



PHILIPS

Philips Interactive Media

Technical Note #103

**Flying Mattes with
CD-i Digital Video**

Jean-Pierre Abello
Advanced Development
Philips Interactive Media

April 18, 1994

Abstract

One of the problems with the Digital Video (DV) plane in CD-i is that it cannot be displayed in front of the base-case planes and, therefore, cannot be used in conjunction with transparency to display portions of a DV screen in the foreground of base-case images. However, an animated DV object or portion of the DV plane can be tracked by a base-case "flying matte" that is updated in synchronization with the DV pictures. This allows you to create the illusion of a DV object moving in front of a base-case plane (which may contain DYUV pictures, run-length animations, etc.).

Publication History:

No revisions.

Note that TN numbers 91 through 100 are reserved for conversion of Philips TSA numbers 1 through 10 to the technical note catalog.

To receive Philips Interactive Media technical notes and other publications, or for more information, please contact the person below who is designated for your area.

From Europe and the Middle East:

Hein Zegers
Developer Support
Philips Interactive Media Centre
Maastrichterstraat 63
B-3500 Hasselt
Belgium
Fax: +32 11 242168
Internet ID: hein@pimc.be

From the USA and Asia:

Lucy Lediaev
Developer Support
Philips Interactive Media
11050 Santa Monica Boulevard
Los Angeles, CA 90025
USA
Fax: +1 310 477 4953
Internet ID: lucy@aimla.com
CompuServe ID: 72056,1130

Copyright © 1994 Philips Interactive Media.
All rights reserved.

This document is not to be duplicated or distributed without written permission from Philips Interactive Media.

Table of Contents

Overview	1
The Flying Matte Technique	1
Playing the Flying Mattes In Synchronization With The DV Plane	2
PIC Signal	2
INTERRUPT Signal	2
Picture Rates and Display Standard	3
Matte LCT Buffering Architecture	3
Timing Diagram	4
Signals Used	4
Operations Performed on Reception of the INTERRUPT Signal	4
Moving The Flying Mattes in Synchronization With The DV Plane	5
DV Position	5
Matte Position	5
Enhanced LCT Buffering Architecture	5
Timing Diagram	6
Experimentation with cdi_dvx	7
Overview	7
Command Line Options	8
References	9

Flying Mattes with CD-i Digital Video

Overview

One of the problems with the Digital Video (DV) plane in CD-i is that it cannot be displayed in front of the base-case planes and, therefore, cannot be used in conjunction with transparency to display portions of a DV screen in the foreground of base-case images.

However, since base-case planes do have access to mattes, the converse mechanism can be used to achieve the same result. An animated DV object or portion of the DV plane can be tracked by a base-case "flying matte" that is updated in synchronization with the DV pictures, with the base-case limitation of a maximum of 8 matte transitions per line.

This allows you to create the illusion of a DV object moving in front of a base-case plane (which may contain DYUV pictures, run-length animations, etc.).

Furthermore, the DV plane and the corresponding flying mattes can be synchronously repositioned on the screen during playback to allow for interesting traveling effects. This even allows for a possible implementation of a "DV sprite."

The above techniques, however, require strict synchronization between the flying mattes and the DV plane at the video scan granularity, to avoid possible misalignments or temporal offsets between the mattes and the DV pictures.

This following section of this document describes a technique for doing flying mattes over Digital Video in CD-i. This technique has been tested and validated with the help of the utility `cdi_dvx`, which is presented on page 9 as an example implementation.

The Flying Matte Technique

The flying matte technique requires strict synchronization between the Digital Video plane and the base-case plane used for the flying mattes.

The general organization of the base-case plane is an Field Control Table (FCT) linked to a main Line Control Table (LCT), which is in turn linked to one of several matte LCTs that contain `cp_matte()` instructions corresponding to successive DV pictures to be tracked by the flying mattes. The matte characteristics can be loaded in memory under various formats, either at set-up time or while the video is played from disc. One possibility is to load preformatted `cp_matte()` instructions to be written into the matte LCTs before the flying matte is displayed. If the vertical size of the mattes does not vary during the play, then a better method is to update only the horizontal offsets of the `cp_matte()` instructions; this is both considerably faster and less memory consuming. For the purpose of this paper, however, only the first technique will be considered, because it is more generic.

Synchronization of the base-case plane with the Digital Video plane to play the flying mattes requires that the main LCT be re-linked to one of the matte LCTs during the field display period just preceding the display of the matte. Because of the current limitations of the CD-i Digital Video cartridge, this is subject to certain difficulties, as explained in the following section.

Moving the flying mattes with the Digital Video plane to create a "DV sprite" is also subject to synchronization problems, and is covered in the section on moving flying mattes on page 5.

Playing the Flying Mattes In Synchronization With The DV Plane

PIC Signal

The current CD-i Digital Video Cartridges provide only one signal, the PIC signal, that can be used for synchronizing the mattes with the DV pictures. Unfortunately, this signal is sent at the end of the display FCT, and, therefore, can be received by the application only after the DV picture starts displaying (see reference 2, IX.4.6.3, IX.8.2.3.1.25). This means that the PIC signal comes too late to use it as a trigger for re-linking the main LCT to the matte LCT containing the corresponding flying matte. The matte would, at the earliest, be displayed during the next display field period, causing a very noticeable temporal offset between the flying mattes and the DV pictures.

For that reason, it is necessary to try to predict when the next DV picture will be displayed, based on the knowledge of the relation between the DV picture rate and the field display rate.

INTERRUPT Signal

The Green Book provides two methods of video scan synchronization. Both methods require a `cp_sig()` instruction to be present in the display LCT to generate a line interrupt when the LCT line is reached. The first method uses `dc_ssig()` to enable a signal to be sent to the application when a line interrupt is detected. The delay in receiving the signal might, however, be significant. The second method uses a "line_event" semaphore that is polled by the function `_ev_wait()` until the event is pulsed to 1. For the purpose of this paper, only the first method has been implemented. The line event method would probably give faster results and could also be used. The fastest method would be to use Jon Piesing's Video Scan Synchronization technique (reference 3), which would lead to the minimum delay in receiving the INTERRUPT Signal. Experimentation with these other techniques is left to the reader.

In my implementation, on reception of the PIC signal, `dc_ssig()` enables reception of the INTERRUPT Signal during the next display field period, which precedes the display field for the next DV picture. When the INTERRUPT Signal is received, one field display period before the display of the DV picture, the main LCT is re-linked to the next matte LCT, so that both the matte and the DV picture will be displayed during the same scan period.

Picture Rates and Display Standard

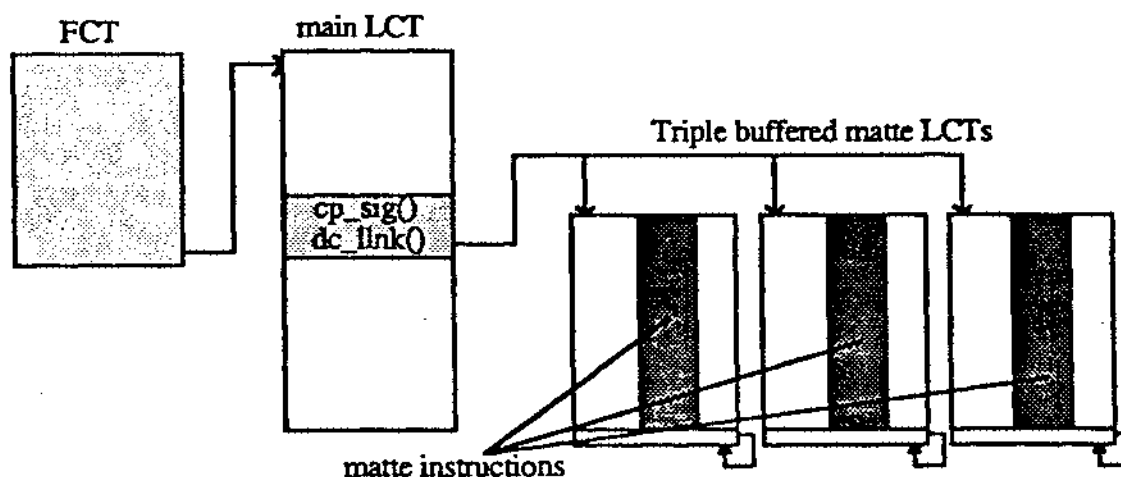
To make this kind of prediction, you must have a steady rate of two display fields per DV picture. There is a difference in frequency rates between the NTSC and PAL display standards. You must encode Digital Video at different picture rates for these different formats. The only picture rates that give exactly two fields per picture are:

NTSC	29.97
PAL	25

Other picture rates do not display two fields per picture on these systems, because the video decoder adds or drops fields to maintain the desired picture rate. Knowledge of the field rate conversion algorithms would be necessary in order to allow you to make this type of prediction based on other picture rates. Unfortunately, these algorithms are not specified in the Green Book, and can vary from decoder to decoder.

Matte LCT Buffering Architecture

The diagram below shows a possible architecture for buffering the matte LCTs for the synchronization of the mattes to the DV pictures:



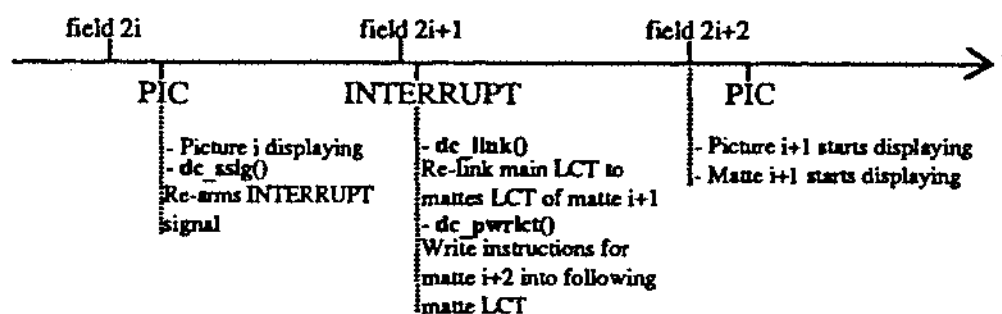
The characteristics of this architecture include:

- An FCT linked to a main LCT, which in turn is linked to one of three triple-buffered matte LCTs containing the matte instructions, as explained in the timing considerations further below.
- A `dc_lnk()` instruction in the main LCT placed on a line number corresponding to the vertical position of the matte on the screen.

- A `cp_sig()` instruction placed on the same line as the `dc_llnk()` instruction in the main LCT, and not above. This is necessary for synchronization to the display field following reception of the PIC signal. This ensures that the INTERRUPT signal is not received before passing the `dc_llnk()` instruction in the main LCT; otherwise, the lower portion of the display might show instructions from the next matte LCT, one field display period too early.
- The height of the matte LCTs are the same as the tallest flying matte in the animation. The ends of the matte LCTs are linked back to themselves to cover the rest of the screen below the mattes.

Timing Diagram

The timing diagram below shows the relative time positions of the signals and the various operations that have to be performed on their reception:



Signals Used

- The PIC signal is received during the first field display period for DV picture *i*, and is used primarily as a time-base for re-enabling the INTERRUPT signal with `dc_ssig()`.
- The INTERRUPT signal is received during the following field display period, slightly after the main LCT line containing the `cp_sig()` instruction is reached, and before the next DV picture, *i+1*, is displayed.

Operations Performed on Reception of the INTERRUPT Signal

- The main LCT is re-linked with `dc_llnk()` to the next triple-buffered matte LCT, which already contains matte instructions corresponding to DV picture *i+1*, so that it will be displayed during the next field display period at the same time as DV picture *i+1*.
- The matte instructions corresponding to the DV picture *i+2* are written into the corresponding matte LCT with `dc_pwrkt()`.

Depending on the size of the mattes (number of lines and number of `cp_matte()` instructions per line), `dc_pwrkt()` can take more or less time to complete, and in most cases, may end after the next field starts displaying, especially if the matte position is close to the bottom of the screen. For that reason, the matte LCTs must be triple-buffered. If they were just double-buffered, then the next matte LCT would start displaying before `dc_pwrkt()` is finished writing matte instructions

into it. This could cause incorrect matte data to be displayed in the lower section of the display. By triple-buffering the matte LCTs, `dc_pwrlct()` has at least 2 field display periods available in which to write the matte instructions corresponding to DV picture `i+2`; this is enough even for large mattes.

Moving The Flying Mattes in Synchronization With The DV Plane.

In addition to the above mechanisms, it is also possible to change the position of the DV pictures during replay. To allow the flying mattes to follow the movement of the DV pictures, the following mechanisms must be added to the above technique.

DV Position

The position where the pictures are displayed in the DV plane can be changed with `mv_pos()`, `mv_org()`, or `mv_window()`. My implementation uses `mv_pos()` because of its simplicity.

To allow for an accurate prediction of when the DV plane repositioning will take place, the scroll flag in `mv_pos()` must be set to 0. This guarantees that the position can change only on the next DV picture. A scroll flag of 1 in `mv_pos()` would allow the position to change either on the next display field or the next DV picture, depending on the amplitude of the movement. Thus, in some cases, the position of the DV picture, rather than the position of the mattes, could be changed during a different field display period.

Matte Position

The horizontal position of the mattes can be changed by adding a low-resolution offset to the x position in the `cp_matte()` instructions written by `dc_pwrlct()`.

The vertical position can be changed by changing the physical line number of the `dc_llnk()` instruction in the main LCT.

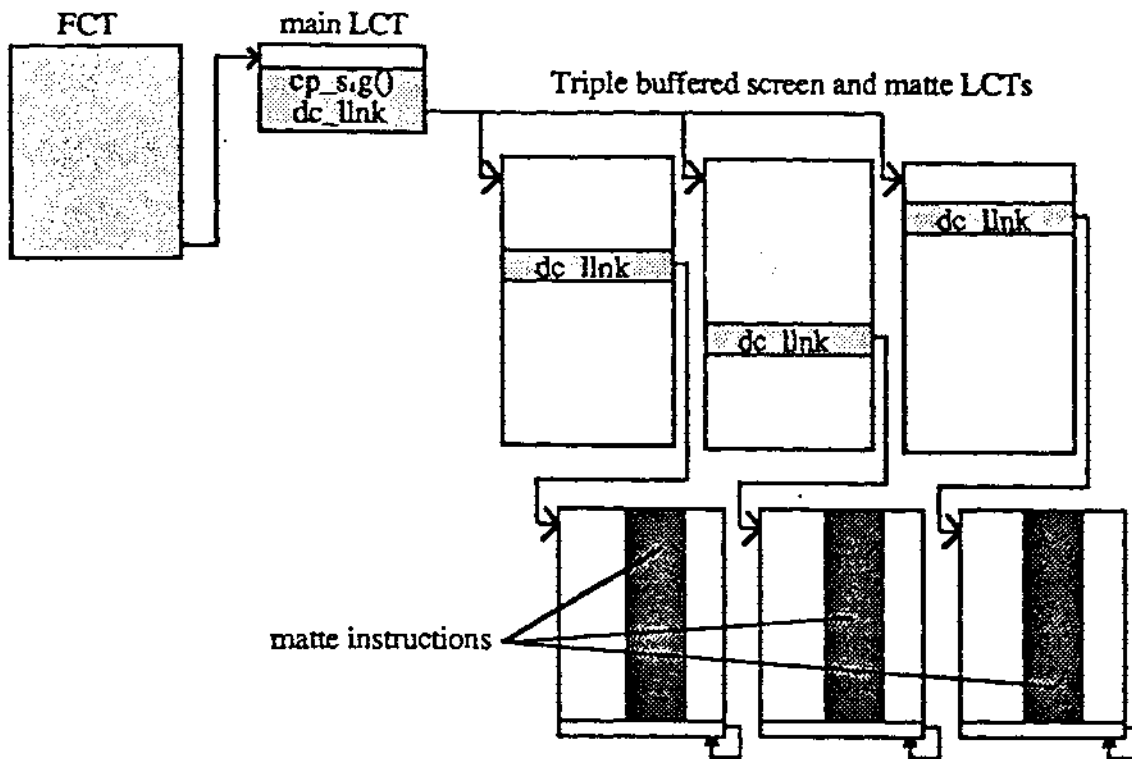
Since the line containing the `dc_llnk()` instruction in the main LCT can change on every DV picture, the INTERRUPT signal may now be received at variable times. It is possible to avoid that by also triple-buffering the main LCT, as explained below.

Enhanced LCT Buffering Architecture

The `cp_sig()` instruction can be placed in a small main LCT of minimal size, used only for linking to one of three screen LCTs containing the actual `dc_llnk()` instruction linking to the corresponding matte LCT.

It is possible to do without the small main LCT, but the FCT would have to be re-linked periodically to the screen LCTs using `dc_flnk()`, which is much slower than `dc_llnk()`. Therefore, it is faster to use the implementation mentioned above.

It is also possible to double-buffer the main LCT, but doing so adds complexity to the mechanism that re-links screen LCTs to the triple-buffered matte LCTs. Thus, it is simpler and faster to triple-buffer the main LCT in the same fashion as the matte LCTs.

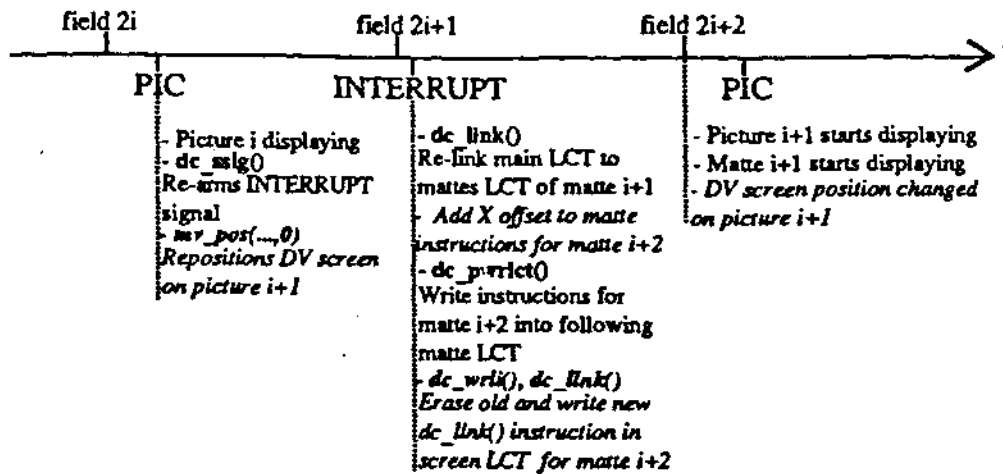


The characteristics of this architecture include:

- The height of the main LCT should be at least 2 physical lines to allow linking to a screen LCT. It should also be as short as possible, to allow the flying matte to be positioned as close as possible to the top of the display.
- The height of the screen LCTs is required to be at least the height of the screen minus the height of the two lines of the main LCT and minus the height of the matte LCTs.
- The line number of the `dc_lnk()` instruction inside the screen LCTs now determines the vertical position of the flying mattes on the screen.
- Each time the line position of a `dc_lnk()` instruction is changed inside a screen LCT, it is necessary to erase the old `dc_lnk()` instruction. This can block the flying matte from traveling down the screen, as shown in the timing diagram below.

Timing Diagram

The following enhanced timing diagram repeats everything present in the old diagram plus the new operations needed to make the flying mattes travel on the display in unison with the DV pictures.



- On reception of the PIC signal, `mv_pos()` is called with a scroll flag of 0 for positioning picture $i+1$ on the DV screen. This change will happen on the field $2i+2$, regardless of the amplitude of the movement.

- On reception of the INTERRUPT signal, the horizontal position of the matte can be changed by adding an x coordinate offset to the `cp_matte()` instructions. The vertical position can be changed by moving the `dc_llnk()` instruction to the correct y coordinate line inside the screen LCT. Since this is done for the matte corresponding to picture $i+2$, and not to picture $i+1$ which will be shown next, the matte position coordinates also have to be double buffered. This will prevent `mv_pos()` from using the wrong coordinates for the next picture. That way, the most recent set of coordinates is used for the matte and DV picture, $i+2$, while the previous set of coordinates is used for the matte and DV picture, $i+1$, to be shown next.

- On the display field following reception of the INTERRUPT signal, DV picture $i+1$ is shown at the new correct position in the DV plane, and the corresponding matte is displayed at the same place in the base-case plane.

Experimentation with `cdi_dvx`

Overview

The program `cdi_dvx` is a small utility designed for playing and experimenting with flying mattes in conjunction with linear or interactive Digital Video streams.

In version 1.0 of `cdi_dvx`, the flying mattes are displayed through a DYUV image in plane B over a DV stream playing from disc or a seamless branching memory DV loop. The matte instructions are loaded from a matte file, and are limited to a maximum of two `cp_matte()` instructions per LCT line. The DV screen and the flying mattes can be moved using the remote controller or a mouse.

You must generate pre-formatted matte instructions for all frames based on your knowledge of the matte characteristics for each frame. These instructions make up the matte file.

4 and 5, cited below, and go beyond the purpose of this paper.

The program `cdi_dvx`, including its source code, OS-9 executable and sample assets, is available from PIMA Developer's Services at (310) 444-6158.

Command Line Options

Here is a printout of the command-line help page for `cdi_dvx v1.0`:

```
Digital Video Explorer v1.0 by Jean-Pierre Abelle - March 31, 1994
Copyright 1994, Philips Interactive Media of America.
Usage: cdi_dvx <options> <mpeg_rtf>
<options> can be any combination of:
  -c cdfs_channel          Cdfs channel for loading data
  -s sector_offset        Cdfs sector offset for lseek()
  -v video_stream          Stream number for mpeg video
  -a audio_stream          Stream number for mpeg audio
  -bv video_byte_offset   Video entry point byte offset
  -ba audio_byte_offset   Audio entry point byte offset
  -pv video_pcl_buffers   Nbr of video pcl buffers (min 1)
  -pa audio_pcl_buffers   Nbr of audio pcl buffers (min 1)
  -mv video_buffers_mem_color A = PlaneA, B = PlaneB, S = SysRam
  -ma audio_buffers_mem_color A = PlaneA, B = PlaneB, S = SysRam
  -o x_ory_y_ory          DV DCM x and y origin offsets from (0,0)
  -u                      Unsyncronized audio/video
  -M scroll_flag           mv_pos() moves the DV window with the mouse
  -t                      Report timing information
  -DV video_delta_value   Video delta value in units of 45KHz.
  -DA audio_delta_value   Audio delta value in units of 45KHz.
  -dv video_intro.std video_body.std Video intro and body sink std files
  -da audio_intro.std audio_body.std Audio intro and body sink std files
  -J                      Use mv_jump and ma_jump instead of iso patch
  -f matte.ops width height frames Matte opcode file, width, height, and frames
  -i image.dyu            Flying Matte DYUV PAL background image
  -V                      Verbose mode (status, signals, etc)
  -VV                     Very Verbose mode (maximum feedback)
  -h                      Help on Seamless Loops and Flying Mattes
  <mpeg_rtf> is either of the following:
    file.rtf              To play a RTF straight off the disc
    intro_file.rtf body_file.rtf To seamlessly loop sinked RTFs from ram
```

Example Command Lines

For playing flying mattes over a linear DV stream:

```
cdi_dvx -V -t stream.rtf -i image.dyu -f loop.ops 96 96 90
```

For playing and moving flying mattes over a linear DV stream:

```
cdi_dvx -V -t stream.rtf -i image.dyu -f loop.ops 96 96 90
```

For playing flying mattes over a seamless branching DV memory loop:

```
cdi_dvx -V -t -dv intro.std loop.std intro.rtf loop.rtf -i image.dyu -f loop.ops 96 96
90
```

For playing and moving flying mattes over a seamless branching DV loop:

```
cdi_dvx -V -t -M 0 -dv intro.std loop.std intro.rtf loop.rtf -i image.dyu -f loop.ops 96 96
90
```

Since there is a possible, player dependent, one-pixel horizontal offset between the DV plane and the base-case planes, sometimes it may be necessary to adjust the alignment of these planes with the `-o` option, as shown below:

```
cdi_dvx -V -t -dv intro.std loop.std intro.rtf loop.rtf -i image.dyu -f loop.ops 96 96 90 -o
2 0
```

1. "Information technology - Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbit/s - Part 1: Systems," ISO/IEC 11172-1, First Edition, August 1, 1993.
2. "CD-I Full Functional Specification: Full Motion Extension," Philips Consumer Electronics BV, March 1993.
3. "A New Method for Video Scan Synchronization in CD-i," Technical Note #89, Jon Piesing, Philips Research Labs, Redhill, January 4, 1994.
4. "Seamless Branching with CD-i Digital Video," Philips Interactive Media Technical Note #104, Jean-Pierre Abello, Advanced Development, Philips Interactive Media of America, to be published May, 1994.
5. "sink Users Guide", PRL Redhill Technical Note No. 3201, Edition 0.91, sink 0.15, Simon Blanchard, Philips Research Laboratories, February 1994.