PHILIPS

PHILIPS INTERACTIVE MEDIA

Technical Note #88

Status of the Digital Video System

Charles Golvin

December 10, 1993

This technical note describes reported problems with the initial CD-i Digital Video system. These reports consist of both confirmed and potential problems.

Status of the Digital Video System

Overview

The first implementation of the CD-i MPEG extension (Digital Video) to the *CD-i Full Functional Specification* (Green Book) is now released. During the development and finalization of this system, various problems have been reported to the Philips engineers responsible for the development of the hardware and system software. This technical note summarizes the problems that remain in the initial released version of the system.

The problems are divided into two sets: problems that have been confirmed by Philips and those observed by developers, but not yet fully verified by Philips. The first group of problems is summarized in the formal Philips IMS bug list for the Digital Video system, which is provided as an appendix to this document. The second group is summarized below in the section entitled "Reported Problems." Each problem is summarized and followed by one or more possible workarounds, if known. Many of these problems were discovered by engineers working on linear play-back systems; therefore, the suggested workarounds may seem particularly slanted toward that type of application and may not be suitable for more interactive applications. The workarounds should not be taken as a finite list of possibilities; it is possible that other ways around the problem exist, but were simply not discovered.

Finally, this document also provides some clarification of implementation details in the system that may not be clear from reading the extension document.

Reported Problems

- MD_TimeCd field is not correct during scanning

When playing Digital Video at normal speed, the MPEG video driver updates the MD_TimeCd and MD_TmpRef fields of the motion video descriptor continuously with each decoded picture. When playing in scan mode, the system only updates these fields on a group of pictures (GOP). However, in the current implementation, the MD_TimeCd and MD_TmpRef fields contains the actual time code from the last GOP, not the picture currently displayed.

Workaround: None

Video disturbances when changing speeds

r changing speed from scan speed to single step, a subsequent change of speed to
motion or normal can result in several incorrectly decoded frames. (Due to
blems with **mv_pause** and **mv_cont**, some developers chose to implement
se and continue by changing speed to single step or normal speed, respectively.)

rkaround: None

mv_pause returns device not ready

en **mv_pause** and **mv_cont** are used to control the playback, **mv_pause**
metimes returns error #246 (device not ready). This may be due to **mv_pause**
owing **mv_cont** too closely; in this case, the system has not yet received data to
ume decoding. Once this error has occurred, it is not possible to continue the
ent play.

ound: If this error is returned by **mv_pause**, abort the current **mv_play** (and
associated disc play) and then restart both plays. The subsequent motion video
then starts on the nearest entry point picture available (see the following bug
cerning playing motion video from random locations). Thus, the appearance is
the same as it would be if the pause had succeeded unless this entry point is the
ure immediately following the picture on which the play was paused. Another
sible solution has also been proposed although not verified: **mv_pause** returns
error because the previous **mv_cont** did not continue decoding; therefore
ying **mv_pause** should eventually result in success. Please contact one of the
ple referenced at the end of this document should you experiment with this
unique and be able to provide further information.

mv_cont is called quickly after mv_freeze

r freezing a motion video playback with continuing audio decoding, using
: shortly after the play has been frozen may result in corruption of one or
e frames. This behavior has been seen using filter parameters of both 0 and 1 for
_cont.

rkaround: Because this problem is tied to the period between the **mv_freeze** and
_cont calls, the application may simply impose an artificial delay following
_freeze, during which it is not possible to continue the motion video decoding.
ther possibility is to use a filter parameter of 2 for **mv_cont**; however, in this
, the period before decoding resumes is dependent on the nature of the MPEG
eo stream.

- System shows a frame from the previous play

After aborting a motion video play, the next motion video play may display one picture from the previous stream for one or two fields.

Workaround: The simplest workaround in this case is to hide the motion video display following the **mv_abort** and then to wait for two PIC signals (or detect the discontinuity in the timecode fields in the motion video descriptor) before re-enabling the display. For titles that cannot tolerate blanking of the motion video plane, another possible workaround exists. The bug occurs only when the previous play is aborted while the system is displaying a bidirectionally coded picture (B-frame). While it is not possible for the application to know when a B-frame is displayed, it is possible to know when certain intracoded pictures (I-frames) are displayed. When the GOP (Group of Pictures) event is enabled, it is generated by the system when the first picture in a group of pictures is displayed—this is guaranteed to be an I-frame. In this case, the application, once prepared to abort the current play, must assure that GOP events are enabled and then will abort the play on receipt of the next GOP event. Of course, the disadvantage of this workaround is the lack of immediacy in aborting the current play. With "typical" encoding parameters, this delay may be up to one half second.

- NIS problems

These bugs appear during playback of a stream in which the stream parameters (in particular, image size) change midstream.

1. When the video is aborted during the playback of a segment with a smaller image size and then restarted at normal speed at a location with a larger image size, video disturbances occur at the bottom of the first **two** frames of video in the new play.

   Workaround: When starting a new play at a location with a different image size, delay the call to **mv_show** until the third PIC event is received.

2. When the image size changes while playing back in scan mode, two NIS events are generated. It appears as if one NIS event is generated just before the first picture is displayed with the new image size, and the second is generated before the second picture with the new image size is displayed.

   Workaround: While in scan mode, ignore NIS events that do not correspond to changes in the image size field of the motion video descriptor.

3. When the image size changes while scanning, the first image displayed with the new image size is corrupted.

   Workaround: While in scan mode, hide the first picture with new image sizes.

- Video offset parameter is required for some plays

If a motion video play is begun on a sector containing both an end of sequence code and a start of sequence code, the play will not start in the case that the video offset parameter passed to **mv_play** is zero.

Workaround: Provide a correct video offset parameter

- PIC signals are generated after play is aborted

When a motion video play is aborted with the PIC event enabled, the system continues to generate PIC signals even though no new pictures are being decoded.

Workaround: Disable PIC events after aborting a motion video play or ignore all PIC signals that are not accompanied by changes in the **MD_TimeCd** and/or **MD_TmpRef** fields. Another alternative is to use the method described above: only abort the play on a GOP.

- Video decoding does not start

Sometimes a motion video play simply does not begin decoding pictures. This may have several causes. The first is when the system does not encounter an entry point within the first fifteen video sectors delivered from the disc; the second is due to an error in the system software under certain conditions surrounding the DTS value in the stream. In both cases, one will likely hear some audio decoded.

Workaround: Always be sure that the play starts less than fifteen video sectors before an entry point picture. There is no known workaround for the other case.

- mv_window or mv_pos after mv_pause

If **mv_pause** is used to pause a motion video play, calls to modify the decoding window (**mv_window**) or to reposition the decoded window (**mv_pos**) have no effect.

Workaround: Rather than using **mv_pause**, abort the current play and restart from the nearest entry point.

- Start of play is not on sequence or system header

When the first play on a path to the motion video device (or on a motion video descriptor) does not start on a sequence or system header, the subsequent decoding of B-frames and P-frames after the next sequence header is encountered may be incorrect.

Workaround: When using a path/descriptor combination for the first time, ensure that the first play begins on a sequence or system header. Such a play may need to take place with the motion video plane hidden, for example, with a dummy play at the start of the application.

- Further explanation of Eindhoven bug #28

  Note: *In the following, the references to pixels mean physical pixels; whereas, references to coordinates that are passed to the various SetStat calls mean UCM coordinates.]*

The current decoder hardware decodes B-frames only to the size of the currently selected window plus a border of 16 pixels in all directions. For this reason, when a new decoded window is requested, it is possible, under certain conditions, that the next picture will not be correctly decoded. In order to prevent this potential display of bad data, the motion video driver has been adapted to, under certain conditions, de the motion video display for one or two fields during the period that the lisplay could be corrupted. This blanking of the motion video display is seen when **mv_window** is called with either a change in coordinates less than -31 or greater than 32 in either coordinate, with a scroll flag value of 0.

> ***Note:*** *A scroll flag value of 1 and coordinate changes in the described range cause* ***mv_window*** *to return an error. This is because the request, a coordinate change of more than 16 pixels on the next video retrace, is not allowed by the Green Book extension.*

The behavior of **mv_pos** with respect to this problem in the decoder chip is slightly different than that of **mv_window**. The **mv_pos** function does not return an error when the position move is less than -31 or greater than 32 with a scroll flag of 1; however, as dictated by the Green Book, such a change can take place only on the next picture change. The blanking of the DV screen is dependent on the window size, the amount by which the picture is to be moved, and the scroll flag.

With a scroll flag of 1, the DV screen is never blanked. This can be considered a bug the system; however, it does provide a mechanism to circumvent the driver ehavior, if desired. In particular, because the glitches are only produced when decoding B-frames, it is possible to perform the window repositioning only when I-frames or P-frames are displayed (using the GOP event, by knowledge of the cadence of B-pictures in the stream, or by encoding a stream without B-frames). With a scroll flag of 0, the screen is blanked whenever one of the requested position coordinate changes is less than -30 (not -31) or greater than 32.

In particular, all of this means that it is possible to scroll and pan the image without glitches and without the system blanking the DV display, provided that the window is moved either with a scroll flag of 1 in **mv_pos** or by no more than 30 UCM coordinates in the other cases, and that subsequent moves do not occur prior to the decoding of the next picture.

• **mv_jump** is not correct

The operating system function **mv_jump**, provided for seamlessly gluing together MPEG video streams, is not completely correct. Among the anomalies that have been detected while using this function is that the video frame rate does not remain constant at the rate prescribed in the video stream and, more importantly, that the video decoding may hesitate for some nontrivial amount of time.

Workaround: The application must provide routines to alter the SCR, PTS, and DTS in the stream(s) being played and provide the appearance of a continuous stream of correct information. PIMA is developing a library of routines for this purpose for distribution.

Note: ma_jump functions correctly.

Further Clarification

• Minimum frame rate using **mv_next**

In early investigations with the system, it was possible to play from CD in single step speed and still achieve the full frame rate coded in the stream (e.g., 30 frames per second). Subsequent versions of the system software only allowed a speed of roughly two thirds the coded frame rate. Because the Green Book extension is not clear on what rate an application may expect to achieve in this mode, it has been agreed that an addendum will be provided clarifying this point. Therefore, an application may be assured that it will always be possible to achieve a frame rate of two thirds the coded stream frame rate when playing in this manner. It is up to the application to regulate the actual speed, because systems are not prevented from providing the ability to play at higher rates, up to the full coded rate.

• Flushing of audio PCLs when paused: flush if more than 2 PCLs

When playing audio from disc and **ma_pause** (or **mv_pause** in synchronized mode) is issued, the system may or may not flush the contents of any audio PCLs that have been filled, but have not yet been given to the decoder. If the chain of audio PCLs provided to **ma_play** consists of exactly two PCLs, the system does not flush the old data; in all other cases the data is discarded.

• Mode for **I$Open** must be zero

The mode parameter passed to **I$Open** when opening the MPEG audio and video devices must be zero.

Summary

This note has provided a list of the known problems in the current Philips implementation of the Digital Video system. This list should not be interpreted to mean that no other problems exist in the system; neither should the workarounds recommended be considered exhaustive.

This document is offered to prevent developers from investing time and energy in implementations that are prevented by problems in the system. Developers who encounter further problems in the system are encouraged to submit a clear description of the problem, supported by a program demonstrating the problem. This program should be accompanied by the source code, required data, and any other information needed to reproduce the stated problem. Similarly, any alternative solutions to problems presented here are welcome. Please provide as detailed an explanation as possible.

Problems and solutions should be submitted to one of the following organizations:

From the USA:

Developer Services
Contact: Lex van Sonderen
Philips Interactive Media
11050 Santa Monica Boulevard
Los Angeles, CA 90025
USA
Telephone: +1 310 444 6689
Internet: lex@aimla.com
CompuServe ID: 771552,2204

From Europe or Asia:

Developer Services
Contact: Marc de Krock
Philips Interactive Media Centre
Maastrichterstraat 63
Hasselt B-3500
Belgium
Telephone: +32 11 242167
Internet: mdk@pimc.be

APPENDIX TO TECHNICAL NOTE #88:
LIST OF BUG (REPORTS IN THE MPEG SYSTEM

Version: 1.7 (prerelease)
Date: October 13, 1993

Below the bug number followed by the date of report and the company/department that reported the bug can be found. After that is a description of the bug and a workaround if possible. Finally, the status of the bug and between parentheses is the company that's working on the problem.

002 Nov 2 IMS
System cannot handle User data and Extension data in MPEG video stream.

Workaround: Not possible.
Status: Solution in new hardware only (SPaSE).

028 Nov 27 IMS
Large window repositioning results in artifacts in screen.

Workaround: Unknown.
Status: This is probably the same bug as bug 26. Closely related to bug 26. Both Video controller and decoder use same window registers at same time. Video controller should take over window values one field later than decoder to allow decoder to decode enough macroblocks to display. Driver cannot solve this, microcode modifications will. A workaround is implemented in edition 4 to prevent video controller from displaying undecoded parts. Side effect is that the window may be positioned in the wrong place during one field. A new demo program showed that the workaround is far from correct. Further investigation made clear that no full proof workaround can be made. One option is to mask the problem by hiding the window during some fields. Temporal hide is implemented in version 5. (SPaSE)

045 May 14 IMS
Very rarely the border color changes without sending the mv_borcol command.

Workaround: Overlap border with CD-I plane.
Status: Under investigation. (IMS)

048 June 4 PIMA
Glitches appear when playing in different speeds on a 910 player, not in 220 player. The number of glitches increases with time, sometimes they block up to 16x16 pixels.

Workaround:	Unknown.
Status:	According to submitter it appears to be a heat related problem. PIMA is asked to send cartridge and player serial numbers. Under investigation. (IMS-Hasselt)

049 June 4 PIMA
The first picture of a sequence always results in two PIC-signals.

Workaround:	If on a PIC signal TimeCd and TmpRef are identical to previous values, ignore this signal.
Status:	Testing showed that this statement is only valid when playing in single step or scan speed. The problem is caused by the two step commands that have to be send to the decoder to get the first picture in these speed modes. This results in two picture interrupts but only the first picture is displayed. Under investigation. (IMS)

052 June 4 PIMA
In version 1.0 ROMS, the stepping rate could reach 30 frames per second, in version 1.1 a maximum of 20 frames per second.

Workaround:	An application note will be released that a maximum of 20 frames per second is possible.
Status:	It's not clear why this changed with the new ROMS, but the application should not expect to be able to play the full frame rate in stepping speed. The change due to the ROM update will be investigated.

062 June 1 PIMA
After power on, starting in scanning speed does not offer the first picture which keeps the system in busy mode.

Workaround:	Start play in normal speed with window hidden.
Status:	This bug report slipped through our administration. Under investigation. (IMS)

065 June 16 OptImage
Changing from scan speed to other speeds frequently results in no more decoding.

Workaround:	Not known.
Status:	There seems to be a general problem in starting at random places (packheaders). Another report from Optimage tells about not starting when the first I-picture data is over 15 sectors ahead. Under investigation (IMS).

068 Aug 6 PIMA
When a ss_play is aborted, decoding MPEG video stops and an underflow signal is generated. After that PIC-signals are generated without the display being updated.

Workaround:	Disable PIC signal with mv_trigger, send mv_pause or abort play when the play from disc is stopped.
Status:	When no more data is received, the decoder does not continue decoding. But since it is not paused either, it must redecode B-pictures to allow the application to update the window parameters beyond the borders of the lastly decoded picture. The decoder chip generates a picture interrupt every time the picture is redecoded and the driver sends the PIC-signal. It will be investigated if it is possible to detect whether or not to stop sending PIC-signals in this case.

070 Aug 12 PIMA & Dorking
When (re)starting at random places the system does not always start decoding correctly which results in data overrun on CD.

Workaround:	Not known.
Status:	Under Investigation.

072 Aug 27 IMS
Playing stills always results in a NIS signal before the actual image sizes and picture rate is known. Changing the window after this NIS event has no effect.

Workaround:	Preset image sizes and picture rate with mv_selstrm to prevent the NIS.
Status:	Two problems exist: * When a NIS is detected, the window sizes are checked and updated. This update is not executed until (just before) the new picture is displayed. A newly requested mv_window call is postponed until the previous update is completed. Since a window update during a play becomes effective on the next picture change, this update has no effect unless another still is decoded. * The NIS is generated too soon because of incorrect image sizes shadow register (4 times height iso double height). This problem is fixed in edition 5.

074 Apr 8 PIMA
Sometimes audio stops or does not start at all, mainly on 605 players.

Workaround:	Unknown.
Status:	Initially this bug was reported to Hasselt for being a 605 problem. It has been found in other players too. In the audio driver init routine, an error is found that should make the system not work at all ("does not start at all"). Since it does work sometimes, it means that some kind of interface problem must exist. Further under investigation. (IMS)

075 30 Aug PIMA
When playing for 13 hours 14 minutes, the system 'crashes'. This time coincides with 32 bits worth of 90 kHz ticks.

Workaround:	Unknown.
Status:	In the video driver (and maybe audio driver?) the SCR value is treated as a signed value. When reaching the hex value 7fffffff, no more data is copied to the decoder. To be fixed. (IMS)

076 6 Sep PIMA
Drop frame flag is not provided to the application. This gives problems when calculating with the timecodes in sequences of 29.97 pictures per second.

Workaround:	Do not use sequences with 29.97 picture rate or correct timecodes on specified times.
Status:	It will be investigated if this (and maybe some other) information can be offered to the application without losing backwards compatibility. (IMS)

077 23 Sep IMS
Fast clicking on the forward or reverse button of the control bar of the film engine causes underflow problems, visible as green image distortions.

Workaround:	Unknown.
Status:	Under investigation. (IMS)

078 ?? Sep IMS
Starting play on a sector containing both a sequence end and sequence start code prevents the play from beginning.

Workaround:	Pass byte offset value to sequence start code.
Status:	When an EOS code is in the fifo, the DTS is set invalid. With no valid DTS available, the driver cannot determine when to start decoding. It is not yet clear if starting anyhow (f.i. when buffer full is detected) will make decoding start. Under investigation. (IMS)