to synchronize audio and video during playback.
- □ The *compression layer* includes the compressed audio and video streams.

2.1.2 General Decoding Process

Figure 2-1 shows a generalized decoding system.

The *system decoder* extracts the timing information from the MPEG stream and sends it to the other system components. (Section 2.4, Synchronization, has more information about the use of timing information for audio and video synchronization.) The system decoder also demultiplexes the video and audio streams and sends each to the appropriate decoder. In many applications, the system decoder function is implemented as a software program on the host computer.

The *video decoder* decompresses the video stream as specified in Part 2 of the MPEG standard. (See Section 2.2, Inter-picture Coding, and Section 2.3, Intra-picture Coding, for more information about video compression.)

The *audio decoder* decompresses the audio stream as specified in Part 3 of the MPEG standard.



Figure 2-1 General MPEG Decoding System

2.1.3 Video Stream Data Hierarchy

The MPEG standard defines a hierarchy of data structures in the video stream as shown schematically in Figure 2-2.





Video Sequence

Consists of a sequence header, one or more groups of pictures, and an end-of-sequence code. The video sequence is another term for a video stream as defined above.

Group of Pictures

A series of one or more pictures intended to allow random access into the sequence.

Picture

The primary coding unit of a video sequence. A picture consist of three rectangular matrices representing luminance (Y) and two chrominance (CbCr) values. The Y matrix has an even number of rows and columns. The Cb and Cr matrices are one-half the size of the Y matrix in each direction (horizontal and vertical).

Figure 2-3 shows the relative x-y locations of the luminance and chrominance components. Note that for every four luminance values, there are two associated chrominance values: one Cb value and one Cr value. (The location of the Cb and Cr values is the same, so only one circle is shown in the figure.)





Slice

One or more contiguous macroblocks. The order of the macroblocks within a slice is from left to right and top to bottom.

Slices are important in the handling of errors. If the bitstream contains an error, the decoder can skip to the start of the next slice. Having more slices in the bitstream allows better error concealment, but uses bits that could otherwise be used to improve picture quality.

Macroblock

A 16-pixel by 16-line section of luminance components and the corresponding 8-pixel x 8-line section of the chrominance components. See Figure 2-3 for the spatial location of luminance and chrominance components. A macroblock contains four Y blocks, one Cb block and one Cr block as shown in Figure 2-4. The numbers correspond to the ordering of the blocks in the data stream, with block 1 first.



Figure 2-4 Macroblock Composition

Block

A block is an 8 by 8 set of values of a luminance or chrominance component. Note that a luminance block corresponds to one-fourth as large a portion of the displayed image as does a chrominance block.

2.2 Inter-picture Coding

Much of the information in a picture within a video sequence is similar to information in a previous or subsequent picture. The MPEG standard takes advantage of this temporal redundancy to represent some pictures in terms of their differences from reference picture. This section describes the picture types and explains the techniques used in inter-picture coding.

2.2.1 Picture Types

The MPEG standard specifically defines three types of pictures: intra, predicted, and bidirectional.

Intra Pictures

Intra or I-pictures are coded using only information present in the picture itself. I-pictures provide random access points into the compressed video data. I-pictures use only transform coding and therefore provide moderate compression. I-pictures typically use about two bits per coded pixel.

Predicted Pictures

Predicted or P-pictures are coded with respect to the nearest previous Ior P-picture. This technique is called *forward prediction* and is illustrated in Figure 2-5. Predicted pictures provide more compression and serve as a reference for B-pictures and future P-pictures. P-pictures use motion compensation to provide more compression than is possible with I-pictures. P-pictures can propagate coding errors, since P-pictures can be predicted from previous P-pictures.





Bidirectional Pictures

Bidirectional or B-pictures are pictures that use both a past and future picture as a reference. This technique is called *bidirectional prediction* and is illustrated in Figure 2-6. Bidirectional pictures provide the most compression and do not propagate errors because they are never used as a reference. Bidirectional prediction also decreases the effect of noise by averaging two pictures.





Figure 2-6 Bidirectional Prediction

2.2.2 Video Stream Composition

The MPEG algorithm allows the encoder to choose the frequency and location of I-pictures. This choice is based on the application's need for random accessibility and the location of scene cuts in the video sequence. In applications where random access is important, intra pictures are typically used two times a second.

The encoder also chooses the number of bidirectional pictures between any pair of reference (I or P) pictures. This choice is based on factors such as the amount of memory in the encoder and the characteristics of the material being coded. For a large class of scenes, a workable arrangement is to have two bidirectional pictures separating successive reference pictures. A typical arrangement of I-, P-, and B-pictures is shown in Figure 2-7 in the order in which they are displayed.



The MPEG encoder reorders pictures in the video stream to present the pictures to the decoder in the most efficient sequence. In particular, the reference pictures needed to reconstruct B-pictures are sent before the associated B-pictures. Figure 2-8 demonstrates this ordering for the first section of the example shown above.



Figure 2-8 Video Stream versus Display Ordering

2.2.3 Motion Compensation

Motion compensation is a technique for enhancing the compression of P- and B-pictures by eliminating temporal redundancy. Motion compensation typically improves compression by about a factor of three compared to intra-picture coding. Motion compensation algorithms work at the macroblock level.

When a macroblock is compressed by motion compensation, the compressed file contains this information:

- □ The spatial difference between the reference and the macroblock being coded (*motion vectors*)
- □ The content differences between the reference and the macroblock being coded (*error terms*)

Not all information in a picture can be predicted from a previous picture. Consider a scene in which a door opens. The visual details of the room behind the door cannot be predicted from a previous frame in which the door was closed. When a macroblock in a P-picture cannot be represented by motion compensation, it is coded in the same way as a macroblock in an I-picture, that is, by transform coding techniques (see Section 2.3, Intra-picture Coding).

Macroblocks in a B-picture can be coded using either a previous or future reference picture as a reference, so that four codings are possible:

- □ Intra coding: no motion compensation
- Forward prediction: the closest previous I- or P-picture is used as a reference
- Backward prediction: the closest future I- or P-picture is used as a reference
- Bidirectional prediction: two pictures are used as reference, the closest previous I- or P-picture and the closest future I- or P-picture

Backward prediction can be used to predict uncovered areas that do not appear in previous pictures.

2.3 Intra-picture (Transform) Coding The MPEG transform coding algorithm includes these steps:

Discrete cosine transform (DCT)

Quantization

Run-length encoding

Both image blocks and prediction-error blocks have high spatial redundancy. To reduce this redundancy, the MPEG algorithm transforms 8×8 blocks of pixels or 8×8 blocks of error terms to the frequency domain with the Discrete Cosine Transform (DCT).

Next, the algorithm quantizes the frequency coefficients. Quantization is the process of approximating each frequency coefficient as one of a limited number of allowed values. The encoder chooses a quantization matrix that determines how each frequency coefficient in the 8 x 8 block is quantized. Human perception of quantization error is lower for high spatial frequencies, so high frequencies are typically quantized more coarsely (i.e., with fewer allowed values) than low frequencies.

The combination of DCT and quantization results in many of the frequency coefficients being zero, especially the coefficients for high spatial frequencies. To take maximum advantage of this, the coefficients are organized in a zigzag order to produce long runs of zeros (see Figure 2-9). The coefficients are then converted to a series of run-amplitude pairs, each pair indicating a number of zero coefficients and the amplitude of a non-zero coefficient. These run-amplitude pairs are then coded with a variable-length code, which uses shorter codes for commonly occurring pairs and longer codes for less common pairs.

Some blocks of pixels need to be coded more accurately than others. For example, blocks with smooth intensity gradients need accurate coding to avoid visible block boundaries. To deal with this inequality between blocks, the MPEG algorithm allows the amount of quantization to be modified for each 16 x 16 block of pixels. This mechanism can also be used to provide smooth adaptation to a particular bit rate.



Figure 2-9 Transform Coding Operations

The MPEG standard provides a timing mechanism that ensures synchronization of audio and video. The standard includes two timing parameters used in the MPEG Video Lab: the system clock reference (SCR) and the presentation time stamp (PTS).

2.4 Synchronization

The MPEG system clock running at 90 kHz generates 7.8×10^9 clocks in a 24-hour day. System clock references and presentation time stamps are 33-bit values, which can represent any clock cycle in a 24-hour period.

2.4.1 System Clock References

A system clock reference is a snapshot of the encoder system clock. The SCRs used by the audio and video decoder must have approximately the same value. To keep their values in agreement, SCRs are inserted into the MPEG stream at least as often as every 0.7 seconds by the MPEG encoder, and are extracted by the system decoder and sent to the audio and video decoders as illustrated in Figure 2-10. The video and audio decoders update their internal clocks using the SCR value sent by the system decoder.



Figure 2-10 SCR Flow in MPEG System

2.4.2 Presentation Time Stamps

Presentation time stamps are samples of the encoder system clock that are associated with some video or audio *presentation units*. A presentation unit is a decoded video picture or a decoded audio time sequence. The encoder inserts PTSs into the MPEG stream at least as often as every 0.7 seconds. The PTS represents the time at which the video picture is to be displayed or the starting playback time for the audio time sequence.

The video decoder either deletes or repeats pictures to ensure that the PTS matches the current value of the SCR when a picture with a PTS is displayed. If the PTS is earlier (has a smaller value) than the current SCR, the video decoder discards the picture. If the PTS is later (has a larger value) than the current SCR, the video decoder repeats the display of the picture.

Chapter 3 Installation

The MPEG Video Lab is shipped with all hardware components preinstalled and all software loaded on the system hard disk. This section describes how you connect the major components and start the system for the first time.

The MPEG Video Lab is shipped in three boxes. The contents of each box is summarized in Table 3-1.

3.1 Contents

Table 3-1MPEG Video Lab Shipping Boxes

Box Size	Wt. Ibs(kg)	Contents
30 x 26 x 20	83(37.3)	Vectra PC and cables
18 x 17 x 15	28(12.6)	Monitor
12 x 12 x 12	30(13.6)	Documentation and software

3.2 Installation Procedure

Follow these instructions to install the MPEG Video Lab:

- Unpack all boxes and check the contents against the list in Table 3-1. If anything is missing, contact C-Cube Microsystems immediately.
- ✓ Locate the HP Vectra personal computer, the VGA monitor, video monitor, keyboard, and mouse. Make sure you have access to the rear of the Vectra and the monitors.
- ✓ Connect the cables to the rear panel of the Vectra as shown in Figure 3-1.
- \checkmark Turn on the power on the monitors and the Vectra.

The system should boot up automatically and should end with Windows running.

If the system boots up to Windows, you can proceed to the next chapter to learn how to use the playback software. If it does not boot up to Windows, call C-Cube Microsystems.



Figure 3-1



Chapter 4 Using the Playback Software

This chapter describes the use of the playback software to preview bitstreams on the MPEG Video Lab system. This chapter includes a tutorial and a reference section.

This section teaches you the basic operations of the playback software. You'll get the most benefit from these instructions if you perform the steps as you read. You can complete this tutorial in about 15 minutes.

Actions that you are expected to take are indicated by a check mark as shown in this example:

✓ Click the Cancel button.

Unless otherwise stated, all mouse button operations in this guide use the left mouse button.

Pressing the return key is represented by

<return>

4.1 Tutorial

4.1.1 Starting the Playback Software

The instructions in this section show you how to start the playback software once the system is powered up. If the system is powered up correctly, you should see the screen display shown in Figure 4-1.

		Prog	ram Manaq	ger	
File	Options	Window He	lp		
	VideoLab	Devel	opment		
	DDDDD DDDDD Main Microsof	DDDDD DDDD t QC/Win Games			

Figure 4-1 Playback Software Initial Screen

Proceed with the following steps to work the tutorial.

✓ Double-click anywhere on the icon labeled "VideoLab" using the left mouse button.

Two new windows, the Player and Remote windows, appear on the display as shown in Figure 4-2.



Figure 4-2 Player and Remote Windows

The Player window displays status messages about the operation of the playback software. The Remote window includes the controls used to operate the playback software. Both are standard windows in Windows 3.0, which means that you can move, iconify, close, and open them using Windows commands. (To learn how to use Windows commands, see the Microsoft Windows documentation supplied with your system.)

Examine the Remote window. The captions for all buttons except the New Seq and Quit are shaded gray. A gray-shaded caption shows that a

button is inactive. When a button is inactive, you can't operate it. Only the buttons with black captions are active.

4.1.2 Playing Back a Video Sequence

The instructions in this section show you how to choose a bitstream, start and stop the playback, and set the option for repetitive playback. These operations use the New Seq, Play, and Stop buttons, and the Repeat checkbox. (The New Seq button changes to Stop when you start playing a video sequence.)

✓ Click New Seq.

The Bitstreams Selection window appears as shown in Figure 4-3.

			Bits	streams	
File	Options	Window	Help		
	File <u>N</u> ame:			Directories: c:\windows\bstreams	ОК
	bike%.704 mobil%.704 olympics.704 olympics.544 baloon.480 nflpal.704 nflpal.352	★		c:\	Cancel
	List Files of <u>T</u> All files (*)	ype:		Drives: ☐ c: videolab ↓	
<u></u>					Vi

Figure 4-3 Bitstreams Selection Window

The Bitstreams Selection window shows the bitstreams that are available for previewing.

✔ Double-click on the bitstream "nfl.704".

The Bitstreams Selection window disappears. The Player window displays the message:

Selected file: c\mpeg\bstreams\nfl.704

The Play button in the Remote window is now active as shown in Figure 4-4.



Figure 4-4 Play Button Active

✔ Click Play.
The Player window displays a series of messages:

```
Clearing decoder memory ...
Initializing decoder ...
Microcode being loaded ...
*** Ready to play sequence ***
```

After the last message, the video sequence begins to play on the video monitor. The Remote window has several changes, as shown in Figure 4-5.



Figure 4-5 Remote Window During Play Sequence

The video sequence continues to play, since the Repeat checkbox is active.

Click on the Repeat checkbox.

The X disappears. At the end of the sequence, the final frame remains displayed on the display screen. The Remote window appears as shown in Figure 4-6.



.....

Figure 4-6 Remote Window at End of Video Sequence

✔ Click Play.

The video sequence replays. At the end of the sequence, the last frame again remains displayed, and the Remote window again appears as shown in Figure 4-6.

✓ Click Repeat, then click Play.

The sequence plays again. When the sequence reaches the end, it immediately repeats from the beginning.

✔ Click Stop.

The sequence stops immediately.

✓ Press F2 on the keyboard.

The messages appear in the Player window, and the sequence begins to play as before. Pressing F2 is equivalent to clicking Play.

✓ Press F1.

The playback stops. Pressing F1 is equivalent to clicking Stop.

Each button has an equivalent function key as shown by the legends in the Remote window. In the rest of this tutorial, the instructions use the window buttons and the mouse.

4.1.3 Pausing, Resuming, and Single-frame Advancing

The instructions in this section show how to pause the video sequence, advance the sequence one frame at a time, and restart (resume) the playback. These operations use the Pause, Advance, and Resume buttons. (The Pause button changes to Resume when you pause the sequence.)

- ✓ Select the bitstream "baloon.480" using the New Seq button and the Bitstreams Selection window as described earlier.
- ✓ Make sure the Repeat checkbox is still selected.
- ✓ Click Play.

The sequence begins to play.

✓ Click Pause.

The sequence immediately stops. The Remote window changes as shown in Figure 4-7.

✓ Click Advance several times.

The sequence advances by one frame each time you click Advance.

✓ Click Resume.

The sequence resumes. The Remote window returns to the configuration shown in Figure 4-4 above.

You can let the sequence continue to play for the operations in the next section.



VL-08

Figure 4-7 Remote Window During Pause

4.1.4 Fast-Forward and Slow-Motion

The instructions in this section show you how to play back the video sequence at fast-forward or slow-motion rates. These operations use the Fast Fwd and Slow buttons.

✔ Click Fast Fwd.

The sequence plays at fast-forward speed.

✔ Click Play.

The sequence returns to the normal speed.

✓ Click Slow.

The sequence plays back at slow-motion speed.

✔ Click Fast Fwd.

The sequence changes to fast-forward speed. You can go directly between Slow and Fast Fwd in either direction. ✔ Click Stop.

The sequence stops.

4.1.5 Ending a Playback Session

When you have finished a session, quit the playback software using the Quit button.

✔ Click Quit.

The Player and Remote windows disappear. You can either leave the system in Windows, or exit Windows back to DOS. To exit Windows and return to DOS, complete the rest of this section.

✓ Click File in the Program Manager menu bar.

The File menu appears as shown in Figure 4-8.

File	
<u>N</u> ew Open Move	Enter
<u>C</u> opy Delete Properties	Del
<u>R</u> un	
E <u>x</u> it Windov	vs
	VL-12

Figure 4-8 File Menu

Click Exit Windows.

The File menu disappears and the Exit Windows dialog box appears as shown in Figure 4-9.



Figure 4-9 Exit Windows Dialog Box

✔ Click OK.

The Exit Windows dialog box disappears, the screen is cleared, and the DOS prompt appears.

This section provides a complete reference for the operation of the playback software. The subsections are:

- □ The Development Window
- □ The Player Window
- D The Remote Window
- □ The Bitstreams Selection Window

4.2.1 The Development Window

Figure 4-10 shows the Development window.

- □ To start a playback session, double-click the "VideoLab" icon.
- □ To return to DOS, select Exit Windows from the File Menu, then click OK.

4.2 Reference Guide

Reference Guide



VL-09

Figure 4-10 Development Window Operations

4.2.2 The Player Window

Figure 4-11 shows the Player window. Each pulldown menu in the Player window is described below.

		MPEG Player	
File	Options	Help	
			-

Figure 4-11 The Player Window

File

The File Menu includes these choices:

Open

Activates the Bitstreams Selection window.

About

Displays the version number of the playback software.

Exit

Exits the playback software and returns to Windows.

Options

The Options menu includes these choices:

Bitstream Info

If a bitstream has been selected, checking Bitstream Info displays information about the bitstream. If no bitstream has been selected, checking the Bitstream Info causes the Bitstreams Select window to be displayed.

Help

To access the Player window help, click Help. The window displays a list of topics for which help is available.

Choose a topic by clicking on the topic name. Exit Help by selecting Exit from the File menu of the Help window.

4.2.3 The Remote Window

Figure 4-12 shows the Remote window.



Figure 4-12 The Remote Window

New Seq /Stop (F1)

Clicking New Seq or pressing F1 causes the Bitstreams Selection window to appear. During playback, the New Seq button changes to Stop. This button (either New Seq or Stop) is always active.

Clicking Stop or pressing F1 causes the playback to stop. To restart a stopped playback, press Play.

Play (F2)

Clicking Play or pressing F2 causes playback to begin. Play is inactive when no sequence has been selected or when a sequence is playing back at normal speed, and is active otherwise.

Pause/Resume (F3)

Clicking Pause or pressing F3 causes the playback to halt (pause) at the current frame and the Pause button to change to Resume. Pause is active whenever a playback is in progress.

Clicking Resume or pressing F3 causes the playback to resume and the Resume button to change to Pause. Resume is active whenever the playback is frozen.

Advance (F4)

Clicking Advance or pressing F4 causes the playback to advance a single frame. Advance is active only when the playback is frozen.

Fast Fwd (F5)

Clicking Fast Fwd or pressing F5 causes the playback to proceed at fastforward speed. Fast Fwd is active when a playback is in progress. (To stop fast-forward playback, click Play.)

Slow (F6)

Clicking Slow or pressing F6 causes the playback to proceed at slowmotion speed. Slow is active when a playback is in progress. (To stop slow-motion playback, click Play.)

Quit (F7)

Clicking Quit or pressing F7 ends the playback software: the Player and Remote windows disappear and the Development window is visible.

Repeat (F8)

Clicking on the checkbox to the left of Repeat or pressing F8 toggles repetitive playback. When an X appears in the checkbox, playback repeats from the beginning when the sequence completes. When no X appears in the checkbox, playback stops at the end of the sequence.

PAL

Clicking on the checkbox to the left of PAL selects PAL format for the decompressed video sequence.

4.2.4 The Bitstreams Selection Window

Figure 4-13 shows the Bitstreams Selection window. Each part of the window is described below.

		•	Bits	streams	
File	Options	Window	Help		
	File <u>N</u> ame:		1	<u>D</u> irectories: c:\windows\bstreams	ОК
	bike%.704 mobil%.704 olympics.704 olympics.544 baloon.480	Ť		c:\	Cancel
	nflpal.352 List Files of <u>T</u> All files (*)	ype:		Drives: ☐ c: videolab ↓	
L					1

Figure 4-13 The Bitstreams Selection Window

File Name

The file name entered here is the bitstream to be selected for playback. The list below the field displays the files in the current directory that match the search criteria specified in the *List Files of Type* field.

You can single-click on a bitstream name in the list or type in the bitstream name. Double-clicking on a bistream name has the same effect as single-clicking on the name followed by clicking OK.

List Files of Type

This file specification determines which files in the current directory are displayed in the file name list above. To change the specification, click

on the down arrow. A list of specifications appears. Scroll to the desired specification, then click on the specification to select it.

Directories

The current directory is shown here, followed by a graphical representation of the pathname. You can double-click on the file folder icons to move in the directory structure.

Drives

The current drive is shown here. To change the drive, click on the down arrow to display a list of drives. Then click on the desired drive to select it.

Read Only

This checkbox determines whether files are opened as read only or read/ write. Since VideoLab does not change bitstream files, the state of this checkbox has no effect on the operation.

OK Button

Clicking this button selects the bitstream specified by the *Drive*, *Directories*, and *File Name* fields. The Bitstreams Selection window disappears.

Cancel Button

Clicking this button causes the Bitstreams Selection window to disappear without selecting a bitstream.

The playback software has the capability to play scripts. A script is a combination of bitstream files that the playback software plays back in sequence. This section describes how to create and run scripts.

4.3 Using Scripts

4.3.1 Creating a Script

A script is a text file containing one or more command lines. The syntax of a command line is

filename repetitions

where *filename* is the complete DOS filename including path for the bitstream file, and *repetitions* is the number of times the bitstream is to be played.

The commands that follow show a sample script called *demo.seq*.

```
\windows\bstreams\olympics.704 2
```

```
\windows\bstreams\nfl.704 3
```

```
\windows\bstreams\olympics.704 2
```

In this example, the playback software proceeds as follows:

- □ Loads microcode for 320 x 240 resolution bitstreams.
- □ Plays *olympics*.704 two times.
- □ Plays nfl.704 three times.
- □ Plays *olympics*.704 two times.

You must observe the bitstream naming conventions given in Appendix A for proper operation of scripts.

You can create a script using any ASCII text editor such as the Notepad program supplied with Windows. Script files must end with the extension *.seq*.

4.3.2 Running a Script

Once you have created a script, you run it the same way that you run a bitstream. The procedure is:

- ✓ Click New Seq.
- ✓ Select the script using the Bitstreams Selection window.
- ✓ Click Play.

The playback software loads the first microcode and begins playing back the first bitstream. It continues until all bitstreams have been played back. If you have the Repeat checkbox checked, the script repeats from the beginning when the last bitstream has been played for the last time.

Chapter 5 The MPEG Video Decoder Board

This chapter describes the MPEG Video Decoder Board that is supplied with the MPEG Video Lab. The sections in this chapter are:

- □ Overview
- Functional Description
- Operation Overview
- Programmer's Model
- Other System Applications
- D Specifications

The C-Cube Microsystems MPEG Video Decoder Board is a 32-bit EISA card based on a single-chip MPEG decoder processor. The board performs real-time decompression of CIF/SIF- and CCIR 601-format bitstreams into RS-170 RGB analog video signals. It forms the video playback and preview element of the MPEG Video Lab.

5.1 Overview In the Video Lab system, the MPEG Video Decoder Board can support data rates of up to 8 Mbits/sec. In addition to the CCIR/NTSC monitor supplied with Video Lab, the MPEG Video Decoder Board can support interlaced monitors that use the CCIR/PAL, square-pixel NTSC, and square-pixel PAL display formats. See Table 5-3 for more information.

The MPEG Video Decoder Board supports two basic resolution modes: interpolated CIF/SIF and non-interpolated. Interpolated CIF/SIF provides for more efficient coding while maintaining reasonable image quality, while the non-interpolated mode provides the highest image quality at the expense of requiring a large, fast storage medium.

5.2 Functional Description

Figure 5-1 shows a block diagram of the MPEG Video Decoder Board. Each major functional element is described in a separate section below.



Figure 5-1 MPEG Video Decoder Board Block Diagram

5.2.1 MPEG Processor

A single-chip MPEG processor performs the decoding function on the MPEG Video Decoder Board. The processor's operation is controlled by microcode stored in the DRAM. The processor includes four ports: host, DRAM, code, and video.

The 32-bit host port is used for initialization and control of the processor and the DRAM. The 64-bit DRAM port transfers data between the MPEG processor and the DRAM. The 8-bit code port is a dedicated write-only port serviced by the coded data FIFO output. The 8-bit video port is the output for the decoded video data.

5.2.2 EISA Interface

The EISA interface connects the EISA bus to the MPEG processor host port. The EISA interface includes a set of I/O mapped control registers, described in Section 5.4, Programmer's Model.

5.2.3 Coded Data FIFO

The 4-Kbyte coded data FIFO buffers bitstream data from the EISA bus. The input to the FIFO is 32 bits wide. The 32-bit word allows the host processor (the Vectra PC in the MPEG Video Lab system) to use EISA burst transfers for maximum throughput. The output of the FIFO is 8 bits wide to match the input port of the MPEG processor. A two-signal handshake controls the rate at which 8-bit words are transferred from the FIFO to the MPEG processor.

5.2.4 DRAM

The DRAM stores the microcode for the MPEG processor and provides working storage for data during the decoding process.

The MPEG Video Decoder Board can support up to 4 Mbytes of DRAM, arranged in two banks of 256K by 64 bits. The DRAM devices used are 1 Mbit, enhanced fast-page mode, 256K by 4 bits. The board supplied with the MPEG Video Lab system includes the full complement of 4 Mbytes.

The MPEG processor provides all DRAM control signals including address and RAS/CAS signals. DRAM timing is controlled entirely by the MPEG processor.

5.2.5 Color Space Converter

The color space converter (CSC) converts the YCbCr output of the MPEG processor into the digital RGB data used to drive the video DACs. The CSC also includes bandwidth limiting filters for the YCbCr data and performs the interpolation for the downsampled CbCr data. Four different chrominance filter modes can be set using the Configuration/Status register, as described in Section 5.4, Programmer's Model.

5.2.6 Video DAC

The output of the color space converter is fed to the video DAC. The video DAC converts 24-bit RGB data into three analog RGB video signals compatible with 75-ohm cabling. The RGB signals and composite sync are combined on a DB-9 connector at the rear of the MPEG Video Decoder Board.

5.2.7 Raster Timing Generator

The raster timing generator controls the timing of the video output components, the line store, color space converter, and video DAC. The raster timing generator can be programmed for a video clock rate of 24.5454, 27.0000, or 29.5000 MHz using the VF field in the Configuration/Status register (see Section 5.4.2 Configuration/Status Register, for more information).

5.2.8 Line Store

The MPEG Video Decoder Board runs the MPEG processor at a system clock rate of 30 MHz to achieve high decoder throughput. However, the video clock rate produced by the raster timing generator is slower than the 3 MHz MPEG processor clock. The line store is a 1-Kbyte dual-port buffer that decouples the MPEG processor and video clocks and allows the processor to operate at a higher clock rate than the video output subsystem without data loss.

5.3 Operation Overview

The MPEG Video Decoder Board is a micro-programmable MPEG decoder pipeline. The host (the Vectra PC in the MPEG Video Lab system) initializes the board and the on-board MPEG processor, loads the microcode into the on-board DRAM, starts the processor, and writes the coded bitstream to the coded data FIFO. The MPEG processor reads the bitstream from the FIFO, decodes it, and sends the video signals to the video output. At startup, the host driver software program polls the EISA slots for the EISA Product ID register (EPIR). It reads the value from the EPIR, initializes the board registers, and posts the board address to the C-Cube resident driver. (The resident driver is a terminate-and-stay-resident, or TSR, program.)

The host driver then loads bootstrap code into the decoder instruction memory (IMEM) and loads the rest of the microcode into the DRAM. The microcode is self-loading so that routines are swapped into the IMEM as needed.

Once the initialization is complete, the host starts the MPEG processor by writing to a dedicated register. The host writes coded data to the coded data FIFO and monitors the status of the FIFO by reading the flags in the Status register. The FIFO can also be configured to generate an interrupt on the not-half-full condition to request more data.

The MPEG processor reads bitstream data from the FIFO and writes it into a circular buffer in the DRAM. The decoding units internal to the MPEG processor parse and decode the bitstream stored in the circular buffer into completed video rasters. The MPEG processor then transfers the decoded images from DRAM to an internal video FIFO, optionally interpolating from CIF/SIF to full-screen resolution in the process.

The decoding and video output processes compete for DRAM access, so the competing requests must be arbitrated by the MPEG processor. The priorities from highest to lowest are:

- □ Video output
- DRAM refresh
- Bitstream decoding

The DRAM throughput is a key system performance parameter. C-Cube has maximized DRAM throughput by using a 64-bit bus and running the memory in page mode whenever possible.

The MPEG processor writes video data from the internal video FIFO into the line store at the system clock rate of 30 MHz. The video data progresses to the color space converter for conversion from YCbCr to RGB and then to the video DACs for conversion into analog RGB signals.

5.4 Programmer's Model

The MPEG Video Decoder Board defines addresses in both the EISA I/O space and the memory space. Figure 5-2 shows the address map. The I/O space addresses are absolute; the memory map spaces are relative to the value in the Base Address register. All addresses are given in hexadecimal notation.



5.4.1 EISA Product ID Register

Register Type:	I/O	
Read/Write:	Read only	
Address:	0x480	
Size:	32 bits	
31		0
	0x 0C63 020A	

The EISA Product ID register identifies the board to the host software driver. When this register is read, it returns the ID value 0x 0C63 020A. This value is fixed in firmware and is determined by rules set forth in the EISA specification.

5.4.2 Configuration/Status Register

Regi	ster	Туре	:		I/O					
Read	l/Wr	ite:]	Read/write (bits 6 and 7 are read only)					
Add	ress:			(0x48	4				
Size					8 bits	5				
7	6	5	4	3	2	1	0			
Res	NE	Res	V	F	C	F	ME			

The Configuration/Status register provides control over several board parameters. Bits 5 and 7 are reserved. When these bits are read, they return zero. When they are written, the written value is ignored. The other bit mnemonics have the following meanings:

NEFIFO Not Empty (bit 6)Read onlyWhen NE is set to 1, the coded data FIFO contains at leastone code word. At reset, NE is cleared to 0.

VFVideo Format Select (bits 4:3)R/WThis field selects the video output format as shown in Table 5-1. (See Table 5-3 later in this chapter for more information

about the video output parameters for the video formats.) At reset, the VF field is set to 00 (format = CCIR NTSC).

Table 5-1VF Field Values

VF Value	Video Output Format
00	704 x 480 or 352 x 240
01	704 x 576 or 352 x 288
10	544 x 480
11	480 x 480

CF Chrominance Filter Select (bits 2:1) R/W

This field selects the cutoff frequency for the chrominance filter in the color space converter as shown in Table 5-2. At reset, the *CF* field is set to 00 (cutoff frequency = 1.1 MHz).

Table 5-2

CF Field Values

CF Value	Chrominance Cutoff Frequency (MHz)
00	1.1
01	1.5
10	1.3
11	1.8

MEMemory Map Enable (bit 0)R/WWhen ME is set to 1, the memory map registers are enabled.When ME is cleared to 0, the memory map registers are disabled. At reset, ME is cleared to 0.

5.4.3 Base Address Register

Register Read/Wr Address: Size:	Type: ite:	I/O Write only 0x488 32 bits					
31	24	23		12	11		0
Ad	dH		AddL			Don't Care	

The Base Address register defines a 4-Kbyte segment in the EISA memory space for the MPEG processor, DRAM, and coded data FIFO registers. Bits 11:0 are ignored and can be written with any value. The other bit mnemonics have the following meanings:

AddH Address, High Byte (bits 31:24) Write only

This field contains the 1's complement of the most significant byte (MSB) of the address. Driver software must complement these bits in the base address value before it is loaded into the register.

AddL Address, Low Byte (bits 23:12) Write only

The AddL field contains the lower bits of the address in standard (uncomplemented) form.

5.4.4 Decoder RREG and GREG Registers

Register Type:	Memory map
Read/Write:	Read/write
Address:	RREG: Base address + 0x000
	GREG: Base address + 0x200
Size:	32 bits x 128 entries each



The Decoder RREG and GREG registers are internal to the MPEG processor.

5.4.5 DRAM Address Register

Register Type:	Memory map	
Read/Write:	Read/write	
Address:	Base address + 0x480	
Size:	32 bits	
31		0
	DRAM Address	· · · · · · · · · · · · · · · · · · ·

DRAM Address is the index for host-DRAM data transfers. The host loads this register with the target DRAM address before initiating a DRAM data access. The host initiates the DRAM data access by reading from or writing to the DRAM Data register.

5.4.6 DRAM Data Register

Register Type:	Memory map	
Read/Write:	Read/write	
Address:	Base address + 0x484	
Size:	32 bits	
31		0
	DRAM Data	

The DRAM Data register is the data port for host–DRAM data transfers.

To request a DRAM read operation, the host reads from this register. The MPEG processor gets the data from the DRAM address pointed to by the DRAM Address register and loads it into the DRAM Data register, where it is read by the host.

To perform a DRAM write operation, the host writes the data into the DRAM Data register. The MPEG processor transfers the data from this register to the DRAM address pointed to by the DRAM Address register.

5.4.7 Coded Data FIFO Reset Register

Memory map
Write only
Base address + 0x488
8 bits
0
t care

The host resets the coded data FIFO by writing any value to this register. Data in the coded data FIFO is flushed and the pointer is reset to the first location in the FIFO. The coded data FIFO is also reset by the host computer's EISA reset signal.

5.4.8 Coded Data FIFO

Register Type:	Memory map
Read/Write:	Write only
Address:	Base address + 0x800
Size:	1024 32-bit words (4096 bytes)



The host places data in the coded data FIFO by writing a 32-bit word to any address in the coded data FIFO address space. The code data word must be aligned to a long-word boundary. Regardless of the address used, the 32-bit word is pushed into the FIFO. Eight- and 16-bit operations are not supported; 32-bit writes must be used.

5.5 Other System Applications

While this guide is designed for the Video Lab system, the MPEG Video Decoder Board can be used in other applications. This section describes the design considerations for system applications of the MPEG Video Decoder Board and provides some general installation guidelines.

5.5.1 System Design Considerations

The C-Cube Microsystems MPEG Video Decoder Board can be integrated into any 386- or 486-based system that uses the EISA bus. The system should include an SCSI-interface hard disk with a seek time of 16 ms or less and a capacity of at least 300 Mbytes, with 600 Mbytes the recommended size.

The C-Cube Microsystems MPEG Video Decoder Board uses 32-bit wide buses for all critical data paths and also supports no-wait-state EISA memory-mapped I/O and burst transfers. Systems designed for 8 Mbit/sec performance should include an EISA DMA/SCSI disk controller.

Special care must be given to the disk format to minimize delays due to the disk seek time. MPEG bitstream files should be stored in contiguous disk locations and the disk interleave factor should be set appropriate to its track-to-track seek time and rotation latency. Most systems will use a 1:1 interleave.

5.5.2 Installation Guidelines

Warning: When you install the MPEG Video Decoder Board in the host computer, keep in mind that the computer contains potentially hazardous voltages and currents. Always follow these precautions:

- □ Turn off the main power to the host computer.
- Remove metal jewelry (rings, watches, etc.) that could come into contact with live electrical components.

The host computer and the MPEG Video Decoder Board contain staticsensitive components that can be permanently damaged by electrostatic discharge. Observe these precautions to minimize the danger of damage:

□ Do not remove the MPEG Video Decoder Board from its anti-static wrapping until you're ready to install it.

- □ Leave the host computer's power cord plugged in to a grounded receptacle to provide a ground return for static charge buildup.
 Double-check that the power switch is turned off.
- Attach a conductive wrist strap from your wrist to the computer chassis. The strap should have a resistance of 100 Kohm to
 1 Megohm and should make a secure connection to your wrist. If no such strap is available, take the precaution of touching the chassis (the power supply housing is best) as you install or remove the MPEG Video Decoder Board.
- □ Always handle the MPEG Video Decoder Board by its edges. Do not touch the metal fingers of the EISA edge connector.

The MPEG Video Decoder Board may be installed in any EISA slot, except slot #2. However, in systems with marginal cooling (e.g., those that draw cooling air through the power supply), the board should be located away from other add-in boards that are sensitive to heat or that generate large amounts of heat.

The MPEG Video Decoder Board is shipped with jumpers J3, J4, J5, and J6 open (no interrupts). Contact C-Cube for information about using the interrupts.

Note: The MPEG Video Decoder Board TSR driver interrupt is 0x85. The MPEG Video Decoder Board is mapped at memory locations 0xD800 – 0xDBFF.

The MPEG Video Decoder Board generates and uses radio-frequency (RF) energy. To prevent electromagnetic interference problems, the board should be connected to the video display monitor with the cable supplied with the board. Also, the EISA computer system should have all shielding in place including the top cover and blank slot filler brackets and plates.

5.6 Specifications

This section describes the specifications of the MPEG Video Decoder Board. It presents general and interface specifications.

5.6.1 General Specifications

Form Factor

EISA, full-length, 32-bit slave-only bus, single slot.

Connectors

- \square EISA card edge
- □ Analog video output (DB-9 RGB)

Configuration Options

- □ Video formats: 704 x 480 (352 x 240), 704 x 576 (352 x 288), 544 x 480, and 480 x 480
- □ Four chrominance cutoff frequencies: 1.1, 1.3, 1.5, and 1.8 MHz
- □ Two interrupts, each with two strap options (IRQ5/12 for FIFO Half-Empty, IRQ3/11 for MPEG decoder interrupt)

Regulatory Agency Approval

Tested and verified to comply with the limits for a Class B device, pursuant to Subpart J, Part 15 of the FCC Rules.

Environmental

- Operating temperature: 0 to 75 degrees C
- □ Storage temperature: -55 to 80 degrees C
- □ Humidity: 8% to 85%
- \square Power: +5 volts ± 5% @ 3.6 A nominal

5.6.2 Interface Specifications

The MPEG Video Decoder Board includes two external connectors: EISA card-edge, and 9-pin video output. The EISA edge connector pinout is specified in the EISA standard.

The analog video output connector is a DB-9 female connector with pins for R, G, B signals plus the composite synchronization signal CSYNC. The RGB signal levels are in accordance with the RS-170 specification. The CSYNC signal is at least 2 volts peak-to-peak into a 75-ohm load. Figure 5-3 shows the pin designations for the analog video output connector.



VL-26

Figure 5-3 Analog Video Output Connector

5.6.3 Video Display Formats

The MPEG Video Decoder Board supports the four display formats described in Table 5-3.

Format	Clock	Cik/	Pixels/line		Lines/frame		Frames/sec
	(MHz)	line	total	disp	total	disp	
704 x 480	27.0000	1716	720	704	525	480	29.97
704 x 576	27.0000	1728	720	704	625	576	25.00
544 x 480	27.0000	1716	720	544	525	480	29.97
480 x 480	27.0000	1716	720	480	525	480	29.97

Table 5-3 Video Display Formats

Specifications

Appendix A Encoding Guidelines

This appendix presents useful information for creating coded bitstreams that are to be decoded and played back on Video Lab. It describes the compliance of Video Lab to the MPEG draft standard and presents bitstream naming conventions.

The MPEG Video Decoder Board in Video Lab supports the MPEG draft standard (ISO-IEC/JTC1 SC29, dated November 22, 1991) with the following restrictions:

The long Huffman codes of Tables 2-B.5e and 2-B.5f are not supported. Instead, the escape codes listed in Table 2-B.5g are supported.

- □ The 28-bit escape mechanism of Table 2-B.5g is not supported.
- \square The quantized coefficients must be in the range -128 to 127.
- Macro blocks cannot be skipped (MBA mechanism) when immediately followed by an intra macro block. Consequently, MBA greater than 1 can only be found on non-intra blocks.

A.1 MPEG Draft Standard Compliance

- The MPEG Video Lab does not support bidirectional frames for 704 frame resolution.
- \square The macroblock address of an intra block cannot be greater than 1.

In addition, the current revision of the MPEG Video Decoder Board firmware places these restrictions on the coded image:

- □ Slices must start at the left edge.
- □ The number of macro blocks per picture width must be even.

The usual naming convention for MPEG bitstreams used in Video Lab is:

name.res

where *name* is a descriptive name of 1 to 8 alpha characters, and *res* is one of the following extensions: SEQ, .352, .480, .544, or .704

A.2 Bitstream Naming Conventions

Appendix B MPEG Video Lab Software Files

This appendix lists the files provided with the MPEG Video Lab. The directory structure for the MPEG Video Lab software is shown in Figure B-1. Each directory is further explained below.

\(root)

This directory contains this file used by MPEG Video Lab:

□ AUTOEXEC.BAT: This file includes a command to start the terminate-and-stay-resident (TSR) program VLBDDRVR.EXE, the driver for the MPEG Video Decoder Board.

\VIDEOLAB:

This directory includes the following files:

- □ *VIDEOLAB.EXE*: The executable version of the demo application supplied with the MPEG Video Lab.
- □ *VLBDDRVR.EXE*: The executable version of the driver for the MPEG Video Decoder Board.

B.1 Software Files



Figure B-1 **Directory Structure Showing MPEG Video Lab Files**

\VIDEOLAB\BSTREAMS:

This directory contains the MPEG-encoded bitstream files. Table B-1 shows the files that are supplied with MPEG Video Lab.

Table B-1 **Bitstream Files Supplied with MPEG Video Lab**

Filename	Resolution
BIKE%.7041	704 x 480 @ 30 Hz
MOBIL%.704 ¹	704 x 480 @ 30 Hz
OLYMPICS.704	704 x 480 @ 30 Hz
OLYMPICS.544	544 x 480 @ 30 Hz
OLYMPICS.480	480 x 480 @ 30 Hz
BALOON.480	480 x 480 @ 30 Hz
NFLPAL.704	704 x 576 @ 25 Hz
NFLPAL.352	352 x 288 @ 25 Hz
NFL.704	704 x 480 @ 30 Hz
NFL.352	352 x 240 @ 30 Hz

1. See Section B.2.

VIDEOLABCODE:

This directory contains the microcode that is downloaded to the MPEG processor on the MPEG Video Decoder Board.

\VIDEOLAB\SRC:

This directory contains the source code for the demo application VIDEOLAB.EXE

\VIDEOLAB\TSR:

This directory contains the source code for the board driver VLBD-DRVR.EXE

WINDOWS:

This directory contains this file used by MPEG Video Lab:

□ *VIDEOLAB.INI*: Contains references to microcode files used by the MPEG processor. Also determines the directory in which the microcode and bitstream files are contained.

Previous versions of the MPEG Video Lab supported bitstreams coded at 704 resolution. These encoded bitstreams can still be used with Version 1.2, provided the file name includes a % sign in the name as in BIKE%.704. Be sure to modify the file names of old 704-resolution interlaced files if you want to use them with Version 1.2 of Video Lab.

B.2 704 Files From Previous Versions

•

.

Errata for MPEG Video Lab User's Guide

November 5, 1993

This document describes changes to the C-Cube MPEG Video Lab User's Guide, PN 92-6000-001.

Note the following typographical conventions:

- □ In most cases, material that is being deleted is shown with a strikethrough mark, for example: this comes out.
- □ New material is always underlined, for example: this is new.

The changes described in this document apply to MPEG Video Lab systems with the HP NetServer 4/33 LM personal computer. (The original manual applies to systems with the HP Vectra personal computer.)

Applicability

1
`hanter 1	Replace Section 1.1 and Figure 1-1 with the following:
	The basic MPEG Video Lab system shown in Figure 1-1 is based on the <u>HP NetServer 4/33 LM</u> personal computer, which also uses the EISA bus. The standard configuration includes the following components (supplier is Hewlett-Packard except where noted oth- erwise in parentheses):
	Eight <u>32</u> Mbytes RAM
	Super VGA monitor
	Extended keyboard with 101 keys
	D Two-button mouse
	Floppy disk drive: 3.5-inch, 1.44 Mbyte capacity
	 SCSI hard disk: 670 Mbytes unformatted (Micropolis 1624)
	DAT tape drive: 2 Gbytes (Maynard® MaynStream®)
	Super VGA graphics card
	MPEG CL950 Video Decoder Board (C-Cube Microsystems)
	 MPEG CL450 Video Decoder Board (optional – C-Cube Mi- crosystems)
	 Board driver software (C-Cube Microsystems)
	□ Ethernet card (3Com W/RG58)
	D PC-NFS networking software (Sun Microsystems)
	Deserializer
	 MS-DOS® 5.0 and Windows® 3.0 (Microsoft®), with Win- dows running in 386 extended mode
	Sample bitstreams
	Complete documentation

2 C-Cube Microsystems



Figure 1-1 Video Lab Components

Change the bulleted list in Section 1.3 to read:

- Installing and Managing Your HP NetServer LM Series
 (Hewlett-Packard)
- MaynStream DOS User's Manual (Maynard)
- □ MS-DOS 5.0 User's Guide & Reference (Hewlett-Packard)
- □ *MS-DOS 5.0 Getting Started* (Hewlett-Packard)
- □ Windows User's Guide (Microsoft)
- D Windows Questions & Answers (Microsoft)

-

ļ





Note: When the CL450 Decoder Board is installed, the VGA monitor plugs in the CL450 board. The dummy jumper plug supplied with the system must be installed in the NetServer VGA when the system is booted. The video monitor can be plugged into either the CL450 or CL950 board.



Add Appendix C, MPEG Decoder Player for MS-DOS.

Appendix C MPEG Decoder Player for MS-DOS

This document describes the installation and use of the MPEG Decoder Player for MS-DOS, referred to in this manual as the DOS Player. The DOS Player allows you to decode and playback MPEG bitstreams using the C-Cube MPEG CL950 Video Decoder Board.

Overview

This procedure assumes that:

- The MPEG CL950 Video Decoder Board is installed in your system
- An RGB video monitor is connected to the MPEG CL950 Video Decoder Board.
- □ Your system has at least one MPEG encoded bitstream

If you don't have any encoded bitstreams, you can still perform the installation steps, but you can't verify the correct operation of the board and the software. Software Installation This procedure describes the installation of the DOS Player. It assumes that your system is running DOS.

- 1. Insert the DOS Player diskette in the floppy drive. (The procedure assumes that your floppy drive is drive B:)
- 2. Enter the command

C:> xcopy b: c:\vlab_dos\ /s /e

All the files on the diskette are copied to the directory VLAB_DOS on drive C:. The directory has the structure shown in Figure 1:



Figure 1 DOS Player Directory Structure

3. Make the changes to the AUTOEXEC.BAT file as shown in Figure 2.



Figure 2 Changes to AUTOEXEC.BAT

The SET line creates the environment variable DB950, which sets the software interrupt to be installed. This interrupt may be changed if your system has a conflict.

4. Reboot the system.

The following message is displayed, indicating the successful installation of the driver.

```
MPEG Decoder Board Driver Version 2.2A
```

(c) Copyright 1990,1991 C-Cube Microsystems

To remove the MPEG Decoder Board driver, run MPEGBDRV.EXE a second time.

This procedure shows how to modify the batch files supplied with the DOS Player to point to the proper microcode and location of your bitstream file. The examples show the changes needed for a 544 x 480 NTSC bitstream named OLYMP.544, which is located in \VLAB_DOS\DEMO\BS.

Modifying the Batch Files **Table 1**

B Files

1. Using any ASCII text editor, open the B file corresponding to the resolution and format of your bitstream file as shown in Table 1. For example, the correct file to edit for a bitstream recorded at 544 x 480 resolution (NTSC format) is

_	Resolution	Format	File Name	-
	352 x 240	NTSC	demo352n.b	
	352 x 288	PAL	demo352p.b	
	480 x 480	NTSC	demo480n.b	
	544 x 480	NTSC	demo544n.b	
	704 x 480	NTSC	demo704n.b	
	704 x 576	PAL	demo704p.b	

\VLAB_DOS\DEMO\CODE\DEMO544N.B.

2. Change the line shown in Figure 3 to point to the correct microcode file as shown by Table 2. Note that the encoding order of file (standard or bottom field first) determines which microcode file you use.



Figure 3 Changes to DEM0544N.B

۰

	Resolution	Format	Standard Order	Old Order ¹
Î	352 x 240	NTSC	uc352.240	N/A
	352 x 288	PAL	uc352.288	N/A
	480 x 480	NTSC	iuc480.480	old480.480
	544 x 480	NTSC	iuc544.480	old544.480
	704 x 480	NTSC	iuc704.480	old704.480
	704 x 576	PAL	iuc704.576	N/A

Table 2 DOS Player Microcode Files

1. Old order is bottom field first.

3. Edit the proper DEMOS file to point to the bitstream as shown in Figure 4. Table 3 lists the DEMOS files.

Note: You play more than one sequence just by adding more lines in the DEMOS file. All files must use the same microcode.

```
bf \vlab_dos\demo\olymp.544 0 0 10 10
```



Number of times to play the bitstream file

Figure 4 Changes to DEMOSN.544

Table 3DEMOS Files

Resolution	Format	File Name	
352 x 240	NTSC	demosn.352	
352 x 288	PAL	demosp.352	
480 x 480	NTSC	demosn.480	
544 x 480	NTSC	demosn.544	
704 x 480	NTSC	demosn.704	
704 x 576	PAL	demosp.704	

Operation This section explains how to decode and display the OLYMP.544 file described in the previous section.

1. Make sure the directory is set to \VLAB_DOS\DEMO.

2. Enter the command

```
544N <return>
```

The screen displays:

MPEG Decoder Board Player Version 2.2A (c) Copyright 1990,1991 C-Cube Microsystems

MPEG Decoder Board Found

The DOS Player then installs the microcode and starts decoding and displaying the bitstream.

3. Control the decoding and display using the function keys shown in Table 4.

Table 4 Control Keys

Key	Function	Description
F1	Freeze	Freeze the decoding and display
F2	Single-step	Advance the video sequence by one frame
F3	Play	Start or resume play at normal speed
F4	Fast forward	Play the sequence at twice normal speed
F5	Slow motion	Play the sequence at one-half normal speed
Q <return></return>	Abort	Stop the decoding and display

4. When you have finished viewing the sequence, press Q <return> to end the decoding and display.

Warning: Be sure that you stop the application before you power-down the system.

Error Messages

The MPEG Decoder for DOS issues the following error messages:

ERROR - DRIVER NOT LOADED

The MPEGBDRV.EXE was not loaded or the board was not found by the driver.

Bitstream Restrictions

If you do not see the "MPEG Decoder Board Found" statement when you load the MPEG.EXE file, try powering the system down and reinserting the MPEG Decoder Board.

ERROR - No script file specified

The MPEG.EXE file requires a script file on the command line for execution (i.e., DEMO352.B).

Error opening filename

If the player can not find the script file that you specified, it will respond with this error message. Quit the player (typing Q <return>) and start again.

If the application appears as though it is running, and there is no video on the monitor, repeat the Installation steps.

In its current version, the MPEG Decoder Board II implements all of the MPEG Committee Draft specifications with these exceptions:

- □ The "long" Huffman codes of Table B5e and B5f are not supported and the escape mechanism (Table B5g) is to be used instead.
- The 28 bits escape mechanism of Table B5g is not supported and the magnitude of the quantized coefficients should not exceed 127.
- Macro blocks cannot be skipped (MBA mechanism), when immediately followed by an intra MacroBlock. As a consequence MBA larger than 1 can only be found on Non-Intra blocks.

There is essentially no impact on picture quality as long as those constraints are observed at the encoder.

In addition there are a few limitations in the current revision of the firmware:

- \Box Slices have to start at the left edge.
- \Box The count of MB per picture width must be even.

Bitstream Restrictions

C-7

Bitstream Restrictions

C-CUBE MICROSYSTEMS

C-CUBE MPEG VIDEO LAB USER'S GUIDE







C-CUBE MICROSYSTEMS

1778 McCarthy Blvd. Milpitas, CA 95035 Tel: (408) 944-6300 Fax: (408) 944-6314