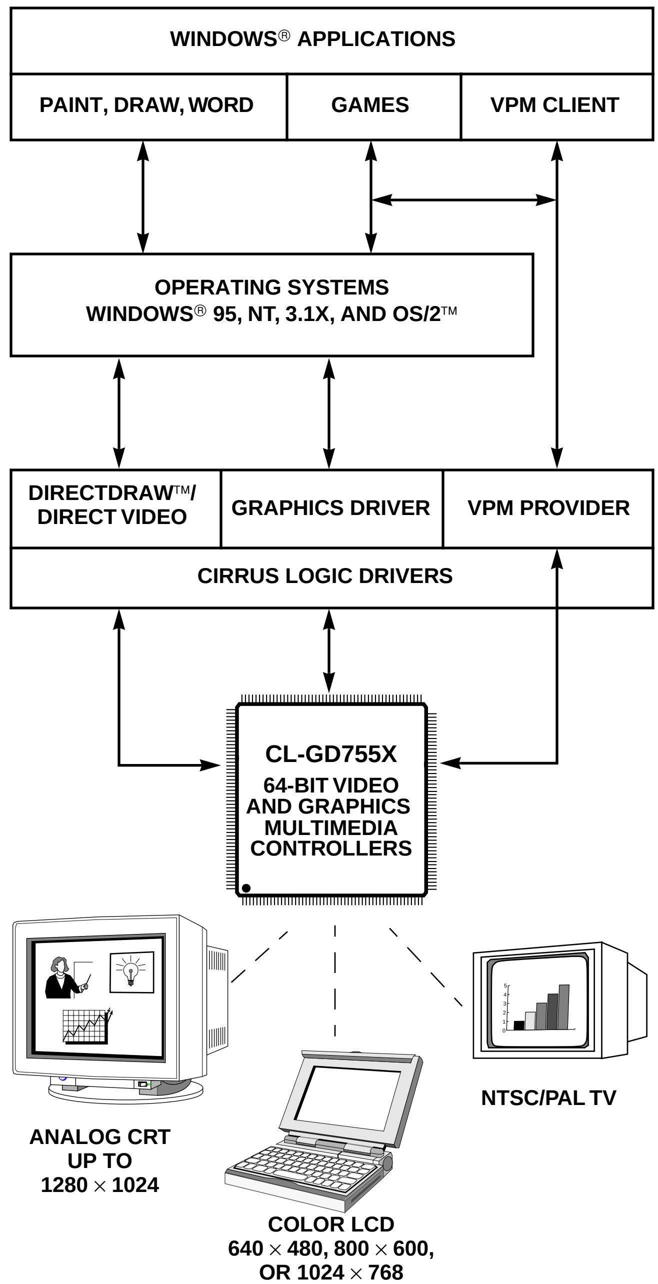
CL-GD7555 and CL-GD7556

Advance Software Reference Manual

64-Bit Video and Graphics LCD/CRT Controllers Multimedia Software Drivers and Utilities for Microsoft® 95, NT, v3.1x, and OS/2™

SYSTEM DEVELOPER TOOLS

- **Drivers**
  - — Microsoft® Windows® 95
  - — DirectDraw™ for Windows® 95 with DirectVideo™ and Direct 3D™
  - — Microsoft Windows NT™ 3.51, 4.0
  - — Microsoft® Windows™ 3.1x
  - — Microsoft®/Intel® DCI™
  - — OS/2® v3.0
  - — Display Switching – hot key support
  - — Cirrus Logic® VPM (video port manager) provider
  - — AutoCAD® v12 and v13 for Windows® with Autoshade® and 3D Studio®
  - — AutoCAD® v11 through v13 for DOS with Autoshade® and 3D Studio®
- **VGA BIOS**
  - — 48-Kbyte Standard VGA and Extended VGA BIOS VESA® VBE (VGA BIOS extensions) 1.2, DDC-2B, and PM (power management)
  - — TV-out support for internal and external TV encoder
  - — VBE 1.2 (CL-GD7555) and VBE 2.0 (CL-GD7556)
  - — TV-out support for internal TV encoder (CL-GD7555 only)
  - — Supports PCI subsystem ID of PC 97 requirement (CL-GD7556 only)
- **Developer Utilities**
  - — OEMSI — DOS®-based BIOS-customization utility
  - — REGS7555 — Windows®-based register editor
  - — Gamma correction utility for flat panel or CRT
- **End User Utilities**
  - — Windows® 95 configuration utilities
  - — WinMode — Windows® 3.1 and OS/2™ configuration utility
  - — CLMode — DOS®-based configuration utility
  - — Color and brightness adjustment utility for video window and background graphics display



SOFTWARE SUPPORT

Operating System and Application Software Drivers

Software Drivers[a]	Resolution Supported	Number of Colors
Microsoft® Windows® v3.1x, Microsoft® Windows® 95™, and Microsoft® Windows NT™ v3.51, v4.0	SimulSCAN™ 640 × 480, 800 × 600 SimulSCAN™ 1024 × 768 CRT Only 1280 × 1024	16.8M 64K 256
Microsoft®/Intel® DCI™	SimulSCAN™ 640 × 480, 800 × 600, 1024 × 768	16.8M video
DirectDraw™ (with DirectVideo™ and Direct 3D™) for Windows® 95 (Resolutions supported in addition to the standard set listed in the row above.)	SimulSCAN™ 320 × 200, 320 × 240, 512 × 384 and 640 × 400	16.8M 16.8M
Cirrus Logic VPM (Video Port Manager) for DCI and DirectDraw™/DirectVideo™/Direct 3D™	SimulSCAN™ 640 × 480, 800 × 600 and 1024 × 768	256 graphics with 16.8M video
OS/2™ v3.0	SimulSCAN™ 640 × 480, 800 × 600 SimulSCAN™ 1024 × 768 CRT Only 1280 × 1024	16.8M 64K 256
AutoCAD® v11 and v13 (DOS® and Windows®) with Autoshade® and 3D Studio®	SimulSCAN™ 640 × 480, 800 × 600 SimulSCAN™ 1024 × 768 CRT Only 1280 × 1024	16.8M 64K 256

[a] Driver support for additional applications is provided by independent software vendors either with specific drivers or through VESA mode support. For more information concerning independent driver support, contact the software manufacturer.

VGA BIOS

Feature	*Benefit*
■ 48-Kbyte BIOS	❒ Optimum performance with VGA and VESA® extended mode support. ❒ Design options for the best combination of performance and functionality.
■ VESA® VBE (VGA BIOS extensions) v1.2, DDC-2B, and power management	❒ Compatible with industry standards for extended mode support beyond VGA, intelligent monitor sensing, and power-management control.

Software Utilities

Utility	Function
Windows® 95 Utilities	Windows® 95 based utilities for graphic mode and display type configurations (includes foreign language translations)
WinMode CLMode	Windows® and DOS utilities for graphic mode and display type configurations (includes foreign language translations, selectable from within the utility)
Auto-Resolution Switching Pan and Scroll	Windows® 3.1x applications for automatically switching display resolutions (for example, from a 1024 × 768 CRT to lower-resolution 640 × 480 or 800 × 600 LCDs) without relaunching Windows, or for creating a larger-size virtual window area in display memory which is viewed by panning and scrolling a smaller-size window within that virtual area
OEMSI	BIOS-customization utility for OEM development use
REG755X Register Editor	Windows-based VGA register viewer/editor and V-Port/Video window-display-configuration utility for OEM development use
Color Correction Utility	OEM utility sets the values in the Graphics CLUT (Color Lookup Table) to compensate for the color/brightness variations of LCD flat panels or CRT monitors
Video-window Color and Brightness Adjustment	End-user utility adjusts for the color difference of the video source, as well as the brightness in the resulting video window.

TABLE OF CONTENTS

1 TECHNICAL REFERENCE ROADMAP
(Guide to Technical Design and Support Information)

The purpose of this chapter is to give an overview of the CL-GD755X Cirrus Logic Video and Graphics LCD/CRT controller software reference, hardware reference and support material, and to provide pointers to the source of applicable information.

1.1 Software Reference Manual

This document contains the detailed description of the following software (also, see Table of Contents):

- Drivers
  - — Microsoft® Windows® 95™
    - — DirectDraw™ for Windows® 95™ with DirectVideo™ and Direct 3D™ support
  - — Microsoft® Windows NT™ 3.51 and 4.0
  - — Microsoft® Windows® v3.1x
    - — Microsoft®/Intel® DCI™
  - — Display Switching – Hot key support
  - — OS/2® v1.2, WARP v3.0
  - — Cirrus Logic VPM (Video Port Manager) Provider
  - — AutoCAD® v12 and v13 for Windows
    - — with Autoshade® and 3D Studio®
  - — AutoCAD® v10 – v13 for DOS
    - — with Autoshade® and 3D Studio®
- VGA BIOS Specification
  - — 48-Kbyte VGA and Extended VGA BIOS
    VESA® VBE (VGA BIOS extensions) 1.2,
    DDC-2B, and PM (power management)
- Developer Utilities
  - — OEMSI – DOS-based BIOS-customization utility
  - — REGS755X – Windows-based register editor
  - — Gamma correction utility to program graphics CLUT for flat panel or CRT variables
- End User Utilities
  - — Windows 95 configuration utilities
  - — WinMode – Windows 3.1x configuration utility
  - — CLMode – DOS-based configuration utility
  - — Color and brightness adjustment utility for video window

1.2 Hardware Reference Manual

The Hardware Reference Manual contains the detailed description of CL-GD755X. It covers:

- Ordering information

- Physical package dimensions
- dc and ac electrical specifications
- Detailed Register descriptions with bit-by-bit breakdown
- Complete list of VGA and Extended-VGA Modes supported
- Flat-panel interface description and programming requirements
- PCI bus interface definitions
- Display-memory description and configuration requirements

1.3 Applications Book

The Applications Book contains hardware related examples of typical designs; such as:

- Typical pc board designs
- Special analog and digital circuit design and layout techniques
- Display memory design and layout techniques
- Panel Interface Guide (PIG) with detailed description to most available DSTN and TFT LCD panels
- Testing information and procedures
- Application Alerts – Special device or circuit conditions that have been found to effect some applications

1.4 Release Notes

The Release Notes are shipped with the software and contain:

- License requirements
- Installation instructions
- Bugs and problems corrected since last release
- Any reported, but not yet fixed, bugs or problems
- Product Deficiency Report form – mechanism for reporting problems to the factory

1.5 Applications/Engineering Bulletin Board System

The Applications/Engineering BBS is available to all Cirrus Logic customers. It is accessible in the USA at;

(USA) 510-440-9080 or 510-440-0394
Modem settings: 14,400 baud, NO parity, 8 Data bits, 1 Stop bit.
The software is available in the conferences on the specific subjects.

Before downloading software, the user must join the conference(s) for the product(s) that are of interest. The BBS provides a means to:

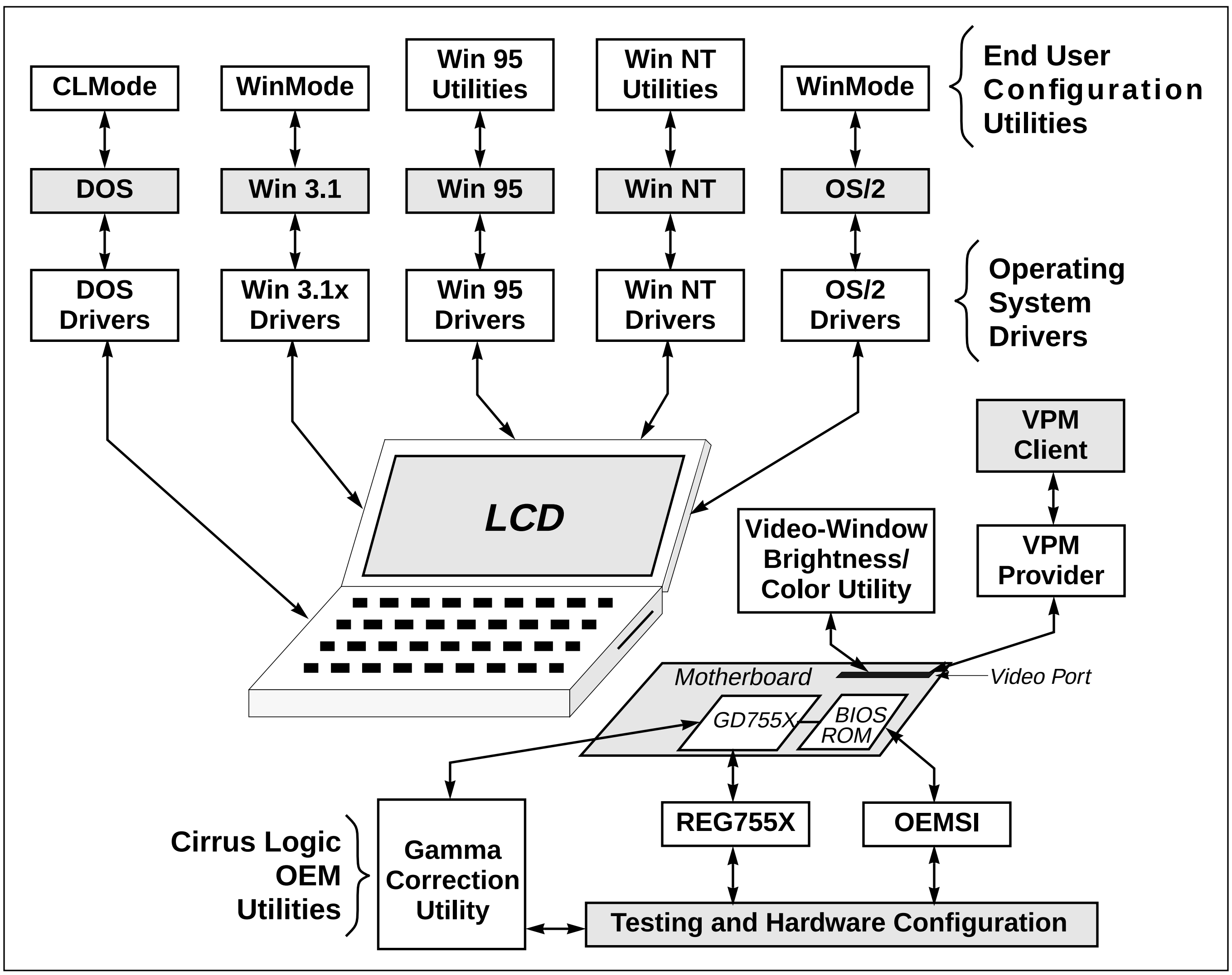
- Download new versions of the available software
- Get access to continuously updated information, like the Panel Interface Guide

1.6 Driver and Utilities System Interface

The Cirrus Logic operating system drivers and utilities supply the interface for the OEM designers and the end users to get access to the flat panel and CRT monitor. The Cirrus Logic software drivers are installed on the hard disk with the operating system to provide access for the operating system to the CL-GD755X LCD/CRT controller. The VPM Provider was created as a standard interface to the video port to simplify video-client applications. The specific capabilities of these software drivers is described in Chapter 6.

The end-user configuration utilities were created by Cirrus Logic to be supplied with the computer system. These utilities provide the interface to the operating system, so the user can change the flat panel and CRT monitor style and configuration. Such things as display resolution, color depth, simultaneous flat panel and CRT, or TV and CRT are selectable by the user. Details for these utilities are contained in Chapters 7 through 10.

The OEM utilities were created to give the developers an easy to use interface to the CL-GD755X LCD/CRT controller hardware. The use and syntax for these utilities is contained in Chapters 11 through 13. The VGA BIOS, defined in detail in Chapter 4, can be modified by the system integrator by using the OEMSI utility defined in Chapter 5.



1.7 Where to Look for Design and Support Information

For basic CL-GD755X initialization, operating capabilities and options, go to:

- *Software Reference Manual, Chapter 4,* The CL-GD755X VGA BIOS.

For installation information, go to:

- *Software Reference Manual, Chapter 2*
- *Release Notes and Readme files,* supplied with the software

For a brief description of the available software, go to:

- *Software Reference Manual, Chapter 3*

For the latest flat-panel support and interface information, go to:

- *Applications Manual,* Panel Interface Guide
- Cirrus Logic, *Bulletin Board System (BBS)*, Panel Interface Guide

For dc and ac electrical specifications, go to:

- *Hardware Reference Manual, Chapter 13*

For a list of terms, acronyms go to:

- *Hardware Reference Manual, GLOSSARY*
- *Software Reference Manual, GLOSSARY*

For information on the chip and system manufacturing tests, go to:

- *Hardware Reference Manual, Appendix D,* Signature Generator setup and testing procedure
- *Hardware Reference Manual, Appendix E,* Pin-Scan testing

For package and pin-layout information, go to:

- *Hardware Reference Manual, Chapter 1,* Pin and ball-grid diagrams for all packages
- *Hardware Reference Manual, Chapter 2,* Pin (ball-grid) descriptions and functions
- *Hardware Reference Manual, Chapter 14,* Typical package diagrams and dimensions for all packages
- *Hardware Reference Manual, Appendix J,* Ball-Grid Array contact configuration
- Cirrus Logic, *Package Information Guide,* for the latest package diagrams and dimensions for all packages

For troubleshooting information, go to:

- *Applications Manual,*

2 GETTING STARTED

2.1 Software Procurement and Licensing

The Cirrus Logic LCD/CRT Graphics-controller development and utilities software is distributed to our OEM customers as part of the CL-GD755X product procurement. Initial copies of the software must be acquired through the Cirrus Logic Sales or Graphics Marketing departments. Subsequent releases and fixes to the software are available through the same sources or through the BBS system. Registered user can access the BBS system at:

(USA) 510-440-9080 or 510-440-0394

Modem settings: 14,400 baud, NO parity, 8 Data bits, 1 Stop bit.

The software is available in the conferences on the specific subjects.

A License for Distribution of Executable Software must be acquired from Cirrus Logic to distribute the display drivers and utility software to end users. Such license is required for each Cirrus Logic LCD/CRT Controller product with which the software is shipped. General modifications, enhancements, updates, and new releases of the software for each product are deemed included within the definition "Licensed Software".

The software may be used freely by the original owner of the Drivers and Utilities package shipped by Cirrus Logic, but no duplication, distribution or sub-licensing is authorized without a formally executed license.

Blank copies of the Cirrus Logic "License for Distribution of Executable Software" are shipped with every software release package. Copies of the license agreement are also available from the Cirrus Logic Sales offices.

2.2 Hardware and Software System Configuration

The minimum system configuration needed to effectively use the Cirrus Logic LCD/CRT Graphics-controller development and utilities software is:

'486 or Pentium based Computer with 32-bit PCI bus interface

640 x 480, 800 x 600 or 1024 x 768, Color, Dual-scan (DSTN) or TFT flat panel

VGA Multisync CRT color monitor is optional

Cirrus Logic LCD/CRT Graphics Controller with the VGA BIOS in EPROM

1 Mbyte or 2 Mbytes of Display Memory (64-bits wide)

Hard disk with a minimum of 2 Mbyte allocated for the graphics software and utilities

MS-DOS or PC-DOS

Windows 3.1x, Windows 95, Windows NT 3.51, or OS/2 operating system

2.2.1 VGA BIOS Requirements

The VGA BIOS, described in Chapter 4, is specifically configured for the CL-GD755X. The VGA BIOS requires 48 Kbytes of EPROM space. The only variations that are allowed are defined in Chapter 4, Section 4.5, "Configuring the CL-GD755X VGA BIOS". The software drivers use this BIOS to configure and run the displays under the appropriate operating systems.

2.3 Installation Guide

The detailed installation instructions for each display driver are included with the software release. Follow the instructions carefully to be sure that each display driver is correctly installed. Read Me files are included with all software releases which contain last-minute changes and updates to the other files.

▲ WARNING Before beginning – Not all video modes are available on all systems. When an extended mode driver is installed for a video mode that is not available, the application program does not function properly. There are a number of things that determine the list of available video modes. Some of these include the current monitor or flat-panel type, the amount of installed memory, and the revision of the LCD/CRT graphics controller. To determine which modes are available before beginning the driver installation, it is recommended that the user run the CLMode program and examine the list of available video modes. The CLMode program is described in Chapter 8 of this manual.

During the installation phase, many products from other companies are referenced. Be aware that other brand and product names are the trademarks or registered trademarks of their respective companies.

2.3.1 Release Notes

Release Notes are included with every software release. These notes provide an up-to-date status of the software by describing:

- The version(s) on the CL-GD755X chip that are supported by the latest version of the software
- The software changes since the last release
- Problems or bugs that have been fixed since last release
- Known, or reported and verified, problems that have not yet been fixed.

2.3.2 TSRFONT Driver

Some DOS application programs bypass the BIOS and directly draw characters to the screen. Programs which directly draw characters using the 8 x 14 font from the BIOS will appear to be writing incorrect data to the screen. Typically this can occur in programs which offer a selection to use a graphics 25 or 34 line display mode. Other programs may appear to cut off the descenders of characters like "y" and "j".

Running the TSRFONT driver will make a full 8 x 14 size character set available to these programs, and should correct display errors discussed above which were observed in these programs.

To run the TSRFONT driver, type TSRFONT [Enter]. If you want to run it automatically when you turn on your computer, add it to your AUTOEXEC.BAT file.

3 SOFTWARE DESCRIPTIONS

The purpose of this chapter is to list the entire suite of Cirrus Logic software for the CL-GD755X and briefly outline the function of each. (The software is discussed in more detail in the cross-referenced chapters.)

VGA BIOS (Refer to Chapter 4)

- The VGA BIOS (basic input / output system) provides the firmware that controls the basic functions of the CL-GD755X LCD/CRT graphics controller. The utilities and applications make calls to the BIOS to make changes in the display configurations.
- The VGA BIOS includes:
  - Standard IBM VGA BIOS support
  - Extended functions to support panel sizes up to 800 x 600 with 16M colors and 1024 x 768 with 64K colors, and CRT sizes up to 1280 x 1024 with 256 colors.
  - SimulSCAN, simultaneous flat panel and CRT operation
  - VESA compatible power saving modes
  - Hot-key and change-on-the-fly support

OEMSI Utility (Refer to Chapter 5)

- OEMSI is an editor for the VGA BIOS. It allows the OEM manufacturer to download the contents of the BIOS EPROM, make changes to the code, and rewrite the EPROM.
- The OEMSI specification in Chapter 5 includes a sample printout of the BIOS showing the organization and definitions used in the BIOS tables.

Drivers (Refer to Chapter 6)

- **Windows 3.1x**
  - These drivers work with the Windows 3.1x operating system to allow the user to configure the display properties controlled by the CL-GD755X.
  - These drivers include support for:
    - Extended functions to support panel sizes up to 800 x 600 with 16M colors, 1024 x 768 with 64K colors, and CRTs up to 1280 x 1024 with 256 colors.
    - Microsoft/Intel DCI provider support
    - SimulSCAN, simultaneous flat panel and CRT operation
    - Video window up to full screen size and with 16M colors
    - Auto resolution support for attached CRT monitors
    - Display switching on-the-fly
    - Changing resolution on-the-fly
    - Hot-key support
  - The WinMode utility is used to access the drivers to re-configure the display.
- **Windows 95**
  - These drivers work with the Windows 95 operating system to allow the user to configure the display properties controlled by the CL-GD755X.
  - These drivers include support for:
    - Extended functions to support panel sizes up to 800 x 600 with 16M colors, 1024 x 768 with 64K colors, and CRTs up to 1280 x 1024 with 256 colors.

- — DirectDraw with DirectVideo and Direct 3D support
- — SimulSCAN, simultaneous flat panel and CRT operation
- — Video window up to full screen size and with 16M colors
- — Auto resolution support for attached CRT monitors
- — Display switching on-the-fly
- — Changing resolution on-the-fly
- — Hot-key support

— The Windows 95 utilities are used to access the drivers and re-configure the display.

- **Windows NT 3.51, 4.0**
  - — These drivers work with the Windows NT operating system to allow the user to configure the display properties controlled by the CL-GD755X.
  - — These drivers include support for:
    - — Extended functions to support panel sizes up to 800 x 600 with 16M colors, 1024 x 768 with 64K colors, and CRTs up to 1280 x 1024 with 256 colors.
    - — SimulSCAN, simultaneous flat panel and CRT operation
    - — Video window up to full screen size and with 16M colors
    - — Auto resolution support for attached CRT monitors
    - — Hot-key support
  - — The standard Windows NT 3.51, 4.0 utilities are used to access the drivers and re-configure the display.
- **VPM (Video Port Manager) Provider**
  - — The VPM is an API, developed by Cirrus Logic, which allows an application (client) to easily manage any video port and its associated display hardware without knowledge of vendor-specific details.
  - — The VPM interface is designed to be a non-hardware-specific interface for controlling a video data port tied directly to a graphics frame buffer or display.
  - — The VPM provides information about the capabilities and limitations of the hardware in a standard manner to enable a VPM client to utilize the hardware to its maximum extent without directly communicating with the hardware.
- **OS/2 v3.0**
  - — These drivers work with the OS/2 operating system to allow the user to configure the display properties controlled by the CL-GD755X.
  - — These drivers include support for:
    - — Extended functions to support panel sizes up to 800 x 600 with 16M colors, 1024 x 768 with 64K colors, and CRTs up to 1280 x 1024 with 256 colors.
    - — SimulSCAN, simultaneous flat panel and CRT operation
    - — Video window up to full screen size and with 16M colors
    - — Auto resolution support for attached CRT monitors
    - — Hot-key support
- **AutoCAD** v12, v13 for Windows
  - — Cirrus Logic provides support for the Windows versions of AutoCAD through the standard Windows 3.1x and Windows 95 drivers.
  - — These drivers include support for AutoShade and 3D-Studio
- **AutoCAD** v10 - v13 for DOS
  - — These drivers were created by Panacea, Inc. for Cirrus Logic to support the DOS versions of AutoCAD.
  - — These drivers include support for AutoShade and 3D-Studio

WinMode Utility for Windows 3.1x (Refer to Chapter 7)

- WinMode is a Windows-based utility that configures the graphics system for Windows 3.1x.
- WinMode has support for 18 languages. (See Section 7.2, Multiple Language Support)
- WinMode allows the user to:
  - — Select a flat panel, CRT monitor, SimulSCAN (both panel and CRT), or TV (with external TV codec)
  - — Control the monitor refresh rates, resolution, number of available colors, large or normal font sizes and font-cache size.
  - — Run the flat panel at higher than normal resolutions by panning and scrolling the display to view off-screen data

CLMode Utility (Refer to Chapter 8)

- CLMode is a DOS-based utility that configures the graphics system. Since it runs under DOS, it can be used to pre-configure or test the Windows operating environments prior to installation of the Windows drivers.
- CLMode has support for 18 languages. (See Section 8.2, Multiple Language Support)
- CLMode allows the user to:
  - — Select a flat panel, CRT monitor, SimulSCAN (both panel and CRT), or TV (with external TV codec)
  - — Control the CRT monitor refresh rates, resolution, and number of available colors.
  - — Control the flat-panel resolution, and number of available colors.
  - — Preview the chosen modes for size, color, and timing compatibility with the displays

Windows 95 Utilities (Refer to Chapter 9)

- These are Windows-based utilities that configure the graphics system for Windows 95.
- These utilities have support for 18 languages.
- These utilities allow the user to:
  - — Select a flat panel, CRT monitor, SimulSCAN (both panel and CRT), or TV (with external TV codec)
  - — Control the monitor refresh rates, resolution, number of available colors, large or normal font sizes and font-cache size.
  - — Control the flat-panel resolution, and number of available colors.

Windows NT 3.51 Utilities

- There are no special utilities for Windows NT 3.51; the standard utilities perform the necessary adjustments.

Regs755X Utility (Refer to Chapter 11)

- A Windows based utility used by the designers to view and edit the register contents of the CL-GD755X.
- This register editor gives the user easy access to VGA, MVA (Motion Video), and Extension registers.
- It provides access to YUV, AccuPak, and DYUV compression options.
- It provides an Auto Adjust option to adjust all effected registers, when a new function is selected.

Brightness/color Adjustment Utility for the Video Window (Refer to Chapter 12)

- This utility allows the end user to adjust the brightness and color quality of the video window.

Graphics Brightness/color Correction Utility (Refer to Chapter 12)

- This utility allows the OEM computer designers to modify the CL-GD755X's graphics CLUT (color look-up table) to compensate for the brightness and color variation between panel types and panel manufacturers. The CLUT is loaded by the drivers during initialization; it is normally not changed by the end users.

Appendixes

A BitBLT Engine

— This appendix describes the operation and programming of the BitBLT engine.

— It explains how to use the BitBLT engine for:

  — Moving or duplicating areas on the display
  — Color expansion of graphics data
  — Color expansion of text data in graphics write modes
  — Pattern fills and clipping
  — System to screen BitBLTs

B Hardware Cursor

— This appendix describes the operation of the 32 (32 x 32) or 16 (64 x 64) hardware cursors (mouse pointers). It explains how to select one of hardware cursors, store the cursors, set the colors, and move the cursors.

— The hardware cursor eliminates the need for application software to save and restore the screen data as the on-screen cursor position changes.

— The hardware cursor has the highest display priority in the display subsystem.

C Hardware Icon

— This appendix describes the operation of the hardware-driven icon window. It explains how to select one of hardware icons, store the icons, set the colors, and move the icons.

— The hardware icon eliminates the need for application software to save and restore the screen data as the on-screen icon position changes.

— Icon pixel sizes can be 64 x 64, 64 x 128, 128 x 64, or 128 x 128.

— Up to eight 64 x 64 icons can be stored, but only four can be displayed at any one time.

D Color Expansion and Extended Write Modes

— This appendix describes how to increase color-expansion performance using Extended Write modes 4 and 5, and the BitBLT engine.

E True-Color Modes

— This appendix describes the programming requirements for the standard 8-bit VGA and the RGB 5-5-5, 5-6-5, and 8-8-8 true-color modes.

F QA Test Procedures

— This appendix describes testing procedures used by the Quality Assurance department to verify the functionality of the software described in this manual.

4 THE CL-GD7555 and CL-GD7556 VGA BIOS

(Version 1.00)

4.1 Introduction

This document specifies the external software interface of the Cirrus Logic CL-GD755X VGA BIOS (Basic Input/Output System). The CL-GD755X VGA BIOS is fully compatible with the standard IBM VGA BIOS and the Interrupt 10h interrupt services. The CL-GD755X VGA BIOS requires 48 Kbytes of ROM space.

This document describes the interface of the Cirrus Logic CL-GD755X VGA BIOS extended functions, most of which are designed to be compatible with other Cirrus Logic products. However, this document does not describe the standard Interrupt 10h functions. For detailed information on these functions, refer to various other documents, which are listed in Section 4.14 on page 60.

4.1.1 Main Features of the CL-GD755X VGA BIOS

The CL-GD755X VGA BIOS is a high-quality, feature-rich firmware product designed to take maximum advantage of the CL-GD755X 64-Bit Video and Graphics LCD/CRT Drivers and Utilities, especially in the areas of display quality, power management, and video performance. This product has the following features:

- Compatible with VGA BIOS
- Support for SimulSCAN™ (that is, the Cirrus Logic proprietary operation that allows for the simultaneous display of data on both a flat panel and a CRT monitor)
- Support for the CL-GD755X power-management modes
- Support for the CL-GD755X display enhancement features
- Support for VGA BIOS daughterboard (segment C000h) or motherboard (segment E000h) implementation
- Support for integration with system BIOS
- Support for switchless configuration
- Can be customized without source code through the use of utility programs such as the Cirrus Logic OEMSI Utility, OEMSI.EXE
- Provides VESA-compatible modes and VBE 1.2 (CL-GD7555) and VBE 2.0 (CL-GD7556) function support
- Supports multiple CRT monitor refresh rates

4.1.2 Cirrus Logic Related Documents

The following documents are available from Cirrus Logic as of the time of the publication of this document.

- *CL-GD7555 Application Book*, Version 1.0 or later versions.
- *CL-GD7555 Hardware Reference Manual*, Version 1.2, October 1996, or later versions.
- *CL-GD7556 Hardware Reference Manual*, Version 0.6, November 1996, or later versions.
- *OEMSI, The OEM System Integration Utility,* described in Chapter 5 of this manual.

4.1.3 Other Related Documents

The following documents provide general information about standard VGA BIOS and related topics.

- *Advanced Programmer's Guide to Super VGAs.*
  George Sutty and Steve Blair.
  Brady, 1990.
  ISBN 0-13-010455-8.
- *EGA/VGA – A Programmer's Reference Guide*.
  Bradley Dyck Kliewer.
  McGraw-Hill, 1989.
  ISBN 0-07-035089-2.
- *PC & PS/2 Video Systems.*
  Richard Wilton.
  Microsoft Press, 1987.
  ISBN 1-55615-103-9.
- *System BIOS for IBM PC/XT/AT Computers and Compatibles*.
  Phoenix Technologies, Ltd.
  Addison-Wesley, 1989.
  ISBN 0-201-51806-6.

4.2 Implementation Overview

The sections that follow describe specific information relating to the particular way the CL-GD755X LCD/CRT VGA controller and its VGA BIOS are implemented. Physically, there are two ways to implement the CL-GD755X LCD/CRT VGA controller and its VGA BIOS:

1) Implement the CL-GD755X and its VGA BIOS on an add-in daughterboard that connects to a motherboard

2) Implement the CL-GD755X and its VGA BIOS by direct placement on a motherboard.

Normally, motherboard designs incorporate the VGA BIOS with the system BIOS. The CL-GD755X VGA BIOS requires 48 Kbytes of ROM space. All extra functionality, such as EGA-compatible fonts and full VESA VBE 1.2 functionality, is contained within the 48-Kbyte image. For system implementations that do not have ROM space, it is recommended that a third-party VGA BIOS be considered. For more information, refer to Section 4.4.1.1.

4.2.1 Implementation of the VGA BIOS

When the CL-GD755X VGA controller and its VGA BIOS are implemented, the following occur:

1. The CL-GD755X VGA BIOS can be relocated in either segment C000h or segment E000h.
2. The VGA subsystem boot process is initialized in the standard method.
3. The ROM header of the CL-GD755X VGA BIOS, "55h,AAh", is identified by the system BIOS first, and a length indicator of the ROM BIOS size immediately follows it.
4. The indicated length checksums to zero, modulo 256.
5. The system BIOS makes a far call to offset 3h of the segment where the CL-GD755X VGA BIOS resides in order to initiate the boot process of the VGA subsystem.
6. The CL-GD7556 BIOS will initialize PCI subsystem ID at the begining of the POST.
7. For the CL-GD755X LCD/CRT VGA subsystem to coexist with other graphics controller subsystems, the planar system BIOS must provide CGA and MDA support (that is, INT 42h interrupt services). The system BIOS must handle all requirements of CGA and MGA graphics controllers.
8. The CL-GD755X bus interface follows the convention of the *PCI Bus Specification*, Revision 2.0. At offset 18h–19h of the ROM is a pointer to the PCI bus data structure. This structure contains the string ‘PCIR’, which is followed by the Cirrus Logic vendor identification and the CL-GD755X device identifier.

The PCI bus implementation allows the expansion ROM to discard the initialization code at POST (power-on self-test). Currently, the CL-GD755X initialization code is less than 1 Kbyte in size and is not discarded.

In some environments, the INT 10h vector is not available. In these cases, for the CL-GD755X VGA subsystem to operate, a direct entry point is provided to far call to ROM BIOS functions without using a software interrupt.

4.2.2 VGA BIOS Interrupt-Handler Entry Point (Offset 0034h)

In the PCI Bus system, the data word that contains the offset of the direct entry point resides at either E000:0034h or C000:0034h, as designed by the OEM.

This data word points to the instruction immediately following the STI ‘Enable Interrupt’ instruction. To allow IRET instructions to execute correctly, those applications that use the direct entry point must push the system flag into the stack before they make a far call to VGA ROM BIOS.

In an SMI (System Management Interrupt) environment, the SMI handler can use the offset address of the direct entry point to set up the INT 10h interrupt vector to allow INT 10h software interrupts. However, unless the interrupt table and BIOS data area are correctly initialized, only certain VGA BIOS functions can be used (such as power-save functions and the save/restore video state function).

4.2.3 Implementation with Extended Display Mode Function Support

The CL-GD755X VGA BIOS supports text output functions for all standard display modes and extended text display modes. This release of the Cirrus Logic VGA BIOS supports text output for extended 16- and 256-color graphics display modes. For extended graphics display modes, the following functions are supported:

- INT 10h functions 0h, Fh, 1Ch
- VESA VBE functions
- Cirrus Logic extended functions

Most programs using extended graphics display modes utilize driver programs that do not rely on VGA BIOS support beyond the display mode set. An application can determine if text output is supported at run time, through the VESA display mode information structures.

4.3 Power-Management Support

For the Cirrus Logic VGA BIOS to support power management in various processor environments, the following items must be considered:

- Direct access to the VGA BIOS Power-Management functions must be provided, without requiring that the interrupt table or BIOS data area in lower-address memory be valid. This situation can occur when switching processor modes, such as when an SMI occurs. (For more information, refer to the description of offset 0034h in Section 4.2.2.)
- The VGA BIOS must be completely re-entrant for the Power-Management functions (INT 10h, AH=12h, BL=94h) and the Save State/Restore State function (INT 10h, AH=1Ch). For example, the Save State function (INT 10h, AX=1C01h) can be called during a Set Video Mode call (INT 10h, AH=00h).
- Interrupts must never be explicitly enabled with an STI instruction, except at the VGA BIOS INT 10h entry point. For more information, refer to the description of offset 0034h in Section 4.2.2.
- No INT instructions must be used by the VGA BIOS for the Power-Management functions (INT 10h, AH=12h, BL=94h) and the Save State/Restore State function (INT 10h, AH=1Ch).
- Support must be provided for the Power-Management functions to reduce power consumption by a display device when the system is not in use.

For two program examples that show the Zero-Volt Suspend/Resume function on the VGA subsystem, refer to the program 755XSAV.ASM and 755XRES.ASM on the Cirrus Logic BIOS and Utilities Disk.

4.4 Configuring the CL-GD755X VGA BIOS

This section discusses how to configure the Cirrus Logic VGA BIOS.

4.4.1 Selecting the Correct VGA BIOS

The Cirrus Logic VGA BIOS is designed to be flexible and to adapt to most system configurations. As described in the following sections, the Cirrus Logic VGA BIOS receives configuration information through multiple hooks to the system BIOS and utility programs.

However, for those instances in which the Cirrus Logic VGA BIOS does not meet the requirements of a specific system, the OEM should consider the alternatives that follow:

4.4.1.1 Alternative 1: Third-Party VGA BIOS

One alternative for OEMs who need to select a VGA BIOS that meets the requirements of a specific system is to obtain a third-party VGA BIOS. While Cirrus Logic does not guarantee or recommend VGA BIOS products from either Phoenix Technologies or SystemSoft, their addresses are provided here as a convenience. (Both Phoenix Technologies and SystemSoft have representatives in both local and international locations.)

- Phoenix Technologies Ltd.
  2770 De La Cruz Blvd., Santa Clara, CA 95050 USA
  Phone: (408) 654-9000
  Fax: (408) 452-1985
  e-Mail: notebios@ptltd.com.
- System Soft Corporation
  2 Vision Drive, Natick, MA 01760-2059 USA
  Phone: (508) 651-0088
  Fax: (508) 651-8188.

4.4.1.2 Alternative 2: Modify Cirrus Logic VGA BIOS Source Code

A second alternative for OEMs who want to configure a VGA BIOS that meets the requirements of a specific system is to obtain a source license for the Cirrus Logic VGA BIOS and maintain the source code for themselves.

▲ **WARNING:** The Cirrus Logic VGA BIOS source code is complex and space-limited. Only OEMs with extensive VGA BIOS programming experience should attempt to modify the source code.

4.4.2 Customizing the VGA BIOS – OEMSI Utility Overview

(See Chapter 5 for Detailed Information About OEMSI)

This section is an overview of the Cirrus Logic OEMSI Utility, which is used to customize the Cirrus Logic VGA BIOS for specific requirements.

By using the OEMSI Utility, an OEM can change a wide variety of Cirrus Logic VGA BIOS defaults and features without having to access the Cirrus Logic VGA BIOS source code.

Using the OEMSI Utility to customize the Cirrus Logic VGA BIOS consists of the following steps:

1. Creating (exporting) a data file
2. Editing the data file
3. Importing the modified data file
4. Creating a BIOS ROM
5. Testing the new, modified Cirrus Logic VGA BIOS

The OEMSI Utility has a minimal user interface. The operation of the OEMSI program is controlled by command line parameters with the following format:

```
OEMSI option data-file BIOS-binary-file [ENTER]
```

To obtain a full list of available OEMSI options, type:

```
OEMSI [ENTER]
```

In general, there are two basic options that must be used to modify the Cirrus Logic VGA BIOS binary image.

Option	Description
`-bo or -b`	Export OEM data from the Cirrus Logic VGA BIOS binary file to the data file
`-i`	Import reconfigured data to the Cirrus Logic VGA BIOS binary file from the data file

4.4.3 Changing the Sign-On Message

To change the sign-on message, do the following:

1. Use the OEMSI Utility -b option to export the data file.
2. Use a text editor to edit the data file for the following modifications:

   2a. To disable the Cirrus Logic copyright message, search for the label CIRRUS_SIGN_ON_YN and change the variable from Y to N.

   2b. To display a custom sign-on message, search for the label CUSTOMER_SIGN_ON_YN and change the variable from N to Y.

   2c. Locate the label CUSTOMER_SIGN_ON_MSG. Enter custom text between the labels BEGIN and END.

   2d. To change the location of either the Cirrus Logic copyright message or your custom sign-on message, search for these labels:

   CIRRUS_SIGN_ON_ROW
   CIRRUS_SIGN_ON_COL
   CUSTOMER_SIGN_ON_ROW
   CUSTOMER_SIGN_ON_COL

   Change the row and column entries to the appropriate starting location.

4.4.4 Changing the Flat Panel Type

4.4.4.1 Default Flat Panel Types

Currently, the Cirrus Logic VGA BIOS can support 16 different flat panel types. However, even within the same flat panel type, there can be variations among flat panel manufacturers. Consequently, for each setting for a flat panel type, the VGA BIOS can default only to a flat panel that is from a particular manufacturer and that is used for testing by Cirrus Logic Quality Assurance.

The default flat panel types are listed in the table that follows. Each panel table entry describes the type of flat panel supported by the Cirrus Logic VGA controller. When configuring for a specific flat panel, start with one of these basic panel types, which are provided as a starting point for popular flat panel configurations. (To obtain default information for each flat panel type, refer to the release notes for the current Cirrus Logic VGA BIOS.)

NOTE: For a given type of flat panel, specific panels typically require tuning, due to differences between manufacturers.

Table 4-1. Default Flat Panel Types

Flat Panel Table Entry	Flat Panel Type[a]	Flat Panel Description	
		Flat Panel Data Format	Spatial Resolution (Column x Rows, in Pixels)
Panel Table Entry 1	0	DSTN XGA (CL-GD7556 only)	1024 x 768
Panel Table Entry 2	1	24-bit XGA TFT color (XGA TFT)	1024 x 768
Panel Table Entry 3	2	Dual-Scan STN color (VGA DSTN)	640 x 480
Panel Table Entry 4	3	18-bit TFT color (VGA TFT)	640 x 480
Panel Table Entry 5	4[b]	Reserved	–
Panel Table Entry 6	5[b]	Reserved	–
Panel Table Entry 7	6	Dual-Scan STN color (SVGA DSTN)	800 x 600
Panel Table Entry 8	7	18-bit TFT color (SVGA TFT)	800 x 600
Panel Table Entry 9	8	Panel types 8–15 definitions are reserved for use by the OEM. They are accessed through the INT15h interface or by using the OEMSI Utility.	
Panel Table Entry 10	9		
Panel Table Entry 11	10		
Panel Table Entry 12	11		
Panel Table Entry 13	12		
Panel Table Entry 14	13		
Panel Table Entry 15	14		
Panel Table Entry 16	15		

[a] After POST, the flat-panel types can be modified by using the INT10 9Ch interface.

4.4.4.2 Changing the Default Flat-Panel Type

So that the Cirrus Logic VGA BIOS does not restrict the association between the flat panel type and the flat panel to be used, OEMs can use the OEMSI Utility to change the default type of flat panel to a replacement type. To change the default type, use OEMSI Utility with the `-b` option to export the data file.

OEMs can obtain parameters for other flat panel models as follows:

1. OEMs who:
   - 1a. Do not have an OEM BBS account can obtain an account by providing their Cirrus Logic representative with their login name and a request to have their BBS account upgraded to OEM-access level.
   - 1b. Do have an OEM BBS account can access the Cirrus Logic BBS at either:

     USA (510) 440-9080
     USA (510) 440-0394
2. Join the 755X conference.
3. Enter the file area for BIOS.
4. Download the necessary flat panel parameters and flat panel interface guides.

4.4.4.3 Determining and Configuring the Flat Panel Type

If the boot process is a:

- *Cold* boot process, the flat panel type information is determined during the boot process.
- *Warm* boot process, and if OTHER_BIN_BYTE17[1] is set to 1, the CL-GD755X VGA BIOS re-determines the flat panel type during the boot process.

NOTE: To obtain exact information concerning OTHER_BIN_BYTE17[1], refer to the Cirrus Logic OEMSI documentation for the CL-GD755X.

Three-Level Interface for Determining and Configuring the Flat Panel Type

To select the correct flat panel type at POST, the CL-GD755X VGA BIOS provides a three-level interface for determining and configuring the flat panel type.

<u>Level 1</u>.

The first level of the interface is the INT 15h interface. If the INT 15h is the desired interface, it can be enabled by setting the READ_DIP_SWITCH[4] parameter to 1. (This parameter's default value is 0.)

NOTE: To obtain exact information concerning the READ_DIP_SWITCH[4] command, refer to the Cirrus Logic OEMSI documentation for the CL-GD755X.

The INT 15h interface occurs as follows:

Output:	AX	=	448Fh (This value can be changed with the Cirrus Logic OEMSI Utility.)	
Input:	AH	=	0	(Function returns successfully, and AL= Panel type.)
			A value other than 0	(Function fails, and the VGA BIOS goes to level 2.)

<u>Level 2</u>.

If the first level of the interface fails (that is, AH has a value other than 0), the Cirrus Logic VGA BIOS goes to this level 2 to obtain its hard-coded flat panel type information as follows:

1. The Cirrus Logic OEMSI Utility can be used to clear one bit in the OEM data byte, READ_DIP_SWITCH[7].
2. The VGA BIOS checks to see if this bit is clear.
3. If the bit is cleared, the VGA BIOS uses the hard-coded information that is in the lower 4 bits of the same OEM data byte, READ_DIP_SWITCH[3:0], to determine the type of flat panel that is being used.

<u>Level 3</u>.

If none of above two interfaces have been enabled, the third level of the interface is used. With this level 3 interface, the VGA BIOS uses information from the hardware switches to determine the type of flat panel being used and to configure the VGA BIOS for this flat panel type.

Alternative Methods for Configuring the Flat Panel Type

After determining the flat panel type, to configure that type, an alternative to using the three-level interface is to overwrite the configuration setting of the type. This overwrite automatically takes place at the last stage of POST and just before enabling the display device (that is, the CRT monitor or flat panel being used).

To overwrite the setting, the VGA BIOS makes a function call, 448Eh, of the system BIOS function INT 15h. In the function call, the system BIOS can call Cirrus Logic extended VGA BIOS function 9Ch to set the flat panel type. (The only restriction is that the function 9Ch is required to perform in CRT_ONLY mode.)

Replacing a Flat Panel Type Setting

The VGA BIOS has default flat panel type settings. To use a flat panel type setting that is different from the type to which the VGA BIOS defaults, the OEMSI Utility must be used to modify the table entry for the flat panel type. The table, which must be modified for the correct entry number, can be found by the label:

%%%[tab]PANEL

4.4.5 Configuring the DRAM

The Cirrus Logic VGA BIOS configures the MCLK (memory clock) frequency as shown in the VGA Display tables starting on page 21. The CL-GD7555 VGA BIOS assumes maximum performance conditions, and therefore sets the MCLK to 80 MHz. When a 3.3-V VDD is used, the MCLK is changed to 66 MHz in response to the “voltage control” modification using the OEMSI utility. The CL-GD7556 VGA BIOS assumes a 3.3-V VDD and sets MCLK to 80MHz.

For speeds approaching these limits, the type of DRAM that must be used is 60-ns EDO (Extended Data-Out) DRAM, which is also known as Hyper-Page-Mode DRAM.

The VGA BIOS is tuned for operation with the above-mentioned MCLK speeds and memory types. Changing the memory speed or type affects the performance of the graphics system and also affects the available memory bandwidth for accelerated video operations. To change the DRAM speed or type:

- Follow directions in the Cirrus Logic application notes for DRAM configuration.
- Test all changes for FIFO underflow and overflow in graphics.
- Test all changes for correct video operation for all supported flat panel types.
- Test all configurations and refresh rates involving the CRT monitor.

NOTE: No MCLK_CONFIG_TABLE in CL-GD7556 VGA BIOS, modify SR1F in HW_CONFIG_TABLE.

To modify the MCLK speed, in the MCLK_CONFIG_TABLE, modify SR1F for the desired memory clock speed as shown in Table 4-2. The SR1F value in the table that is used for 66 MHz operation is 25h, and the SR1F value that used for 80 MHz operation is 2Dh. The length and duty-cycle of the RAS cycle is adjusted by using Extension registers GR18[2] to change the DRAM type, and SRF[2] and SR20[6] to change the number of MCLKs as shown in Table 4-3.

The MCLK table that follows provides examples with a reference frequency of 14.31818 MHz as determined by an MCLK value of [(14.31818 MHz/8) x (SR1F)]. No parameter checking is performed on the value for the memory clock.

▲ **WARNING:** Do not exceed the CL-GD755X or DRAM specifications.

Table 4-2. MCLK Frequency Select

MCLK Frequency (MHz)	SRIF		
	(Hex)	(Decimal)	(Binary)
44.744	19h	25	0001-1001
50.114	1Ch	28	0001-1100
60.852	22h	34	0010-0010
66.2216	25h	37	0010-0101
80.539	2Dh	45	0010-1101

Table 4-3. DRAM RAS Cycle Timing Adjustment

DRAM Type	GR18[2]	SR20[6]	SRF[2]	Length of RAS# Cycle	Type Of RAS# Cycle
Fast-page Mode	0	0	0	7 MCLKs	4 MCLKs low and 3 MCLKs high
	0	0	1	6 MCLKs	3.5 MCLKs low and 2.5 MCLKs high
EDO Hyper-page Mode	1	0	0	8 MCLKs	5 MCLKs low and 3 MCLKs high
	1	1	0	9 MCLKs	5 MCLKs low and 4 MCLKs high

4.4.6 Configuration of External Signaling

This section describes the configuration of the external signaling (that is, the interface from the VGA BIOS to the system BIOS). The CL-GD755X VGA BIOS signals the system BIOS during the POST VGA subsystem initialization and during each Video Set Mode function call.

The system BIOS can respond to these signal calls by calling back the VGA BIOS to configure any special options. (Although the system BIOS is not required to respond to these signal calls, in certain implementations, a response can be desirable.)

The signal calls are made as INT 15h calls, with a function value loaded into AX. The default values for these function calls are compatible with previous Cirrus Logic products. However, the OEM can use the OEMSI Utility to modify these values to use the function calls that are most compatible with their particular system implementation.

4.4.6.1 External Signaling Interface during POST VGA Subsystem Initialization

This section describes the interface between the VGA BIOS and the system BIOS during the POST VGA subsystem initialization.

After a cold or warm boot, during VGA subsystem initialization, the VGA BIOS initializes the CL-GD755X to a default state. Upon completion of VGA subsystem initialization, the VGA BIOS displays a sign-on message and returns control to the system BIOS.

VGA Subsystem Initialization After a Cold Boot

For a cold boot process, just before the VGA BIOS turns on the display device, the following occur:

1. The VGA BIOS calls INT 15h with AX = 448Eh (SIGNAL_INITIALIZATION_COMPLETE) to signal to the system that the VGA subsystem is completely initialized. (The system BIOS is not required to respond to the SIGNAL_INITIALIZATION_COMPLETE signal call.)
2. At this point, custom configuration calls can be made. The OEM can define functionality in response to the SIGNAL_INITIALIZATION_COMPLETE function call from the VGA BIOS. The system BIOS can call back the VGA BIOS through INT 10h to configure the VGA subsystem to a custom state. Custom configuration information can include the following:
   - Core voltage level
   - CRT monitor refresh rate and maximum vertical size parameters
   - Flat panel type (if the display device is in CRT-only mode)
   - Initial display state
3. After the system completes the configuration of the VGA subsystem to a custom state, it returns control to the VGA BIOS.
4. The VGA BIOS then displays its sign-on message and exits to the system initialization code.

VGA Subsystem Initialization After a Warm Boot

For a warm boot process, the SIGNAL_INITIALIZATION_COMPLETE call can be made if the appropriate bit in the OEM data is set to a 1. Otherwise, all current user options and the flat panel type remain the same. If a CRT monitor is previously active, it is checked to ensure it is still attached. If the CRT monitor is no longer attached, the VGA BIOS boots in the panel-only mode.

4.4.6.2 External Signaling Interface during Video Set Mode Function Call

This section describes the interface between the VGA BIOS and the system BIOS during the Video Set Mode function call.

During each VGA BIOS Video Set Mode call (INT 10h, AH = 00h), as described in the process that follows, the VGA BIOS signals the system by making three INT 15h function calls.

- The INT 15h function call, AX = 4490h (SIGNAL_MINIMUM_VOLTAGE)
- The INT 15h function call, AX = 0F964h (SETMODE_BEGIN)
- The INT 15h function call, AX = 0F965h (SETMODE_END)

NOTE: The OEMSI Utility can be used to suppress these INT 15h function calls.

INT 15h Function Call Number 1

Before the VGA BIOS validates the requested display mode setting, it makes the first of the three INT 15h function calls, AX = 4490h (SIGNAL_MINIMUM_VOLTAGE), which signals the voltage requirement for the display mode. Other registers used in the function call are the following:

BL = The Cirrus Logic display mode number to be set

BH = Voltage level
(BH = 0 for an operation that requires 5 V)
(BH = 1 for an operation that requires a minimum of 3.3 V)

The SIGNAL_MINIMUM_VOLTAGE function call:

- Is intended to be used for dynamic power switching. (For a sample implementation of the SIGNAL_MINIMUM_VOLTAGE function for dynamic power switching, refer to Section 4.14 on page 60)
- Does not require a response by the system BIOS.

INT 15h Function Call Numbers 2 and 3

After validating the display mode number, two INT 15h function calls remain in the SET_VIDEO_MODE process.

- The second INT 15h function call, AX = 0F964h, signals the SETMODE_BEGIN.
- The third INT 15h function call, AX = 0F965h, signals the SETMODE_END.

Both function calls pass control back to the system BIOS for it to make necessary adjustments relating to the display mode.

4.5 Error Codes

During POST, the VGA BIOS can produce a beep error code, listed below.

Beep Error Code	Error Condition
1 long beep, followed by 3 short beeps	RAM test failure

4.6 Blocking

The Cirrus Logic VGA BIOS performs several checks. After the VGA BIOS performs the checks, it blocks (that is, restricts) the actions discussed in Section 4.6.1 and Section 4.6.2. In determining if a display mode or a type of display device is to be blocked, factors that the VGA BIOS can evaluate include the following:

- The amount of display memory
- MCLK speed (that is, the speed of the memory clock)
- Voltage level to the CL-GD755X
- Flat panel type
- Flat panel display screen spatial resolution
- CRT monitor display screen spatial resolution
- CRT monitor vertical refresh rate. (In the CRT-only mode, the VGA BIOS selects the highest refresh rate supported by the current configuration.)

4.6.1 Display Mode Blocking

After the VGA BIOS performs its checks, the VGA BIOS blocks attempts to change to a different display mode if the target display mode cannot be properly displayed upon the requested display device (that is, either a CRT monitor or a flat panel).

4.6.2 Display Device Type Blocking

After the VGA BIOS performs its checks, the VGA BIOS also blocks attempts to change to a different display device if the target display device cannot support the current display mode. For most (but not all) products that use the CL-GD755X, if the CL-GD755X scratch pad Extension register bit SRA[5] is set to 1, the VGA BIOS does not block attempts to change the type of display device. In this case, the blocking is bypassed. However, to protect the flat panel, the power to the flat panel is not cycled on.

The application program that controls the setting of CL-GD755X Extension register bit SRA[5] must also make the necessary adjustments before turning on the flat panel. For example, the Cirrus Logic Windows 3.1x drivers and Windows 95 driver use this SRA[5] bit to allow the type of display device to be changed as part of the Resolution-change-on-the-fly scheme.

4.6.2.1 Display Device Type Blocking: 640 x 480 (VGA) Flat Panels

When the flat panel used with the CL-GD755X has a spatial resolution of 640 x 480, the VGA BIOS blocks the following attempts:

- When the CL-GD755X is in panel-only mode, the VGA BIOS blocks attempts to set display modes with spatial resolutions of 800 x 600 or greater.
- When the CL-GD755X is in CRT-only mode and the CRT monitor is set for a display mode with a spatial resolution of 800 x 600 or greater, the VGA BIOS blocks attempts to switch to the panel-only mode or to SimulSCAN.
- When the CL-GD755X is in panel-only mode or SimulSCAN, the VGA BIOS blocks attempts to switch to display modes that a specific flat panel does not support.

Since the limiting factor is the 640 x 480 spatial resolution of the flat panel, when the CL-GD755X is in the panel-only mode or SimulSCAN, the VGA BIOS does not block attempts to switch to the CRT-only mode.

4.6.2.2 Display Device Type Blocking: 800 x 600 (SVGA) Flat Panels

When the flat panel used with the CL-GD755X has a spatial resolution of 800 x 600, the VGA BIOS blocks the following attempts:

- When the CL-GD755X is in panel-only mode, the VGA BIOS blocks attempts to set a display mode with a spatial resolution greater than 800 x 600.
- When the CL-GD755X is in CRT-only mode and the CRT monitor is set for a display mode with a spatial resolution of 1024 x 768 or greater, the VGA BIOS blocks attempts to switch to the panel-only mode or to SimulSCAN.
- When the CL-GD755X is in panel-only mode or SimulSCAN, the VGA BIOS blocks attempts to switch to display modes that a specific flat panel does not support.
- When the CL-GD755X is in panel-only mode and the display mode spatial resolution is 800 x 600, the VGA BIOS blocks attempts to switch to a CRT monitor that is configured to a maximum spatial resolution of 640 x 480.
- When the CL-GD755X is in panel-only mode and the display mode spatial resolution is 800 x 600, the VGA BIOS blocks attempts to switch to SimulSCAN mode when the CRT monitor is a fixed-frequency CRT that is configured to a maximum spatial resolution of 640 x 480.
- When the Expanded Display mode operation is not available for a requested display mode, the VGA BIOS automatically disables the Expanded Display mode option. In this case, the Non-Expanded Display mode is used until a display mode is set that supports the Expanded Display mode option. (For a description of the Expanded Display mode and the Centering Display mode, refer to Section 3.3.18 in the *CL-GD755X Advance Hardware Reference Manual*.)

In general, the VGA BIOS tries to accommodate all option requests supported by the current VGA subsystem configuration, but if the option request involves changing to:

- A display mode that is not supported, the VGA BIOS blocks changes to the display mode setting.
- A display device type that is not supported, the VGA BIOS blocks changes to the display device type.

4.6.2.3 Display Device Type Blocking: 1024 x 768 (XGA) Flat Panels

When the flat panel used with the CL-GD755X has a spatial resolution of 1024 x 768, the VGA BIOS blocks the following attempts:

- When the CL-GD755X is in panel-only mode, the VGA BIOS blocks attempts to set a display mode with a spatial resolution greater than 1024 x 768.
- When the CL-GD755X is in CRT-only mode and the CRT monitor is set for a display mode with a spatial resolution of 1280 x 1024, the VGA BIOS blocks attempts to switch to the panel-only mode or to SimulSCAN.
- When the CL-GD755X is in panel-only mode or SimulSCAN, the VGA BIOS blocks attempts to switch to display modes that a specific panel does not support.
- When the CL-GD755X is in panel-only mode, the VGA BIOS blocks attempts to switch to a CRT monitor that is configured to a maximum spatial resolution of either 800 x 600 or 640 x 480, when the spatial resolution of the flat panel display mode is greater than the configured maximum spatial resolution of the CRT monitor.
- When the CL-GD755X is in panel-only mode and the display mode spatial resolution is 1024 x 768, the VGA BIOS blocks attempts to switch to SimulSCAN mode when the CRT monitor is a fixed-frequency CRT monitor that is configured to a maximum spatial resolution of either 640 x 480 or 800 x 600.
- When the Expanded Display mode operation is not available for a requested display mode, the VGA BIOS automatically temporarily disables the Expanded Display mode option. In this case, the Centering Display mode is used until a display mode is set that supports the Expanded Display mode option. (For a description of the Expanded Display mode and the Centering Display mode, refer to Section 3.3.18 in the *CL-GD7555 Hardware Reference Manual* and Section 3.3.19 in the *CL-GD7556 Hardware Reference Manual.)*

In general, the VGA BIOS tries to accommodate all option requests supported by the current VGA subsystem configuration, but if the option request involves changing to:

- A display mode that is not supported, the VGA BIOS blocks changes to the display mode setting.
- A display device type that is not supported, the VGA BIOS blocks changes to the display device type.

4.7 VESA Support

The CL-GD755X VGA BIOS supports the following VESA functions:

- Full VESA VBE 1.2 implementation
- VBE/PM (DPMS)
- VBE/DDC2B functions 0 and 1

For specific questions about VESA functions, refer to the following VESA documentation:

- *VESA Super VGA Standard, version 1.2*
- *VESA BIOS Extensions/Power Management (VBE/PM), version 1.0*
- *VESA BIOS Extensions/Display Data Channel (VBE/DDC) Standard, version 1.0*
- *VESA VBE 2,0 TSR*

Direct any questions on topics related to VESA standards by writing to VESA at:

- America On-Line. Use the keyword VESA and post your message on the VESA Message Board.
- VESA FAX number: USA [408) 435-8225
- VESA mailing address: Video Electronics Standards Association
  2150 North First Street, Suite 440
  San Jose, CA 95131-2029

4.8 CRT-Only Display Modes

This section lists tables for standard VGA and Cirrus Logic Extended VGA CRT-only display modes.

The following symbols and abbreviations are used for display modes in this section:

- – A minus sign (or no sign at all) after a display mode number indicates CGA display mode. (Table 4-4 only.)
- * EGA display mode. (Table 4-4 only.)
- + VGA display mode. (Table 4-4 only.)
- ‡ Indicates a 32K direct-color packed-pixel display mode
- i Indicates a display mode that uses an interlaced frequency
- tbd to be determined

4.8.1 Standard VGA CRT-Only Display Modes

The CL-GD755X VGA BIOS supports all standard VGA CRT-only display modes, listed in Table 4-4.

Table 4-4. Standard VGA CRT-Only Display Modes

Standard VGA Display Mode No. (hex)	VESA Display Mode Number	Color Depth	Characters × Rows	Character Cell (pixels)	Spatial Resolution (pixels)	Type of Display Mode	Horizontal Frequency (kHz)	Vertical Frequency (Hz)
00, 01	–	16/256K	40 × 25	8 × 8	320 × 200	Text	31.5	70
00*, 01*	–	16/256K	40 × 25	8 × 14	320 × 350	Text	31.5	70
00+, 01+	–	16/256K	40 × 25	9 × 16	360 × 400	Text	31.5	70
02, 03	–	16/256K	80 × 25	8 × 8	640 × 200	Text	31.5	70
02*, 03*	–	16/256K	80 × 25	8 × 14	640 × 350	Text	31.5	70
02+, 03+	–	16/256K	80 × 25	9 × 16	720 × 400	Text	31.5	70
04, 05	–	4/256K	40 × 25	8 × 8	320 × 200	Graphics	31.5	70
06	–	2/256K	80 × 25	8 × 8	640 × 200	Graphics	31.5	70
07*	–	Mono.	80 × 25	9 × 14	720 × 350	Text	31.5	70
07+	–	Mono.	80 × 25	9 × 16	720 × 400	Text	31.5	70
0D	–	16/256K	40 × 25	8 × 8	320 × 200	Graphics	31.5	70
0E	–	16/256K	80 × 25	8 × 8	640 × 200	Graphics	31.5	70
0F	–	Mono.	80 × 25	8 × 14	640 × 350	Graphics	31.5	70
10	–	16/256K	80 × 25	8 × 14	640 × 350	Graphics	31.5	70
11	–	2/256K	80 × 30	8 × 16	640 × 480	Graphics	31.5	60
12	–	16/256K	80 × 30	8 × 16	640 × 480	Graphics	31.5	60
13	–	256/256K	40 × 25	8 × 8	320 × 200	Graphics	31.5	70

4.8.2 Cirrus Logic Extended VGA CRT-Only Display Modes

Table 4-5 shows the extended VGA and VESA display modes that the CL-GD755X VGA BIOS supports.

- Within the table:
  - — The ‘Minimum MCLK’ column gives the recommended memory clock frequency at which the CL-GD755X can run without adverse effects to functionality. Better benchmarks can be achieved with an MCLK frequency higher than the frequency specified.
  - — Some of the display modes are not supported by all CRT monitors.
  - — Unless otherwise noted, the display modes are supported with 1 Mbyte of display memory.
- Text display mode 54h is actually 1056 x 344 addressable pixels. However, it uses 1056 x 350 timing.
- Graphics display modes 11h and 12h are high-refresh derivations of the standard VGA CRT-only display modes 11h and 12h in Table 4-4. The graphics display modes 11h and 12h in this table have been enhanced by Cirrus Logic for a higher vertical frequency.
- Graphics display mode 6Ah must be used, rather than graphics display mode 58h, for application programs to retain compatibility with other VGA BIOS products.
- For the CRT monitor type that is selected, the fastest vertical refresh rate is automatically used.
  - — i Indicates a display mode that uses an interlaced frequency

Table 4-5. Cirrus Logic Extended VGA CRT-Only Display Modes

Extended VGA Display Mode No. (hex)	VESA Display Mode No. (hex)	Color Depth	Charac-ters × Rows	Char-acter Cell (pixels)	Spatial Resolution (pixels)	Horizontal Freq. (kHz)	Vertical Freq. (Hz)	VCLK (MHz)	Mini-mum MCLK (MHz)	Required CVDD Voltage (Min.)
Text Display Modes										
14	–	16/256K	132 × 25	8 × 16	1056 × 400	31.5	70	41.5	50	3.3
54	10A	16/256K	132 × 43	8 × 8	1056 × 350	31.5	70	41.5	50	3.3
55	109	16/256K	132 × 25	8 × 14	1056 × 350	31.5	70	41.5	50	3.3
Graphics Display Modes										
11	–	2/256K	80 × 30	8 × 16	640 × 480	37.9	72	31.5	50	3.3
						37.5	75	31.5	50	3.3
						43.269	85	36	50	3.3
12	–	16/256K	80 × 30	8 × 16	640 × 480	37.9	72	31.5	50	3.3
						37.5	75	31.5	50	3.3
						43.269	85	36	50	3.3
58, 6A	102	16/256K	100 × 37	8 × 16	800 × 600	35.2	56	36	50	3.3
						37.8	60	40	50	3.3
						48.1	72	50	50	3.3
						46.875	75	50	50	3.3

Table 4-5. Cirrus Logic Extended VGA CRT-Only Display Modes *(cont.)*

Extended VGA Display Mode No. (hex)	VESA Display Mode No. (hex)	Color Depth	Charac-ters × Rows	Char-acter Cell (pixels)	Spatial Resolution (pixels)	Horizontal Freq. (kHz)	Vertical Freq. (Hz)	VCLK (MHz)	Mini-mum MCLK (MHz)	Required CVDD Voltage (Min.)
5C	103	256/256K	100 × 37	8 × 16	800 × 600	35.2	56	36	50	3.3
						37.9	60	40	50	3.3
						48.1	72	50	50	3.3
						46.875	75	50	50	3.3
						53.674	85	56.25	50	3.3
5D	104	16/256K	128 × 48	8 × 16	1024 × 768	35.5	43i	44.9	50	3.3
						48.3	60	65	50	3.3
						56	70	75	50	3.3
						58	72	77	50	3.3
						60	75	78.75	50	3.3
5E	100	256/256K	80 × 25	8 × 16	640 × 400	31.5	70	25	50	3.3
5F	101	256/256K	80 × 30	8 × 16	640 × 480	31.5	60	25	50	3.3
						37.9	72	31.5	50	3.3
						37.5	75	31.5	50	3.3
						43.269	85	36	50	3.3
60	105	256/256K	128 × 48	8 × 16	1024 × 768	35.5	43i	44.9	50	3.3
						48.3	60	65	50	3.3
						56	70	75	50	3.3
						58	72	77	50	3.3
						60	75	78.75	50	3.3
						68.677	85	94.5	50	5.0[a]
64	111	64K	–	–	640 × 480	31.5	60	25	50	3.3
						37.9	72	31.5	50	3.3
						37.5	75	31.5	50	3.3
						43.269	85	36	50	3.3
65	114	64K	–	–	800 × 600	35.2	56	36	50	3.3
						37.8	60	40	50	3.3
						48.1	72	50	50	3.3
						46.875	75	50	50	3.3
						53.674	85	56.25	50	3.3
66	110	32K[b]	–	–	640 × 480	31.5	60	25	50	3.3
						37.9	72	31.5	50	3.3
						37.5	75	31.5	50	3.3
						43.269	85	36	50	3.3

Table 4-5. Cirrus Logic Extended VGA CRT-Only Display Modes *(cont.)*

Extended VGA Display Mode No. (hex)	VESA Display Mode No. (hex)	Color Depth	Charac-ters × Rows	Char-acter Cell (pixels)	Spatial Resolution (pixels)	Horizontal Freq. (kHz)	Vertical Freq. (Hz)	VCLK (MHz)	Mini-mum MCLK (MHz)	Required CVDD Voltage (Min.)
67	113	32K[b]	128 x 48	8 × 16	800 × 600	35.2	56	36	50	3.3
						37.8	60	40	50	3.3
						48.1	72	50	50	3.3
						47	75	49.5	50	3.3
						53.674	85	56.25	50	3.3
68[c]	116	32K[b]	128 x 48	8 × 16	1024 × 768	35.5	43i	44.9	50	3.3
						48.2	60	65	50	3.3
						56	70	75	60	3.3
						60	75	78.7	60	3.3
						68.677	85	94.5	80	5.0[a]
69[de]	106	32K	160 × 64	8 × 16	1280 × 1024	46.4	43i	78.75	80	3.3
						63.98	60	108	80	3.3
6C[c]	106	16/256K	160 × 64	8 × 16	1280 × 1024	48	43i	75	50	3.3
6D[c]	107	256/256K	160 × 64	8 × 16	1280 × 1024	48	43i	75	50	3.3
						64	60	108	80	5.0[a]
6D[d]	107	256	160 × 64	8 × 16	1280 × 1024	80	75	135	80	3.3
71	112	16M	80 × 30	8 × 16	640 × 480	31.5	60	25	50	3.3
						37.9	72	31.5	50	3.3
						37.5	75	31.5	50	3.3
						43.269	85	36	50	3.3
74[c]	117	64K	128 × 48	8 × 16	1024 × 768	35.5	43i	44.9	50	3.3
						48.3	60	65	50	3.3
						56	70	75	60	3.3
						60	75	78.7	60	3.3
						68.677	85	94.5	80	5.0[a]
75[de]	11A	64K	128 × 48	8 × 16	1280 × 1024	46.4	43i	78.75	80	3.3
						63.98	60	108	80	3.3
78[c]	115	16M	100 x 37	8 × 16	800 × 600	35.2	56	36	50	3.3
						37.8	60	40	50	3.3
						48.1	72	50	60	3.3
						47	75	49.5	60	3.3
						53.674	85	56.25	66	3.3

Table 4-5. Cirrus Logic Extended VGA CRT-Only Display Modes *(cont.)*

Extended VGA Display Mode No. (hex)	VESA Display Mode No. (hex)	Color Depth	Charac-ters × Rows	Char-acter Cell (pixels)	Spatial Resolution (pixels)	Horizontal Freq. (kHz)	Vertical Freq. (Hz)	VCLK (MHz)	Mini-mum MCLK (MHz)	Required CVDD Voltage (Min.)
79[de]	118	16M	128 × 48	8 × 16	1024 × 768	35.5	43i	44.9	80	3.3
						48.4	60	65	80	3.3
						56.5	70	75	80	3.3

[a] The CL-GD7556 operates at a VDD = 3.3 V.
[b] This display mode is 32K direct-color packed-pixel.
[c] A minimum of 2 Mbytes of display memory are required to support all of the capabilities of this display mode.
[d] For CL-GD7556 only.
[e] A minimum of 4 Mbytes of display memory are required.

4.9 Panel-Only and SimulSCAN™ Display Modes

The CL-GD755X VGA BIOS supports panel-only and SimulSCAN operations with the standard VGA display modes listed in Section 4.8.1. In addition, the CL-GD755X VGA BIOS supports panel-only and SimulSCAN operations with the extended display modes listed in this section.

The following symbols and abbreviations are used for display modes in this section:

DSTN	Dual-scan STN flat panel
TFT	Active matrix flat panel
tbd	to be determined

4.9.1 Panel-Only and SimulSCAN 640 x 480 (VGA) Display Modes

For 640 x 480 flat panels, the Cirrus Logic VGA BIOS supports panel-only and SimulSCAN operations with the standard VGA display modes listed in Section 4.8 and the extended VGA display modes listed in Table 4-9.

- Within the table, the ‘Minimum MCLK’ column gives the recommended memory clock frequency at which the CL-GD755X can run without adverse effects to functionality. Better benchmarks can be achieved with an MCLK frequency higher than the frequency specified.
- DSTN flat panels require display memory for frame accelerator functionality.
- For SimulSCAN operation:
  - — Both the flat panel and the CRT monitor must be configured at a minimum to support the resolution of a given SimulSCAN display mode.
  - — The Expanded Display mode is disabled by default.

Table 4-6. Panel-Only and SimulSCAN™ Display Modes for 640 x 480 Flat Panels

Extended VGA Display Mode Number (hex)	VESA Display Mode Number (hex)	Color Depth	Charac-ters × Rows	Character Cell (pixels)	Spatial Resolution (pixels)	Type of Flat Panel	VCLK (MHz)	Mini-mum MCLK (MHz)	Required CVDD Voltage (Min.)
5E	100	256/256K	80 × 25	8 × 16	640 × 400	DSTN/TFT	25	50	3.3
5F	101	256/256K	80 × 30	8 × 16	640 × 480	DSTN/TFT	25	50	3.3
64	111	64K	80 × 30	8 × 16	640 × 480	DSTN/TFT	25	50	3.3
66[a]	110	32K[a]	80 × 30	8 × 16	640 × 480	DSTN/TFT	25	50	3.3
71[b]	112	16M	80 × 30	8 × 16	640 × 480	DSTN[b] TFT	25	50	3.3

[a] This display mode is 32K direct-color packed-pixel.

[b] A minimum of 2 Mbytes of display memory are required to support a DSTN panel in this display mode.

4.9.2 Panel-Only and SimulSCAN 800 x 600 (SVGA) Display Modes

For 800 x 600 flat panels, the Cirrus Logic VGA BIOS supports panel-only and SimulSCAN operations with the standard VGA display modes listed in Section 4.8.1 and the extended VGA display modes listed in Table 4-7.

- Within the table, the 'Minimum MCLK' column gives the recommended memory clock frequency at which the CL-GD755X can run without adverse effects to functionality. Better benchmarks can be achieved with an MCLK frequency higher than the frequency specified.
- DSTN flat panels require display memory for frame accelerator functionality.
- On 800 x 600 flat panels, when expansion to 800 x 600 is disabled, display modes with resolutions less than 800 x 600 are displayed at a 640 x 480 resolution.
- For SimulSCAN operation with a CRT monitor and an 800 x 600 flat panel:
  - Both the CRT monitor and the flat panel must be configured at a minimum to support the resolution of a given display mode during SimulSCAN operation.
  - For the 800 x 600 flat panels, all resolutions use 800 x 600 timing. As a result, even if an 800 x 600 flat panel displays a 640 x 480 display mode, the SimulSCAN operation is not allowed when using a CRT monitor configured for a maximum resolution of 640 x 480.

Table 4-7. Panel-Only and SimulSCAN™ Display Modes for 800 x 600 Flat Panels

Extended VGA Display Mode No. (hex)	VESA Display Mode No. (hex)	Color Depth	Charac-ters × Rows	Charac-ter Cell (pixels)	Spatial Resolution (pixels)	Expand from 640 × 480 to 800 × 600?	Type of Flat Panel	VCLK (MHz)	Mini-mum MCLK (MHz)	Required CVDD Voltage (Min.)
58, 6A[a]	102	16/256K	100 × 37	8 × 16	800 × 600	–	DSTN	40	60	3.3
							TFT	40	50	3.3
5C	103	256/256K	100 × 37	8 × 16	800 × 600	–	DSTN	40	60	3.3
							TFT	40	50	3.3
5E	100	256/256K	80 × 25	8 × 16	640 × 400	Yes	DSTN	40	60	3.3
							TFT	40	50	3.3
5F	101	256/256K	80 × 30	8 × 16	640 × 480	Yes	DSTN	40	60	3.3
							TFT	40	50	3.3
64	111	64K	80 × 30	8 × 16	640 × 480	Yes	DSTN	40	60	3.3
							TFT	40	50	3.3
65[b]	114	64K	100 × 37	8 × 16	800 × 600	–	DSTN[b]	40	60	3.3
							TFT	40	50	3.3
66	110	32K[c]	80 × 30	8 × 16	640 × 480	Yes	DSTN	40	60	3.3
							TFT	40	50	3.3
67[b]	113	32K[c]	100 × 37	8 × 16	800 × 600	–	DSTN[b]	40	60	3.3
							TFT	40	50	3.3
71[b]	112	16M	80 × 30	8 × 16	640 × 480	Yes	DSTN[b]	40	66	3.3
							TFT	40	50	3.3

Table 4-7. Panel-Only and SimulSCAN™ Display Modes for 800 x 600 Flat Panels *(cont.)*

Extended VGA Display Mode No. (hex)	VESA Display Mode No. (hex)	Color Depth	Charac-ters × Rows	Charac-ter Cell (pixels)	Spatial Resolution (pixels)	Expand from 640 × 480 to 800 × 600?	Type of Flat Panel	VCLK (MHz)	Mini-mum MCLK (MHz)	Required CVDD Voltage (Min.)
78[d]	115	16M	100 × 37	8 × 16	800 × 600	–	DSTN	40	66	3.3
							TFT	40	50	3.3

[a] Graphics display mode 6Ah must be used, rather than graphics display mode 58h, for application programs to retain compatibility with other VGA BIOS products.

[b] A minimum of 2 Mbytes of display memory are required to support a DSTN panel in this display mode.

[c] This display mode is 32K direct-color packed-pixel.

[d] A minimum of 2 Mbytes of display memory are required to support all of the capabilities of this display mode.

4.9.3 Panel-Only and SimulSCAN 1024 x 768 (XGA) Display Modes

For 1024 x 768 flat panels, the Cirrus Logic VGA BIOS supports panel-only and SimulSCAN operations with the standard VGA display modes listed in Section 4.8.1 and the extended VGA display modes in Table 4-8.

- Within the table, the 'Minimum MCLK' column gives the recommended memory clock frequency at which the CL-GD755X can run without adverse effects to functionality. Better benchmarks can be achieved with an MCLK frequency higher than the frequency specified.
- For SimulSCAN operation with 1024 x 768 flat panels, both the CRT monitor and the flat panel must be configured at a minimum to support the resolution of a given display mode during SimulSCAN operation.

Table 4-8. Panel-Only and SimulSCAN Display Modes for 1024 x 768 Flat Panels

Extended VGA Display Mode Number (hex)	VESA Display Mode Number (hex)	Color Depth	Characters × Rows	Character Cell (pixels)	Spatial Resolution (pixels)	Expand from 640 × 480 to 800 × 600?	Type of Flat Panel	VCLK (MHz)	Minimum MCLK (MHz)	Required CVDD Voltage (Min.)
58, 6A[a]	102	16/256K	100 × 37	8 × 16	800 × 600	–	TFT/DSTN	65	50	3.3
5C	103	256/256K	100 × 37	8 × 16	800 × 600	–	TFT/DSTN	65	50	3.3
5D	104	16/256K	128 × 48	8 × 16	1024 × 768	–	TFT/DSTN	65	50	3.3
5E	100	256/256K	80 × 25	8 × 16	640 × 400	Yes	TFT/DSTN	65	50	3.3
5F	101	256/256K	80 × 30	8 × 16	640 × 480	Yes	TFT/DSTN	65	50	3.3
60	105	256/256K	128 × 48	8 × 16	1024 × 768	–	TFT/DSTN	65	50	3.3
64	111	64K	80 × 30	8 × 16	640 × 480	Yes	TFT/DSTN	65	50	3.3
65	114	64K	100 × 37	8 × 16	800 × 600	–	TFT/DSTN	65	50	3.3
66[b]	110	32K	80 × 30	8 × 16	640 × 480	Yes	TFT/DSTN	65	50	3.3
67[b]	113	32K	100 × 37	8 × 16	800 × 600	–	TFT/DSTN	65	50	3.3
68[b,c]	116	32K	128 × 48	8 × 16	1024 × 768	–	TFT/DSTN	65	50	3.3
71	112	16M	80 × 30	8 × 16	640 × 480	Yes	TFT/DSTN	65	50	3.3
74[c]	117	64K	128 × 48	8 × 16	1024 × 768	–	TFT/DSTN	65	50	3.3
78[d]	115	16M	100 × 37	8 × 16	800 × 600	–	TFT	65	50	3.3

[a] Graphics display mode 6Ah must be used, rather than graphics display mode 58h, for application programs to retain compatibility with other VGA BIOS products.

[b] This display mode is 32K direct-color packed-pixel.

[c] A minimum of 2 Mbytes of display memory are required to support all of the capabilities of this display mode.

[d] A minimum of 2 Mbytes of display memory are required to support all of the capabilities of this display mode.

4.10 TV-Output Modes Supported in the VGA BIOS

The CL-GD7555 supports TV-Out through the use of an external TV Encoder (i.e., AITech 1106KLC VSPro or Chrontel CH7001). The CL-GD7556 supports TV-Out through the use of an external TV Encoder, or with its own internal TV encoding which drives an external analog encoder (i.e., AD720 RGB-to-NTSC Encoder).

4.10.1 TV-Out Support Using an External TV Encoder (Applies to Both CL-GD7555 and CL-GD7556)

The CL-GD755X VGA BIOS can support TV-output timing relationships within the core 48K BIOS, when an external TV Encoder is attached to the CL-GD755X. Both NTSC and PAL formats are included. The NTSC (National Television Standards Committee) is the video standard that is used primarily in North America for home television sets. The PAL (Phase Alternation Line) standard is the video standard that is used primarily in Europe for home television sets. Only one set of PAL timing parameters can be contained within the BIOS. The PAL timing parameters can be altered by using the OEMSI utility described in Chapter 5.

A CRT can be used simultaneously in the External TV-Out mode, if the TV Encoder can use the same timing relationships as the CRT. When a CRT is not connected, an external 75 ohm termination resistor must be added to each of the R,G and B outputs. See Applications Note 7555-AN-13 “Using the VGA-to-NTSC/PAL Encoder with the CL-GD755X LCD/CRT Controller” for details on the capabilities and limitations of the external TV-out function.

When external TV-out is enabled, the CL-GD755X VGA BIOS supports operations with the Standard VGA display modes listed in Table 4-4 on page 4-20 and the Extended VGA display modes listed in Table 4-9.

Table 4-9. TV-Out Extended Display Modes

Extended VGA Display Mode Number (hex)	VESA Display Mode Number (hex)	Color Depth	Characters × Rows	Character Cell (pixels)	Spatial Resolution (pixels)	VCLK (MHz)	Minimum MCLK (MHz)	Required CVDD Voltage (Min.)
5E	100	256/256K	80 × 25	8 × 16	640 × 400	25	50	3.3
5F	101	256/256K	80 × 30	8 × 16	640 × 480	25	50	3.3
64	111	64K	80 × 30	8 × 16	640 × 480	25	50	3.3
66[a]	110	32K[a]	80 × 30	8 × 16	640 × 480	25	50	3.3
71	112	16M	80 × 30	8 × 16	640 × 480	25	50	3.3

[a] This display mode is 32K direct-color packed-pixel.

4.10.1.1 Enabling/Disabling the External TV-Output Function

The **default** TV-Out function is different for the CL-GD7555 and CL-GD7556. See Section 4.11.5.3 on page 51 for details on the Set/Get TV Output function.

CL-GD7555

By default, **External** TV-Out is **disabled** in the CL-GD7555 VGA BIOS. When an external TV Encoder is installed in the computer, the TV-Out support must be enabled by using the OEMSI utility to set OTHER_BIN_BYTE_01[6] to '1'.

CL-GD7556

By default, **Internal** TV-Out is **enabled** in the CL-GD7556 VGA BIOS (i.e., OTHER_BIN_BYTE_01[6:5] is '11'). When an External TV Encoder is installed in the computer, the OEMSI utility must be used to set OTHER_BIN_BYTE_01[5] to '0'. When no TV-Out is supported, the OEMSI utility must be used to set OTHER_BIN_BYTE_01[6] to '0'.

4.10.2 TV-Out Support Using the Internal TV Encoding (CL-GD7556 Only)

The CL-GD7556 VGA BIOS supports TV-Output devices (that is, NTSC or PAL display devices such as a standard home television set) using an internal TV encoding scheme and an external analog encoder. When this internal TV-Out function is enabled, only the TV can be connected to the computer. The flat panel will be disabled.

An external 75 ohm termination resistor must be added to each of the R,G and B outputs, when the Internal TV-Out function is enabled.

The CL-GD7556 VGA BIOS can be used for the following types of Internal TV-Output display modes:

- All TV-Output display modes with a resolution of 640 x 480 or less and 16M colors or less can be displayed.
- All TV-Output display modes with timings for 400- and 480-scanline display modes can be displayed.

The CL-GD7556 VGA BIOS supports the Internal TV-out display modes listed in Table 4.10. In the table, the term:

- 'Excellent' indicates the display quality is considered appropriate for television displays.
- 'Good' indicates the display quality is considered acceptable for occasional use on television displays, but some flickering or decimation of single-pixel horizontal lines can occur.
- 'Limitations' indicates the display quality is affected by the size of the image, and portions of the image are clipped on the top and/or bottom of the television display screen. Also, some flickering or decimation of single-pixel horizontal lines can occur.

Table 4-10. Cirrus Logic TV Output Display Modes

Extended VGA Display Mode Number (hex)	VESA Display Mode Number (hex)	Number of Colors	Characters × Rows	Character Cell (pixels)	Display Screen Format, Column × Row (pixels)	Display Quality
01	–	16/256K	40 × 25	8 × 8	320 × 200	Excellent
03	–	16/256K	80 × 25	8 × 8	640 × 200	Good
12	–	16/256K	80 × 30	8 × 16	640 × 480	Limitations

Table 4-10. Cirrus Logic TV Output Display Modes

Extended VGA Display Mode Number (hex)	VESA Display Mode Number (hex)	Number of Colors	Characters × Rows	Character Cell (pixels)	Display Screen Format, Column × Row (pixels)	Display Quality
13	–	256/256K	40 × 25	8 × 8	320 × 200	Excellent
5E	100	256/256K	80 × 25	8 × 16	640 × 400	Good
5F	101	256/256K	80 × 30	8 × 16	640 × 480	Limitations
64	111	64K	–	–	640 × 480	Limitations
66	110	32K	–	–	640 × 480	Limitations

As shown above, the default CL-GD7556 VGA BIOS enables the Internal TV-Output display modes (i.e., OTHER_BIN_BYTE_01[6:5] is ‘11’). When an External TV Encoder is installed in the computer, the OEMSI utility must be used to set OTHER_BIN_BYTE_01[5] to ‘0’. When no TV-Out is supported, the OEMSI utility must be used to disable the TV Output display mode support by clearing OTHER_BIN_BYTE_01[6] to ‘0’.

4.11 Interrupt 10h Interface Extensions

The CL-GD755X VGA BIOS supports all standard VGA BIOS Interrupt 10h interrupt services. In addition, the BIOS provides extensive support for various features of the CL-GD755X 64-Bit Video and Graphics LCD/CRT Drivers and Utilities. These functions are available as extended functions under Interrupt 10h.

The standard VGA BIOS Interrupt 10h interrupt services, which are described in IBM documentation, are not described in this document. All extended function calls preserve the host CPU registers, except those that are used to pass information from the Cirrus Logic VGA BIOS.

4.11.1 Extended Function Summary

Table 4-11 provides a summary of the extended functions that the CL-GD755X VGA BIOS provides.

NOTES:

1) Function calls are used to set options that affect the specific display device types listed in the ‘Display Device Type’ column. Within a Display Device Type column:
   - 1a) A (✓) indicates the Extended Function has an effect on the display device type.
   - 1b) A (blank) indicates the Extended Function has no effect on the display device type.
   - 1c) Except for function 92h, functions can be called from any display device type.
   - 1d) When the display device type is set for TV output, function 92h has no effect.When the display device type is set for a flat panel, the options that are selected by functions that affect flat panel displays take effect.

Table 4-11. Summary of Extended Function Calls

AH Register	BL Register	Name of Extended Function Call	Page Number within This Document	Display Device Type			
				Panel Only	CRT Only	Simul-SCAN	TV Output
12h	80h	Inquire VGA Controller Type	4-33	✓	✓	✓	✓
12h	81h	Inquire VGA BIOS Version Number	4-34	✓	✓	✓	✓
12h	82h	Inquire CL-GD755X Design Revision Code	4-34	✓	✓	✓	✓
12h	83h	Auto Display Toggle	4-39	✓	✓	✓	
12h	85h	Return Amount of Installed Memory	4-34	✓	✓	✓	✓
12h	86h	Select Memory Page	4-46	✓	✓	✓	✓
12h	8Fh	Expanded Display Mode	4-36	✓		✓	
12h	92h	Set Display Device Type	4-38	✓	✓	✓	
12h	94h	Power-Save Modes	4-44	✓	✓	✓	✓
12h	94h	Quick Subsystem Initialization	4-45	✓			
12h	94h	Quick Setup Mode	4-45	✓			
12h	9Ah	Inquire CL-GD755X User Options	4-35	✓	✓	✓	✓
12h	9Bh	Voltage Operation (CL-GD7555 only)	4-46	✓	✓	✓	✓
12h	9Ch	Set or Inquire Flat Panel Information	4-40	✓	✓	✓	✓
12h	A0h	Query Display Mode Availability	4-36	✓	✓	✓	✓
12h	A1h	Read CRT Monitor Type	4-49	✓	✓	✓	✓
12h	A4h	Set CRT Monitor Type (Vertical)	4-50		✓		
12h	A6h	Inquire Linear Frame Accelerator Address	4-36	✓	✓	✓	✓
12h	B0h	Set/Get TV Output	4-51	✓	✓	✓	✓
12h	B8h	Set Hardware Icon Function	4-41	✓	✓	✓	✓
4Fh	10h	Display Power-Management Capabilities	4-47	✓	✓	✓	✓
4Fh	15h	Display Device Identification Extensions	4-52	✓	✓	✓	✓

4.11.2 Inquiry Functions

This section describes inquiry functions.

4.11.2.1 Inquiry Function: Inquire VGA Controller Type

This inquiry function provides a mechanism for software to determine the type of Cirrus Logic VGA controller, its silicon revision number, and its corresponding hardware capabilities. (The VGA types that are of interest to readers of this document are given in **bold** type.)

Input: AH = 12h
AL = 00h
BL = 80h

Output: AL = VGA Controller type

0h	=	No extended alternate select support
1h	=	Reserved
2h	=	CL-GD510/520
3h	=	CL-GD610/620
4h	=	CL-GD5320
5h	=	CL-GD6410
6h	=	CL-GD5410
7h	=	CL-GD6420
8h	=	CL-GD6412
10h	=	CL-GD5401
11h	=	CL-GD5402
12h	=	CL-GD5420
13h	=	CL-GD5422
14h	=	CL-GD5424
15h	=	CL-GD5426
16h	=	CL-GD5420r1
17h	=	CL-GD5402r1
18h	=	CL-GD5428
19h	=	CL-GD5429
1Ah	=	CL-GD5425
20h	=	CL-GD6205
21h	=	CL-GD6215
22h	=	CL-GD6225
23h	=	CL-GD6235
24h	=	CL-GD6245
30h	=	CL-GD5432
31h	=	CL-GD5434
32h	=	CL-GD5430 and CL-GD5440
33h	=	CL-GD5434 Rev E and F
36h	=	CL-GD5436
40h	=	CL-GD6440
41h	=	CL-GD7542
42h	=	CL-GD7543
43h	=	CL-GD7541
44h	=	CL-GD7548
46h	**=**	**CL-GD7555**
49h	**=**	**CL-GD7556**

AH = Reserved
BL = Silicon revision number

0-7Fh	=	Silicon revision
80h	=	Silicon revision number not available

4.11.2.2 Inquiry Function: Inquire VGA BIOS Version Number

This inquiry function provides a mechanism for software to determine the VGA BIOS version number.

Input:	AH	=	12h
	AL	=	00h
	BL	=	81h
Output:	AH	=	Major VGA BIOS version number
	AL	=	Minor VGA BIOS version number

Example: For example, if the VGA BIOS version is 1.02, the AH output is 01 and the AL output is 02.

4.11.2.3 Inquiry Function: Inquire CL-GD755X Design Revision Code

This inquiry function provides a mechanism for software to determine the revision of the CL-GD755X silicon.

Input:	AH	=	12h
	AL	=	00h
	BL	=	82h
Output:	AX	=	CL-GD755X revision, in ASCII code.

Example: For example, if the CL-GD755X silicon revision is BC, the output for AH is ‘B’ and the output for AL is ‘C’.

4.11.2.4 Inquiry Function: Return Amount of Installed Display Memory

This inquiry function returns the amount of installed display memory, as well as the minimum amount of available display memory present (in 64-Kbyte units).

NOTES:

1) On some systems, use of the frame accelerator feature or other features reduces the amount of available video memory.
2) This function must not be used to determine the availability of display modes. [The availability of a display mode depends upon other factors such as (1) if the display device type is either a CRT monitor or a flat panel or, (2) if the display device is a flat panel, the type of flat panel that is used.] Instead, to determine if a display mode is available, use the Query Display Mode Availability function A0h.
3) Some older Cirrus Logic products do not use the AH register to return the amount of available memory. On those products, when this function is used, the value does not change and remains 12h.

Input:	AH	=	12h
	AL	=	00h
	BL	=	85h
Output:	AH	=	Amount of display memory available, given in 64-Kbyte units.
	AL	=	Amount of display memory installed, given in 64-Kbyte units.

4.11.2.5 Inquiry Function: Inquire CL-GD755X User Options

For the ‘Inquire CL-GD755X User Options’ inquiry function, subfunction 00h returns the current status of user options, defined below.

Input:	AH	=	12h			
	AL	=	00h			
	BL	=	9Ah			
Output:	AX	=	Bits 7:0	=	Reserved	
			Bit 8	=	Display mode (‘0’ = Panel-only) (‘1’ = CRT-only or SimulSCAN)
			Bit 9	=	Expanded Text Display mode (‘0’ = Enable) (‘1’ = Disable)
			Bit 10	=	Expanded Graphics Display mode (‘0’ = Enable) (‘1’ = Disable)
			Bits 14:11	=	Reserved
			Bit 15	=	Voltage operation (‘0’ = 5.0 V) (‘1’ = 3.3 V)
	BX	=	Bits 8:0	=	Reserved
			Bit 9	=	SimulSCAN (‘0’ = Enable) (‘1’ = Disable)
			Bit 10	=	Reserved
			Bit 11	=	Reserved
			Bits 13:12	=	Vertical frequency for a 640 x 480 display mode
			Bits 15:14	=	Reserved
	CX	=	Bit 0	=	Reserved
			Bits 3:1	=	Vertical frequency for a 1280 x 1024 display mode
			Bits 7:4	=	Maximum vertical resolution
			Bits 11:8	=	Vertical frequency for a 800 x 600 display mode
			Bits 15:12	=	Vertical frequency for a 1024 x 768 display mode
	DX	=	Bits 7:0	=	Standby power timer (in minutes)
			Bits 11:8	=	Power save (Refer to the Power-Save function, function 94h.)
			Bits 15:12	=	Reserved

For the ‘Inquire CL-GD755X User Options’ inquiry function, subfunction 01h returns the extended user options, defined below. (While subfunction 01h is not currently used, it is reserved for future use.)

Input:	AH	=	12h		
	AL	=	01h		
	BL	=	9Ah		
Output:	AH	=	Bits 15:8	=	Reserved
	AL	=	Bits 7:4	=	Reserved
			Bits 3:0	=	Flat Panel Type (in hex) - (see Table 4-1 on page 4-8)
	BX	=	Bits 15:0	=	Reserved
	CX	=	Bits 15:0	=	Reserved
	DX	=	Bits 15:0	=	Reserved

4.11.2.6 Inquiry Function: Query Display Mode Available

This inquiry function determines when a display mode is currently available and returns pointers to standard and supplemental parameters.

NOTES:

1) For information on the parameter table structure, refer to Section 4.12.3.
2) Output in AH bits 1 and 2 reflect the state of the current hardware and not the state of current software options.
3) To inquire about the current display option settings, use function 9Ah.

Input:	AH	=	12h
	BL	=	A0h
	AL	=	Bit 7 ‘0’ = Display mode number is in AL bits 6:0 ‘1’ = VESA display mode number is in SI
	AL	=	Bits 6:0 Display mode number (from 0 to 7Fh)
	SI	=	VESA display mode number
Output:	AH	=	Bit 0 (‘0’ = Display mode is not supported.) (‘1’ = Display mode is supported.) Bit 1 (‘0’ = CRT monitor display mode is disabled.) (‘1’ = CRT monitor display mode is enabled.) Bit 2 (‘0’ = Flat panel display mode is disabled.) (‘1’ = Flat panel display mode is enabled.) Bit 3 (‘0’ = Flat panel display is not supported for requested display mode.) (‘1’ = Flat panel display is supported for requested display mode.)
	DS:SI=		Pointer to standard display mode parameters or FFFF:FFFFh if standard parameters are undefined for this display mode.
	ES:DI=		Pointer to supplemental display mode parameters or FFFF:FFFFh if supplemental parameters are undefined for this display mode.
	BX	=	Reserved. Value is changed by this function call.

4.11.2.7 Inquiry Function: Inquire Linear Frame Accelerator Address

This inquiry function provides a mechanism for software to determine the address of the linear frame accelerator. (PCI bus systems can use standard INT 1Ah calls to determine the address of the linear frame accelerator.)

NOTE: This function is removed from this specification.

4.11.3 Flat Panel Functions

This section describes flat panel functions that can be modified by the CL-GD755X VGA BIOS.

4.11.3.1 Flat Panel Function: Expanded Display Mode Function

This flat panel function controls the vertical and horizontal expansion of text and graphics display modes, so that as much as possible of the flat panel display screen is filled. When the flat panel is the active display device, this function takes effect immediately.

This option applies to the SimulSCAN mode and the panel-only mode.

On VGA / SVGA / XGA panels, for display modes that have a spatial resolution that is:

- Equal to the spatial resolution of the flat panel, the expansion algorithm does not apply.
- Less than the spatial resolution of the flat panel, the expansion algorithm does apply, and there is an expansion effect.
- For 640 x 480 (VGA), 800 x 600 (SVGA) or for 1024 x 768 (XGA) panels, the non-expanded mode of the display is centered automatically. (Refer to Section 4.11.3.2.)

Set register SR28[7] =1 to block this function from executing, if expansion enable/disable switching is not safe. The BIOS default is SR28[7]=0 which allows text and graphics expansion to be set normally.

Input:	AH =	12h	
	BL =	8Fh	
	AL[6:0]=	0	Enable the Expanded Text/Graphics Display mode.
		1	Disable the Expanded Text/Graphics Display mode.
		2	Enable the Expanded Text Display mode only.
		3	Disable the Expanded Text Display mode only.
		4	Enable the Expanded Graphics Display mode only.
		5	Disable the Expanded Graphics Display mode only.
	AL[7] =	Inquire about the status of Expanded Display mode.	
Output:	AL[0] =	‘0’	Text Expansion Enabled
		‘1’	Text Expansion Disables
	AL[1] =	‘0’	Graphics Expansion Enabled
		‘1’	Graphics Expansion Disabled

4.11.3.2 Flat Panel Function: Set Display Device Type Function

This flat panel function selects and controls one of the following types of display device operations:

- Panel-only operation
- CRT-only operation
- SimulSCAN operation (both a panel and a CRT monitor are operating)

This panel function takes effect immediately. In using this panel function, the type of display device operation can change only if both of the following conditions are true:

- The CL-GD755X VGA BIOS display device output type is set to VGA.
- The target display device can display the current display mode.

Otherwise, if both of the conditions are not present, no display device detection occurs during the execution of this panel function (that is, there is no attempt to validate the presence of a CRT monitor).

When switching from a panel-only operation to an operation involving a CRT monitor, the Read Monitor/Type function A1h can be used to determine the existence of a CRT monitor before calling the Set Display Device Type function to switch to the CRT monitor.

Normally, for most (but not all) Cirrus Logic CL-GD754X and CL-GD755X products, when scratch pad Extension register bit SRA[5] is set to 1, the VGA BIOS does not block attempts to change the type of display device. Instead, the blocking is bypassed. However, to protect the flat panel, power to the flat panel is not cycled on.

NOTE: Whichever application program (such as the Cirrus Logic Windows driver) controls the setting of CL-GD755X Extension register bit SRA[5] must also make all necessary adjustments, before turning on the flat panel.

Input:	AH	=	12h	
	BL	=	92h	
	AL	=	0	Enable the flat panel as the current display device.
			1	Enable the CRT monitor as the current display device.
			2	Enable the SimulSCAN operation.
Output:	AH	=	Current display device type (0 = Flat panel) (1 = CRT monitor) (2 = SimulSCAN)	

4.11.3.3 Flat Panel Function: Auto Display Toggle Function

This flat panel function automatically changes the current display device to another display device, using a pre-determined sequence of display device types. A CRT monitor must be connected in the configuration for the change to take place. The pre-determined sequence is:

LCD -> CRT -> SIMULSCAN -> LCD -> CRT -> and so on

The pre-determined sequence can be broken by the following conditions:

- If no CRT monitor is present, then the ‘CRT’ and ‘SimulSCAN’ display device types are skipped in the sequence.
- If the current display mode is not suitable for the next display device, the sequence is broken.
- If the function 92h "Set Display Device Type" is used, the sequence is broken.

This flat panel function does not attempt to remember the last display device type that it toggled to. It always reads the current display device type before determining the target display device type.

For more information on using the Auto Display Toggle function, refer to a description of function 92h.

Input:	AH	=	12h
	BL	=	83h
Output:	AH	=	Current display device type (0 = Flat panel) (1 = CRT monitor) (2 = SimulSCAN)

4.11.3.4 Flat Panel Function: Set Flat Panel Type or Inquire Flat Panel Information Function

This flat panel function is normally used during POST to set the flat panel type. This function can be used instead of reading panel-type switches on the motherboard.

By setting bit 7 of AL to 1, the flat panel type setting is bypassed and an inquiry is performed for current flat panel information. This set function takes effect only for the CRT-only mode.

IMPORTANT: The CL-GD755X VGA BIOS does not check or validate the number of the flat panel type.

Input:	AH	=	12h
	BL	=	9Ch
	AL	=	Bits 6:0 Flat panel type number (such as from 00h to 0Fh)
	AL	=	Bit 7 ‘0’ = Set flat panel type that is indicated in AL[6:0]. ‘1’ =Inquire about information on the current flat panel type. (AL[6:0] ignored.)
Output:	AL	=	Bit 0 (‘0’ = Color display) (‘1’ = Monochrome display) Bits 2:1 (‘00’ = Single-Scan STN) (‘01’ = Dual-Scan STN) (‘10’ = TFT) (‘11’ = Reserved) Bit 3 Reserved Bits 5:4 TFT flat panel data format (‘00’ = 9 bit) (‘01’ = 12 bit) (‘10’ = 18 bit) (‘11’ = 24 bit) Bits 7:6 Reserved
	BX	=	Horizontal size of flat panel
	CX	=	Vertical size of flat panel

4.11.3.5 Flat Panel Function: Set Hardware Icon Function

This flat panel function is used to manipulate the hardware icon function. This flat panel function is implemented for the VGA BIOS software revision 1.20 and beyond.

Set Hardware Icon Function 0: Set the Hardware Icon Properties

Input: AH = 12h
BL = B8h

AL = 00h Set the hardware icon properties
BH = Bits 1:0 = Select hardware icon number x (where x = 0, 1, 2, or 3)
Bits 7:2 = Reserved
CL = Bit 0
('0' = Disable hardware icon number *x*)
('1' = Enable hardware icon number *x*)
Bit 1
('0' = Four-color display mode)
('1' = Three-colors-and-transparent display mode)
Bit 2
('0' = Hardware icon blinks at a steady state)
('1' = Hardware icon blinks at a state that is one-half the text-cursor blink rate)
Bit 3
('0' = Disable horizontal expansion of hardware icon image on the display screen)
('1' = Enable horizontal expansion of hardware icon image on the display screen)
Bit 4
('0' = Disable vertical expansion of hardware icon image on the display screen)
('1' = Enable vertical expansion of hardware icon image on the display screen)
Bit 5
('0' = Select Memory Map #0 for hardware icon number *x*.)
('1' = Select Memory Map #1 for hardware icon number *x*.)
Bit 7:6 = Reserved

Output: Nothing

Set Hardware Icon Function 1: Program the Color of the Hardware Icon

Input: AH = 12h
BL = B8h

AL = 01h Program the color of the hardware icon
ES:DI= Pointer to CLUT (color look-up table) buffer, where XXh = a color value from 00h to 3Fh.

	Red (Byte 0)	Green (Byte 1)	Blue (Byte 2)
(CLUT 0)	XXh	XXh	XXh
(CLUT 1)	XXh	XXh	XXh
(CLUT 2)	XXh	XXh	XXh
(CLUT 3)	XXh	XXh	XXh

Output: Nothing

Set Hardware Icon Function 2: Set the Hardware Icon Position

Input: AH = 12h
BL = B8h

AL = 02h Set the hardware icon position.
CX = X, the horizontal pixel position
DX = Y, the vertical pixel position

Output: Nothing

Set Hardware Icon Function 3: Load the Pattern for the Hardware Icon x

Input: AH = 12h
BL = B8h

AL = 03h Load the pattern for hardware icon x.
BH = Bit 0
('0' = Memory Map 0 selected for hardware icon x)
('1' = Memory Map 1 selected for hardware icon x)
Bits 2:1 = Select hardware icon number x (x = 0, 1, 2, or 3)
Bits 7:3 = Reserved
CH = The starting scanline to be updated for the requested hardware icon x.
CL = The number of scanlines to be updated for the requested hardware icon x.
ES:DI= Pointer to the hardware icon data buffer

Output: Nothing

Set Hardware Icon Function 4: Save the Video State and Set Up Access Environment

Input: AH = 12h
BL = B8h

AL = 04h Save the video state and set up the access environment for the hardware icon memory map.
BH = Bit 0
('0' = Memory Map 0 selected for hardware icon x)
('1' = Memory Map 1 selected for hardware icon x)
Bits 2:1 = Select hardware icon number x (x = 0, 1, 2, or 3)
Bits 7:3 = Reserved
ES:DI= 130-byte buffer for saving the video state of the hardware icon. (This buffer is automatically terminated with the appendage 0000h.)

Output: ES:DI= The address of the requested hardware icon memory map.

NOTE: On return, the value in the registers ES:DI is automatically replaced by the value of the address of the requested hardware icon memory map.The video state that will be saved is those registers altered by setting up access environment. This will be used with function 5.

Set Hardware Icon Function 5: Restore the Video State

Input:	AH	=	12h	
	BL	=	B8h	
	AL	=	05h	Restore the video state that was saved by the 04h function.
	ES:DI	=	130-byte buffer of function 04h that saved the video state of the hardware icon. (This buffer is automatically terminated with the appendage 0000h.)	
Output:	ES:DI	=	The address of the requested hardware icon memory map.	

Figure 4-1 demonstrates the organization of the hardware icon as it is stored within a buffer.

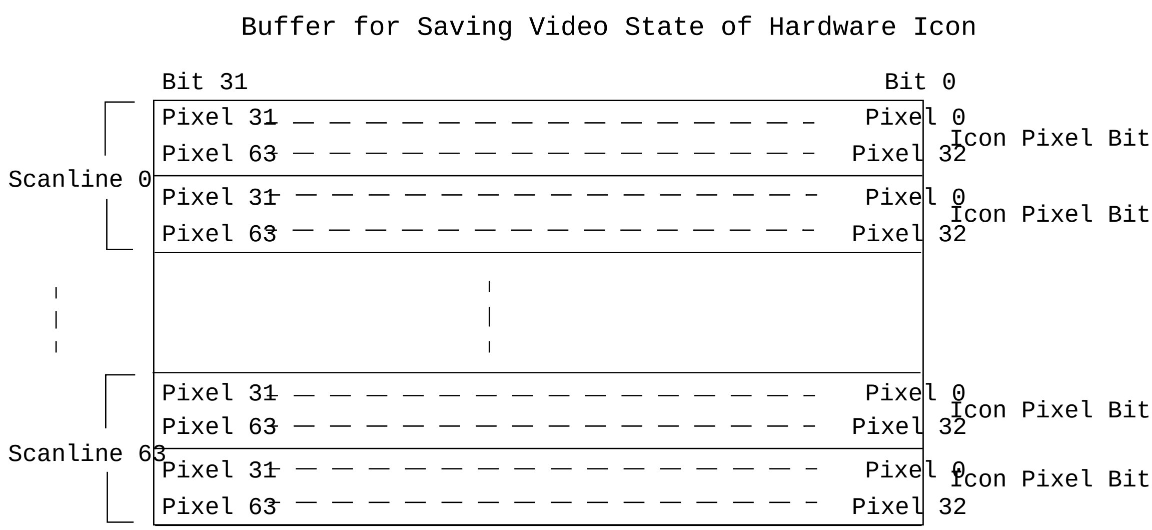


Figure 4-1. Hardware Icon (64 Pixels × 64 Pixels) Stored in a Buffer

The Table 4-12 demonstrates the selection of bits for a particular hardware icon.

Table 4-12. Hardware Icon Memory Map

Bits 2:1 (Hardware Icon)	Bit 0 (Memory Map for Selected Hardware Icon)
Hardware Icon #0	Memory Map #0
Hardware Icon #0	Memory Map #1
Hardware Icon #1	Memory Map #0
Hardware Icon #1	Memory Map #1
Hardware Icon #2	Memory Map #0
Hardware Icon #2	Memory Map #1
Hardware Icon #3	Memory Map #0
Hardware icon #3	Memory Map #1

4.11.4 Power-Management Functions

This section discusses the Cirrus Logic CL-GD755X power-management functions.

4.11.4.1 Power-Management Function: Power-Save Modes

This section discusses the three Cirrus Logic power-save modes (Standby and Suspend modes). The function for the power-save modes is given below.

Input:	AH	=	12h	
	BL	=	94h	
	BH	=	Time interval in minutes 0 – 15 (only for AL = 3h, 4h, or 7h)	
	AL	=	0h	Disable power-save modes.
			1h	Enable Standby mode immediately.
			2h	Enable Suspend mode immediately.
			3h	Enable Standby mode timer for no CPU display memory access.
			4h	Enable Standby mode timer for no port 60h or ACTI pin activity. (The ACTI pin is usually connected to the keyboard controller.)
			7h	Enable Standby mode timer for no CPU display memory access, port 60, or ACTI pin activity.
			8–0Bh	Reserved
			0Ch	Enable Suspend mode immediately and disable HSYNC and VSYNC signals to the CRT monitor.

Output: Nothing.

Standby Mode

The Cirrus Logic Standby mode is an automatic power-save feature. When no activity is detected on the keyboard or there is no attempt to access display memory for a pre-programmed time period, the Standby mode is entered. The Standby mode conserves power in the following manner:

- By not executing CRT monitor refresh
- By stopping the RAMDAC and then bringing it up only if a RAMDAC I/O access occurs
- By switching off power to the flat panel
- By not clocking the CRT Controller
- By disabling HSYNC (but a 32-kHz VSYNC signal to a CRT monitor remains active)

(For information on VESA DPMS (display power-management specification) power-saving states, refer to Section 4.11.4.6 or appropriate VESA documentation.)

Suspend Mode

The Cirrus Logic Suspend mode provides the lowest power consumption. The Suspend mode can be entered in one of two ways:

- Hardware-controlled Suspend mode
  - The hardware-controlled Suspend mode can be entered through the SUSPI pin. However, when the hardware-controlled Suspend mode is entered through the SUSPI pin, the Suspend mode must also be exited through the SUSPI pin.
  - The hardware-controlled Suspend mode does not allow display memory or I/O accesses.
- The software-controlled Suspend mode can be entered and exited through the VGA BIOS function 94h.
  - The software-controlled Suspend mode allows I/O accesses.

All internal clocks, except the internally generated memory refresh mode, are stopped and the panel is powered off. VSYNC is then disabled and a 32-kHz HSYNC signal remains active to the CRT monitor.

4.11.4.2 Power Management Function: Quick Subsystem Initialization

This power-management function performs a minimal initialization of the VGA subsystem.

Upon completion of this function, the VGA subsystem is awake and has initialized and configured some of the hardware. However, the VGA subsystem is not in a mode for outputting characters, and so forth. At this point, the only operations that can be performed on the VGA subsystem are the following:

- The VGA Save/Restore State function (AH = 1Ch) can be used to restore a previously saved state.
- The Quick Setup Mode function (AX = 1206h, BL = 94h) can be used to initialize the VGA subsystem to a state where error and status messages can be displayed on the flat panel.

The System Reset Status Flag (0:472h) is examined by this function. To ensure complete initialization of the VGA subsystem, this flag must be set to a value other than 1234h. No other memory outside of the VGA BIOS address space is read to or written by this function.

Input:	AH	=	12h	
	BL	=	94h	
	AL	=	05h	Quick Subsystem Initialization
Output:	AL	=	Nothing	

4.11.4.3 Power-Management Function: Quick Set Mode

This power-management function performs a minimal VGA mode initialization to set up the VGA subsystem to allow simple status and error messages to be displayed.

This function mainly sets up the specific text display mode environment. It sets the registers and loads the 8 x 16 font image into display memory. To blank the display screen, this function clears the first 4 Kbytes of display memory. Then it allows application programs to manipulate B800:000h for displaying the desired text image. This function must be used only when there is no available INT 10h interrupt service.

Input:	AH	=	12h	
	BL	=	94h	
	AL	=	06h	Quick Set mode
Output:	AL	=	Nothing	

4.11.4.4 Power-Management Function: Select Memory Page

This power-management function selects a specified 64-Kbyte page and maps it to A000h. The purpose of this function is to allow the system BIOS to generate a generic routine for reading and writing display memory as part of a save or restore VGA subsystem routine that is associated with a zero-volt suspend implementation.

Before calling this function, the SVGA state must be saved using either the standard INT 10h function 1Ch or the VESA function 4F04h. The SVGA state must be restored after completing this function. Otherwise, this function alters the current register settings. To determine the number of 64-Kbyte units of display memory installed in the system, use the Return Installed Memory function 85h.

Input:	AH	=	12h
	BL	=	86h
	AL	=	Page number to select
Output:	AL	=	Nothing

4.11.4.5 Power-Management Function: Voltage Level (CL-GD7555 only)

This power-management function allows the VGA BIOS to dynamically optimize its environment for the voltage level that is being applied to the VGA subsystem.

(Note: This function is removed from CL-GD7556 BIOS)

This function:

- Determines whether certain display modes are allowed.
- Is meant to provide a signal to the VGA BIOS to give it information on the current voltage level.
  — For the current display mode, except for MCLK, the VGA BIOS does not change either the timing or the parameters. Using the MCLK value set in the OEMSI Utility MCLK_CONFIG_TABLE, the system sets the MCLK to the correct frequency for the appropriate voltage level.
- Controls the usage of one of the PROG pins. By using the OEMSI Utility, one of the three available PROG[2:0] pins can be configured to control the power-management voltage level function.
  — For 3.3-V settings, the selected PROG pin is toggled low (that is, Extension register SR2F[6] = 0).
  — For 5.0-V settings, the selected PROG pin is toggled high (that is, Extension register SR2F[6] = 1).
  — The current voltage state can be read back in function 9Ah. For implementing dynamic, mode-dependent voltage selection using this function, refer to Section 4.14 on page 60.
  — Through the OEMSI Utility, the OEM can instruct the CL-GD755X VGA BIOS to bypass the PROG pin programming for this function.

Input:	AH	=	12h	
	BL	=	9Bh	
	AL	=	0	Enable 5.0-V operation.
			1	Enable 3.3-V operation.
Output:	AL	=	Nothing	

4.11.4.6 Power-Management Function: VESA Display Power-Management Signaling

The three power-management functions in this section are defined by VESA as VBE/PM version 1.0, which is a software interface for VESA-defined DPMS (Display Power-Management Signaling). These three functions are included to specify the Cirrus Logic implementation of display power management.

Report Display Power-Management Capabilities

This power-management function reports the display power-management capabilities that are available for the current display device type. If the revision of the CL-GD755X does not support display power-management capabilities, this function is not supported.

Input:	AH	=	4Fh	VESA Extension
	AL	=	10h	VBE/PM Services
	BL	=	00h	Report VBE/PM Capabilities
	ES:DI=		Null pointer. Must be 0000:0000h. (Reserved for future use.)	
Output:	AL	=	4Fh	Function is supported.
	!	=	4Fh	Function is not supported.
	AH	=	00h	Function call is successful.
		=	01h	Function call is not successful.
	BH	=	Support for power-saving state signals for current display device type.	
			(0 = There is no support for power-saving state signals.)	
			(1 = There is support for power-saving state signals.)	
			Bit 0	Standby mode
			Bit 1	Suspend mode
			Bit 2	Power is off
			Bit 3	Reduced On (Not supported in current specification.)
			Bits 7:4	Reserved
	BL	=	VBE/PM version number	
			Bits 3:0	Minor VBE/PM version number
			Bits 7:4	Major VBE/PM version number
	ES:DI=		Unchanged	

Set Display Power State

This power-management function sets the power state of the current display.

- The Cirrus Logic function 94h, option 0 (the Disable Power-Save Modes function) serves the same purpose as the VESA function 10h, subfunction 1, option 0 (Set Display Power State function).
- For maximum power savings, place the CL-GD755X into a power-save mode, using the Cirrus Logic Power-Save Modes function 94h.
- If function 92h (Set Display Device Type function) is used to change the display device type while the display device is in a power-management state other than 'Power on', the power state is changed to 'Power on'.

Input:	AH	=	4Fh	VESA Extension
	AL	=	10h	VBE/PM Services
	BL	=	01h	Set Display Power State
	BH	=	Requested power state (00h = Power is on) (01h = Standby mode) (02h = Suspend mode) (04h = Power is off) (08h = Reserved for Reduced On)	

Output:	AL	=	4Fh	Function is supported.
	!	=	4Fh	Function is not supported.
	AH	=	00h	Function call is successful.
		=	01h	Function call is not successful.

Get Display Power State

This power-management function returns the current requested display power state. If the revision of the CL-GD755X does not support display power-management capabilities, this function is not supported.

Input:	AH	=	4Fh	VESA Extension
	AL	=	10h	VBE/PM Services
	BL	=	02h	Get Display Power State

Output:	AL	=	4Fh	Function is supported.
	!	=	4Fh	Function is not supported.
	AH	=	00h	Function call is successful.
		=	01h	Function call is not successful.
	BH	=	Current requested power state (00h = Power is on) (01h = Standby mode) (02h = Suspend mode) (04h = Power is off) (08h = Reserved for Reduced On)	

4.11.5 CRT Monitor Functions

4.11.5.1 CRT Monitor Function: Read CRT Monitor Type

This CRT monitor function uses analog sensing to determine the type of CRT monitor that is being detected.

Input:	AH	=	12h		
	AL	=	00h		
	BL	=	A1h	Read CRT monitor type	
Output:	BH	=	Reserved		
	BL	=	CRT monitor type		
			0h	=	Color CRT monitor
			1h	=	Monochrome CRT monitor
			2h	=	No CRT monitor

4.11.5.2 CRT Monitor Function: Set CRT Monitor Type (Vertical)

This CRT monitor function sets the CRT monitor type in terms of vertical timings. The CRT monitor type information is used by the VGA BIOS:

- To determine which frequency to use when selecting an extended display mode.
- To define what display mode resolutions are available.

The vertical CRT monitor type information can be read back by using function 9Ah.

Input: AH = 12h
BL = A4h
AL = Bits 3:0 = Maximum vertical resolution, in scanlines
(0h = 480 scanlines)
(1h = 600 scanlines)
(2h = 768 scanlines)
(3h = 1024 scanlines)
(4h:0Fh = Reserved)
Bits 5:4 = Frequency for a 640 x 480 display mode
(0h = 60 Hz. This is the standard frequency.)
(1h = 72 Hz. This is a high-refresh frequency.)
(2h = 75 Hz. This is a high-refresh frequency.)
(3h = 85 Hz. This is a high-refresh frequency.)
Bits 7:6 = Reserved
BH = Bits 3:0 = Frequency for a 800 x 600 display mode
(0h = 56 Hz)
(1h = 60 Hz)
(2h = 72 Hz)
(3h = 75 Hz)
(4h = 85 Hz)
(5h–0Fh = Reserved)
Bits 7:4 = Frequency for a 1024 x 768 display mode
(0h = 43.5 Hz or 87 Hz for an interlaced display mode)
(1h = 60 Hz)
(2h = 70 Hz)
(3h = 72 Hz)
(4h = 75 Hz)
(5h = 85 Hz)
(6h–0Fh = Reserved)
CH = Bits 3:0 = Reserved
Bits 7:4 = Frequency for a 1280 x 1024 display mode
(0h = 43.5 Hz or 87 Hz for an interlaced display mode)
(1h = 60 Hz)
(4h–Fh = Reserved)
CL = Bits 7:0 = Reserved
DX = Bits 15:0 = Reserved

Output: Nothing

4.11.5.3 CRT Monitor Function: Set/Get TV Output

This CRT monitor function selects the output type for a display device.

- The VGA output type is defined as the output type configured by the Set Display Device Type function of Section 4.11.3.2 on page 38.
- An NTSC or PAL output type is possible for those CL-GD755X systems that are configured with an external TV Encoder, or for those CL-GD7556 systems that are configured with an external RGB-to-NTSC/PAL analog encoder.

For an NTSC or PAL output type, the Set/Get TV Output function is used to alter the display device output type for either NTSC or PAL timings. The output type and the display timings are changed immediately.

The TV Output is valid only for display modes that have a resolution either at, or below, 640 x 480 (see Section 4.10.1 on page 29). Once this Set/Get TV Output function is used to select a ‘TV Out’ output type (that is, either an NTSC or a PAL output type), calls to the Set Display Device Type function of Section 4.11.3.2 are ignored until the Set/Get TV Output function returns the VGA output type.

CL-GD7556 Internal TV-Out Function

By default, the CL-GD7556 VGA BIOS assumes support for the internal TV encoder is enabled, and an external RGB-to-NTSC/PAL analog encoder is attached to the CL-GD7556.

CL-GD7555 and CL-GD7556 External TV-Out Function

By default, the VGA BIOS assumes that no external TV Encoder is attached to the CL-GD755X. As a result, for those systems that are configured with an external TV Encoder, the OEMSI Utility is used to enable the support of the external TV Encoder circuitry (see Section 4.10.1.1 on page 30). A standard home television set can be the boot device, if bits are properly set by using the OEMSI Utility.

To perform the Set TV Output function, use the parameters described in the table that follows.

- If TV Circuitry is not supported, BH[0] is cleared to 0 upon return from this function.

To perform the Get TV Output function and to determine the current output type for a display device:

- In the Set/Get TV Output function, AL, use a value of 0Fh. If TV Circuitry is not supported, BH[0] is cleared to 0 upon return from this function.

Input:	AH	=	12h	
	BL	=	B0h	
	AL	=	Bits 3:0	Display Device Output Type (0h = VGA Output Type – no TV Out) (1h = NTSC Output Type) (2h = PAL Output Type) (3h:Eh = Reserved) (Fh = Request status of TV Out type)
			Bits 7:4	Reserved
Output:	AL	=	Bits 3:0	Display Device Output Type (0h = VGA Output Type – no TV Out) (1h = NTSC Output Type) (2h = PAL Output Type) (3h:Fh = Reserved)
			Bits 7:4	Reserved
	BH	=	Bit 0	‘0’ = TV Circuitry is not supported. ‘1’ = TV Circuitry is supported.
			Bit 7:1	Reserved

4.11.5.4 CRT Monitor Function: Display Device Identification Extensions

The CRT monitor functions that are implemented are defined by the VESA VBE/DDC specification, version 1.0, which is a software interface to DDC. These functions are included to specify the Cirrus Logic DDC implementation, which supports only DDC2B.

OEMs who do not wish to support DDC can use the OEMSI Utility to disable the DDC interface.

Report DDC Capabilities

This CRT monitor function reports the DDC capabilities available with the current CRT monitor.

Input:	AH	=	4Fh	VESA Extension
	AL	=	15h	VBE/DDC Services
	BL	=	00h	Report DDC Capabilities
	CX	=	00h	Controller Unit Number (00 = Primary Controller)
	ES:DI=		Null pointer. Must be 0000:0000h in version 1.0. (Reserved for future use.)	
Output:	AL	=	4Fh	Function is supported.
	!	=	4Fh	Function is not supported.
	AH	=	00h	Function call is successful.
			01h	Function call is not successful.
	BH	=	Approximate time in seconds, rounded up, to transfer one EDID (Extended Display ID) block, which consists of 128 bytes.	
	BL	=	DDC supported by both the CRT monitor display device and the CL-GD755X controller. Bit 0 (‘0’ = DDC1 is not supported.) (‘1’ = DDC1 is supported.) Bit 1 (0 = DDC2 is not supported.) (1 = DDC2 is supported.) Bit 3 (‘0’ = Display screen is not blanked during a data transfer.) (‘1’ = Display screen is blanked during a data transfer.)	

Read EDID (Extended Display Identification)

This CRT monitor function reads the EDID (the VESA Extended Display Identification) and returns the ID code in the buffer that is passed in the function input.

Input:	AH	=	4Fh	VESA Extension
	AL	=	15h	VBE/DDC Services
	BL	=	01h	Read EDID
	CX	=	00h	Controller Unit Number (00 = Primary Controller)
	DX	=	00h	EDID block number. Zero is the only valid value in version 1.0.
	ES:DI	=	Pointer to area in which the 128-byte EDID block (a byte stream) is returned.	
Output:	AL	=	4Fh	Function is supported.
	!	=	4Fh	Function is not supported.
	AH	=	00h	Function call is successful.
		=	01h	Function call is not successful.
	BH	=	Unchanged	
	CX	=	Unchanged	
	ES:DI	=	Pointer to area in which the 128-byte EDID block (a byte stream) is returned.	

4.12 Extended Display Modes in RAM

4.12.1 Extensions to the Save Area Table

The Cirrus Logic VGA BIOS versions support standard VGA-compatible display modes and have a set of extended display modes. OEMs can add new display modes to the system or can redefine existing display modes that are in the BIOS ROM by manipulating the VGA BIOS Save-Area table, pointed to by 0040:00A8h. The Save-Area table is located in the BIOS ROM after the system boots. Any changes to the Save-Area table must be made in a RAM copy.

Cirrus Logic has extended the definition of the Save-Area Table with 'negative' offsets that point to parameters that are defined by Cirrus Logic. The VGA-compatible table, along with the defined extensions, follows.

Offset	TypeDescription
-1Ch	WordReserved
-1Ah	WordReserved
-18h	DoublewordReserved
-14h	DoublewordPointer to next negative offset table in linked list
-10h	WordSet to 04h if offset -14h is a valid pointer. Set to 00h if this link is the last in RAM. To block all ROM-based modes, set this field to 04h and set offset -14h to 0:0.
-0Eh	WordSize of supplemental table
-0Ch	DoublewordPointer to extended display mode supplemental parameters
-08h	DoublewordPointer to extended display mode standard parameters
-04h	WordNumber of extended display modes
-02h	Word'RV' (revision) identifier
00h	DoublewordPointer to standard display mode standard parameters
04h	DoublewordDynamic save-area pointer (that is, the palette-save area)
08h	DoublewordAlpha mode auxiliary character-generator pointer
0Ch	DoublewordGraphics mode auxiliary character-generator pointer
10h	DoublewordSecondary save pointer
14h	DoublewordReserved and set to zero
18h	DoublewordReserved and set to zero

4.12.2 Cirrus Logic VGA BIOS Processing

The Cirrus Logic VGA BIOS determines what display mode to select by processing a linked list of extended display mode supplemental parameter tables, while evaluating factors such as the following:

- Size of display memory
- CRT monitor type or flat panel type
- Frequency of MCLK
- The voltage level of CVCC, the core voltage to the CL-GD755X

From the top down, the Cirrus Logic VGA BIOS services a request to set a display mode once all the evaluated factors are satisfied. For example, a display mode that has multiple horizontal frequencies must have the frequencies sequentially ordered, with the highest frequencies at the top and the lowest frequencies at the bottom. This order ensures that the Cirrus Logic VGA BIOS always sets the correct display mode for a given CRT monitor or panel type.

To satisfy a request to set a display mode, the VGA BIOS always looks first for the RAM-defined links. If the VGA BIOS cannot find a display mode based on the current configuration of the VGA subsystem, the VGA BIOS then scans the ROM tables.

Display modes can be added to the VGA BIOS by manipulating the structure previously described.

- If new display modes are to be added to the VGA BIOS by defining the display modes in RAM, a TSR (terminate and stay resident) utility need only to modify the negative offsets given in Section 4.12.1 (that is, modifications are made for higher refreshes of previously defined display modes or for entirely new display modes).
- If display modes are to be redefined, special care must be taken. If a TSR modifies a particular frequency of a display mode that has higher frequencies that are already defined in the BIOS ROM, all frequencies must be redefined in RAM.

4.12.3 Extended Display Mode Supplemental Parameters

The table that follows describes what the Cirrus Logic VGA BIOS expects in the supplemental structure previously discussed.

Offset	Type	Description
00h	Byte	Standard display mode number
01h	Word	VESA display mode number
03h	Word	Horizontal resolution
05h	Word	Vertical resolution
07h	Byte	Bits per color
08h	Byte	Character width
09h	Byte	Character height
0Ah	Byte	VESA memory model (Defined in VESA function 1)
0Bh	Byte	VESA display mode attributes (Defined in VESA function 1)
0Ch	Byte	Index to refresh rate fix-up table (used by the Cirrus Logic VGA BIOS). FFh means that the refresh rate is the same as the refresh rate in the standard parameter table entry.
0Dh	Byte	Pointer to standard parameter table
0Eh	Byte	External/General register MISC (3C2h) value
0Fh	Byte	Bit 0 = Minimum voltage ('0' = 5 V minimum) – (applicable to CL-GD7555 only) ('1' = 3.3 V minimum) Bits 6:1 = Extension register SR1F minimum MCLK required Bit 7 = Reserved
10h	Byte	Display memory that is required, in units of 64-Kbyte blocks
11h	Byte	Bits 3:0 = Maximum vertical resolution required Bits 6:4 = Required vertical frequency for specified resolution Bit 7 = Reserved
12h	Word	Offset address of CRT Controller fix-up table entry

(continued)

14h	Byte	Flat panel support information
		Bits 2:0 = For dual-scan panels, the maximum panel size supported for the display mode.
		('000' = No support for panel)
		('001' = 640 x 480 panel)
		('010' = 800 x 600 panel)
		('011' = 1024 x 768 panel)
		('1xx' = Reserved)
		Bit 3 = Reserved
		Bits 6:4 = For TFT panels, the maximum panel size supported for the display mode.
		('000' = No support for panel)
		('001' = 640 x 480 panel)
		('010' = 800 x 600 panel)
		('011' = 1024 x 768 panel)
		('1xx' = Reserved)
		Bit 7 = Reserved
15h	Byte	Minimum MCLK settings for 800 x 600 DSTN flat panel
		Bits 5:0= Extension register SR1F minimum MCLK required
		Bit 7:6 = Reserved
16h	Byte	Extension register SRE, VCLK3 numerator
17h	Byte	Extension register SR1E, VCLK3 denominator
18h	Byte	Extension register SR7, Extended Sequencer Mode Control
19h	Byte	Extension register GRB, Graphics Controller Mode Extensions
1Ah	Byte	Extension register CR19, Interlace End
1Bh	Byte	Extension register CR1A, Miscellaneous Control
1Ch	Byte	Extension register CR1B, Extended Display Control
1Dh	Byte	Extension register GR18, EDO RAM Control
1Eh	Byte	Extension register CR8E, Hardware Debug #1
1Fh	Byte	Extension register HDR, Hidden DAC

4.12.4 CRT Controller Fix-Up Table Parameters

The table that follows describes what the VGA BIOS expects in the CRT Controller Fix-Up table discussed in Section 4.12.3. This table structure is used to change the refresh rate for extended display modes. In this manner, the table structure for an extended display mode can be compressed by eliminating the redundant information of refresh rates that are used across resolutions and which are color-depth independent.

Offset	Size	Standard CRT Controller Register	Register Name
00	1 Byte	CR0	Horizontal Total
01	1 Byte	CR3	Horizontal Blanking End
02	1 Byte	CR4	Horizontal Sync Start
03	1 Byte	CR5	Horizontal Sync End
04	1 Byte	CR6	Vertical Total
05	1 Byte	CR7	Overflow
06	1 Byte	CR9	Character Cell Height
07	1 Byte	CR10	Vertical Sync Start
08	1 Byte	CR12	Vertical Display End
09	1 Byte	CR15	Vertical Blanking Start
0A	1 Byte	CR16	Vertical Blanking End
0B	1 Byte	CR11	Vertical Sync End
0C	1 Byte	CR1C	Sync. Adjust and GENLOCK

4.13 Naming Conventions for VGA BIOS Releases

This section describes naming conventions for Cirrus Logic VGA BIOS releases. In general, Cirrus Logic VGA BIOS products are named using the following format:

X.YZ[letter] [α | Β] NN for example: 1.21c Β02

Table 4-13 defines symbols used for this naming convention. The most important naming convention for VGA BIOS products is the Alpha/Beta level designation. Either of these designations indicate that the VGA BIOS product must not be distributed to end users. This naming convention is used for two reasons:

- The naming convention conveys in a consistent manner the significance of changes between each release of the VGA BIOS software. (To determine changes to the VGA BIOS since it was last released, refer to the release notes that accompany each released product.)
- The naming convention indicates the amount of testing that has gone into the product.

Table 4-13. Descriptions of Naming Conventions for Cirrus Logic VGA BIOS Releases

Symbol	Definition	Description of Naming Convention
X	Software architecture version number	Describes changes to VGA BIOS core functionality, such as the following: – Addition of new architecture – Removal of a key functionality A change to this numbers signifies a full Quality Assurance cycle has been performed.
Y	Major version number	Describes significant bug fixes that affect functionality or changes in the VGA BIOS functionality or specification, such as the following: – New product identification numbers – Support for new members of a chip family A change to this numbers signifies a full Quality Assurance cycle has been performed.
Z	Minor version number	Describes maintenance releases and minor bug fixes to the code. A change to this number signifies a full Quality Assurance cycle has been performed.
letter	Point release designation	Describes minor bug fixes that can be isolated in code of minor complexity, or describes VGA BIOS specification or functionality changes that take place only in data and which do not affect the code. Releases with this designation receive phase 1 Quality Assurance cycle and testing only in the area of change.
α	Alpha level	Designates the product has received little or no Quality Assurance testing, and is available only for engineering purposes or to demonstrate concepts. Absence of the Alpha (or Beta) designation indicates a product is of release quality.
Β	Beta level	Designates the product has completed a phase 1 Quality Assurance cycle and is available for both evaluation and test purposes. However, the product is not yet of release quality. Absence of the Beta (or Alpha) designation indicates a product is of release quality.
NN	Revision or build level	Designates the build level of Alpha or Beta designation. (Alpha and Beta build designations are separate.)

4.14 Dynamic Power-Switching (CL-GD7555 Only)

This is an overview of a possible way to implement a power-switching method that is sensitive to the display mode that is being used. In this example, the implementation requires that many system-dependent variables be determined, by coordinating the efforts of the CL-GD7555 VGA BIOS, the system BIOS, and the system hardware.

As described in Section 4.4.6, the CL-GD7555 VGA BIOS can be configured to make a system BIOS call while selecting the proper refresh rate within the Video Set Mode function. The system BIOS can determine if there needs to be a change in the voltage level to the video circuitry. If a change is necessary, the system BIOS must perform the voltage change. Power switching must be smooth and without glitches to avoid losing power to the CL-GD7555 and to avoid corrupting CL-GD7555 registers. After changing the voltage, the system BIOS can signal the VGA BIOS to inform it of the new voltage and return to the VGA BIOS for completion of the Video Set Mode function.

Alternatively, when signaling the CL-GD7555 VGA BIOS to inform it of the new voltage, the VGA BIOS sets the level on the PROG pin that was selected from one the three available PROG[2:0] pins. The level is set either to a high level (when 5 V is requested) or to a low level (when 3.3 V is requested). In this manner, a voltage supply circuit can be controlled by a PROG pin to provide the correct voltage. As a result, this dynamic switching of power allows application programs to set display modes that require 5 V, and yet the system can retain the power savings of a 3.3-V operation, whenever 3.3-V operation is possible. No user inputs, ‘hot keys’, or utility programs are required.

NOTES:

1) Another possible power-switching application would be to operate a system on 3.3-V battery power and then switch to 5-V power when connected to an AC adapter.
2) In general, display modes that require a 5-V operation are higher-resolution CRT monitor display modes that are typically used while connected to AC power.

4.14.1 Dynamic Power-Switching Example

In the example that follows, an application program sets a CRT monitor display screen that has a 1024 x 768 spatial resolution for a 16-color display mode (5Dh). This example makes the following assumptions:

- The CRT monitor that is used selects a 75-Hz refresh rate.
- The display mode used requires 5 V.
- A 3.3-V operation supports lower refresh rates.

NOTE: OEMs can determine acceptable voltage levels from the display mode tables. However, it is not sufficient to determine voltage requirements solely from the display mode number.

Table 4-14. Example for Implementing Dynamic Power-Switching (CL-GD7555 Only)

Step	Command or Action	Performer	Description of Action
1	INT 10h AX = 005Dh	Application program	The application program calls the Video Set Mode function.
2	Begin Video Set Mode	Cirrus Logic VGA BIOS	1. The VGA BIOS looks up the table entry for the display mode to be set. 2. The VGA BIOS determines the refresh rate from the CRT monitor type. 3. The VGA BIOS determines the necessary voltage level from the extended mode supplemental parameter table.
3	INT 15h AX = 4490h BX = 005Dh	Cirrus Logic VGA BIOS	1. The VGA BIOS calls the system BIOS. (The default function value, 4490h, can be changed using the OEMSI Utility.) 2. Within the system BIOS, BH[0] indicates the voltage requirements. a. For this example, the voltage requirement is 5 V. b. For an example that requires 3.3 V, BX would be 015Dh.
4	INT 10h AX = 1200h BL = 9Bh	System BIOS	1. The system BIOS relays the current voltage level on the CL-GD755X PROG pin to the VGA BIOS. 2. The VGA BIOS uses the output on the PROG pin to set the desired voltage, which is in AL.
5	OUT voltage_port, level	System BIOS	The voltage level is changed in one of two ways: 1. The system BIOS changes the voltage level by writing to the system port. 2. The selected PROG pin selects the correct voltage level.
6	Voltage level change	System hardware	The system hardware changes the level of the voltage that goes to the video circuitry. (This implementation is system-dependent.)
7	IRET	Cirrus Logic VGA BIOS	The VGA BIOS returns.
8	IRET	System BIOS	The system BIOS returns. After waiting for an appropriate time for the voltage to stabilize, the system BIOS returns to the VGA BIOS.
9	Continue Video Set Mode	Cirrus Logic VGA BIOS	The VGA BIOS checks the internal variables for the proper voltage level.
10	INT 15h, AX = F964h	Cirrus Logic VGA BIOS	The VGA BIOS signals to the system BIOS that the VGA BIOS is starting the set display mode process.
11	IRET	System BIOS	The system BIOS returns. After waiting for an appropriate time for the voltage to stabilize, the system BIOS returns to the VGA BIOS.
12	Display mode is set.	Cirrus Logic VGA BIOS	If the necessary voltage level is: 1. Present, the VGA BIOS sets the requested display mode. 2. Not present, the VGA BIOS exits without setting the requested display mode.
13	INT 15h, AX = F965h	Cirrus Logic VGA BIOS	The VGA BIOS signals to the system BIOS that the VGA BIOS has completed setting a display mode.
14	IRET	System BIOS	The system BIOS returns. After waiting for an appropriate time for the voltage to stabilize, the system BIOS returns to the VGA BIOS.
15	IRET	Cirrus Logic VGA BIOS	The VGA BIOS returns.
16	Continue	Application program	The application program continues.

Notes

5 OEMSI UTILITY

(Version 2.10 B8)

The OEMSI (OEM System Integration) utility is used to customize the Cirrus Logic VGA BIOS (see Chapter 4) to support specific system requirements. A wide variety of changes can be made to the VGA BIOS to modify the default behavior and features of the LCD/CRT Controller, without requiring access to the BIOS source code.

The OEMSI utility has a minimal user interface. It is driven by parameters specified from the command line. See Section 5.2 for a complete description of these parameters.

5.1 Using the OEMSI Utility

Using the OEMSI utility consists of five steps shown in the following paragaphs:

5.1.1 Create a Data File

To create a new data file for customizing, use an appropriate parameter from the listing in Section 5.3. The OEMSI utility generates (exports) a text file based on the current BIOS image in ROM at either the C000h or E000h address, or from a binary file specified on the command line.

5.1.2 Edit the Data File

The extracted data file must be altered to reflect the desired changes in the features and behavior of the VGA controller. The data file structure is shown in Section 5.3. Care must be taken to edit only the desired ‘1’ or ‘0’ bits in the specified registers. OEMSI does not have the ability to detect illegal register settings. The original register bits identified with an ‘x’ are masked out and changes are ignored, when the import file is generated.

5.1.3 Import the Modified Data File

The OEMSI utility reads (imports) the modified data file and combines it with the existing BIOS binary file. It then creates a new BIOS binary file.

5.1.4 Create (load) the BIOS ROM/EPROM

The new BIOS image must be programmed into the ROM/EPROM which is used to initialize the LCD/CRT Controller.

5.1.5 Test the Modified VGA BIOS

The final step in creating a new VGA BIOS is a thorough test of the modified program. It is important to test all of the standard functions, as well as the modified sections.

5.2 OEMSI Syntax – Command-Line Options

The operation of the OEMSI program is controlled by command parameters. The format for the command line can be either:

```
OEMSI option-1 data-file BIOS-binary-file
```

where `option-1` can be:

ParameterDescription

`-bo or -b` export OEM data from BIOS binary file to data file.

`-bp` export Parameter Table from BIOS binary file to data file.

`-bf` export Font Pattern from BIOS binary file to data file.

`-ba` export ALL from BIOS binary file to data file.

`-i` import configured data to BIOS binary file from data file.

or:

```
OEMSI option-2 data-file
```

where `option-2` can be:

ParameterDescription

`-co or -c` export OEM data from C000h ROM BIOS to data file.

`-cp` export Parameter Table from C000h ROM BIOS to data file.

`-cf` export Font Pattern from C000h ROM BIOS to data file.

`-ca` export ALL from C000h ROM BIOS to data file.

`-eo or -e` export OEM data from E000h ROM BIOS to data file.

`-ep` export Parameter Table from E000h ROM BIOS to data file.

`-ef` export Font Pattern from E000h ROM BIOS to data file.

`-ea` export ALL from E000h ROM BIOS to data file.

5.3 OEMSI Output File Format

```
REM   This OEMSI-generated text output file
REM   allows user to change the BIOS
REM   configuration.
REM   - Consult OEMSI manual for detailed
REM   definition (reference is to this manual).
REM
REM   BIOS TYPE/VERSION: CL-GD7555/6 VGA BIOS
REM   Version x.xx
REM   OEMSI VERSION: 2.10 B8
REM   BIOS SUPPORTED BY OEMSI: GD64XX,GD754X
REM   OUTPUT DATA FROM: 'File name' of BIOS
REM   binary image/ROM image
REM   CREATED AT: 'day' 'month' 'year' 'time'
REM

DOC   THE VALUES OUTPUT FROM YOUR BIOS
DOC   SOFTWARE MAY NOT BE REFLECTED EXACTLY IN
DOC   THIS DOCUMENT. THIS DOCUMENT PROVIDES
DOC   ONLY THE DEFINITION OF EACH FIELD.
DOC   *** Warning *** Do not use the text file
DOC   printed here as a source file for
DOC   creating a new BIOS program.
DOC
DOC   THE REFERENCE MATERIAL REQUIRED :
DOC   CL-GD7555/6 Hardware Reference Manual
DOC
DOC   EXAMPLES: COMMAND LINE SYNTAX
DOC   To export binary image:
DOC   "oemsi -b datafile vga.bin"
DOC   To import binary image:
DOC   "oemsi -i datafile vga.bin"

REM   DATA FORMAT : binary byte

%%%   BIN_BYTE

REM
REM   Enable BIOS reading dip switch during
REM   POST

READ_DIP_SWITCH=[1011-0000]

DOC   bit 7: 0=don't ...; 1=use dip switches
DOC          for panel
DOC   bit 6: Reserved
DOC   bit 5: 0=don't ...; 1=suppress setmode
DOC          int 15h
DOC   bit 4: 0=don't ...; 1=suppress panel
DOC          type int 15h
DOC   bit[3-0]: panel type if bit 7=0

REM
REM   Other binary byte data item(s)
REM

OTHER_BIN_BYTE_01=[0000-0000]

DOC bit 7: Reserved
DOC bit 6: 0 = all TV output support disabled
DOC          (BIOS default for CL-GD7555)
DOC          1=TV output support enabled
DOC          (BIOS default for CL-GD7556)
DOC   bit 5: 0=External TV Encoder supported
             (CL-GD7556 only)
DOC          1=Internal TV encoding supported
DOC          (BIOS default for CL-GD7556 only)
DOC   bit 4-0 : Reserved

OTHER_BIN_BYTE_02=[0000-0000]

DOC   bit 7: 0=BIOS is not relocated by
DOC          hardcode;
DOC          1=BIOS is relocated by hardcode.
DOC   bit 6: Reserved
DOC   bit 5: 0=monochrome monitor POST at
DOC          mode 7;
DOC          1=monochome POST at mode 3+sum
DOC          to-gray
DOC   bit[4-0]:  Reserved

OTHER_BIN_BYTE_03=[1001-0000]
OTHER_BIN_BYTE_04=[0100-0100]

DOC   OTHER_BIN_BYTE_03&04: Int 15h 4490h
DOC   call for
DOC   Signal the minimum_voltage required for
DOC   the requested video mode.
DOC   The call is gated by the
DOC   READ_DIP_SWITCH[5].

OTHER_BIN_BYTE_05=[1000-1110]
OTHER_BIN_BYTE_06=[0100-0100]

DOC   OTHER_BIN_BYTE_05&06: Int 15h 448Eh
DOC   call for
DOC   signal_initialization_complete.

OTHER_BIN_BYTE_07=[1000-1111]
OTHER_BIN_BYTE_08=[0100-0100]

DOC   OTHER_BIN_BYTE_07&08: Int 15h 448Fh
DOC   call for
DOC   Get the OEM panel type. The call is
DOC   gated by the READ_DIP_SWITCH[4].

OTHER_BIN_BYTE_09=[0110-0100]
OTHER_BIN_BYTE_10=[1111-1001]

DOC   OTHER_BIN_BYTE_09&10: Int 15h F964h
DOC   call for
DOC   Signal the beginning of video set mode.
DOC   The call is gated by the
DOC   READ_DIP_SWITCH[5].

OTHER_BIN_BYTE_11=[0110-0101]
OTHER_BIN_BYTE_12=[1111-1001]

DOC   OTHER_BIN_BYTE_11&12: Int 15h F965h
DOC   call to Signal the beginning of video
DOC   set mode. The call is gated by the
DOC   READ_DIP_SWITCH[5].
OTHER_BIN_BYTE_13=[0110-0110]
OTHER_BIN_BYTE_14=[1111-1001]
```

```
DOC   OTHER_BIN_BYTE_13&14: INT 15 F966h call
DOC   Sub-function BL: 0x00 signal for 75 ohm
DOC   rersitor, TV support, BIOS sets
DOC   BH: 0x00=enable, 0x01=disable resistor

OTHER_BIN_BYTE_15=[0000-0000]
OTHER_BIN_BYTE_16=[0000-0000]

DOC   OTHER_BIN_BYTE_15&16: Reserved

OTHER_BIN_BYTE_17=[0100-0101]

DOC   bit 0: 0=disable DDC interface
DOC          1=enable  DDC interface
DOC   bit 1 : 0=skip re-latch panel type,
DOC          at warmboot
DOC          1=do re-latch panel type,
DOC          at warmboot
DOC   bit 2: 0=No control PROG1 pin in
DOC          switching voltage function call
DOC          1=Control on PROG1 pin in
DOC          switching voltage function call
DOC          (CL-GD7555 only)
DOC   bit 7-3: Reserved

REM   DATA FORMAT : decimal byte

%%%   DEC_BYTE

REM
REM   Where to display BIOS sign on message
REM

CIRRUS_SIGN_ON_ROW=[00]
CIRRUS_SIGN_ON_COL=[05]
CUSTOMER_SIGN_ON_ROW=[ff]
CUSTOMER_SIGN_ON_COL=[ff]
AFTER_POST_ROW=[ff]
AFTER_POST_COL=[ff]

DOC   CIRRUS_SIGN_ON and CUSTOMER_SIGN_ON
DOC   specify the initial cursor row and
DOC   column position to display the Cirrus
DOC   Logic and customer sign on messages
DOC   during POST.
DOC   AFTER_POST specifies where the cursor
DOC   will be at the end of POST.
DOC   To set the location to current cursor
DOC   position use "ff".


REM   DATA FORMAT : hexadecimal byte

%%%   HEX_BYTE

REM
REM   BIOS user option
REM

USER_OPTION_1=[00]

DOC   bit[7-0]: Reserved

USER_OPTION_2=[01]

DOC   bit 7-6: High Refresh, 640x480 vertical
DOC            freq
DOC            0=60Hz, 1=72Hz, 2=75Hz, 3=85Hz
DOC   bit 5: Reserved - Should be Zero (0)
DOC   bit 4-1: Reserved
DOC   bit 0: 0=LCD, 1=CRT or SimulSCAN
DOC            display mode

USER_OPTION_3=[00]
DOC      bit 7-3: Reserved
DOC      bit 2-0: 1280x1024 vertical frequency

USER_OPTION_4=[02]
DOC      bit 7-2: Reserved
DOC      bit 1: 0=SimulSCAN    1=no SimulSCAN
DOC      bit 0: 0=5.0 voltage  1=3.3 voltage

USER_OPTION_5=[24]
DOC      bit 7-5: 1024x768 vertical frequency
DOC      bit 4-2: 800x600 vertical frequency
DOC      bit 1-0: Reserved

USER_OPTION_6=[30]
DOC      bit 7-4: Maximum resolution
DOC      bit 3-2: Reserved
DOC      bit 1-0: TV mode (0=vga,1=NTSC,2=PAL)

REM   DATA FORMAT : Y or N

%%%   Y_OR_N

REM
REM   Whether to display BIOS sign on message
REM

CIRRUS_SIGN_ON_YN=[Y]

DOC   Controls whether the Cirrus Logic
DOC   sign-on message should be displayed
DOC   during POST.

CUSTOMER_SIGN_ON_YN=[N]

DOC   Controls whether the customer
DOC   sign-onmessage should be displayed
DOC   during POST.

REM   DATA FORMAT : ASCII string

%%%   STRING

REM
REM   Customer sign on message
REM

CUSTOMER_SIGN_ON_MSG:
BEGIN
END
DOC   This text goes between the BEGIN & END
DOC   lines & can be separated into
DOC   several lines.  Max length is 80 bytes,
DOC   including spaces & null lines.
```

```
%%%  PANEL

DOC  DATA FORMAT : hexadecimal format,
DOC  except where noted

REM
REM  Panel dependent configuration registers
REM

DOC  Modify, to replace the BIOS default
DOC  panel types.
DOC  DATA FORMAT : hexadecimal format, except
DOC  where noted

REM  PANEL table entry 1 (panel type 0)
DOC  Reserved Panel Type

PANEL_TYPE=[00]
DOC  bit 15-10 : Reserved
DOC  bit 9: 0=Set default enable graphics
DOC         expand for this panel type.
DOC         1=Set default disable graphics
DOC         expand for this panel type.
DOC  bit 8: 0=Set default enable text expand
DOC         for this panel type.
DOC         1=Set default disable text
DOC         expand for this panel type.
DOC  bit 7: 0=Allow SimulSCAN
DOC         1=No SimulSCAN
DOC  bit 6-3: Reserved
DOC  bit 2-1: Panel Information
DOC           0=Single Scan STN
DOC           1=Dual Scan STN
DOC           2=TFT
DOC           3=Reserved
DOC  bit 0: 0=Color panel
DOC         1=Monochrome panel

PANEL_FEATURE1=[00]
DOC  bit 15-0: Reserved

PANEL_FEATURE2=[00]
DOC  bit 15-0: Reserved

PANEL_MAX_HORIZONTAL_RES=[00]
DOC  Decimal data format.
DOC  The horizontal size of the panel.

PANEL_VERTICAL_RES=[00]
DOC  Decimal data format.
DOC  The vertical size of the panel.

PANEL_OTHER_BYTE_01=[00]
DOC  bit 7-4: Reserved
DOC  bit 3: 1 = Deduct one 64k unit from
DOC         total video memory in CRT-only.
DOC         0 = No deduction for CRT-only
DOC         mode.
DOC  bit 2-0: Deduct number of 64k units
DOC         from total video memory in
DOC  LCD & SimulSCAN modes.

PANEL_CONFIG_TABLE:

DOC  NOTICE, for register information:
DOC  See "CL-GD7555/6 Advanced Hardware
DOC  Reference Manual"
DOC  BINARY DATA FORMAT.  Bit locations
DOC  marked as "x" can not be modified.

     SR0D=[0000-0000]
DOC  SR0D: Numerator of panel video clock
     SR1D=[0000-0000]
DOC  SR1D: Denominator of panel video clock
     SR2F=[xxxx-0000]
DOC  SR2F: Half-Frame Accelerator FIFO
DOC  Threshold
     SR32=[xxxx-x000]
DOC  SR32: Half-Frame Accelerator FIFO in
DOC  Video Window
     CR81=[x000-0000]
DOC  CR81: Flat Panel Text Attributes
     CR82=[x000-x000]
DOC  CR82: Flat Panel Graphics Attributes
     CR83=[0000-0000]
DOC  CR83: Flat Panel Type
     CR84=[0000-0000]
     CR85=[0000-0000]
     CR86=[0000-0000]
     CR87=[0000-0000]
     CR88=[0000-0000]
     CR89=[x00x-xxxx]
     CR8C=[0000-0000]
     CR8E=[0000-0xxx]
DOC  CR8E[7]: Vsync Cross-talk reduction
DOC  (Rev. BC & up)
DOC  CR8E[6]: Hsync Cross-talk reduction
DOC  (Rev. BC & up)
     CR90=[0000-0000]
     CR91=[0000-0000]
DOC  CR83..CR91: Registers for flat panel
DOC  type controls
     CRA0=[0000-0000]
     CRA1=[0000-0000]
     CRA2=[0000-0000]
     CRA3=[0000-0000]
     CRA4=[0000-0000]
     CRA5=[0000-0000]
     CRA6=[0000-0000]
     CRA7=[0000-0000]
     CRA8=[0000-0000]
     CRA9=[0000-0000]
     CRAA=[0000-0000]
     CRAB=[0000-0000]
     CRAC=[0000-0000]
     CRAD=[0000-0000]
     CRAE=[0000-0000]
     CRAF=[0000-0000]
DOC  CRA0..CRAF: Registers for flat panel
DOC  horizontal controls
     CRB0=[0000-0000]
     CRB1=[x000-0000]
     CRB2=[0000-0000]
     CRB3=[0000-0000]
     CRBB=[0000-0000]
     CRBC=[0000-0000]
```

```
     CRBD=[0000-0000]
     CRBE=[0000-0000]
     CRBF=[0000-0000]
DOC  CRB0..CRBF: Registers for flat panel
DOC  vertical controls

     ERXX=[---------]
DOC  ERXX: end of table

REM  PANEL table entry 2 (panel type 1)

DOC  1024x768 Color TFT (NEC)

PANEL_TYPE=[314]
PANEL_FEATURE1=[00]
PANEL_FEATURE2=[00]
PANEL_MAX_HORIZONTAL_RES=[1024]
PANEL_VERTICAL_RES=[768]
PANEL_OTHER_BYTE_01=[08]
PANEL_CONFIG_TABLE:
     SR0D=[0011-0110]
     SR1D=[0001-1000]
     SR2F=[xxxx-0000]
     SR32=[xxxx-x000]
     CR81=[x101-0101]
     CR82=[x101-x011]
     CR83=[1010-0110]
     CR84=[0010-0000]
     CR85=[0100-0000]
     CR86=[0100-0001]
     CR87=[0111-0000]
     CR88=[0100-0100]
     CR89=[x11x-xxxx]
     CR8C=[0000-0000]
     CR8E=[0000-0xxx]
     CR90=[0000-0000]
     CR91=[0000-0000]
     CRA0=[1010-0110]
     CRA1=[1001-1100]
     CRA2=[0101-0000]
     CRA3=[0100-1100]
     CRA4=[1001-0011]
     CRA5=[1001-0000]
     CRA6=[0100-0111]
     CRA7=[0100-0011]
     CRA8=[1000-0100]
     CRA9=[1000-0110]
     CRAA=[0011-1111]
     CRAB=[0011-1011]
     CRAC=[1000-1111]
     CRAD=[1111-1111]
     CRAE=[0101-0100]
     CRAF=[0000-1001]
     CRB0=[0010-0100]
     CRB1=[x001-1011]
     CRB2=[0000-0000]
     CRB3=[0000-0100]
     CRBB=[1111-1110]
     CRBC=[0000-0000]
     CRBD=[0000-0000]
     CRBE=[0001-0000]
     CRBF=[0000-0000]
     ERXX=[---------]

REM  PANEL table entry 3 (panel type 2)

DOC  640x480 DSTN Color (general)

PANEL_TYPE=[312]
PANEL_FEATURE1=[00]
PANEL_FEATURE2=[00]
PANEL_MAX_HORIZONTAL_RES=[640]
PANEL_VERTICAL_RES=[480]
PANEL_OTHER_BYTE_01=[09]
PANEL_CONFIG_TABLE:
     SR0D=[0001-1011]
     SR1D=[0001-1110]
     SR2F=[xxxx-0110]
     SR32=[xxxx-x110]
     CR81=[x011-0000]
     CR82=[x011-x000]
     CR83=[0000-0100]
     CR84=[0000-0000]
     CR85=[0000-0000]
     CR86=[0000-0000]
     CR87=[0111-0000]
     CR88=[0100-0100]
     CR89=[x11x-xxxx]
     CR8C=[0000-0000]
     CR8E=[1000-0xxx]
     CR90=[0000-0000]
     CR91=[0101-1010]
     CRA0=[0101-1111]
     CRA1=[0101-1010]
     CRA2=[0010-1101]
     CRA3=[0000-1011]
     CRA4=[0000-0000]
     CRA5=[0000-0000]
     CRA6=[0000-0000]
     CRA7=[0000-0000]
     CRA8=[0000-0000]
     CRA9=[0000-0000]
     CRAA=[0000-0000]
     CRAB=[0000-0000]
     CRAC=[1000-1111]
     CRAD=[1001-1111]
     CRAE=[1100-0110]
     CRAF=[0000-0100]
     CRB0=[0000-1011]
     CRB1=[x001-0010]
     CRB2=[0000-0000]
     CRB3=[0000-0100]
     CRBB=[1110-0010]
     CRBC=[0000-1010]
     CRBD=[0000-0000]
     CRBE=[0000-1000]
     CRBF=[0000-0000]
     ERXX=[---------]

REM  PANEL table entry 4 (panel type 3)

DOC  640x480 TFT Color (NEC 18-bit)

PANEL_TYPE=[314]
PANEL_FEATURE1=[00]
PANEL_FEATURE2=[00]
```

```
PANEL_MAX_HORIZONTAL_RES=[640]
PANEL_VERTICAL_RES=[480]
PANEL_OTHER_BYTE_01=[08]
PANEL_CONFIG_TABLE:
      SR0D=[0001-1011]
      SR1D=[0001-1110]
      SR2F=[xxxx-0000]
      SR32=[xxxx-x000]
      CR81=[x011-0000]
      CR82=[x011-x000]
      CR83=[0010-1000]
      CR84=[0010-0000]
      CR85=[0100-0000]
      CR86=[0100-0001]
      CR87=[0111-0000]
      CR88=[0110-0110]
      CR89=[x11x-xxxx]
      CR8C=[0000-0000]
      CR8E=[0000-0xxx]
      CR90=[0000-0000]
      CR91=[0000-0000]
      CRA0=[0101-1111]
      CRA1=[0101-1010]
      CRA2=[0010-1101]
      CRA3=[0000-1011]
      CRA4=[0000-0000]
      CRA5=[0000-0000]
      CRA6=[0000-0000]
      CRA7=[0000-0000]
      CRA8=[0000-0000]
      CRA9=[0000-0000]
      CRAA=[0000-0000]
      CRAB=[0000-0000]
      CRAC=[1000-1111]
      CRAD=[1001-1111]
      CRAE=[1100-0110]
      CRAF=[0000-0100]
      CRB0=[0000-1011]
      CRB1=[x001-0010]
      CRB2=[0000-0000]
      CRB3=[0000-0100]
      CRBB=[1101-1110]
      CRBC=[0000-0000]
      CRBD=[1110-1100]
      CRBE=[0000-1001]
      CRBF=[0000-0000]
      ERXX=[---------]

REM   PANEL table entry 5 (panel type 4)

DOC   Reserved Panel Type

PANEL_TYPE=[00]
PANEL_FEATURE1=[00]
PANEL_FEATURE2=[00]
PANEL_MAX_HORIZONTAL_RES=[00]
PANEL_VERTICAL_RES=[00]
PANEL_OTHER_BYTE_01=[00]
PANEL_CONFIG_TABLE:
      SR0D=[0000-0000]
      SR1D=[0000-0000]
      SR2F=[xxxx-0000]
      SR32=[xxxx-x000]
      CR81=[x000-0000]
      CR82=[x000-x000]
      CR83=[0000-0000]
      CR84=[0000-0000]
      CR85=[0000-0000]
      CR86=[0000-0000]
      CR87=[0000-0000]
      CR88=[0000-0000]
      CR89=[x00x-xxxx]
      CR8C=[0000-0000]
      CR8E=[0000-0xxx]
      CR90=[0000-0000]
      CR91=[0000-0000]
      CRA0=[0000-0000]
      CRA1=[0000-0000]
      CRA2=[0000-0000]
      CRA3=[0000-0000]
      CRA4=[0000-0000]
      CRA5=[0000-0000]
      CRA6=[0000-0000]
      CRA7=[0000-0000]
      CRA8=[0000-0000]
      CRA9=[0000-0000]
      CRAA=[0000-0000]
      CRAB=[0000-0000]
      CRAC=[0000-0000]
      CRAD=[0000-0000]
      CRAE=[0000-0000]
      CRAF=[0000-0000]
      CRB0=[0000-0000]
      CRB1=[x000-0000]
      CRB2=[0000-0000]
      CRB3=[0000-0000]
      CRBB=[0000-0000]
      CRBC=[0000-0000]
      CRBD=[0000-0000]
      CRBE=[0000-0000]
      CRBF=[0000-0000]
      ERXX=[---------]

REM   PANEL table entry 6 (panel type 5)

DOC   Reserved Panel Type

PANEL_TYPE=[00]
PANEL_FEATURE1=[00]
PANEL_FEATURE2=[00]
PANEL_MAX_HORIZONTAL_RES=[00]
PANEL_VERTICAL_RES=[00]
PANEL_OTHER_BYTE_01=[00]
PANEL_CONFIG_TABLE:
      SR0D=[0000-0000]
      SR1D=[0000-0000]
      SR2F=[xxxx-0000]
      SR32=[xxxx-x000]
      CR81=[x000-0000]
      CR82=[x000-x000]
      CR83=[0000-0000]
      CR84=[0000-0000]
      CR85=[0000-0000]
      CR86=[0000-0000]
      CR87=[0000-0000]
      CR88=[0000-0000]
```

```
     CR89=[x00x-xxxx]
     CR8C=[0000-0000]
     CR8E=[0000-0xxx]
     CR90=[0000-0000]
     CR91=[0000-0000]
     CRA0=[0000-0000]
     CRA1=[0000-0000]
     CRA2=[0000-0000]
     CRA3=[0000-0000]
     CRA4=[0000-0000]
     CRA5=[0000-0000]
     CRA6=[0000-0000]
     CRA7=[0000-0000]
     CRA8=[0000-0000]
     CRA9=[0000-0000]
     CRAA=[0000-0000]
     CRAB=[0000-0000]
     CRAC=[0000-0000]
     CRAD=[0000-0000]
     CRAE=[0000-0000]
     CRAF=[0000-0000]
     CRB0=[0000-0000]
     CRB1=[x000-0000]
     CRB2=[0000-0000]
     CRB3=[0000-0000]
     CRBB=[0000-0000]
     CRBC=[0000-0000]
     CRBD=[0000-0000]
     CRBE=[0000-0000]
     CRBF=[0000-0000]
     ERXX=[---------]

REM  PANEL table entry 7 (panel type 6)

DOC  800x600 DSTN Color (general)

PANEL_TYPE=[312]
PANEL_FEATURE1=[00]
PANEL_FEATURE2=[00]
PANEL_MAX_HORIZONTAL_RES=[800]
PANEL_VERTICAL_RES=[600]
PANEL_OTHER_BYTE_01=[09]
PANEL_CONFIG_TABLE:
     SR0D=[0010-1010]
     SR1D=[0001-1110]
     SR2F=[xxxx-0110]
     SR32=[xxxx-x110]
     CR81=[x101-0101]
     CR82=[x101-x011]
     CR83=[0000-0101]
     CR84=[0000-0000]
     CR85=[0000-0000]
     CR86=[0000-0000]
     CR87=[0111-0000]
     CR88=[0100-0100]
     CR89=[x11x-xxxx]
     CR8C=[0000-0000]
     CR8E=[1000-0xxx]
     CR90=[0000-0000]
     CR91=[0101-1010]
     CRA0=[1000-0010]
     CRA1=[0111-1011]
     CRA2=[0011-1111]
     CRA3=[0100-0011]
     CRA4=[0111-0011]
     CRA5=[0111-0011]
     CRA6=[0011-0111]
     CRA7=[0011-1110]
     CRA8=[0110-0111]
     CRA9=[0110-0111]
     CRAA=[0011-0001]
     CRAB=[0000-1111]
     CRAC=[0000-1001]
     CRAD=[1100-0111]
     CRAE=[0000-1100]
     CRAF=[0010-1001]
     CRB0=[0110-1111]
     CRB1=[x001-0010]
     CRB2=[0101-1011]
     CRB3=[0000-1111]
     CRBB=[0101-1001]
     CRBC=[0000-1011]
     CRBD=[0000-0000]
     CRBE=[0001-0000]
     CRBF=[0000-0000]
     ERXX=[---------]

REM  PANEL table entry 8 (panel type 7)

DOC  800x600 TFT Color (NEC 18-bit)

PANEL_TYPE=[314]
PANEL_FEATURE1=[00]
PANEL_FEATURE2=[00]
PANEL_MAX_HORIZONTAL_RES=[800]
PANEL_VERTICAL_RES=[600]
PANEL_OTHER_BYTE_01=[08]
PANEL_CONFIG_TABLE:
     SR0D=[0010-1010]
     SR1D=[0001-1110]
     SR2F=[xxxx-0000]
     SR32=[xxxx-x000]
     CR81=[x101-0101]
     CR82=[x101-x011]
     CR83=[0010-1001]
     CR84=[0010-0000]
     CR85=[0100-0000]
     CR86=[0100-0001]
     CR87=[0111-0000]
     CR88=[0110-0110]
     CR89=[x11x-xxxx]
     CR8C=[0000-0000]
     CR8E=[0000-0xxx]
     CR90=[0000-0000]
     CR91=[0000-0000]
     CRA0=[1000-0010]
     CRA1=[0111-1011]
     CRA2=[0011-1111]
     CRA3=[0100-0011]
     CRA4=[0111-0011]
     CRA5=[0111-0011]
     CRA6=[0011-0111]
     CRA7=[0011-1110]
     CRA8=[0110-0111]
     CRA9=[0110-0111]
     CRAA=[0011-0001]
```

```
      CRAB=[0000-1111]
      CRAC=[0000-1001]
      CRAD=[1100-0111]
      CRAE=[0000-1100]
      CRAF=[0010-1001]
      CRB0=[0110-1111]
      CRB1=[x001-0010]
      CRB2=[0101-1011]
      CRB3=[0000-1111]
      CRBB=[0101-1001]
      CRBC=[0000-0000]
      CRBD=[0000-0000]
      CRBE=[0001-0000]
      CRBF=[0000-0000]
      ERXX=[---------]

REM   PANEL table entry 9 (panel type 8)

DOC   Reserved for custom panel type.

PANEL_TYPE=[00]
PANEL_FEATURE1=[00]
PANEL_FEATURE2=[00]
PANEL_MAX_HORIZONTAL_RES=[00]
PANEL_VERTICAL_RES=[00]
PANEL_OTHER_BYTE_01=[00]
PANEL_CONFIG_TABLE:
      SR0D=[0000-0000]
      SR1D=[0000-0000]
      SR2F=[xxxx-0000]
      SR32=[xxxx-x000]
      CR81=[x000-0000]
      CR82=[x000-x000]
      CR83=[0000-0000]
      CR84=[0000-0000]
      CR85=[0000-0000]
      CR86=[0000-0000]
      CR87=[0000-0000]
      CR88=[0000-0000]
      CR89=[x00x-xxxx]
      CR8C=[0000-0000]
      CR8E=[0000-0xxx]
      CR90=[0000-0000]
      CR91=[0000-0000]
      CRA0=[0000-0000]
      CRA1=[0000-0000]
      CRA2=[0000-0000]
      CRA3=[0000-0000]
      CRA4=[0000-0000]
      CRA5=[0000-0000]
      CRA6=[0000-0000]
      CRA7=[0000-0000]
      CRA8=[0000-0000]
      CRA9=[0000-0000]
      CRAA=[0000-0000]
      CRAB=[0000-0000]
      CRAC=[0000-0000]
      CRAD=[0000-0000]
      CRAE=[0000-0000]
      CRAF=[0000-0000]
      CRB0=[0000-0000]
      CRB1=[x000-0000]
      CRB2=[0000-0000]
      CRB3=[0000-0000]
      CRBB=[0000-0000]
      CRBC=[0000-0000]
      CRBD=[0000-0000]
      CRBE=[0000-0000]
      CRBF=[0000-0000]
      ERXX=[---------]

REM   PANEL table entry 10 (panel type 9)

DOC   Reserved for custom panel type.

PANEL_TYPE=[00]
PANEL_FEATURE1=[00]
PANEL_FEATURE2=[00]
PANEL_MAX_HORIZONTAL_RES=[00]
PANEL_VERTICAL_RES=[00]
PANEL_OTHER_BYTE_01=[00]
PANEL_CONFIG_TABLE:
      SR0D=[0000-0000]
      SR1D=[0000-0000]
      SR2F=[xxxx-0000]
      SR32=[xxxx-x000]
      CR81=[x000-0000]
      CR82=[x000-x000]
      CR83=[0000-0000]
      CR84=[0000-0000]
      CR85=[0000-0000]
      CR86=[0000-0000]
      CR87=[0000-0000]
      CR88=[0000-0000]
      CR89=[x00x-xxxx]
      CR8C=[0000-0000]
      CR8E=[0000-0xxx]
      CR90=[0000-0000]
      CR91=[0000-0000]
      CRA0=[0000-0000]
      CRA1=[0000-0000]
      CRA2=[0000-0000]
      CRA3=[0000-0000]
      CRA4=[0000-0000]
      CRA5=[0000-0000]
      CRA6=[0000-0000]
      CRA7=[0000-0000]
      CRA8=[0000-0000]
      CRA9=[0000-0000]
      CRAA=[0000-0000]
      CRAB=[0000-0000]
      CRAC=[0000-0000]
      CRAD=[0000-0000]
      CRAE=[0000-0000]
      CRAF=[0000-0000]
      CRB0=[0000-0000]
      CRB1=[x000-0000]
      CRB2=[0000-0000]
      CRB3=[0000-0000]
      CRBB=[0000-0000]
      CRBC=[0000-0000]
      CRBD=[0000-0000]
      CRBE=[0000-0000]
      CRBF=[0000-0000]
      ERXX=[---------]
```

```
REM   PANEL table entry 11 (panel type 10)

DOC   Reserved for custom panel type.

PANEL_TYPE=[00]
PANEL_FEATURE1=[00]
PANEL_FEATURE2=[00]
PANEL_MAX_HORIZONTAL_RES=[00]
PANEL_VERTICAL_RES=[00]
PANEL_OTHER_BYTE_01=[00]
PANEL_CONFIG_TABLE:
      SR0D=[0000-0000]
      SR1D=[0000-0000]
      SR2F=[xxxx-0000]
      SR32=[xxxx-x000]
      CR81=[x000-0000]
      CR82=[x000-x000]
      CR83=[0000-0000]
      CR84=[0000-0000]
      CR85=[0000-0000]
      CR86=[0000-0000]
      CR87=[0000-0000]
      CR88=[0000-0000]
      CR89=[x00x-xxxx]
      CR8C=[0000-0000]
      CR8E=[0000-0xxx]
      CR90=[0000-0000]
      CR91=[0000-0000]
      CRA0=[0000-0000]
      CRA1=[0000-0000]
      CRA2=[0000-0000]
      CRA3=[0000-0000]
      CRA4=[0000-0000]
      CRA5=[0000-0000]
      CRA6=[0000-0000]
      CRA7=[0000-0000]
      CRA8=[0000-0000]
      CRA9=[0000-0000]
      CRAA=[0000-0000]
      CRAB=[0000-0000]
      CRAC=[0000-0000]
      CRAD=[0000-0000]
      CRAE=[0000-0000]
      CRAF=[0000-0000]
      CRB0=[0000-0000]
      CRB1=[x000-0000]
      CRB2=[0000-0000]
      CRB3=[0000-0000]
      CRBB=[0000-0000]
      CRBC=[0000-0000]
      CRBD=[0000-0000]
      CRBE=[0000-0000]
      CRBF=[0000-0000]
      ERXX=[---------]

REM   PANEL table entry 12 (panel type 11)

DOC   Reserved for custom panel type.

PANEL_TYPE=[00]
PANEL_FEATURE1=[00]
PANEL_FEATURE2=[00]
PANEL_MAX_HORIZONTAL_RES=[00]
PANEL_VERTICAL_RES=[00]
PANEL_OTHER_BYTE_01=[00]
PANEL_CONFIG_TABLE:
      SR0D=[0000-0000]
      SR1D=[0000-0000]
      SR2F=[xxxx-0000]
      SR32=[xxxx-x000]
      CR81=[x000-0000]
      CR82=[x000-x000]
      CR83=[0000-0000]
      CR84=[0000-0000]
      CR85=[0000-0000]
      CR86=[0000-0000]
      CR87=[0000-0000]
      CR88=[0000-0000]
      CR89=[x00x-xxxx]
      CR8C=[0000-0000]
      CR8E=[0000-0xxx]
      CR90=[0000-0000]
      CR91=[0000-0000]
      CRA0=[0000-0000]
      CRA1=[0000-0000]
      CRA2=[0000-0000]
      CRA3=[0000-0000]
      CRA4=[0000-0000]
      CRA5=[0000-0000]
      CRA6=[0000-0000]
      CRA7=[0000-0000]
      CRA8=[0000-0000]
      CRA9=[0000-0000]
      CRAA=[0000-0000]
      CRAB=[0000-0000]
      CRAC=[0000-0000]
      CRAD=[0000-0000]
      CRAE=[0000-0000]
      CRAF=[0000-0000]
      CRB0=[0000-0000]
      CRB1=[x000-0000]
      CRB2=[0000-0000]
      CRB3=[0000-0000]
      CRBB=[0000-0000]
      CRBC=[0000-0000]
      CRBD=[0000-0000]
      CRBE=[0000-0000]
      CRBF=[0000-0000]
      ERXX=[---------]

REM   PANEL table entry 13 (panel type 12)

DOC   Reserved for custom panel type.

PANEL_TYPE=[00]
PANEL_FEATURE1=[00]
PANEL_FEATURE2=[00]
PANEL_MAX_HORIZONTAL_RES=[00]
PANEL_VERTICAL_RES=[00]
PANEL_OTHER_BYTE_01=[00]
PANEL_CONFIG_TABLE:
      SR0D=[0000-0000]
      SR1D=[0000-0000]
      SR2F=[xxxx-0000]
      SR32=[xxxx-x000]
      CR81=[x000-0000]
```

```
      CR82=[x000-x000]
      CR83=[0000-0000]
      CR84=[0000-0000]
      CR85=[0000-0000]
      CR86=[0000-0000]
      CR87=[0000-0000]
      CR88=[0000-0000]
      CR89=[x00x-xxxx]
      CR8C=[0000-0000]
      CR8E=[0000-0xxx]
      CR90=[0000-0000]
      CR91=[0000-0000]
      CRA0=[0000-0000]
      CRA1=[0000-0000]
      CRA2=[0000-0000]
      CRA3=[0000-0000]
      CRA4=[0000-0000]
      CRA5=[0000-0000]
      CRA6=[0000-0000]
      CRA7=[0000-0000]
      CRA8=[0000-0000]
      CRA9=[0000-0000]
      CRAA=[0000-0000]
      CRAB=[0000-0000]
      CRAC=[0000-0000]
      CRAD=[0000-0000]
      CRAE=[0000-0000]
      CRAF=[0000-0000]
      CRB0=[0000-0000]
      CRB1=[x000-0000]
      CRB2=[0000-0000]
      CRB3=[0000-0000]
      CRBB=[0000-0000]
      CRBC=[0000-0000]
      CRBD=[0000-0000]
      CRBE=[0000-0000]
      CRBF=[0000-0000]
      ERXX=[---------]

REM   PANEL table entry 14 (panel type 13)

DOC   Reserved for custom panel type.

PANEL_TYPE=[00]
PANEL_FEATURE1=[00]
PANEL_FEATURE2=[00]
PANEL_MAX_HORIZONTAL_RES=[00]
PANEL_VERTICAL_RES=[00]
PANEL_OTHER_BYTE_01=[00]
PANEL_CONFIG_TABLE:
      SR0D=[0000-0000]
      SR1D=[0000-0000]
      SR2F=[xxxx-0000]
      SR32=[xxxx-x000]
      CR81=[x000-0000]
      CR82=[x000-x000]
      CR83=[0000-0000]
      CR84=[0000-0000]
      CR85=[0000-0000]
      CR86=[0000-0000]
      CR87=[0000-0000]
      CR88=[0000-0000]
      CR89=[x00x-xxxx]
      CR8C=[0000-0000]
      CR8E=[0000-0xxx]
      CR90=[0000-0000]
      CR91=[0000-0000]
      CRA0=[0000-0000]
      CRA1=[0000-0000]
      CRA2=[0000-0000]
      CRA3=[0000-0000]
      CRA4=[0000-0000]
      CRA5=[0000-0000]
      CRA6=[0000-0000]
      CRA7=[0000-0000]
      CRA8=[0000-0000]
      CRA9=[0000-0000]
      CRAA=[0000-0000]
      CRAB=[0000-0000]
      CRAC=[0000-0000]
      CRAD=[0000-0000]
      CRAE=[0000-0000]
      CRAF=[0000-0000]
      CRB0=[0000-0000]
      CRB1=[x000-0000]
      CRB2=[0000-0000]
      CRB3=[0000-0000]
      CRBB=[0000-0000]
      CRBC=[0000-0000]
      CRBD=[0000-0000]
      CRBE=[0000-0000]
      CRBF=[0000-0000]
      ERXX=[---------]

REM   PANEL table entry 15 (panel type 14)

DOC   Reserved for custom panel type.

PANEL_TYPE=[00]
PANEL_FEATURE1=[00]
PANEL_FEATURE2=[00]
PANEL_MAX_HORIZONTAL_RES=[00]
PANEL_VERTICAL_RES=[00]
PANEL_OTHER_BYTE_01=[00]
PANEL_CONFIG_TABLE:
      SR0D=[0000-0000]
      SR1D=[0000-0000]
      SR2F=[xxxx-0000]
      SR32=[xxxx-x000]
      CR81=[x000-0000]
      CR82=[x000-x000]
      CR83=[0000-0000]
      CR84=[0000-0000]
      CR85=[0000-0000]
      CR86=[0000-0000]
      CR87=[0000-0000]
      CR88=[0000-0000]
      CR89=[x00x-xxxx]
      CR8C=[0000-0000]
      CR8E=[0000-0xxx]
      CR90=[0000-0000]
      CR91=[0000-0000]
      CRA0=[0000-0000]
      CRA1=[0000-0000]
      CRA2=[0000-0000]
      CRA3=[0000-0000]
```

```
      CRA4=[0000-0000]
      CRA5=[0000-0000]
      CRA6=[0000-0000]
      CRA7=[0000-0000]
      CRA8=[0000-0000]
      CRA9=[0000-0000]
      CRAA=[0000-0000]
      CRAB=[0000-0000]
      CRAC=[0000-0000]
      CRAD=[0000-0000]
      CRAE=[0000-0000]
      CRAF=[0000-0000]
      CRB0=[0000-0000]
      CRB1=[x000-0000]
      CRB2=[0000-0000]
      CRB3=[0000-0000]
      CRBB=[0000-0000]
      CRBC=[0000-0000]
      CRBD=[0000-0000]
      CRBE=[0000-0000]
      CRBF=[0000-0000]
      ERXX=[---------]

REM   PANEL table entry 16 (panel type 15)

DOC   Reserved for custom panel type.

PANEL_TYPE=[00]
PANEL_FEATURE1=[00]
PANEL_FEATURE2=[00]
PANEL_MAX_HORIZONTAL_RES=[00]
PANEL_VERTICAL_RES=[00]
PANEL_OTHER_BYTE_01=[00]
PANEL_CONFIG_TABLE:
      SR0D=[0000-0000]
      SR1D=[0000-0000]
      SR2F=[xxxx-0000]
      SR32=[xxxx-x000]
      CR81=[x000-0000]
      CR82=[x000-x000]
      CR83=[0000-0000]
      CR84=[0000-0000]
      CR85=[0000-0000]
      CR86=[0000-0000]
      CR87=[0000-0000]
      CR88=[0000-0000]
      CR89=[x00x-xxxx]
      CR8C=[0000-0000]
      CR8E=[0000-0xxx]
      CR90=[0000-0000]
      CR91=[0000-0000]
      CRA0=[0000-0000]
      CRA1=[0000-0000]
      CRA2=[0000-0000]
      CRA3=[0000-0000]
      CRA4=[0000-0000]
      CRA5=[0000-0000]
      CRA6=[0000-0000]
      CRA7=[0000-0000]
      CRA8=[0000-0000]
      CRA9=[0000-0000]
      CRAA=[0000-0000]
      CRAB=[0000-0000]
      CRAC=[0000-0000]
      CRAD=[0000-0000]
      CRAE=[0000-0000]
      CRAF=[0000-0000]
      CRB0=[0000-0000]
      CRB1=[x000-0000]
      CRB2=[0000-0000]
      CRB3=[0000-0000]
      CRBB=[0000-0000]
      CRBC=[0000-0000]
      CRBD=[0000-0000]
      CRBE=[0000-0000]
      CRBF=[0000-0000]
      ERXX=[---------]

%%%   XMODE

REM
REM   Extension registers used to set up
REM   extended video mode
REM

DOC   DATA FORMAT : hexadecimal format,
DOC   except where noted

REM   XMODE table entry 1

DOC   extended mode 14, 1056x400x16colorsx70Hz

XMODE_NUMBER=[14]
DOC   The Cirrus's extended mode number.

XMODE_VESA_NUMBER=[00]
DOC   The VESA video mode number.

XMODE_HORIZONTAL_RES=[1056]
DOC   The horizontal resolution for this
DOC   extended mode, measured in pixels.

XMODE_VERTICAL_RES=[400]
DOC   The vertical resolution for this
DOC   extended mode, measured in pixels.
XMODE_COLOR=[04]
DOC   The number of colors supported by this
DOC   extended mode, in bits per pixel.

XMODE_CHAR_WIDTH=[08]
DOC   The character cell width.

XMODE_CHAR_HEIGHT=[10]
DOC    The character cell height.

XMODE_FEATURE_BYTE_01=[00]
DOC   The memory model of the VESA VBE
DOC   Interface. Information is internally
DOC   used by the BIOS.

XMODE_FEATURE_BYTE_02=[0e]
DOC   The mode attribute of the VESA VBE
DOC   Interface.
DOC   Information is internally used by the
DOC   BIOS.
```

```
XMODE_FEATURE_BYTE_03=[ff]
DOC   The index to refresh rate table for the
DOC   requested refresh rate.
DOC   The value 0FFh means that the requested
DOC   refresh rate is the
DOC   same as the refresh rate in generic
DOC   standard parameter for the
DOC   resolution.
DOC   Information is internally used by the
DOC   BIOS.

XMODE_FEATURE_BYTE_04=[02]
DOC   Generic table entry of standard
DOC   parameter table for the
DOC   requested resolution.

XMODE_FEATURE_BYTE_05=[6f]
DOC   Miscellaneous Output Register 3C2h value

XMODE_OTHER_BYTE_01=[39]
DOC   bit 7: reserved
DOC   bit 6-1: the minimum memory clock
DOC           required for the requested mode.
DOC   bit 0: minimum voltage required for the
DOC           requested mode -
DOC           0=5V; 1=3V.

XMODE_OTHER_BYTE_02=[04]
DOC   Minimum video memory requirements for
DOC   the requested mode.

XMODE_OTHER_BYTE_03=[00]
DOC   The monitor type used, where
DOC   bit 7: reserved
DOC   bit 6-4: Required frequency to support
DOC           mode
DOC   bit 3-0: Maximum vertical resolution
DOC           required
DOC   Information is internally used by the
DOC   BIOS.

XMODE_OTHER_BYTE_04=[88]
XMODE_OTHER_BYTE_05=[1d]
DOC   Bytes 05 & 04 are the offset address of
DOC   the refresh rate table entry for the
DOC   requested refresh rate. Check
DOC   XMODE_OTHER_BYTE_03
DOC   value for 0FFh, this means offset
DOC   address was not used.
DOC   This field can be directly used by
DOC   application when theentry address of
DOC   supplement table returned from Cirrus's
DOC   extended BIOS function call, function
DOC   A0h.

XMODE_OTHER_BYTE_06=[00]
DOC   Panel support information, where
DOC   bit 7: reserved
DOC   bit 6-4: For dual scan panel types -
DOC           DOC Max. panel size support for
DOC           the requested mode.
DOC           0=No support on any panels
DOC           1=640x480 panel
DOC           2=800x600 panel or less
DOC           3=1024x768 panel or less
DOC           4-7=Reserved
DOC   bit 3: reserved
DOC   bit 2-0: For NON dual scan panel types -
DOC           Max. panel size support for the
DOC           requested mode.
DOC           0=No support on any panels
DOC           1=640x480 panel
DOC           2=800x600 panel or less
DOC           3=1024x768 panel or less


XMODE_OTHER_BYTE_07=[00]
DOC   800x600 DSTN panel information, where
DOC   bit 7-6: reserved
DOC   bit 5-0: Minimum memory clock required
DOC           (00h = no mini.)

XMODE_CONFIG_TABLE:
DOC   NOTICE, for register information:
DOC   See "CL-GD7555/6 Advanced Hardware
DOC   Reference Manual"
DOC   BINARY DATA FORMAT: Bit locations
DOC   marked as "x" are not modified.

      SR0E=[0101-1010]
DOC   SR0E: VCLK3 Numerator Value for the
DOC   requested mode
      SR1E=[0011-1110]
DOC   SR1E: VCLK3 Denominator and Post Scalar
DOC   Value for the requested mode
      SR07=[0000-0000]
DOC   SR07: Extended Sequencer Mode
      GR0B=[xxx0-0000]
DOC   GR0B: Graphics Controller Mode
DOC   Extensions
      CR19=[0000-0000]
DOC   CR19: Interface End Register
      CR1A=[0000-0000]
DOC   CR1A: Miscellaneous Control Register
      CR1B=[0000-0000]
DOC   CR1B: Extended Display Controls Register
      GR18=[xx0x-0xxx]
DOC   GR18: EDO DRAM Control
      CR8E=[xxxx-x1xx]
DOC   CR8E: Bit[2]=1 Bypass DAC, so there
DOC   will be no gamma adjustment
      xxFF=[0000-0000]
DOC   HDR: Hidden DAC Register 3C6h
      ERXX=[---------]
DOC   ERXX: end of table

REM   XMODE table entry 2

DOC   extended mode 54, 1056x350x16colorsx70Hz

XMODE_NUMBER=[54]
XMODE_VESA_NUMBER=[10a]
XMODE_HORIZONTAL_RES=[1056]
XMODE_VERTICAL_RES=[350]
XMODE_COLOR=[04]
XMODE_CHAR_WIDTH=[08]
```

```
XMODE_CHAR_HEIGHT=[08]
XMODE_FEATURE_BYTE_01=[00]
XMODE_FEATURE_BYTE_02=[0e]
XMODE_FEATURE_BYTE_03=[ff]
XMODE_FEATURE_BYTE_04=[03]
XMODE_FEATURE_BYTE_05=[af]
XMODE_OTHER_BYTE_01=[39]
XMODE_OTHER_BYTE_02=[04]
XMODE_OTHER_BYTE_03=[00]
XMODE_OTHER_BYTE_04=[88]
XMODE_OTHER_BYTE_05=[1d]
XMODE_OTHER_BYTE_06=[00]
XMODE_OTHER_BYTE_07=[00]
XMODE_CONFIG_TABLE:
     SR0E=[0101-1010]
     SR1E=[0011-1110]
     SR07=[0000-0000]
     GR0B=[xxx0-0000]
     CR19=[0000-0000]
     CR1A=[0000-0000]
     CR1B=[0000-0000]
     GR18=[xx0x-0xxx]
     CR8E=[xxxx-x1xx]
     xxFF=[0000-0000]
     ERXX=[---------]

REM  XMODE table entry 3

DOC extended mode 55, 1056x350x16colorsx70Hz

XMODE_NUMBER=[55]
XMODE_VESA_NUMBER=[109]
XMODE_HORIZONTAL_RES=[1056]
XMODE_VERTICAL_RES=[350]
XMODE_COLOR=[04]
XMODE_CHAR_WIDTH=[08]
XMODE_CHAR_HEIGHT=[0e]
XMODE_FEATURE_BYTE_01=[00]
XMODE_FEATURE_BYTE_02=[0e]
XMODE_FEATURE_BYTE_03=[ff]
XMODE_FEATURE_BYTE_04=[04]
XMODE_FEATURE_BYTE_05=[af]
XMODE_OTHER_BYTE_01=[39]
XMODE_OTHER_BYTE_02=[04]
XMODE_OTHER_BYTE_03=[00]
XMODE_OTHER_BYTE_04=[88]
XMODE_OTHER_BYTE_05=[1d]
XMODE_OTHER_BYTE_06=[00]
XMODE_OTHER_BYTE_07=[00]
XMODE_CONFIG_TABLE:
     SR0E=[0101-1010]
     SR1E=[0011-1110]
     SR07=[0000-0000]
     GR0B=[xxx0-0000]
     CR19=[0000-0000]
     CR1A=[0000-0000]
     CR1B=[0000-0000]
     GR18=[xx0x-0xxx]
     CR8E=[xxxx-x1xx]
     xxFF=[0000-0000]
     ERXX=[---------]

REM  XMODE table entry 4

DOC  extended mode 5C, 800x600x256colorsx85Hz

XMODE_NUMBER=[5c]
XMODE_VESA_NUMBER=[103]
XMODE_HORIZONTAL_RES=[800]
XMODE_VERTICAL_RES=[600]
XMODE_COLOR=[08]
XMODE_CHAR_WIDTH=[08]
XMODE_CHAR_HEIGHT=[10]
XMODE_FEATURE_BYTE_01=[04]
XMODE_FEATURE_BYTE_02=[1e]
XMODE_FEATURE_BYTE_03=[07]
XMODE_FEATURE_BYTE_04=[05]
XMODE_FEATURE_BYTE_05=[2f]
XMODE_OTHER_BYTE_01=[39]
XMODE_OTHER_BYTE_02=[08]
XMODE_OTHER_BYTE_03=[41]
XMODE_OTHER_BYTE_04=[a6]
XMODE_OTHER_BYTE_05=[1e]
XMODE_OTHER_BYTE_06=[33]
XMODE_OTHER_BYTE_07=[22]
XMODE_CONFIG_TABLE:
     SR0E=[0111-1010]
     SR1E=[0011-1110]
     SR07=[0000-0001]
     GR0B=[xxx0-0000]
     CR19=[0000-0000]
     CR1A=[0000-0000]
     CR1B=[0010-0010]
     GR18=[xx0x-0xxx]
     CR8E=[xxxx-x1xx]
     xxFF=[0000-0000]
     ERXX=[---------]

REM  XMODE table entry 5


DOC extended mode 5C, 800x600x256colorsx75Hz

XMODE_NUMBER=[5c]
XMODE_VESA_NUMBER=[103]
XMODE_HORIZONTAL_RES=[800]
XMODE_VERTICAL_RES=[600]
XMODE_COLOR=[08]
XMODE_CHAR_WIDTH=[08]
XMODE_CHAR_HEIGHT=[10]
XMODE_FEATURE_BYTE_01=[04]
XMODE_FEATURE_BYTE_02=[1e]
XMODE_FEATURE_BYTE_03=[06]
XMODE_FEATURE_BYTE_04=[05]
XMODE_FEATURE_BYTE_05=[2f]
XMODE_OTHER_BYTE_01=[39]
XMODE_OTHER_BYTE_02=[08]
XMODE_OTHER_BYTE_03=[31]
XMODE_OTHER_BYTE_04=[f0]
XMODE_OTHER_BYTE_05=[1d]
XMODE_OTHER_BYTE_06=[33]
XMODE_OTHER_BYTE_07=[22]
XMODE_CONFIG_TABLE:
     SR0E=[0101-0011]
     SR1E=[0011-0000]
     SR07=[0000-0001]
```

```
     GR0B=[xxx0-0000]
     CR19=[0000-0000]
     CR1A=[0000-0000]
     CR1B=[0010-0010]
     GR18=[xx0x-0xxx]
     CR8E=[xxxx-x1xx]
     xxFF=[0000-0000]
     ERXX=[---------]

REM  XMODE table entry 6

DOC extended mode 5C, 800x600x256colorsx72Hz

XMODE_NUMBER=[5c]
XMODE_VESA_NUMBER=[103]
XMODE_HORIZONTAL_RES=[800]
XMODE_VERTICAL_RES=[600]
XMODE_COLOR=[08]
XMODE_CHAR_WIDTH=[08]
XMODE_CHAR_HEIGHT=[10]
XMODE_FEATURE_BYTE_01=[04]
XMODE_FEATURE_BYTE_02=[1e]
XMODE_FEATURE_BYTE_03=[05]
XMODE_FEATURE_BYTE_04=[05]
XMODE_FEATURE_BYTE_05=[2f]
XMODE_OTHER_BYTE_01=[39]
XMODE_OTHER_BYTE_02=[08]
XMODE_OTHER_BYTE_03=[21]
XMODE_OTHER_BYTE_04=[e3]
XMODE_OTHER_BYTE_05=[1d]
XMODE_OTHER_BYTE_06=[33]
XMODE_OTHER_BYTE_07=[22]
XMODE_CONFIG_TABLE:
     SR0E=[0110-0101]
     SR1E=[0011-1010]
     SR07=[0000-0001]
     GR0B=[xxx0-0000]
     CR19=[0000-0000]
     CR1A=[0000-0000]
     CR1B=[0010-0010]
     GR18=[xx0x-0xxx]
     CR8E=[xxxx-x1xx]

     xxFF=[0000-0000]
     ERXX=[---------]

REM  XMODE table entry 7

DOC extended mode 5C, 800x600x256colorsx60Hz

XMODE_NUMBER=[5c]
XMODE_VESA_NUMBER=[103]
XMODE_HORIZONTAL_RES=[800]
XMODE_VERTICAL_RES=[600]
XMODE_COLOR=[08]
XMODE_CHAR_WIDTH=[08]
XMODE_CHAR_HEIGHT=[10]
XMODE_FEATURE_BYTE_01=[04]
XMODE_FEATURE_BYTE_02=[1e]
XMODE_FEATURE_BYTE_03=[04]
XMODE_FEATURE_BYTE_04=[05]
XMODE_FEATURE_BYTE_05=[2f]
XMODE_OTHER_BYTE_01=[39]
XMODE_OTHER_BYTE_02=[08]
XMODE_OTHER_BYTE_03=[11]
XMODE_OTHER_BYTE_04=[d6]
XMODE_OTHER_BYTE_05=[1d]
XMODE_OTHER_BYTE_06=[33]
XMODE_OTHER_BYTE_07=[22]
XMODE_CONFIG_TABLE:
     SR0E=[0101-0001]
     SR1E=[0011-1010]
     SR07=[0000-0001]
     GR0B=[xxx0-0000]
     CR19=[0000-0000]
     CR1A=[0000-0000]
     CR1B=[0010-0010]
     GR18=[xx0x-0xxx]
     CR8E=[xxxx-x1xx]
     xxFF=[0000-0000]
     ERXX=[---------]


REM  XMODE table entry 8

DOC extended mode 5C, 800x600x256colorsx56Hz

XMODE_NUMBER=[5c]
XMODE_VESA_NUMBER=[103]
XMODE_HORIZONTAL_RES=[800]
XMODE_VERTICAL_RES=[600]
XMODE_COLOR=[08]
XMODE_CHAR_WIDTH=[08]
XMODE_CHAR_HEIGHT=[10]
XMODE_FEATURE_BYTE_01=[04]
XMODE_FEATURE_BYTE_02=[1e]
XMODE_FEATURE_BYTE_03=[ff]
XMODE_FEATURE_BYTE_04=[05]
XMODE_FEATURE_BYTE_05=[2f]
XMODE_OTHER_BYTE_01=[39]
XMODE_OTHER_BYTE_02=[08]
XMODE_OTHER_BYTE_03=[01]
XMODE_OTHER_BYTE_04=[88]
XMODE_OTHER_BYTE_05=[1d]
XMODE_OTHER_BYTE_06=[33]
XMODE_OTHER_BYTE_07=[22]
XMODE_CONFIG_TABLE:
     SR0E=[0111-1110]
     SR1E=[0011-0011]
     SR07=[0000-0001]
     GR0B=[xxx0-0000]
     CR19=[0000-0000]
     CR1A=[0000-0000]
     CR1B=[0010-0010]
     GR18=[xx0x-0xxx]
     CR8E=[xxxx-x1xx]
     xxFF=[0000-0000]
     ERXX=[---------]


REM  XMODE table entry 9

DOC extended mode 5D, 1024x768x16colorsx75Hz

XMODE_NUMBER=[5d]
XMODE_VESA_NUMBER=[104]
```

```
XMODE_HORIZONTAL_RES=[1024]
XMODE_VERTICAL_RES=[768]
XMODE_COLOR=[04]
XMODE_CHAR_WIDTH=[08]
XMODE_CHAR_HEIGHT=[10]
XMODE_FEATURE_BYTE_01=[03]
XMODE_FEATURE_BYTE_02=[1e]
XMODE_FEATURE_BYTE_03=[0c]
XMODE_FEATURE_BYTE_04=[06]
XMODE_FEATURE_BYTE_05=[2f]
XMODE_OTHER_BYTE_01=[39]
XMODE_OTHER_BYTE_02=[06]
XMODE_OTHER_BYTE_03=[42]
XMODE_OTHER_BYTE_04=[24]
XMODE_OTHER_BYTE_05=[1e]
XMODE_OTHER_BYTE_06=[33]
XMODE_OTHER_BYTE_07=[00]
XMODE_CONFIG_TABLE:
     SR0E=[0010-1100]
     SR1E=[0001-0000]
     SR07=[0000-0000]
     GR0B=[xxx0-0000]
     CR19=[0100-1010]
     CR1A=[0000-0000]
     CR1B=[0010-0010]
     GR18=[xx0x-1xxx]
     CR8E=[xxxx-x1xx]
     xxFF=[0000-0000]
     ERXX=[---------]


REM  XMODE table entry 10

DOC extended mode 5D, 1024x768x16colorsx72Hz

XMODE_NUMBER=[5d]
XMODE_VESA_NUMBER=[104]
XMODE_HORIZONTAL_RES=[1024]
XMODE_VERTICAL_RES=[768]
XMODE_COLOR=[04]
XMODE_CHAR_WIDTH=[08]
XMODE_CHAR_HEIGHT=[10]
XMODE_FEATURE_BYTE_01=[03]
XMODE_FEATURE_BYTE_02=[1e]
XMODE_FEATURE_BYTE_03=[0b]
XMODE_FEATURE_BYTE_04=[06]
XMODE_FEATURE_BYTE_05=[2f]
XMODE_OTHER_BYTE_01=[39]
XMODE_OTHER_BYTE_02=[06]
XMODE_OTHER_BYTE_03=[32]
XMODE_OTHER_BYTE_04=[17]
XMODE_OTHER_BYTE_05=[1e]
XMODE_OTHER_BYTE_06=[33]
XMODE_OTHER_BYTE_07=[00]
XMODE_CONFIG_TABLE:
     SR0E=[0110-0010]
     SR1E=[0010-0100]
     SR07=[0000-0000]
     GR0B=[xxx0-0000]
     CR19=[0100-1010]
     CR1A=[0000-0000]
     CR1B=[0010-0010]
     GR18=[xx0x-1xxx]
     CR8E=[xxxx-x1xx]
     xxFF=[0000-0000]
     ERXX=[---------]


REM  XMODE table entry 11

DOC extended mode 5D, 1024x768x16colorsx70Hz

XMODE_NUMBER=[5d]
XMODE_VESA_NUMBER=[104]
XMODE_HORIZONTAL_RES=[1024]
XMODE_VERTICAL_RES=[768]
XMODE_COLOR=[04]
XMODE_CHAR_WIDTH=[08]
XMODE_CHAR_HEIGHT=[10]
XMODE_FEATURE_BYTE_01=[03]
XMODE_FEATURE_BYTE_02=[1e]
XMODE_FEATURE_BYTE_03=[0a]
XMODE_FEATURE_BYTE_04=[06]
XMODE_FEATURE_BYTE_05=[ef]
XMODE_OTHER_BYTE_01=[39]
XMODE_OTHER_BYTE_02=[06]
XMODE_OTHER_BYTE_03=[22]
XMODE_OTHER_BYTE_04=[0a]
XMODE_OTHER_BYTE_05=[1e]
XMODE_OTHER_BYTE_06=[33]
XMODE_OTHER_BYTE_07=[00]
XMODE_CONFIG_TABLE:
     SR0E=[0110-1110]
     SR1E=[0010-1010]
     SR07=[0000-0000]
     GR0B=[xxx0-0000]
     CR19=[0100-1010]
     CR1A=[0000-0000]
     CR1B=[0010-0010]
     GR18=[xx0x-1xxx]
     CR8E=[xxxx-x1xx]
     xxFF=[0000-0000]
     ERXX=[---------]

REM  XMODE table entry 12

DOC  extended mode 5D, 1024x768x16colorsx60Hz

XMODE_NUMBER=[5d]
XMODE_VESA_NUMBER=[104]
XMODE_HORIZONTAL_RES=[1024]
XMODE_VERTICAL_RES=[768]
XMODE_COLOR=[04]
XMODE_CHAR_WIDTH=[08]
XMODE_CHAR_HEIGHT=[10]
XMODE_FEATURE_BYTE_01=[03]
XMODE_FEATURE_BYTE_02=[1e]
XMODE_FEATURE_BYTE_03=[09]
XMODE_FEATURE_BYTE_04=[06]
XMODE_FEATURE_BYTE_05=[2f]
XMODE_OTHER_BYTE_01=[39]
XMODE_OTHER_BYTE_02=[06]
XMODE_OTHER_BYTE_03=[12]
XMODE_OTHER_BYTE_04=[fd]
XMODE_OTHER_BYTE_05=[1d]
XMODE_OTHER_BYTE_06=[33]
```

```
XMODE_OTHER_BYTE_07=[00]
XMODE_CONFIG_TABLE:
     SR0E=[0011-1011]
     SR1E=[0001-1010]
     SR07=[0000-0000]
     GR0B=[xxx0-0000]
     CR19=[0100-1010]
     CR1A=[0000-0000]
     CR1B=[0010-0010]
     GR18=[xx0x-1xxx]
     CR8E=[xxxx-x1xx]
     xxFF=[0000-0000]
     ERXX=[---------]

REM  XMODE table entry 13

DOC  extended mode 5D,
DOC  1024x768x16colorsx87Hz/interlaced

XMODE_NUMBER=[5d]
XMODE_VESA_NUMBER=[104]
XMODE_HORIZONTAL_RES=[1024]
XMODE_VERTICAL_RES=[768]
XMODE_COLOR=[04]
XMODE_CHAR_WIDTH=[08]
XMODE_CHAR_HEIGHT=[10]
XMODE_FEATURE_BYTE_01=[03]
XMODE_FEATURE_BYTE_02=[1e]
XMODE_FEATURE_BYTE_03=[ff]
XMODE_FEATURE_BYTE_04=[06]
XMODE_FEATURE_BYTE_05=[2f]
XMODE_OTHER_BYTE_01=[39]
XMODE_OTHER_BYTE_02=[06]
XMODE_OTHER_BYTE_03=[02]
XMODE_OTHER_BYTE_04=[88]
XMODE_OTHER_BYTE_05=[1d]
XMODE_OTHER_BYTE_06=[00]
XMODE_OTHER_BYTE_07=[00]
XMODE_CONFIG_TABLE:
     SR0E=[0101-0101]
     SR1E=[0011-0110]
     SR07=[0000-0000]
     GR0B=[xxx0-0000]
     CR19=[0100-1010]
     CR1A=[0000-0001]
     CR1B=[0010-0010]
     GR18=[xx0x-0xxx]
     CR8E=[xxxx-x1xx]
     xxFF=[0000-0000]
     ERXX=[---------]

REM  XMODE table entry 14

DOC  extended mode 5E, 640x400x256colorsx70Hz

XMODE_NUMBER=[5e]
XMODE_VESA_NUMBER=[100]
XMODE_HORIZONTAL_RES=[640]
XMODE_VERTICAL_RES=[400]
XMODE_COLOR=[08]
XMODE_CHAR_WIDTH=[08]
XMODE_CHAR_HEIGHT=[10]
XMODE_FEATURE_BYTE_01=[04]
XMODE_FEATURE_BYTE_02=[1e]
XMODE_FEATURE_BYTE_03=[ff]
XMODE_FEATURE_BYTE_04=[08]
XMODE_FEATURE_BYTE_05=[6f]
XMODE_OTHER_BYTE_01=[39]
XMODE_OTHER_BYTE_02=[04]
XMODE_OTHER_BYTE_03=[00]
XMODE_OTHER_BYTE_04=[88]
XMODE_OTHER_BYTE_05=[1d]
XMODE_OTHER_BYTE_06=[33]
XMODE_OTHER_BYTE_07=[22]
XMODE_CONFIG_TABLE:
     SR0E=[0110-0110]
     SR1E=[0011-1011]
     SR07=[0000-0001]
     GR0B=[xxx0-0000]
     CR19=[0000-0000]
     CR1A=[0000-0000]
     CR1B=[0010-0010]
     GR18=[xx0x-0xxx]
     CR8E=[xxxx-x1xx]
     xxFF=[0000-0000]
     ERXX=[---------]

REM  XMODE table entry 15

DOC  extended mode 5F, 640x480x256colorsx60Hz

XMODE_NUMBER=[5f]
XMODE_VESA_NUMBER=[101]
XMODE_HORIZONTAL_RES=[640]
XMODE_VERTICAL_RES=[480]
XMODE_COLOR=[08]
XMODE_CHAR_WIDTH=[08]
XMODE_CHAR_HEIGHT=[10]
XMODE_FEATURE_BYTE_01=[04]
XMODE_FEATURE_BYTE_02=[1e]
XMODE_FEATURE_BYTE_03=[ff]
XMODE_FEATURE_BYTE_04=[00]
XMODE_FEATURE_BYTE_05=[ef]
XMODE_OTHER_BYTE_01=[39]
XMODE_OTHER_BYTE_02=[05]
XMODE_OTHER_BYTE_03=[00]
XMODE_OTHER_BYTE_04=[88]
XMODE_OTHER_BYTE_05=[1d]
XMODE_OTHER_BYTE_06=[33]
XMODE_OTHER_BYTE_07=[22]
XMODE_CONFIG_TABLE:
     SR0E=[0110-0110]
     SR1E=[0011-1011]
     SR07=[0000-0001]
     GR0B=[xxx0-0000]
     CR19=[0000-0000]
     CR1A=[0000-0000]
     CR1B=[0010-0010]
     GR18=[xx0x-0xxx]
     CR8E=[xxxx-x1xx]
     xxFF=[0000-0000]
     ERXX=[---------]

REM  XMODE table entry 16

DOC  extended mode 60,
```

```
DOC   1024x768x256colorsx85Hz

XMODE_NUMBER=[60]
XMODE_VESA_NUMBER=[105]
XMODE_HORIZONTAL_RES=[1024]
XMODE_VERTICAL_RES=[768]
XMODE_COLOR=[08]
XMODE_CHAR_WIDTH=[08]
XMODE_CHAR_HEIGHT=[10]
XMODE_FEATURE_BYTE_01=[04]
XMODE_FEATURE_BYTE_02=[1e]
XMODE_FEATURE_BYTE_03=[0d]
XMODE_FEATURE_BYTE_04=[06]
XMODE_FEATURE_BYTE_05=[2f]
XMODE_OTHER_BYTE_01=[38]
XMODE_OTHER_BYTE_02=[0c]
XMODE_OTHER_BYTE_03=[52]
XMODE_OTHER_BYTE_04=[99]
XMODE_OTHER_BYTE_05=[1e]
XMODE_OTHER_BYTE_06=[33]
XMODE_OTHER_BYTE_07=[00]
XMODE_CONFIG_TABLE:
      SR0E=[0111-0000]
      SR1E=[0010-0010]
      SR07=[0000-0001]
      GR0B=[xxx0-0000]
      CR19=[0100-1010]
      CR1A=[0000-0000]
      CR1B=[0010-0010]
      GR18=[xx0x-1xxx]
      CR8E=[xxxx-x1xx]
      xxFF=[0000-0000]
      ERXX=[---------]

REM   XMODE table entry 17

DOC   extended mode 60,
DOC   1024x768x256colorsx75Hz

XMODE_NUMBER=[60]
XMODE_VESA_NUMBER=[105]
XMODE_HORIZONTAL_RES=[1024]
XMODE_VERTICAL_RES=[768]
XMODE_COLOR=[08]
XMODE_CHAR_WIDTH=[08]
XMODE_CHAR_HEIGHT=[10]
XMODE_FEATURE_BYTE_01=[04]
XMODE_FEATURE_BYTE_02=[1e]
XMODE_FEATURE_BYTE_03=[0c]
XMODE_FEATURE_BYTE_04=[06]
XMODE_FEATURE_BYTE_05=[2f]
XMODE_OTHER_BYTE_01=[39]
XMODE_OTHER_BYTE_02=[0c]
XMODE_OTHER_BYTE_03=[42]
XMODE_OTHER_BYTE_04=[24]
XMODE_OTHER_BYTE_05=[1e]
XMODE_OTHER_BYTE_06=[33]
XMODE_OTHER_BYTE_07=[00]
XMODE_CONFIG_TABLE:
      SR0E=[0010-1100]
      SR1E=[0001-0000]
      SR07=[0000-0001]
      GR0B=[xxx0-0000]
      CR19=[0100-1010]
      CR1A=[0000-0000]
      CR1B=[0010-0010]
      GR18=[xx0x-1xxx]
      CR8E=[xxxx-x1xx]
      xxFF=[0000-0000]
      ERXX=[---------]

REM   XMODE table entry 18

DOC   extended mode 60,
DOC   1024x768x256colorsx72Hz

XMODE_NUMBER=[60]
XMODE_VESA_NUMBER=[105]
XMODE_HORIZONTAL_RES=[1024]
XMODE_VERTICAL_RES=[768]
XMODE_COLOR=[08]
XMODE_CHAR_WIDTH=[08]
XMODE_CHAR_HEIGHT=[10]
XMODE_FEATURE_BYTE_01=[04]
XMODE_FEATURE_BYTE_02=[1e]
XMODE_FEATURE_BYTE_03=[0b]
XMODE_FEATURE_BYTE_04=[06]
XMODE_FEATURE_BYTE_05=[2f]
XMODE_OTHER_BYTE_01=[39]
XMODE_OTHER_BYTE_02=[0c]
XMODE_OTHER_BYTE_03=[32]
XMODE_OTHER_BYTE_04=[17]
XMODE_OTHER_BYTE_05=[1e]
XMODE_OTHER_BYTE_06=[33]
XMODE_OTHER_BYTE_07=[00]
XMODE_CONFIG_TABLE:
      SR0E=[0110-0010]
      SR1E=[0010-0100]
      SR07=[0000-0001]
      GR0B=[xxx0-0000]
      CR19=[0100-1010]
      CR1A=[0000-0000]
      CR1B=[0010-0010]
      GR18=[xx0x-1xxx]
      CR8E=[xxxx-x1xx]
      xxFF=[0000-0000]
      ERXX=[---------]

REM   XMODE table entry 19

DOC   extended mode 60,
DOC   1024x768x256colorsx70Hz

XMODE_NUMBER=[60]
XMODE_VESA_NUMBER=[105]
XMODE_HORIZONTAL_RES=[1024]
XMODE_VERTICAL_RES=[768]
XMODE_COLOR=[08]
XMODE_CHAR_WIDTH=[08]
XMODE_CHAR_HEIGHT=[10]
XMODE_FEATURE_BYTE_01=[04]
XMODE_FEATURE_BYTE_02=[1e]
XMODE_FEATURE_BYTE_03=[0a]
XMODE_FEATURE_BYTE_04=[06]
XMODE_FEATURE_BYTE_05=[ef]
XMODE_OTHER_BYTE_01=[39]
```

```
XMODE_OTHER_BYTE_02=[0c]
XMODE_OTHER_BYTE_03=[22]
XMODE_OTHER_BYTE_04=[0a]
XMODE_OTHER_BYTE_05=[1e]
XMODE_OTHER_BYTE_06=[33]
XMODE_OTHER_BYTE_07=[00]
XMODE_CONFIG_TABLE:
      SR0E=[0110-1110]
      SR1E=[0010-1010]
      SR07=[0000-0001]
      GR0B=[xxx0-0000]
      CR19=[0100-1010]
      CR1A=[0000-0000]
      CR1B=[0010-0010]
      GR18=[xx0x-1xxx]
      CR8E=[xxxx-x1xx]
      xxFF=[0000-0000]
      ERXX=[---------]

REM   XMODE table entry 20

DOC   extended mode 60,
DOC   1024x768x256colorsx60Hz

XMODE_NUMBER=[60]
XMODE_VESA_NUMBER=[105]
XMODE_HORIZONTAL_RES=[1024]
XMODE_VERTICAL_RES=[768]
XMODE_COLOR=[08]
XMODE_CHAR_WIDTH=[08]
XMODE_CHAR_HEIGHT=[10]
XMODE_FEATURE_BYTE_01=[04]
XMODE_FEATURE_BYTE_02=[1e]
XMODE_FEATURE_BYTE_03=[09]
XMODE_FEATURE_BYTE_04=[06]
XMODE_FEATURE_BYTE_05=[2f]
XMODE_OTHER_BYTE_01=[39]
XMODE_OTHER_BYTE_02=[0c]
XMODE_OTHER_BYTE_03=[12]
XMODE_OTHER_BYTE_04=[fd]
XMODE_OTHER_BYTE_05=[1d]
XMODE_OTHER_BYTE_06=[33]
XMODE_OTHER_BYTE_07=[00]
XMODE_CONFIG_TABLE:
      SR0E=[0011-1011]
      SR1E=[0001-1010]
      SR07=[0000-0001]
      GR0B=[xxx0-0000]
      CR19=[0100-1010]
      CR1A=[0000-0000]
      CR1B=[0010-0010]
      GR18=[xx0x-1xxx]
      CR8E=[xxxx-x1xx]
      xxFF=[0000-0000]
      ERXX=[---------]

REM   XMODE table entry 21

DOC   extended mode 60,
DOC   1024x768x256colorsx87Hz/interlaced

XMODE_NUMBER=[60]
XMODE_VESA_NUMBER=[105]
XMODE_HORIZONTAL_RES=[1024]
XMODE_VERTICAL_RES=[768]
XMODE_COLOR=[08]
XMODE_CHAR_WIDTH=[08]
XMODE_CHAR_HEIGHT=[10]
XMODE_FEATURE_BYTE_01=[04]
XMODE_FEATURE_BYTE_02=[1e]
XMODE_FEATURE_BYTE_03=[ff]
XMODE_FEATURE_BYTE_04=[06]
XMODE_FEATURE_BYTE_05=[2f]
XMODE_OTHER_BYTE_01=[39]
XMODE_OTHER_BYTE_02=[0c]
XMODE_OTHER_BYTE_03=[02]
XMODE_OTHER_BYTE_04=[88]
XMODE_OTHER_BYTE_05=[1d]
XMODE_OTHER_BYTE_06=[00]
XMODE_OTHER_BYTE_07=[00]
XMODE_CONFIG_TABLE:
      SR0E=[0101-0101]
      SR1E=[0011-0110]
      SR07=[0000-0001]
      GR0B=[xxx0-0000]
      CR19=[0100-1010]
      CR1A=[0000-0001]
      CR1B=[0010-0010]
      GR18=[xx0x-0xxx]
      CR8E=[xxxx-x1xx]
      xxFF=[0000-0000]
      ERXX=[---------]

REM   XMODE table entry 22

DOC   extended mode 64, 640x480x64Kcolorsx60Hz

XMODE_NUMBER=[64]
XMODE_VESA_NUMBER=[111]
XMODE_HORIZONTAL_RES=[640]
XMODE_VERTICAL_RES=[480]
XMODE_COLOR=[10]
XMODE_CHAR_WIDTH=[08]
XMODE_CHAR_HEIGHT=[10]
XMODE_FEATURE_BYTE_01=[56]
XMODE_FEATURE_BYTE_02=[1a]
XMODE_FEATURE_BYTE_03=[ff]
XMODE_FEATURE_BYTE_04=[00]
XMODE_FEATURE_BYTE_05=[ef]
XMODE_OTHER_BYTE_01=[39]
XMODE_OTHER_BYTE_02=[0a]
XMODE_OTHER_BYTE_03=[00]
XMODE_OTHER_BYTE_04=[8c]
XMODE_OTHER_BYTE_05=[1e]
XMODE_OTHER_BYTE_06=[33]
XMODE_OTHER_BYTE_07=[22]
XMODE_CONFIG_TABLE:
      SR0E=[0110-0110]
      SR1E=[0011-1011]
      SR07=[0000-0111]
      GR0B=[xxx0-0000]
      CR19=[0000-0000]
      CR1A=[0000-0000]
      CR1B=[0010-0010]
      GR18=[xx0x-0xxx]
      CR8E=[xxxx-x1xx]
```

```
     xxFF=[1100-0001]
     ERXX=[---------]

REM  XMODE table entry 23

DOC  extended mode 65, 800x600x64Kcolorsx85Hz

XMODE_NUMBER=[65]
XMODE_VESA_NUMBER=[114]
XMODE_HORIZONTAL_RES=[800]
XMODE_VERTICAL_RES=[600]
XMODE_COLOR=[10]
XMODE_CHAR_WIDTH=[08]
XMODE_CHAR_HEIGHT=[10]
XMODE_FEATURE_BYTE_01=[56]
XMODE_FEATURE_BYTE_02=[1a]
XMODE_FEATURE_BYTE_03=[07]
XMODE_FEATURE_BYTE_04=[05]
XMODE_FEATURE_BYTE_05=[2f]
XMODE_OTHER_BYTE_01=[39]
XMODE_OTHER_BYTE_02=[0f]
XMODE_OTHER_BYTE_03=[41]
XMODE_OTHER_BYTE_04=[a6]
XMODE_OTHER_BYTE_05=[1e]
XMODE_OTHER_BYTE_06=[33]
XMODE_OTHER_BYTE_07=[22]
XMODE_CONFIG_TABLE:
     SR0E=[0111-1010]
     SR1E=[0011-1110]
     SR07=[0000-0111]
     GR0B=[xxx0-0000]
     CR19=[0000-0000]
     CR1A=[0000-0000]
     CR1B=[0010-0010]
     GR18=[xx0x-0xxx]
     CR8E=[xxxx-x1xx]
     xxFF=[1100-0001]
     ERXX=[---------]
REM  XMODE table entry 24

DOC  extended mode 65, 800x600x64Kcolorsx75Hz

XMODE_NUMBER=[65]
XMODE_VESA_NUMBER=[114]
XMODE_HORIZONTAL_RES=[800]
XMODE_VERTICAL_RES=[600]
XMODE_COLOR=[10]
XMODE_CHAR_WIDTH=[08]
XMODE_CHAR_HEIGHT=[10]
XMODE_FEATURE_BYTE_01=[56]
XMODE_FEATURE_BYTE_02=[1a]
XMODE_FEATURE_BYTE_03=[06]
XMODE_FEATURE_BYTE_04=[05]
XMODE_FEATURE_BYTE_05=[2f]
XMODE_OTHER_BYTE_01=[39]
XMODE_OTHER_BYTE_02=[0f]
XMODE_OTHER_BYTE_03=[31]
XMODE_OTHER_BYTE_04=[f0]
XMODE_OTHER_BYTE_05=[1d]
XMODE_OTHER_BYTE_06=[33]
XMODE_OTHER_BYTE_07=[22]
XMODE_CONFIG_TABLE:
     SR0E=[0101-0011]
     SR1E=[0011-0000]
     SR07=[0000-0111]
     GR0B=[xxx0-0000]
     CR19=[0000-0000]
     CR1A=[0000-0000]
     CR1B=[0010-0010]
     GR18=[xx0x-0xxx]
     CR8E=[xxxx-x1xx]
     xxFF=[1100-0001]
     ERXX=[---------]

REM  XMODE table entry 25

DOC  extended mode 65, 800x600x64Kcolorsx72Hz

XMODE_NUMBER=[65]
XMODE_VESA_NUMBER=[114]
XMODE_HORIZONTAL_RES=[800]
XMODE_VERTICAL_RES=[600]
XMODE_COLOR=[10]
XMODE_CHAR_WIDTH=[08]
XMODE_CHAR_HEIGHT=[10]
XMODE_FEATURE_BYTE_01=[56]
XMODE_FEATURE_BYTE_02=[1a]
XMODE_FEATURE_BYTE_03=[05]
XMODE_FEATURE_BYTE_04=[05]
XMODE_FEATURE_BYTE_05=[2f]
XMODE_OTHER_BYTE_01=[39]
XMODE_OTHER_BYTE_02=[0f]
XMODE_OTHER_BYTE_03=[21]
XMODE_OTHER_BYTE_04=[e3]
XMODE_OTHER_BYTE_05=[1d]
XMODE_OTHER_BYTE_06=[33]
XMODE_OTHER_BYTE_07=[22]
XMODE_CONFIG_TABLE:
     SR0E=[0110-0101]
     SR1E=[0011-1010]
     SR07=[0000-0111]
     GR0B=[xxx0-0000]
     CR19=[0000-0000]
     CR1A=[0000-0000]
     CR1B=[0010-0010]
     GR18=[xx0x-0xxx]
     CR8E=[xxxx-x1xx]
     xxFF=[1100-0001]
     ERXX=[---------]

REM  XMODE table entry 26

DOC  extended mode 65, 800x600x64Kcolorsx60Hz

XMODE_NUMBER=[65]
XMODE_VESA_NUMBER=[114]
XMODE_HORIZONTAL_RES=[800]
XMODE_VERTICAL_RES=[600]
XMODE_COLOR=[10]
XMODE_CHAR_WIDTH=[08]
XMODE_CHAR_HEIGHT=[10]
XMODE_FEATURE_BYTE_01=[56]
XMODE_FEATURE_BYTE_02=[1a]
XMODE_FEATURE_BYTE_03=[04]
XMODE_FEATURE_BYTE_04=[05]
XMODE_FEATURE_BYTE_05=[2f]
```

```
XMODE_OTHER_BYTE_01=[39]
XMODE_OTHER_BYTE_02=[0f]
XMODE_OTHER_BYTE_03=[11]
XMODE_OTHER_BYTE_04=[7f]
XMODE_OTHER_BYTE_05=[1e]
XMODE_OTHER_BYTE_06=[33]
XMODE_OTHER_BYTE_07=[22]
XMODE_CONFIG_TABLE:
     SR0E=[0101-1111]
     SR1E=[0010-0011]
     SR07=[0000-0111]
     GR0B=[xxx0-0000]
     CR19=[0000-0000]
     CR1A=[0000-0000]
     CR1B=[0010-0010]
     GR18=[xx0x-0xxx]
     CR8E=[xxxx-x1xx]
     xxFF=[1100-0001]
     ERXX=[---------]

REM  XMODE table entry 27

DOC  extended mode 65, 800x600x64Kcolorsx56Hz

XMODE_NUMBER=[65]
XMODE_VESA_NUMBER=[114]
XMODE_HORIZONTAL_RES=[800]
XMODE_VERTICAL_RES=[600]
XMODE_COLOR=[10]
XMODE_CHAR_WIDTH=[08]
XMODE_CHAR_HEIGHT=[10]
XMODE_FEATURE_BYTE_01=[56]
XMODE_FEATURE_BYTE_02=[1a]
XMODE_FEATURE_BYTE_03=[ff]
XMODE_FEATURE_BYTE_04=[05]
XMODE_FEATURE_BYTE_05=[2f]
XMODE_OTHER_BYTE_01=[39]
XMODE_OTHER_BYTE_02=[0f]
XMODE_OTHER_BYTE_03=[01]
XMODE_OTHER_BYTE_04=[88]
XMODE_OTHER_BYTE_05=[1d]
XMODE_OTHER_BYTE_06=[33]
XMODE_OTHER_BYTE_07=[22]
XMODE_CONFIG_TABLE:
     SR0E=[0111-1110]
     SR1E=[0011-0011]
     SR07=[0000-0111]
     GR0B=[xxx0-0000]
     CR19=[0000-0000]
     CR1A=[0000-0000]
     CR1B=[0010-0010]
     GR18=[xx0x-0xxx]
     CR8E=[xxxx-x1xx]
     xxFF=[1100-0001]
     ERXX=[---------]

REM  XMODE table entry 28

DOC  extended mode 66, 640x480x32Kcolorsx60Hz

XMODE_NUMBER=[66]
XMODE_VESA_NUMBER=[110]
XMODE_HORIZONTAL_RES=[640]
XMODE_VERTICAL_RES=[480]
XMODE_COLOR=[0f]
XMODE_CHAR_WIDTH=[08]
XMODE_CHAR_HEIGHT=[10]
XMODE_FEATURE_BYTE_01=[16]
XMODE_FEATURE_BYTE_02=[1a]
XMODE_FEATURE_BYTE_03=[ff]
XMODE_FEATURE_BYTE_04=[00]
XMODE_FEATURE_BYTE_05=[ef]
XMODE_OTHER_BYTE_01=[39]
XMODE_OTHER_BYTE_02=[0a]
XMODE_OTHER_BYTE_03=[00]
XMODE_OTHER_BYTE_04=[8c]
XMODE_OTHER_BYTE_05=[1e]
XMODE_OTHER_BYTE_06=[33]
XMODE_OTHER_BYTE_07=[22]
XMODE_CONFIG_TABLE:
     SR0E=[0110-0110]
     SR1E=[0011-1011]
     SR07=[0000-0111]
     GR0B=[xxx0-0000]
     CR19=[0000-0000]
     CR1A=[0000-0000]
     CR1B=[0000-0010]
     GR18=[xx0x-0xxx]
     CR8E=[xxxx-x1xx]
     xxFF=[1100-0000]
     ERXX=[---------]

REM  XMODE table entry 29

DOC  extended mode 67, 800x600x32Kcolorsx85Hz

XMODE_NUMBER=[67]
XMODE_VESA_NUMBER=[113]
XMODE_HORIZONTAL_RES=[800]
XMODE_VERTICAL_RES=[600]
XMODE_COLOR=[0f]
XMODE_CHAR_WIDTH=[08]
XMODE_CHAR_HEIGHT=[10]
XMODE_FEATURE_BYTE_01=[16]
XMODE_FEATURE_BYTE_02=[1a]
XMODE_FEATURE_BYTE_03=[07]
XMODE_FEATURE_BYTE_04=[05]
XMODE_FEATURE_BYTE_05=[2f]
XMODE_OTHER_BYTE_01=[39]
XMODE_OTHER_BYTE_02=[0f]
XMODE_OTHER_BYTE_03=[41]
XMODE_OTHER_BYTE_04=[a6]
XMODE_OTHER_BYTE_05=[1e]
XMODE_OTHER_BYTE_06=[33]
XMODE_OTHER_BYTE_07=[22]
XMODE_CONFIG_TABLE:
     SR0E=[0111-1010]
     SR1E=[0011-1110]
     SR07=[0000-0111]
     GR0B=[xxx0-0000]
     CR19=[0000-0000]
     CR1A=[0000-0000]
     CR1B=[0010-0010]
     GR18=[xx0x-0xxx]
     CR8E=[xxxx-x1xx]
     xxFF=[1100-0000]
```

```
     ERXX=[---------]

REM  XMODE table entry 30

DOC  extended mode 67, 800x600x32Kcolorsx75Hz

XMODE_NUMBER=[67]
XMODE_VESA_NUMBER=[113]
XMODE_HORIZONTAL_RES=[800]
XMODE_VERTICAL_RES=[600]
XMODE_COLOR=[0f]
XMODE_CHAR_WIDTH=[08]
XMODE_CHAR_HEIGHT=[10]
XMODE_FEATURE_BYTE_01=[16]
XMODE_FEATURE_BYTE_02=[1a]
XMODE_FEATURE_BYTE_03=[06]
XMODE_FEATURE_BYTE_04=[05]
XMODE_FEATURE_BYTE_05=[2f]
XMODE_OTHER_BYTE_01=[39]
XMODE_OTHER_BYTE_02=[0f]
XMODE_OTHER_BYTE_03=[31]
XMODE_OTHER_BYTE_04=[f0]
XMODE_OTHER_BYTE_05=[1d]
XMODE_OTHER_BYTE_06=[33]
XMODE_OTHER_BYTE_07=[22]
XMODE_CONFIG_TABLE:
     SR0E=[0101-0011]
     SR1E=[0011-0000]
     SR07=[0000-0111]
     GR0B=[xxx0-0000]
     CR19=[0000-0000]
     CR1A=[0000-0000]
     CR1B=[0010-0010]
     GR18=[xx0x-0xxx]
     CR8E=[xxxx-x1xx]
     xxFF=[1100-0000]
     ERXX=[---------]

REM  XMODE table entry 31

DOC  extended mode 67, 800x600x32Kcolorsx72Hz

XMODE_NUMBER=[67]
XMODE_VESA_NUMBER=[113]
XMODE_HORIZONTAL_RES=[800]
XMODE_VERTICAL_RES=[600]
XMODE_COLOR=[0f]
XMODE_CHAR_WIDTH=[08]
XMODE_CHAR_HEIGHT=[10]
XMODE_FEATURE_BYTE_01=[16]
XMODE_FEATURE_BYTE_02=[1a]
XMODE_FEATURE_BYTE_03=[05]
XMODE_FEATURE_BYTE_04=[05]
XMODE_FEATURE_BYTE_05=[2f]
XMODE_OTHER_BYTE_01=[39]
XMODE_OTHER_BYTE_02=[0f]
XMODE_OTHER_BYTE_03=[21]
XMODE_OTHER_BYTE_04=[e3]
XMODE_OTHER_BYTE_05=[1d]
XMODE_OTHER_BYTE_06=[33]
XMODE_OTHER_BYTE_07=[22]
XMODE_CONFIG_TABLE:
     SR0E=[0110-0101]
     SR1E=[0011-1010]
     SR07=[0000-0111]
     GR0B=[xxx0-0000]
     CR19=[0000-0000]
     CR1A=[0000-0000]
     CR1B=[0010-0010]
     GR18=[xx0x-0xxx]
     CR8E=[xxxx-x1xx]
     xxFF=[1100-0000]
     ERXX=[---------]

REM  XMODE table entry 32

DOC  extended mode 67, 800x600x32Kcolorsx60Hz

XMODE_NUMBER=[67]
XMODE_VESA_NUMBER=[113]
XMODE_HORIZONTAL_RES=[800]
XMODE_VERTICAL_RES=[600]
XMODE_COLOR=[0f]
XMODE_CHAR_WIDTH=[08]
XMODE_CHAR_HEIGHT=[10]
XMODE_FEATURE_BYTE_01=[16]
XMODE_FEATURE_BYTE_02=[1a]
XMODE_FEATURE_BYTE_03=[04]
XMODE_FEATURE_BYTE_04=[05]
XMODE_FEATURE_BYTE_05=[2f]
XMODE_OTHER_BYTE_01=[39]
XMODE_OTHER_BYTE_02=[0f]
XMODE_OTHER_BYTE_03=[11]
XMODE_OTHER_BYTE_04=[7f]
XMODE_OTHER_BYTE_05=[1e]
XMODE_OTHER_BYTE_06=[33]
XMODE_OTHER_BYTE_07=[22]
XMODE_CONFIG_TABLE:
     SR0E=[0101-1111]
     SR1E=[0010-0011]
     SR07=[0000-0111]
     GR0B=[xxx0-0000]
     CR19=[0000-0000]
     CR1A=[0000-0000]
     CR1B=[0010-0010]
     GR18=[xx0x-0xxx]
     CR8E=[xxxx-x1xx]
     xxFF=[1100-0000]
     ERXX=[---------]

REM  XMODE table entry 33

DOC  extended mode 67, 800x600x32Kcolorsx56Hz

XMODE_NUMBER=[67]
XMODE_VESA_NUMBER=[113]
XMODE_HORIZONTAL_RES=[800]
XMODE_VERTICAL_RES=[600]
XMODE_COLOR=[0f]
XMODE_CHAR_WIDTH=[08]
XMODE_CHAR_HEIGHT=[10]
XMODE_FEATURE_BYTE_01=[16]
XMODE_FEATURE_BYTE_02=[1a]
XMODE_FEATURE_BYTE_03=[ff]
XMODE_FEATURE_BYTE_04=[05]
XMODE_FEATURE_BYTE_05=[2f]
```

```
XMODE_OTHER_BYTE_01=[39]
XMODE_OTHER_BYTE_02=[0f]
XMODE_OTHER_BYTE_03=[01]
XMODE_OTHER_BYTE_04=[88]
XMODE_OTHER_BYTE_05=[1d]
XMODE_OTHER_BYTE_06=[33]
XMODE_OTHER_BYTE_07=[22]
XMODE_CONFIG_TABLE:
     SR0E=[0111-1110]
     SR1E=[0011-0011]
     SR07=[0000-0111]
     GR0B=[xxx0-0000]
     CR19=[0000-0000]
     CR1A=[0000-0000]
     CR1B=[0010-0010]
     GR18=[xx0x-0xxx]
     CR8E=[xxxx-x1xx]
     xxFF=[1100-0000]
     ERXX=[---------]

REM  XMODE table entry 34

DOC  extended mode 68,
DOC  1024x768x32Kcolorsx85Hz

XMODE_NUMBER=[68]
XMODE_VESA_NUMBER=[116]
XMODE_HORIZONTAL_RES=[1024]
XMODE_VERTICAL_RES=[768]
XMODE_COLOR=[0f]
XMODE_CHAR_WIDTH=[08]
XMODE_CHAR_HEIGHT=[10]
XMODE_FEATURE_BYTE_01=[16]
XMODE_FEATURE_BYTE_02=[1a]
XMODE_FEATURE_BYTE_03=[0d]
XMODE_FEATURE_BYTE_04=[06]
XMODE_FEATURE_BYTE_05=[2f]
XMODE_OTHER_BYTE_01=[5a]
XMODE_OTHER_BYTE_02=[17]
XMODE_OTHER_BYTE_03=[52]
XMODE_OTHER_BYTE_04=[99]
XMODE_OTHER_BYTE_05=[1e]
XMODE_OTHER_BYTE_06=[33]
XMODE_OTHER_BYTE_07=[00]
XMODE_CONFIG_TABLE:
     SR0E=[0111-0000]
     SR1E=[0010-0010]
     SR07=[0000-0111]
     GR0B=[xxx0-0000]
     CR19=[0100-1010]
     CR1A=[0000-0000]
     CR1B=[0011-0010]
     GR18=[xx0x-1xxx]
     CR8E=[xxxx-x1xx]
     xxFF=[1100-0000]
     ERXX=[---------]

REM  XMODE table entry 35

DOC  extended mode 68, 1024x768x32Kcolorsx75

XMODE_NUMBER=[68]
XMODE_VESA_NUMBER=[116]
XMODE_HORIZONTAL_RES=[1024]
XMODE_VERTICAL_RES=[768]
XMODE_COLOR=[0f]
XMODE_CHAR_WIDTH=[08]
XMODE_CHAR_HEIGHT=[10]
XMODE_FEATURE_BYTE_01=[16]
XMODE_FEATURE_BYTE_02=[1a]
XMODE_FEATURE_BYTE_03=[0c]
XMODE_FEATURE_BYTE_04=[06]
XMODE_FEATURE_BYTE_05=[2f]
XMODE_OTHER_BYTE_01=[45]
XMODE_OTHER_BYTE_02=[17]
XMODE_OTHER_BYTE_03=[42]
XMODE_OTHER_BYTE_04=[24]
XMODE_OTHER_BYTE_05=[1e]
XMODE_OTHER_BYTE_06=[33]
XMODE_OTHER_BYTE_07=[00]
XMODE_CONFIG_TABLE:
     SR0E=[0010-1100]
     SR1E=[0001-0000]
     SR07=[0000-0111]
     GR0B=[xxx0-0000]
     CR19=[0000-0000]
     CR1A=[0000-0000]
     CR1B=[0011-0010]
     GR18=[xx0x-1xxx]
     CR8E=[xxxx-x1xx]
     xxFF=[1100-0000]
     ERXX=[---------]

REM  XMODE table entry 36

DOC  extended mode 68,
DOC  1024x768x32Kcolorsx70Hz

XMODE_NUMBER=[68]
XMODE_VESA_NUMBER=[116]
XMODE_HORIZONTAL_RES=[1024]
XMODE_VERTICAL_RES=[768]
XMODE_COLOR=[0f]
XMODE_CHAR_WIDTH=[08]
XMODE_CHAR_HEIGHT=[10]
XMODE_FEATURE_BYTE_01=[16]
XMODE_FEATURE_BYTE_02=[1a]
XMODE_FEATURE_BYTE_03=[0a]
XMODE_FEATURE_BYTE_04=[06]
XMODE_FEATURE_BYTE_05=[ef]
XMODE_OTHER_BYTE_01=[45]
XMODE_OTHER_BYTE_02=[17]
XMODE_OTHER_BYTE_03=[22]
XMODE_OTHER_BYTE_04=[0a]
XMODE_OTHER_BYTE_05=[1e]
XMODE_OTHER_BYTE_06=[33]
XMODE_OTHER_BYTE_07=[00]
XMODE_CONFIG_TABLE:
     SR0E=[0110-1110]
     SR1E=[0010-1010]
     SR07=[0000-0111]
     GR0B=[xxx0-0000]
     CR19=[0000-0000]
     CR1A=[0000-0000]
     CR1B=[0011-0010]
     GR18=[xx0x-1xxx]
```

```
     CR8E=[xxxx-x1xx]
     xxFF=[1100-0000]
     ERXX=[---------]

REM  XMODE table entry 37

DOC  extended mode 68,
DOC  1024x768x32Kcolorsx60Hz

XMODE_NUMBER=[68]
XMODE_VESA_NUMBER=[116]
XMODE_HORIZONTAL_RES=[1024]
XMODE_VERTICAL_RES=[768]
XMODE_COLOR=[0f]
XMODE_CHAR_WIDTH=[08]
XMODE_CHAR_HEIGHT=[10]
XMODE_FEATURE_BYTE_01=[16]
XMODE_FEATURE_BYTE_02=[1a]
XMODE_FEATURE_BYTE_03=[09]
XMODE_FEATURE_BYTE_04=[06]
XMODE_FEATURE_BYTE_05=[2f]
XMODE_OTHER_BYTE_01=[39]
XMODE_OTHER_BYTE_02=[17]
XMODE_OTHER_BYTE_03=[12]
XMODE_OTHER_BYTE_04=[fd]
XMODE_OTHER_BYTE_05=[1d]
XMODE_OTHER_BYTE_06=[33]
XMODE_OTHER_BYTE_07=[00]
XMODE_CONFIG_TABLE:
     SR0E=[0011-1011]
     SR1E=[0001-1010]
     SR07=[0000-0111]
     GR0B=[xxx0-0000]
     CR19=[0000-0000]
     CR1A=[0000-0000]
     CR1B=[0011-0010]
     GR18=[xx0x-1xxx]
     CR8E=[xxxx-x1xx]
     xxFF=[1100-0000]
     ERXX=[---------]


REM  XMODE table entry 38

DOC  extended mode 68,
DOC  1024x768x32Kcolorsx87Hz/interlaced

XMODE_NUMBER=[68]
XMODE_VESA_NUMBER=[116]
XMODE_HORIZONTAL_RES=[1024]
XMODE_VERTICAL_RES=[768]
XMODE_COLOR=[0f]
XMODE_CHAR_WIDTH=[08]
XMODE_CHAR_HEIGHT=[10]
XMODE_FEATURE_BYTE_01=[16]
XMODE_FEATURE_BYTE_02=[1a]
XMODE_FEATURE_BYTE_03=[ff]
XMODE_FEATURE_BYTE_04=[06]
XMODE_FEATURE_BYTE_05=[2f]
XMODE_OTHER_BYTE_01=[39]
XMODE_OTHER_BYTE_02=[17]
XMODE_OTHER_BYTE_03=[02]
XMODE_OTHER_BYTE_04=[88]
XMODE_OTHER_BYTE_05=[1d]
XMODE_OTHER_BYTE_06=[00]
XMODE_OTHER_BYTE_07=[00]
XMODE_CONFIG_TABLE:
     SR0E=[0101-0101]
     SR1E=[0011-0110]
     SR07=[0000-0111]
     GR0B=[xxx0-0000]
     CR19=[0100-1010]
     CR1A=[0000-0001]
     CR1B=[0011-0010]
     GR18=[xx0x-0xxx]
     CR8E=[xxxx-x1xx]
     xxFF=[1100-0000]
     ERXX=[---------]

REM  XMODE table entry 39

DOC  extended mode 69,
DOC  1280x1024x32Kcolorsx87Hz/interlaced

XMODE_NUMBER=[69]
XMODE_VESA_NUMBER=[119]
XMODE_HORIZONTAL_RES=[1280]
XMODE_VERTICAL_RES=[1024]
XMODE_COLOR=[0f]
XMODE_CHAR_WIDTH=[08]
XMODE_CHAR_HEIGHT=[10]
XMODE_FEATURE_BYTE_01=[16]
XMODE_FEATURE_BYTE_02=[1a]
XMODE_FEATURE_BYTE_03=[0e]
XMODE_FEATURE_BYTE_04=[07]
XMODE_FEATURE_BYTE_05=[2f]
XMODE_OTHER_BYTE_01=[39]
XMODE_OTHER_BYTE_02=[26]
XMODE_OTHER_BYTE_03=[03]
XMODE_OTHER_BYTE_04=[3e]
XMODE_OTHER_BYTE_05=[1e]
XMODE_OTHER_BYTE_06=[00]
XMODE_OTHER_BYTE_07=[00]
XMODE_CONFIG_TABLE:
     SR0E=[0110-1110]
     SR1E=[0010-1010]
     SR07=[0000-0111]
     GR0B=[xxx0-0000]
     CR19=[0110-0000]
     CR1A=[0000-0001]
     CR1B=[0011-0010]
     GR18=[xx0x-0xxx]
     CR8E=[xxxx-x1xx]
     xxFF=[1100-0000]
     ERXX=[---------]

REM  XMODE table entry 40

DOC  extended mode 6A/58,
DOC  800x600x16colorsx75Hz

XMODE_NUMBER=[6a]
XMODE_VESA_NUMBER=[102]
XMODE_HORIZONTAL_RES=[800]
XMODE_VERTICAL_RES=[600]
XMODE_COLOR=[04]
```

```
XMODE_CHAR_WIDTH=[08]
XMODE_CHAR_HEIGHT=[10]
XMODE_FEATURE_BYTE_01=[03]
XMODE_FEATURE_BYTE_02=[1e]
XMODE_FEATURE_BYTE_03=[06]
XMODE_FEATURE_BYTE_04=[05]
XMODE_FEATURE_BYTE_05=[2f]
XMODE_OTHER_BYTE_01=[39]
XMODE_OTHER_BYTE_02=[04]
XMODE_OTHER_BYTE_03=[31]
XMODE_OTHER_BYTE_04=[f0]
XMODE_OTHER_BYTE_05=[1d]
XMODE_OTHER_BYTE_06=[33]
XMODE_OTHER_BYTE_07=[22]
XMODE_CONFIG_TABLE:
     SR0E=[0101-0011]
     SR1E=[0011-0000]
     SR07=[0000-0000]
     GR0B=[xxx0-0000]
     CR19=[0000-0000]
     CR1A=[0000-0000]
     CR1B=[0010-0010]
     GR18=[xx0x-0xxx]
     CR8E=[xxxx-x1xx]
     xxFF=[0000-0000]
     ERXX=[---------]

REM  XMODE table entry 41

DOC  extended mode 6A/58,
DOC  800x600x16colorsx72Hz

XMODE_NUMBER=[6a]
XMODE_VESA_NUMBER=[102]
XMODE_HORIZONTAL_RES=[800]
XMODE_VERTICAL_RES=[600]
XMODE_COLOR=[04]
XMODE_CHAR_WIDTH=[08]
XMODE_CHAR_HEIGHT=[10]
XMODE_FEATURE_BYTE_01=[03]
XMODE_FEATURE_BYTE_02=[1e]
XMODE_FEATURE_BYTE_03=[05]
XMODE_FEATURE_BYTE_04=[05]
XMODE_FEATURE_BYTE_05=[2f]
XMODE_OTHER_BYTE_01=[39]
XMODE_OTHER_BYTE_02=[04]
XMODE_OTHER_BYTE_03=[21]
XMODE_OTHER_BYTE_04=[e3]
XMODE_OTHER_BYTE_05=[1d]
XMODE_OTHER_BYTE_06=[33]
XMODE_OTHER_BYTE_07=[22]
XMODE_CONFIG_TABLE:
     SR0E=[0110-0101]
     SR1E=[0011-1010]
     SR07=[0000-0000]
     GR0B=[xxx0-0000]
     CR19=[0000-0000]
     CR1A=[0000-0000]
     CR1B=[0010-0010]
     GR18=[xx0x-0xxx]
     CR8E=[xxxx-x1xx]
     xxFF=[0000-0000]
     ERXX=[---------]

REM  XMODE table entry 42

DOC  extended mode 6A/58,
DOC  800x600x16colorsx60Hz

XMODE_NUMBER=[6a]
XMODE_VESA_NUMBER=[102]
XMODE_HORIZONTAL_RES=[800]
XMODE_VERTICAL_RES=[600]
XMODE_COLOR=[04]
XMODE_CHAR_WIDTH=[08]
XMODE_CHAR_HEIGHT=[10]
XMODE_FEATURE_BYTE_01=[03]
XMODE_FEATURE_BYTE_02=[1e]
XMODE_FEATURE_BYTE_03=[04]
XMODE_FEATURE_BYTE_04=[05]
XMODE_FEATURE_BYTE_05=[2f]
XMODE_OTHER_BYTE_01=[39]
XMODE_OTHER_BYTE_02=[04]
XMODE_OTHER_BYTE_03=[11]
XMODE_OTHER_BYTE_04=[d6]
XMODE_OTHER_BYTE_05=[1d]
XMODE_OTHER_BYTE_06=[33]
XMODE_OTHER_BYTE_07=[22]
XMODE_CONFIG_TABLE:
     SR0E=[0101-0001]
     SR1E=[0011-1010]
     SR07=[0000-0000]
     GR0B=[xxx0-0000]
     CR19=[0000-0000]
     CR1A=[0000-0000]
     CR1B=[0010-0010]
     GR18=[xx0x-0xxx]
     CR8E=[xxxx-x1xx]
     xxFF=[0000-0000]
     ERXX=[---------]


REM  XMODE table entry 43

DOC  extended mode 6A/58,
DOC  800x600x16colorsx56Hz

XMODE_NUMBER=[6a]
XMODE_VESA_NUMBER=[102]
XMODE_HORIZONTAL_RES=[800]
XMODE_VERTICAL_RES=[600]
XMODE_COLOR=[04]
XMODE_CHAR_WIDTH=[08]
XMODE_CHAR_HEIGHT=[10]
XMODE_FEATURE_BYTE_01=[03]
XMODE_FEATURE_BYTE_02=[1e]
XMODE_FEATURE_BYTE_03=[ff]
XMODE_FEATURE_BYTE_04=[05]
XMODE_FEATURE_BYTE_05=[2f]
XMODE_OTHER_BYTE_01=[39]
XMODE_OTHER_BYTE_02=[04]
XMODE_OTHER_BYTE_03=[01]
XMODE_OTHER_BYTE_04=[88]
XMODE_OTHER_BYTE_05=[1d]
XMODE_OTHER_BYTE_06=[33]
XMODE_OTHER_BYTE_07=[22]
```

```
XMODE_CONFIG_TABLE:
     SR0E=[0111-1110]
     SR1E=[0011-0011]
     SR07=[0000-0000]
     GR0B=[xxx0-0000]
     CR19=[0000-0000]
     CR1A=[0000-0000]
     CR1B=[0010-0010]
     GR18=[xx0x-0xxx]
     CR8E=[xxxx-x1xx]
     xxFF=[0000-0000]
     ERXX=[---------]

REM  XMODE table entry 44

DOC  extended mode 6C,
DOC  1280x1024x16colorsx87Hz/interlaced

XMODE_NUMBER=[6c]
XMODE_VESA_NUMBER=[106]
XMODE_HORIZONTAL_RES=[1280]
XMODE_VERTICAL_RES=[1024]
XMODE_COLOR=[04]
XMODE_CHAR_WIDTH=[08]
XMODE_CHAR_HEIGHT=[10]
XMODE_FEATURE_BYTE_01=[03]
XMODE_FEATURE_BYTE_02=[1a]
XMODE_FEATURE_BYTE_03=[ff]
XMODE_FEATURE_BYTE_04=[07]
XMODE_FEATURE_BYTE_05=[2f]
XMODE_OTHER_BYTE_01=[39]
XMODE_OTHER_BYTE_02=[0a]
XMODE_OTHER_BYTE_03=[03]
XMODE_OTHER_BYTE_04=[88]
XMODE_OTHER_BYTE_05=[1d]
XMODE_OTHER_BYTE_06=[00]
XMODE_OTHER_BYTE_07=[00]
XMODE_CONFIG_TABLE:
     SR0E=[0110-1110]
     SR1E=[0010-1010]
     SR07=[0000-0000]
     GR0B=[xxx0-0000]
     CR19=[0110-0000]
     CR1A=[0000-0001]
     CR1B=[0010-0010]
     GR18=[xx0x-0xxx]
     CR8E=[xxxx-x1xx]
     xxFF=[0000-0000]
     ERXX=[---------]

REM  XMODE table entry 45

DOC  extended mode 6D,
DOC  1280x1024x256colorsx60Hz

XMODE_NUMBER=[6d]
XMODE_VESA_NUMBER=[107]
XMODE_HORIZONTAL_RES=[1280]
XMODE_VERTICAL_RES=[1024]
XMODE_COLOR=[08]
XMODE_CHAR_WIDTH=[08]
XMODE_CHAR_HEIGHT=[10]
XMODE_FEATURE_BYTE_01=[04]
XMODE_FEATURE_BYTE_02=[1a]
XMODE_FEATURE_BYTE_03=[10]
XMODE_FEATURE_BYTE_04=[07]
XMODE_FEATURE_BYTE_05=[2f]
XMODE_OTHER_BYTE_01=[5a]
XMODE_OTHER_BYTE_02=[14]
XMODE_OTHER_BYTE_03=[13]
XMODE_OTHER_BYTE_04=[58]
XMODE_OTHER_BYTE_05=[1e]
XMODE_OTHER_BYTE_06=[00]
XMODE_OTHER_BYTE_07=[00]
XMODE_CONFIG_TABLE:
     SR0E=[0101-0011]
     SR1E=[0001-0110]
     SR07=[0000-0001]
     GR0B=[xxx0-0000]
     CR19=[0000-0000]
     CR1A=[0000-0000]
     CR1B=[0010-0010]
     GR18=[xx0x-1xxx]
     CR8E=[xxxx-x0xx]
     xxFF=[0000-0000]
     ERXX=[---------]

REM  XMODE table entry 46

DOC  extended mode 6D,
DOC  1280x1024x256colorsx87Hz/interlaced

XMODE_NUMBER=[6d]
XMODE_VESA_NUMBER=[107]
XMODE_HORIZONTAL_RES=[1280]
XMODE_VERTICAL_RES=[1024]
XMODE_COLOR=[08]
XMODE_CHAR_WIDTH=[08]
XMODE_CHAR_HEIGHT=[10]
XMODE_FEATURE_BYTE_01=[04]
XMODE_FEATURE_BYTE_02=[1a]
XMODE_FEATURE_BYTE_03=[ff]
XMODE_FEATURE_BYTE_04=[07]
XMODE_FEATURE_BYTE_05=[2f]
XMODE_OTHER_BYTE_01=[39]
XMODE_OTHER_BYTE_02=[14]
XMODE_OTHER_BYTE_03=[03]
XMODE_OTHER_BYTE_04=[88]
XMODE_OTHER_BYTE_05=[1d]
XMODE_OTHER_BYTE_06=[00]
XMODE_OTHER_BYTE_07=[00]
XMODE_CONFIG_TABLE:
     SR0E=[0110-1110]
     SR1E=[0010-1010]
     SR07=[0000-0001]
     GR0B=[xxx0-0000]
     CR19=[0110-0000]
     CR1A=[0000-0001]
     CR1B=[0010-0010]
     GR18=[xx0x-0xxx]
     CR8E=[xxxx-x0xx]
     xxFF=[0000-0000]
     ERXX=[---------]

REM  XMODE table entry 47
```

```
DOC extended mode 71, 640x480x16Mcolorsx60Hz

XMODE_NUMBER=[71]
XMODE_VESA_NUMBER=[112]
XMODE_HORIZONTAL_RES=[640]
XMODE_VERTICAL_RES=[480]
XMODE_COLOR=[18]
XMODE_CHAR_WIDTH=[08]
XMODE_CHAR_HEIGHT=[10]
XMODE_FEATURE_BYTE_01=[a6]
XMODE_FEATURE_BYTE_02=[1a]
XMODE_FEATURE_BYTE_03=[ff]
XMODE_FEATURE_BYTE_04=[00]
XMODE_FEATURE_BYTE_05=[ef]
XMODE_OTHER_BYTE_01=[39]
XMODE_OTHER_BYTE_02=[0f]
XMODE_OTHER_BYTE_03=[00]
XMODE_OTHER_BYTE_04=[88]
XMODE_OTHER_BYTE_05=[1d]
XMODE_OTHER_BYTE_06=[33]
XMODE_OTHER_BYTE_07=[25]
XMODE_CONFIG_TABLE:
     SR0E=[0110-0110]
     SR1E=[0011-1011]
     SR07=[0000-0101]
     GR0B=[xxx0-0000]
     CR19=[0000-0000]
     CR1A=[0000-0000]
     CR1B=[0011-0010]
     GR18=[xx0x-0xxx]
     CR8E=[xxxx-x0xx]
     xxFF=[1110-0101]
     ERXX=[---------]

REM  XMODE table entry 48

DOC  extended mode 74, 1024x768x64Kcolorx85Hz

XMODE_NUMBER=[74]
XMODE_VESA_NUMBER=[117]
XMODE_HORIZONTAL_RES=[1024]
XMODE_VERTICAL_RES=[768]
XMODE_COLOR=[10]
XMODE_CHAR_WIDTH=[08]
XMODE_CHAR_HEIGHT=[10]
XMODE_FEATURE_BYTE_01=[56]
XMODE_FEATURE_BYTE_02=[1a]
XMODE_FEATURE_BYTE_03=[0d]
XMODE_FEATURE_BYTE_04=[06]
XMODE_FEATURE_BYTE_05=[2f]
XMODE_OTHER_BYTE_01=[5a]
XMODE_OTHER_BYTE_02=[18]
XMODE_OTHER_BYTE_03=[52]
XMODE_OTHER_BYTE_04=[99]
XMODE_OTHER_BYTE_05=[1e]
XMODE_OTHER_BYTE_06=[33]
XMODE_OTHER_BYTE_07=[00]
XMODE_CONFIG_TABLE:
     SR0E=[0111-0000]
     SR1E=[0010-0010]
     SR07=[0000-0111]
     GR0B=[xxx0-0000]
     CR19=[0100-1010]
     CR1A=[0000-0000]
     CR1B=[0011-0010]
     GR18=[xx0x-1xxx]
     CR8E=[xxxx-x0xx]
     xxFF=[1100-0001]
     ERXX=[---------]

REM  XMODE table entry 49

DOC  extended mode 74, 1024x768x64Kcolorx75Hz

XMODE_NUMBER=[74]
XMODE_VESA_NUMBER=[117]
XMODE_HORIZONTAL_RES=[1024]
XMODE_VERTICAL_RES=[768]
XMODE_COLOR=[10]
XMODE_CHAR_WIDTH=[08]
XMODE_CHAR_HEIGHT=[10]
XMODE_FEATURE_BYTE_01=[56]
XMODE_FEATURE_BYTE_02=[1a]
XMODE_FEATURE_BYTE_03=[0c]
XMODE_FEATURE_BYTE_04=[06]
XMODE_FEATURE_BYTE_05=[2f]
XMODE_OTHER_BYTE_01=[45]
XMODE_OTHER_BYTE_02=[18]
XMODE_OTHER_BYTE_03=[42]
XMODE_OTHER_BYTE_04=[24]
XMODE_OTHER_BYTE_05=[1e]
XMODE_OTHER_BYTE_06=[33]
XMODE_OTHER_BYTE_07=[00]
XMODE_CONFIG_TABLE:
     SR0E=[0010-1100]
     SR1E=[0001-0000]
     SR07=[0000-0111]
     GR0B=[xxx0-0000]
     CR19=[0000-0000]
     CR1A=[0000-0000]
     CR1B=[0011-0010]
     GR18=[xx0x-1xxx]
     CR8E=[xxxx-x1xx]
     xxFF=[1100-0001]
     ERXX=[---------]

REM  XMODE table entry 50

DOC  extended mode 74, 1024x768x64Kcolorx70Hz

XMODE_NUMBER=[74]
XMODE_VESA_NUMBER=[117]
XMODE_HORIZONTAL_RES=[1024]
XMODE_VERTICAL_RES=[768]
XMODE_COLOR=[10]
XMODE_CHAR_WIDTH=[08]
XMODE_CHAR_HEIGHT=[10]
XMODE_FEATURE_BYTE_01=[56]
XMODE_FEATURE_BYTE_02=[1a]
XMODE_FEATURE_BYTE_03=[0a]
XMODE_FEATURE_BYTE_04=[06]
XMODE_FEATURE_BYTE_05=[ef]
XMODE_OTHER_BYTE_01=[45]
XMODE_OTHER_BYTE_02=[18]
XMODE_OTHER_BYTE_03=[22]
XMODE_OTHER_BYTE_04=[0a]
```

```
XMODE_OTHER_BYTE_05=[1e]
XMODE_OTHER_BYTE_06=[33]
XMODE_OTHER_BYTE_07=[00]
XMODE_CONFIG_TABLE:
     SR0E=[0110-1110]
     SR1E=[0010-1010]
     SR07=[0000-0111]
     GR0B=[xxx0-0000]
     CR19=[0000-0000]
     CR1A=[0000-0000]
     CR1B=[0011-0010]
     GR18=[xx0x-1xxx]
     CR8E=[xxxx-x1xx]
     xxFF=[1100-0001]
     ERXX=[---------]

REM  XMODE table entry 51

DOC  extended mode 74,
DOC  1024x768x64Kcolorsx60Hz

XMODE_NUMBER=[74]
XMODE_VESA_NUMBER=[117]
XMODE_HORIZONTAL_RES=[1024]
XMODE_VERTICAL_RES=[768]
XMODE_COLOR=[10]
XMODE_CHAR_WIDTH=[08]
XMODE_CHAR_HEIGHT=[10]
XMODE_FEATURE_BYTE_01=[56]
XMODE_FEATURE_BYTE_02=[1a]
XMODE_FEATURE_BYTE_03=[09]
XMODE_FEATURE_BYTE_04=[06]
XMODE_FEATURE_BYTE_05=[2f]
XMODE_OTHER_BYTE_01=[39]
XMODE_OTHER_BYTE_02=[18]
XMODE_OTHER_BYTE_03=[12]
XMODE_OTHER_BYTE_04=[fd]
XMODE_OTHER_BYTE_05=[1d]
XMODE_OTHER_BYTE_06=[33]
XMODE_OTHER_BYTE_07=[00]
XMODE_CONFIG_TABLE:
     SR0E=[0011-1011]
     SR1E=[0001-1010]
     SR07=[0000-0111]
     GR0B=[xxx0-0000]
     CR19=[0000-0000]
     CR1A=[0000-0000]
     CR1B=[0011-0010]
     GR18=[xx0x-1xxx]
     CR8E=[xxxx-x1xx]
     xxFF=[1100-0001]
     ERXX=[---------]

REM  XMODE table entry 52

DOC  extended mode 74,
DOC  1024x768x64Kcolorsx87Hz/interlaced

XMODE_NUMBER=[74]
XMODE_VESA_NUMBER=[117]
XMODE_HORIZONTAL_RES=[1024]
XMODE_VERTICAL_RES=[768]
XMODE_COLOR=[10]
XMODE_CHAR_WIDTH=[08]
XMODE_CHAR_HEIGHT=[10]
XMODE_FEATURE_BYTE_01=[56]
XMODE_FEATURE_BYTE_02=[1a]
XMODE_FEATURE_BYTE_03=[08]
XMODE_FEATURE_BYTE_04=[06]
XMODE_FEATURE_BYTE_05=[2f]
XMODE_OTHER_BYTE_01=[39]
XMODE_OTHER_BYTE_02=[18]
XMODE_OTHER_BYTE_03=[02]
XMODE_OTHER_BYTE_04=[31]
XMODE_OTHER_BYTE_05=[1e]
XMODE_OTHER_BYTE_06=[00]
XMODE_OTHER_BYTE_07=[00]
XMODE_CONFIG_TABLE:
     SR0E=[0101-0101]
     SR1E=[0011-0110]
     SR07=[0000-0111]
     GR0B=[xxx0-0000]
     CR19=[0100-1010]
     CR1A=[0000-0001]
     CR1B=[0011-0010]
     GR18=[xx0x-0xxx]
     CR8E=[xxxx-x1xx]
     xxFF=[1100-0001]
     ERXX=[---------]

REM  XMODE table entry 53

DOC  extended mode 75,
DOC  1280x1024x64Kcolorsx87Hz/interlaced

XMODE_NUMBER=[75]
XMODE_VESA_NUMBER=[11a]
XMODE_HORIZONTAL_RES=[1280]
XMODE_VERTICAL_RES=[1024]
XMODE_COLOR=[10]
XMODE_CHAR_WIDTH=[08]
XMODE_CHAR_HEIGHT=[10]
XMODE_FEATURE_BYTE_01=[56]
XMODE_FEATURE_BYTE_02=[1a]
XMODE_FEATURE_BYTE_03=[ff]
XMODE_FEATURE_BYTE_04=[07]
XMODE_FEATURE_BYTE_05=[2f]
XMODE_OTHER_BYTE_01=[45]
XMODE_OTHER_BYTE_02=[28]
XMODE_OTHER_BYTE_03=[03]
XMODE_OTHER_BYTE_04=[88]
XMODE_OTHER_BYTE_05=[1d]
XMODE_OTHER_BYTE_06=[00]
XMODE_OTHER_BYTE_07=[00]
XMODE_CONFIG_TABLE:
     SR0E=[0110-1110]
     SR1E=[0010-1010]
     SR07=[0000-0111]
     GR0B=[xxx0-0000]
     CR19=[0110-0000]
     CR1A=[0000-0001]
     CR1B=[0011-0010]
     GR18=[xx0x-0xxx]
     CR8E=[xxxx-x1xx]
     xxFF=[1100-0001]
     ERXX=[---------]
```

```
REM   XMODE table entry 54

DOC   extended mode 78, 800x600x16Mcolorsx85Hz

XMODE_NUMBER=[78]
XMODE_VESA_NUMBER=[115]
XMODE_HORIZONTAL_RES=[800]
XMODE_VERTICAL_RES=[600]
XMODE_COLOR=[18]
XMODE_CHAR_WIDTH=[08]
XMODE_CHAR_HEIGHT=[10]
XMODE_FEATURE_BYTE_01=[a6]
XMODE_FEATURE_BYTE_02=[1a]
XMODE_FEATURE_BYTE_03=[07]
XMODE_FEATURE_BYTE_04=[05]
XMODE_FEATURE_BYTE_05=[2f]
XMODE_OTHER_BYTE_01=[4b]
XMODE_OTHER_BYTE_02=[17]
XMODE_OTHER_BYTE_03=[41]
XMODE_OTHER_BYTE_04=[a6]
XMODE_OTHER_BYTE_05=[1e]
XMODE_OTHER_BYTE_06=[33]
XMODE_OTHER_BYTE_07=[25]
XMODE_CONFIG_TABLE:
      SR0E=[0111-1010]
      SR1E=[0011-1110]
      SR07=[0000-0101]
      GR0B=[xxx0-0000]
      CR19=[0000-0000]
      CR1A=[0000-0000]
      CR1B=[0011-0010]
      GR18=[xx0x-0xxx]
      CR8E=[xxxx-x1xx]
      xxFF=[1110-0101]
      ERXX=[---------]


REM   XMODE table entry 55

DOC   extended mode 78, 800x600x16Mcolorsx75Hz

XMODE_NUMBER=[78]
XMODE_VESA_NUMBER=[115]
XMODE_HORIZONTAL_RES=[800]
XMODE_VERTICAL_RES=[600]
XMODE_COLOR=[18]
XMODE_CHAR_WIDTH=[08]
XMODE_CHAR_HEIGHT=[10]
XMODE_FEATURE_BYTE_01=[a6]
XMODE_FEATURE_BYTE_02=[1a]
XMODE_FEATURE_BYTE_03=[06]
XMODE_FEATURE_BYTE_04=[05]
XMODE_FEATURE_BYTE_05=[2f]
XMODE_OTHER_BYTE_01=[45]
XMODE_OTHER_BYTE_02=[16]
XMODE_OTHER_BYTE_03=[31]
XMODE_OTHER_BYTE_04=[f0]
XMODE_OTHER_BYTE_05=[1d]
XMODE_OTHER_BYTE_06=[33]
XMODE_OTHER_BYTE_07=[25]
XMODE_CONFIG_TABLE:
      SR0E=[0101-0011]
      SR1E=[0011-0000]
      SR07=[0000-0101]
      GR0B=[xxx0-0000]
      CR19=[0000-0000]
      CR1A=[0000-0000]
      CR1B=[0011-0010]
      GR18=[xx0x-0xxx]
      CR8E=[xxxx-x1xx]
      xxFF=[1110-0101]
      ERXX=[---------]

REM   XMODE table entry 56

DOC   extended mode 78, 800x600x16Mcolorsx72Hz

XMODE_NUMBER=[78]
XMODE_VESA_NUMBER=[115]
XMODE_HORIZONTAL_RES=[800]
XMODE_VERTICAL_RES=[600]
XMODE_COLOR=[18]
XMODE_CHAR_WIDTH=[08]
XMODE_CHAR_HEIGHT=[10]
XMODE_FEATURE_BYTE_01=[a6]
XMODE_FEATURE_BYTE_02=[1a]
XMODE_FEATURE_BYTE_03=[05]
XMODE_FEATURE_BYTE_04=[05]
XMODE_FEATURE_BYTE_05=[2f]
XMODE_OTHER_BYTE_01=[45]
XMODE_OTHER_BYTE_02=[16]
XMODE_OTHER_BYTE_03=[21]
XMODE_OTHER_BYTE_04=[e3]
XMODE_OTHER_BYTE_05=[1d]
XMODE_OTHER_BYTE_06=[33]
XMODE_OTHER_BYTE_07=[25]
XMODE_CONFIG_TABLE:
      SR0E=[0110-0101]
      SR1E=[0011-1010]
      SR07=[0000-0101]
      GR0B=[xxx0-0000]
      CR19=[0000-0000]
      CR1A=[0000-0000]
      CR1B=[0011-0010]
      GR18=[xx0x-0xxx]
      CR8E=[xxxx-x1xx]
      xxFF=[1110-0101]
      ERXX=[---------]

REM   XMODE table entry 57

DOC   extended mode 78, 800x600x16Mcolorsx60Hz

XMODE_NUMBER=[78]
XMODE_VESA_NUMBER=[115]
XMODE_HORIZONTAL_RES=[800]
XMODE_VERTICAL_RES=[600]
XMODE_COLOR=[18]
XMODE_CHAR_WIDTH=[08]
XMODE_CHAR_HEIGHT=[10]
XMODE_FEATURE_BYTE_01=[a6]
XMODE_FEATURE_BYTE_02=[1a]
XMODE_FEATURE_BYTE_03=[04]
XMODE_FEATURE_BYTE_04=[05]
XMODE_FEATURE_BYTE_05=[2f]
```

```
XMODE_OTHER_BYTE_01=[39]
XMODE_OTHER_BYTE_02=[16]
XMODE_OTHER_BYTE_03=[11]
XMODE_OTHER_BYTE_04=[7f]
XMODE_OTHER_BYTE_05=[1e]
XMODE_OTHER_BYTE_06=[33]
XMODE_OTHER_BYTE_07=[25]
XMODE_CONFIG_TABLE:
     SR0E=[0101-1111]
     SR1E=[0010-0011]
     SR07=[0000-0101]
     GR0B=[xxx0-0000]
     CR19=[0000-0000]
     CR1A=[0000-0000]
     CR1B=[0011-0010]
     GR18=[xx0x-0xxx]
     CR8E=[xxxx-x1xx]
     xxFF=[1110-0101]
     ERXX=[---------]

REM  XMODE table entry 58

DOC  extended mode 78, 800x600x16Mcolorsx56Hz

XMODE_NUMBER=[78]
XMODE_VESA_NUMBER=[115]
XMODE_HORIZONTAL_RES=[800]
XMODE_VERTICAL_RES=[600]
XMODE_COLOR=[18]
XMODE_CHAR_WIDTH=[08]
XMODE_CHAR_HEIGHT=[10]
XMODE_FEATURE_BYTE_01=[a6]
XMODE_FEATURE_BYTE_02=[1a]
XMODE_FEATURE_BYTE_03=[ff]
XMODE_FEATURE_BYTE_04=[05]
XMODE_FEATURE_BYTE_05=[2f]
XMODE_OTHER_BYTE_01=[39]
XMODE_OTHER_BYTE_02=[16]
XMODE_OTHER_BYTE_03=[01]
XMODE_OTHER_BYTE_04=[88]
XMODE_OTHER_BYTE_05=[1d]
XMODE_OTHER_BYTE_06=[33]
XMODE_OTHER_BYTE_07=[25]
XMODE_CONFIG_TABLE:
     SR0E=[0111-1110]
     SR1E=[0011-0011]
     SR07=[0000-0101]
     GR0B=[xxx0-0000]
     CR19=[0000-0000]
     CR1A=[0000-0000]
     CR1B=[0011-0010]
     GR18=[xx0x-0xxx]
     CR8E=[xxxx-x1xx]
     xxFF=[1110-0101]
     ERXX=[---------]

REM  XMODE table entry 59

DOC  extended mode 79,
DOC  1024x768x16Mcolorsx70Hz

XMODE_NUMBER=[79]
XMODE_VESA_NUMBER=[118]
XMODE_HORIZONTAL_RES=[1024]
XMODE_VERTICAL_RES=[768]
XMODE_COLOR=[18]
XMODE_CHAR_WIDTH=[08]
XMODE_CHAR_HEIGHT=[10]
XMODE_FEATURE_BYTE_01=[96]
XMODE_FEATURE_BYTE_02=[1a]
XMODE_FEATURE_BYTE_03=[0a]
XMODE_FEATURE_BYTE_04=[06]
XMODE_FEATURE_BYTE_05=[ef]
XMODE_OTHER_BYTE_01=[5a]
XMODE_OTHER_BYTE_02=[24]
XMODE_OTHER_BYTE_03=[22]
XMODE_OTHER_BYTE_04=[0a]
XMODE_OTHER_BYTE_05=[1e]
XMODE_OTHER_BYTE_06=[33]
XMODE_OTHER_BYTE_07=[00]
XMODE_CONFIG_TABLE:
     SR0E=[0110-1110]
     SR1E=[0010-1010]
     SR07=[0000-0101]
     GR0B=[xxx0-0000]
     CR19=[0000-0000]
     CR1A=[0000-0000]
     CR1B=[0011-0010]
     GR18=[xx0x-1xxx]
     CR8E=[xxxx-x1xx]
     xxFF=[1110-0101]
     ERXX=[---------]

REM  XMODE table entry 60

DOC  extended mode 79,
DOC  1024x768x16Mcolorsx60Hz

XMODE_NUMBER=[79]
XMODE_VESA_NUMBER=[118]
XMODE_HORIZONTAL_RES=[1024]
XMODE_VERTICAL_RES=[768]
XMODE_COLOR=[18]
XMODE_CHAR_WIDTH=[08]
XMODE_CHAR_HEIGHT=[10]
XMODE_FEATURE_BYTE_01=[96]
XMODE_FEATURE_BYTE_02=[1a]
XMODE_FEATURE_BYTE_03=[09]
XMODE_FEATURE_BYTE_04=[06]
XMODE_FEATURE_BYTE_05=[2f]
XMODE_OTHER_BYTE_01=[5a]
XMODE_OTHER_BYTE_02=[24]
XMODE_OTHER_BYTE_03=[12]
XMODE_OTHER_BYTE_04=[fd]
XMODE_OTHER_BYTE_05=[1d]
XMODE_OTHER_BYTE_06=[33]
XMODE_OTHER_BYTE_07=[00]
XMODE_CONFIG_TABLE:
     SR0E=[0011-1011]
     SR1E=[0001-1010]
     SR07=[0000-0101]
     GR0B=[xxx0-0000]
     CR19=[0000-0000]
     CR1A=[0000-0000]
     CR1B=[0011-0010]
     GR18=[xx0x-1xxx]
```

```
     CR8E=[xxxx-x1xx]
     xxFF=[1110-0101]
     ERXX=[---------]

REM  XMODE table entry 61

DOC  extended mode 79,
DOC  1024x768x16Mcolorsx87Hz/interlaced

XMODE_NUMBER=[79]
XMODE_VESA_NUMBER=[118]
XMODE_HORIZONTAL_RES=[1024]
XMODE_VERTICAL_RES=[768]
XMODE_COLOR=[18]
XMODE_CHAR_WIDTH=[08]
XMODE_CHAR_HEIGHT=[10]
XMODE_FEATURE_BYTE_01=[96]
XMODE_FEATURE_BYTE_02=[1a]
XMODE_FEATURE_BYTE_03=[08]
XMODE_FEATURE_BYTE_04=[06]
XMODE_FEATURE_BYTE_05=[2f]
XMODE_OTHER_BYTE_01=[45]
XMODE_OTHER_BYTE_02=[24]
XMODE_OTHER_BYTE_03=[02]
XMODE_OTHER_BYTE_04=[31]
XMODE_OTHER_BYTE_05=[1e]
XMODE_OTHER_BYTE_06=[00]
XMODE_OTHER_BYTE_07=[00]
XMODE_CONFIG_TABLE:
     SR0E=[0101-0101]
     SR1E=[0011-0110]
     SR07=[0000-0101]
     GR0B=[xxx0-0000]
     CR19=[0100-1010]
     CR1A=[0000-0001]
     CR1B=[0011-0010]
     GR18=[xx0x-0xxx]
     CR8E=[xxxx-x1xx]
     xxFF=[1110-0101]
     ERXX=[---------]


%%%  HW
REM
REM  Hardware-related configuration
REM  registers: set up during POST
REM

HW_CONFIG_TABLE:

DOC  NOTICE, for register information:
DOC  See "CL-GD7555/6 Advanced Hardware
DOC  Reference Manual"
DOC
DOC  BINARY DATA FORMAT.  Bit locations
DOC  marked as "x" can not be modified.

     SR0B=[0110-0110]
DOC  SR0B: Standard VCLK0 Numerator
     SR1B=[0011-1011]
DOC  SR1B: Standard VCLK0 Denominator
     SR0C=[0101-1011]
DOC  SR0C: Standard VCLK1 Numerator
     SR1C=[0010-1111]
DOC  SR1C: Standard VCLK1 Denominator
     SR0F=[x001-1x01]
DOC  SR0F: DRAM Control, must have
DOC  bits[4:3]=11 for 64-bit access
     SR1F=[0010-0101] (see HW_Config_Table CL-
     GD7556 only)
DOC  SR1F: MCLK Frequency and VCLK Source
     SR12=[xxxx-x000]
DOC  SR12: Hardware Cursor Enable
     SR17=[00xx-x0xx]
DOC  SR17: BitBlt Memory Map IO
     SR20=[0x00-0000]
DOC  SR20: Miscellaneous Control 2
     SR23=[0000-0000]
DOC  SR23: Software Configuration Register 1
     SR25=[xxxx-xx10]
DOC  SR25: FasText (tm) Mode Control
     SR2A=[xxxx-xxx0]
DOC  SR2A: Hardware Icon #0 Control
     SR2B=[xxxx-xxx0]
DOC  SR2B: Hardware Icon #1 Control
     SR2C=[xxxx-xxx0]
DOC  SR2C: Hardware Icon #2 Control
     SR2D=[xxxx-xxx0]
DOC  SR2D: Hardware Icon #3 Control
     SR2F=[0001-xxxx]
DOC  SR2F: Bit[4] = 1 Kill PCLK in CRT-only
     GR18=[0000-0110]
DOC  GR18: EDO RAM Register
     GR31=[0xxx-x00x]
DOC  GR31: BLT Start/Status Register
     CR3C=[xxx0-xxxx]
DOC  CR3C: Video Window Enable
     CR3F=[xxx0-xxxx]
DOC  CR3F[4] = 0 Normal CLUT RAM I/O selected
DOC  CR3F[4] = 1 Video Window Gamma RAM I/O
DOC  selected
     CR40=[xxxx-xx0x]
DOC  CR40: AccuPak Dither Disable
     CR51=[xxx0-0xxx]
DOC  CR51: V-Port Downscaling and Operation
DOC  Enable
     CR80=[xxxx-0010]
DOC  CR80: Power Management Controls
     CR85=[xx0x-xxxx]
DOC  CR85: Bit[5] = 0 Ensure panel will power on
     CR8B=[xxxx-1000]
DOC  CR8B: Standby Timer Control
     CR8D=[1111-1000]
DOC  CR8D: Miscellaneous Flat Panel Control
     CR8F=[0000-0000]
DOC  CR8F: Bit[1] = 0 Enable Internal Current
     Source for DAC (IREF)
DOC  Notice GD7555 Rev. BC must set Bit[1] = 1.
     CRB1=[1xxx-xxxx]
DOC  CRB1: Bit[7] = 1 ****Do Not Change****
AC (IREF)
     ERXX=[---------]
DOC  ERXX: end of table


%%%  MCLK (CL-GD7555 only - see HW_CONFIG_TABLE
```

```
     CL-GD7556)
REM
REM  Additional configuration table
REM

DOC  NOTICE, for register information:
DOC  See "CL-GD7555/6 Advanced Hardware
DOC  Reference Manual"
DOC
DOC  BINARY DATA FORMAT.  Bit locations
DOC  marked as "x" can not be modified.

REM  MCLK table entry 1

DOC  MCLK when system in 3.3 volt operation

MCLK_CONFIG_TABLE:

     SR1F=[0010-0101]
DOC  SR1F: MCLK Frequency and VCLK Source
DOC  Select [3V]
     SR20=[x0xx-xxxx]
DOC  SR20: Bit[6] = Select 9 MCLK RAS#
DOC  Cycles [3V]
     SR0F=[xxxx-x0xx]
DOC  SR0F: Bit[2] = Display Memory RAS# Cycle
     Select [3V]
     ERXX=[---------]
DOC  ERXX: end of table

REM  MCLK table entry 2

DOC  MCLK when system in 5.0 volt operation

MCLK_CONFIG_TABLE:
     SR1F=[0010-1101]
     SR20=[x1xx-xxxx]
     SR0F=[xxxx-x0xx]
     ERXX=[---------]


%%%  TV
REM
REM  Additional configuration table
REM

DOC  TV registers that signal an external TV
DOC  encoder device.
DOC  See OTHER_BIN_BYTE_01 bit[6:5] for
DOC  enabling TV output support.
DOC  See USER_OPTION_6 bits[1:0] for TV as
DOC  boot device.
DOC  See USER_OPTION_2 bit[0] and
DOC  USER_OPTION_4 bit[1] for VGA device,
DOC  if TV was selected as boot device,
DOC  because default is CRT display.

DOC  NOTICE, for register information:
DOC  See "CL-GD7555/6 Advanced Hardware
DOC  Reference Manual"
DOC
DOC  BINARY DATA FORMAT.  Bit locations
DOC  marked as "x" can not be modified.

REM  TV table entry 1

DOC  NTSC 60hz

TV_CONFIG_TABLE:
     SR0D=[0110-0110]
DOC  SR0D: VCLK Numerator
     SR1D=[0011-1011]
DOC  SR1D: VCLK Denominator
     CR81=[x001-0000]
DOC  CR82: Text mode expansion and centering
DOC  attributes
     CR82=[x001-x000]
DOC  CR82: Graphics mode expansion and
DOC  centering attributes
        CR83=[0010-1000]
DOC  CR83: Bits[6:4] (default set TFT flat
DOC  panel type)
DOC  CR83: Bits[3:2] (default set 18-bit
DOC  data format)
DOC  CR83: Bits[1:0] (default set 640x480
DOC  display)
     CRA0=[0101-1111]
DOC  CRA0: CRT Horizontal Total 8-Dot for
DOC  640x480 modes
     CRA1=[0101-1110]
DOC  CRA1: CRT HSync Start 8-Dot for 640x480
DOC  modes
     CRA2=[0010-1101]
DOC  CRA2: CRT Horizontal Total 8-Dot for
DOC  640x400 modes
     CRA3=[0000-1100]
DOC  CRA3: CRT HSync Start 8-Dot for 640x400
DOC  modes
     CRAC=[1000-1111]
     CRAD=[1010-0000]
     CRAE=[1100-0110]
     CRAF=[0000-0100]
DOC  CRA0..CRAF: Registers for horizontal
DOC  controls
     CRB0=[0000-1010]
DOC  CRB0: CRT Vertical Total
     CRB1=[x001-0010]
DOC  CRB1: CRT Vertical Extension
     CRB2=[0000-0000]
DOC  CRB2: CRT Vertical Sync Start
     CRB3=[0000-0100]
DOC  CRB3: CRT Vertical Sync End
     CRBB=[1110-0000]
DOC  CRBB: Vertical size
     CRBC=[0000-0000]
DOC  CRBC: Vertical size Increment
     CRBE=[0000-1001]
DOC  CRBE: Vertical Overflow
     ERXX=[---------]
DOC  ERXX: end of table

REM  TV table entry 2

DOC  PAL 50hz

TV_CONFIG_TABLE:
```

```
     SR0D=[0110-0110]
     SR1D=[0011-1011]
     CR81=[x001-0000]
     CR82=[x001-x000]
     CR83=[0010-1000]
     CRA0=[0101-1111]
     CRA1=[0101-1110]
     CRA2=[0010-1101]
     CRA3=[0000-1100]
     CRAC=[1000-1111]
     CRAD=[1010-0000]
     CRAE=[1100-0110]
     CRAF=[0000-0100]
     CRB0=[0111-0011]
     CRB1=[x001-0010]
     CRB2=[0100-1000]
     CRB3=[0000-0100]
     CRBB=[1110-0000]
     CRBC=[0000-0000]
     CRBE=[0000-1001]
     ERXX=[---------]


%%%  XCLR
REM
REM  Additional configuration table
REM

DOC  NOTICE, for register information:
DOC  See "CL-GD7555/6 Advanced Hardware
DOC  Reference Manual"
DOC
DOC  BINARY DATA FORMAT.  Bit locations
DOC  marked as "x" are not modified.
DOC  Extended color mode fix up table for
DOC  registers other than CRTC.

REM  XCLR table entry 1

DOC  256 color fix up table

XCLR_CONFIG_TABLE:
     SR04=[0000-1110]
DOC  SR4: Memory Mode
     GR05=[0100-0000]
DOC  GR5: Mode
     GR06=[0000-0101]
DOC  GR6: Miscellaneous
     CR13=[0000-0001]
DOC  CR13: Offset (number to left-shift CR13)
     ERXX=[---------]
DOC  ERXX: end of table


REM  XCLR table entry 2

DOC  Direct color fix up table

XCLR_CONFIG_TABLE:
     SR04=[0000-1110]
     GR05=[0100-0000]
     GR06=[0000-0101]
     CR13=[0000-0010]
     ERXX=[---------]

REM  XCLR table entry 3

DOC  16M 640x480 color fix up table

XCLR_CONFIG_TABLE:
     SR04=[0000-1110]
     GR05=[0000-0000]
     GR06=[0000-0101]
     CR13=[0000-1000]
DOC  CR13: Offset (mode 78 & 79 CR13=80h)
     ERXX=[---------]


%%%  FREQ
REM
REM  Additional configuration table
REM

DOC  NOTICE, for register information:
DOC  See "CL-GD7555/6 Advanced Hardware
DOC  Reference Manual"
DOC
DOC  BINARY DATA FORMAT.  Bit locations
DOC  marked as "x" are not modified.
DOC  Extended mode frequency fix up table.
DOC  (Please also refer to
DOC  XMODE_FEATURE_BYTE_03 in XMODE TABLE)

REM  FREQ table entry 1

DOC  640x480 60Hz

FREQ_CONFIG_TABLE:
     ER00=[0101-1111]
DOC  CR0: Horizontal Total
     ER03=[1000-0010]
DOC  CR3: Horizontal Blanking End
     ER04=[0101-0100]
DOC  CR4: Horizontal Sync Start
     ER05=[1001-1111]
DOC  CR5: Horizontal Sync End
     ER06=[0000-1011]
DOC  CR6: Vertical Total
     ER07=[0011-1110]
DOC  CR7: Overflow
     ER09=[x10x-xxxx]
DOC  CR9: Character Cell Height
     ER10=[1110-1010]
DOC  CR10: Vertical Sync Start
     ER12=[1101-1111]
DOC  CR12: Vertical Display End
     ER15=[1110-0111]
DOC  CR15: Vertical Blank Start
     ER16=[0000-0100]
DOC  CR16: Vertical Blank End
     ER11=[1000-1100]
DOC  CR11: Vertical Sync End
     ER1C=[0000-0000]
DOC  CR1C: Sync Adjust and GENLOCK
     ERXX=[---------]
DOC  ERXX: end of table
```

```
REM   FREQ table entry 2

DOC   640x480 72Hz

FREQ_CONFIG_TABLE:
      ER00=[0110-0011]
      ER03=[1000-0110]
      ER04=[0101-0101]
      ER05=[1001-1101]
      ER06=[0000-0110]
      ER07=[0011-1110]
      ER09=[x10x-xxxx]
      ER10=[1110-1000]
      ER12=[1101-1111]
      ER15=[1110-0111]
      ER16=[1111-1111]
      ER11=[1000-1011]
      ER1C=[0000-0101]
      ERXX=[---------]

REM   FREQ table entry 3

DOC   640x480 75Hz

FREQ_CONFIG_TABLE:
      ER00=[0110-0100]
      ER03=[1000-0111]
      ER04=[0101-0100]
      ER05=[1001-1100]
      ER06=[1111-0010]
      ER07=[0001-1111]
      ER09=[x10x-xxxx]
      ER10=[1110-0001]
      ER12=[1101-1111]
      ER15=[1110-0111]
      ER16=[1110-1011]
      ER11=[0000-0100]
      ER1C=[0000-0101]
      ERXX=[---------]

REM   FREQ table entry 4

DOC   640x480 85Hz

FREQ_CONFIG_TABLE:
      ER00=[0110-0011]
      ER03=[1000-0110]
      ER04=[0101-0101]
      ER05=[1001-1011]
      ER06=[1111-1010]
      ER07=[0001-1111]
      ER09=[x10x-xxxx]
      ER10=[1110-0000]
      ER12=[1101-1111]
      ER15=[1110-0111]
      ER16=[1110-1011]
      ER11=[1000-0011]
      ER1C=[0000-0101]
      ERXX=[---------]

REM   FREQ table entry 5

DOC   800x600 60Hz

FREQ_CONFIG_TABLE:
      ER00=[0111-1111]
      ER03=[1000-0010]
      ER04=[0110-1011]
      ER05=[0001-1011]
      ER06=[0111-0010]
      ER07=[1111-0000]
      ER09=[x11x-xxxx]
      ER10=[0101-1000]
      ER12=[0101-0111]
      ER15=[0101-1000]
      ER16=[0111-0010]
      ER11=[1000-1100]
      ER1C=[0000-0100]
      ERXX=[---------]

REM   FREQ table entry 6

DOC   800x600 72Hz

FREQ_CONFIG_TABLE:
      ER00=[0111-1101]
      ER03=[1000-0000]
      ER04=[0110-1101]
      ER05=[0001-1100]
      ER06=[1001-1000]
      ER07=[1111-0000]
      ER09=[x11x-xxxx]
      ER10=[0111-1100]
      ER12=[0101-0111]
      ER15=[0101-1111]
      ER16=[1001-0001]
      ER11=[1000-0010]
      ER1C=[0000-0000]
      ERXX=[---------]

REM   FREQ table entry 7

DOC   800x600 75Hz

FREQ_CONFIG_TABLE:
      ER00=[0111-1111]
      ER03=[1000-0010]
      ER04=[0110-1000]
      ER05=[0001-0010]
      ER06=[0110-1111]
      ER07=[1111-0000]
      ER09=[x11x-xxxx]
      ER10=[0101-1000]
      ER12=[0101-0111]
      ER15=[0101-0111]
      ER16=[0110-1111]
      ER11=[1000-1011]
      ER1C=[0000-0000]
      ERXX=[---------]

REM   FREQ table entry 8

DOC   800x600 85Hz

FREQ_CONFIG_TABLE:
```

```
     ER00=[0111-1111]
     ER03=[1000-0010]
     ER04=[0110-1010]
     ER05=[0001-0010]
     ER06=[0111-0010]
     ER07=[1111-0000]
     ER09=[x11x-xxxx]
     ER10=[0101-1000]
     ER12=[0101-0111]
     ER15=[0101-1000]
     ER16=[0111-0010]
     ER11=[1000-1011]
     ER1C=[0000-0000]
     ERXX=[---------]

REM  FREQ table entry 9

DOC  1024x768 87Hz Interlaced

FREQ_CONFIG_TABLE:
     ER00=[1001-1001]
     ER03=[1001-1100]
     ER04=[1000-0100]
     ER05=[0001-1010]
     ER06=[1001-0110]
     ER07=[0001-1111]
     ER09=[x10x-xxxx]
     ER10=[1000-0000]
     ER12=[0111-1111]
     ER15=[1000-0000]
     ER16=[1001-0110]
     ER11=[1000-0100]
     ER1C=[0000-0000]
     ERXX=[---------]

REM  FREQ table entry 10

DOC  1024x768 60Hz

FREQ_CONFIG_TABLE:
     ER00=[1010-0011]
     ER03=[1000-0110]
     ER04=[1000-0101]
     ER05=[1001-0110]
     ER06=[0010-0100]
     ER07=[1111-1101]
     ER09=[x11x-xxxx]
     ER10=[0000-0010]
     ER12=[1111-1111]
     ER15=[0000-0000]
     ER16=[0010-0100]
     ER11=[1000-1000]
     ER1C=[0000-0000]
     ERXX=[---------]

REM  FREQ table entry 11

DOC  1024x768 70Hz

FREQ_CONFIG_TABLE:
     ER00=[1010-0001]
     ER03=[1000-0100]
     ER04=[1000-0101]
     ER05=[1001-0110]
     ER06=[0010-0100]
     ER07=[1111-1101]
     ER09=[x11x-xxxx]
     ER10=[0000-0010]
     ER12=[1111-1111]
     ER15=[0000-0000]
     ER16=[0010-0100]
     ER11=[1000-1000]
     ER1C=[0000-0000]
     ERXX=[---------]

REM  FREQ table entry 12

DOC  1024x768 72Hz

FREQ_CONFIG_TABLE:
     ER00=[1010-0001]
     ER03=[1000-0100]
     ER04=[1000-0101]
     ER05=[1001-0011]
     ER06=[0010-1010]
     ER07=[1111-1101]
     ER09=[x11x-xxxx]
     ER10=[0001-0010]
     ER12=[1111-1111]
     ER15=[0000-0000]
     ER16=[0010-1010]
     ER11=[1000-1001]
     ER1C=[0000-0000]
     ERXX=[---------]

REM  FREQ table entry 13

DOC  1024x768 75Hz

FREQ_CONFIG_TABLE:
     ER00=[1001-1111]
     ER03=[1000-0010]
     ER04=[1000-0100]
     ER05=[1001-0000]
     ER06=[0001-1110]
     ER07=[1111-0101]
     ER09=[x11x-xxxx]
     ER10=[0000-0000]
     ER12=[1111-1111]
     ER15=[1111-1111]
     ER16=[0001-1110]
     ER11=[1000-0011]
     ER1C=[0000-0000]
     ERXX=[---------]


REM  FREQ table entry 14

DOC  1024x768 85Hz

FREQ_CONFIG_TABLE:
     ER00=[1010-0111]
     ER03=[1000-0100]
     ER04=[1000-1000]
     ER05=[1001-0100]
     ER06=[0010-0101]
```

```
     ER07=[1111-1101]
     ER09=[x11x-xxxx]
     ER10=[0000-0000]
     ER12=[1111-1111]
     ER15=[0000-0000]
     ER16=[0010-0101]
     ER11=[1000-0011]
     ER1C=[0000-0000]
     ERXX=[---------]

REM  FREQ table entry 15

DOC  1280x1024 87Hz Interlaced (mode 69)

FREQ_CONFIG_TABLE:
     ER00=[1011-1101]
     ER03=[1000-0000]
     ER04=[1010-0101]
     ER05=[0001-1010]
     ER06=[0010-1010]
     ER07=[1011-0010]
     ER09=[x11x-xxxx]
     ER10=[0000-1011]
     ER12=[1111-1111]
     ER15=[0000-0000]
     ER16=[0010-1010]
     ER11=[1000-0000]
     ER1C=[0000-0000]
     ERXX=[---------]

REM  FREQ table entry 16

DOC  1280x1024 60Hz (mode 69)

FREQ_CONFIG_TABLE:
     ER00=[1100-0011]
     ER03=[1000-0110]
     ER04=[1010-0101]
     ER05=[0000-0001]
     ER06=[0001-1010]
     ER07=[1011-0010]
     ER09=[x11x-xxxx]
     ER10=[0001-0000]
     ER12=[1111-1111]
     ER15=[0000-0010]
     ER16=[0001-1010]
     ER11=[0000-1001]
     ER1C=[0000-0000]
     ERXX=[---------]

REM  FREQ table entry 17

DOC  1280x1024 60Hz (mode 6D)

FREQ_CONFIG_TABLE:
     ER00=[1100-1110]
     ER03=[1000-0000]
     ER04=[1010-1010]
     ER05=[0001-1001]
     ER06=[0001-0011]
     ER07=[1011-0010]
     ER09=[x11x-xxxx]
     ER10=[0000-0011]
     ER12=[1111-1111]
     ER15=[0000-0000]
     ER16=[0001-0010]
     ER11=[0000-0111]
     ER1C=[0000-0100]
     ERXX=[---------]

REM  FREQ table entry 18

DOC  1280x1024 70hz (mode 6D - reserved)

FREQ_CONFIG_TABLE:
     ER00=[1100-1000]
     ER03=[1000-0000]
     ER04=[1010-1010]
     ER05=[0001-1110]
     ER06=[0001-0100]
     ER07=[1011-0010]
     ER09=[x11x-xxxx]
     ER10=[0000-0001]
     ER12=[1111-1111]
     ER15=[0000-0000]
     ER16=[0001-0010]
     ER11=[1000-0000]
     ER1C=[0000-0111]
     ERXX=[---------]


%%%  E_ENV
REM
REM  Additional configuration table
REM

DOC  This environment table is used for the
DOC  BIOSs located other than C000h segment.
DOC  Please refer to CIRRUS's external BIOS
DOC  function specification.

E_ENV_OTHER_BYTE_01=[00]
E_ENV_OTHER_BYTE_02=[00]
E_ENV_OTHER_BYTE_03=[1f]
E_ENV_OTHER_BYTE_04=[00]
E_ENV_OTHER_BYTE_05=[42]
E_ENV_OTHER_BYTE_06=[1f]
E_ENV_OTHER_BYTE_07=[00]
E_ENV_OTHER_BYTE_08=[e0]
E_ENV_OTHER_BYTE_09=[bc]
E_ENV_OTHER_BYTE_10=[27]
E_ENV_OTHER_BYTE_11=[00]
E_ENV_OTHER_BYTE_12=[e0]
E_ENV_OTHER_BYTE_13=[46]
E_ENV_OTHER_BYTE_14=[00]
E_ENV_OTHER_BYTE_15=[56]
E_ENV_OTHER_BYTE_16=[52]

DOC  Above C_ENV_OTHER_BYTE_01 TO
DOC  C_ENV_OTHER_BYTE_16 are
DOC  CIRRUS's extension.

E_ENV_OTHER_BYTE_17=[0a]
E_ENV_OTHER_BYTE_18=[69]
E_ENV_OTHER_BYTE_19=[00]
```

```
E_ENV_OTHER_BYTE_20=[e0]
E_ENV_OTHER_BYTE_21=[00]
E_ENV_OTHER_BYTE_22=[00]
E_ENV_OTHER_BYTE_23=[00]
E_ENV_OTHER_BYTE_24=[00]
E_ENV_OTHER_BYTE_25=[00]
E_ENV_OTHER_BYTE_26=[00]
E_ENV_OTHER_BYTE_27=[00]
E_ENV_OTHER_BYTE_28=[00]
E_ENV_OTHER_BYTE_29=[00]
E_ENV_OTHER_BYTE_30=[00]
E_ENV_OTHER_BYTE_31=[00]
E_ENV_OTHER_BYTE_32=[00]
E_ENV_OTHER_BYTE_33=[bc]
E_ENV_OTHER_BYTE_34=[68]
E_ENV_OTHER_BYTE_35=[00]
E_ENV_OTHER_BYTE_36=[e0]
E_ENV_OTHER_BYTE_37=[00]
E_ENV_OTHER_BYTE_38=[00]
E_ENV_OTHER_BYTE_39=[00]
E_ENV_OTHER_BYTE_40=[00]
E_ENV_OTHER_BYTE_41=[00]
E_ENV_OTHER_BYTE_42=[00]
E_ENV_OTHER_BYTE_43=[00]
E_ENV_OTHER_BYTE_44=[00]

DOC   Above C_ENV_OTHER_BYTE_17 TO
DOC   C_ENV_OTHER_BYTE_44 are VGA standard.

E_ENV_CONFIG_TABLE:
      ERXX=[---------]


%%%   E_2ENV
REM
REM   Additional configuration table
REM

DOC   The secondary save area table.

E_2ENV_OTHER_BYTE_01=[1a]
E_2ENV_OTHER_BYTE_02=[00]
E_2ENV_OTHER_BYTE_03=[d6]
E_2ENV_OTHER_BYTE_04=[68]
E_2ENV_OTHER_BYTE_05=[00]
E_2ENV_OTHER_BYTE_06=[e0]
E_2ENV_OTHER_BYTE_07=[00]
E_2ENV_OTHER_BYTE_08=[00]
E_2ENV_OTHER_BYTE_09=[00]
E_2ENV_OTHER_BYTE_10=[00]
E_2ENV_OTHER_BYTE_11=[00]
E_2ENV_OTHER_BYTE_12=[00]
E_2ENV_OTHER_BYTE_13=[00]
E_2ENV_OTHER_BYTE_14=[00]
E_2ENV_OTHER_BYTE_15=[00]
E_2ENV_OTHER_BYTE_16=[00]
E_2ENV_OTHER_BYTE_17=[00]
E_2ENV_OTHER_BYTE_18=[00]
E_2ENV_OTHER_BYTE_19=[00]
E_2ENV_OTHER_BYTE_20=[00]
E_2ENV_OTHER_BYTE_21=[00]
E_2ENV_OTHER_BYTE_22=[00]
E_2ENV_OTHER_BYTE_23=[00]
E_2ENV_OTHER_BYTE_24=[00]
E_2ENV_OTHER_BYTE_25=[00]
E_2ENV_OTHER_BYTE_26=[00]
E_2ENV_CONFIG_TABLE:
      ERXX=[---------]


%%%   END_OEM_DATA
```

Notes

6 DRIVERS

The purpose of this chapter is to list the software drivers provided by Cirrus Logic for the CL-GD755X, and describe the function of each driver. Topics discussed include resolution, color depth, active display or displays, CRT-monitor refresh rate, auto-resolution, on-the-fly resolution, and Hot-Key support.

6.1 Operating Systems Driver Support

Drivers are available to support the following operating systems and software applications in the CL-GD755X Software production release. Some of the descriptions in this manual are incomplete at this time, but the software is thoroughly checked out before it is released. Any limitations are described with the “Release Notes” that accompany the software. The product descriptions in this manual will be updated in subsequent revisions.

6.2 Cirrus Logic Driver Availability

The CL-GD755X video and graphics LCD/CRT controller is VGA compatible. The display drivers described in this manual are supplied to improve the resolution, features, and performance for the video and graphics images produced with the Windows and DOS operating systems, and software application programs that run under them.

The detailed installation instructions for each display driver is included with the software release. Follow the instructions carefully to be sure that each display driver is correctly installed. For registered users, updates are available on the Cirrus Logic Bulletin-Board System (BBS). The BBS can be accessed at:

(USA) 510-440-9080 or 510-440-0394
Modem settings: 14,400 baud, NO Parity, 8 Data bits, 1 Stop bit
The software is available in the conferences on the specific subjects.

▲ WARNING Before beginning – Not all video modes are available on all systems. When an extended mode driver is installed for a video mode that is not available, the application program does not function properly. There are a number of things that determine the list of available video modes. Some of these include the current monitor type, the amount of installed memory, and the revision of the VGA controller. To determine which modes are available before beginning the driver installation, it is recommended that the user run the CLMode program and examine the list of available video modes.

6.3 Windows 95 Drivers for the CL-GD755X

The Cirrus Logic Windows 95 drivers are designed to work with the Windows 95 operating system to allow the user to configure the display properties controlled by the CL-GD755X. DirectDraw drivers, with DirectVideo and Direct-3D support, are supplied as part of the Windows 95 Driver files. Access to the configuration utilities is through the Windows 95 ‘Display’ icon in the ‘Control Panel’ window. By choosing the ‘Settings’ and ‘Monitor Refresh’ tabs in the ‘Display Properties’ window, the user can adjust the following functions:

- Installation of new or updated drivers
- Active display or displays
  - Flat panel only
  - CRT monitor only (when a monitor is connected)
  - Flat panel and CRT monitor (SimulSCAN)- (when a monitor is connected)
  - TV only – NTSC or PAL (when a TV is connected through external hardware)
  - External TV Encoder only
- Flat-panel and CRT-monitor resolution
  - 648 x 480
  - 800 x 600
  - 1024 x 768
- Color depth for the chosen resolution (check display memory requirements in Chapters 4.9 and 4.10)
  - 16, 256, 32K, 64K or 16M for 648 x 480 and 800 x 600 flat-panel and/or CRT monitor resolutions
  - 16, 256, 32K and 64K for 1024 x 768 flat-panel and/or CRT monitor resolution
  - 16 and 256 for 1280 x 1024 CRT monitor only resolution
- Monitor refresh rates for each resolution
  - The available refresh rates supported for each resolution appear in listing boxes
  - When new refresh rates are selected, the driver access the VGA BIOS program to identify the new mode, and to load all of the appropriate registers in the CL-GD755X
  - When SimulSCAN is selected, the refresh rates are controlled by the flat panel parameters

6.3.1 Windows 95 Driver Installation

The Windows 95 drivers are supplied on floppy disks or CD-ROM, or through the Cirrus Logic Bulletin Board (BBS). A “readme” file of specific instructions accompanies each version of the software.

6.3.2 Reconfiguring Windows 95

The display functions listed in Section 6.3 can all be modified by the user by using the Windows 95 utilities described in Chapter 9 of this manual.

6.3.2.1 Changing Resolutions On-the-fly

Using the Windows 95 ‘Settings’ tab in the ‘Display Properties’ window, the active resolution of the flat panel or CRT monitor can be changed with the slide bar in the ‘Desktop area’ without rebooting Windows. When SimulSCAN is being used, the resolution is controlled by the flat-panel parameters.

6.3.2.2 Changing Displays On-the-fly

Using the Windows 95 ‘Active Display’ area in the ‘Monitor Refresh’ tab of the ‘Display Properties’ window, the active display can be changed between the flat panel, CRT monitor, or both (SimulSCAN) without rebooting Windows.

Any time an external TV Encoder is either connected or disconnected in the ‘Active Display’ area in the ‘Monitor Refresh’ tab, the utility prompts the user to “restart Windows” before the change takes affect.

6.3.2.3 Changing Active Display with Hot Key

The Cirrus Logic VGA BIOS and Windows 95 Drivers provide the resources necessary for the OEM designer to create Hot-key access for changing the active display. The VGA-BIOS function calls operate within the tolerances defined for the SMI (System Maskable Interrupt). When Hot-key switching from a CRT-only dispaly mode, which is using a higher resolution than the flat panel can support, the resolution of the flat panel (and CRT in SimulSCAN) is changed to the highest resolution supported by the flat panel.

6.3.3 Windows 95 Driver Limitations

1) When using SimulSCAN (simultaneous flat panel and CRT) the refresh rate is the same for both displays, and it is controlled by the flat-panel parameters.

2) When using some low-resolution game modes (ie 640 x 400, etc) or VPM is on, hot-key display switching will be disabled.

6.3.4 Functional Description of Windows 95 Drivers for the CL-GD755X

Driver Filename	Function
7555.INF	The install script file loads all of the drivers into the folder name chosen by the user.
CIRLMM.DRV	Windows 95 driver
CIRL.VXD	Windows 95 VxD
755VPM.DRV	Video-Port-Manager driver
CL_REF.DLL	Windows 95 refresh-rate/display utility
CL_REF.HLP	Windows 95 refresh-rate/display utility help file
CLBIOS16.DLL	16-bit thunk dll to access interrupt calls
CLBIOS32.DLL	32-bit thunk dll to access interrupt calls
CLMMDD32.DLL	32-bit Direct-Draw driver
CLGAMMA.EXE	Color correction utility
CLGAMMA.HLP	Color correction help file
CLGAMMA.TXT	Color correction localization text file
ICCVID.DLL	Cinepak driver
MCIAVI.DRV	MCI driver for AVI
VPM16T.DLL	Video-Port-Manager 16bit thunk driver
VPM32AP.EXE	Video-Port-Manager dummy application
VPM32T.DLL	Video-Port-Manager 32-bit thunk driver
VVPMD.VxD	Video-Port-Manager VxD

6.4 Windows NT 3.51, 4.0 Drivers for the CL-GD755X

The Cirrus Logic Windows NT 3.51, 4.0 drivers are designed to work with the Windows NT 3.51, 4.0 operating system to allow the user to configure the display properties controlled by the CL-GD755X. Access to the configuration utilities is through the Windows NT 3.51, 4.0 ‘Display’ icon in the ‘Control Panel’ window. By choosing the appropriate color depth, resolution, and refresh rates window, the user can adjust the following functions:

- Flat-panel and CRT-monitor resolution
  - 648 x 480
  - 800 x 600
  - 1024 x 768
  - 1280 x 1024 (CRT only)
- Color depth for the chosen resolution (check display memory requirements in Chapter 4.9 and 4.10)
  - 16, 256, 32K, 64K or 16M for 648 x 480 and 800 x 600 flat-panel and/or CRT monitor resolutions
  - 16, 256, 32K and 64K for 1024 x 768 flat-panel and/or CRT monitor resolution
  - 16 and 256 for 1280 x 1024 CRT-only resolution
- Monitor refresh rates for each resolution
  - The available refresh rates supported for each resolution appear in listing boxes
- Installation of new or updated drivers is done using the ‘Change Monitor’ request box.
- Active display or displays
  - Flat panel only
  - CRT monitor only (when a CRT monitor is connected)
  - SimulSCAN (Flat panel and CRT monitor)- (when a CRT monitor is connected)
- NT 3.5.1 has no virtual display (panning/scrolling) support
- NT 4.0 has virtual display (panning/scrolling) support.

6.4.1 Windows NT 3.51, 4.0 Driver Installation

The Windows NT 3.51, 4.0 drivers are supplied on floppy disks or CD-ROM, or through the Cirrus Logic Bulletin Board. A “readme” file of specific installation instructions accompanies each version of the software.

6.4.2 Reconfiguring Windows NT 3.51, 4.0

The display functions listed in Section 6.4 can all be modified by choosing the ‘Display’ icon in the ‘Control Panel’ window. The color depth, resolution, and refresh rates are selectable using slide bars and drop-down listing boxes in the ‘Display’ window. The drivers use the Cirrus Logic VGA BIOS to define the resolution and color-depth capabilities. Only the display options supported by the present display memory and CL-GD755X speed settings are shown in the listing boxes.

A utility in the control panle called “WINNT Control Panel” allows user to switch display type.

6.4.2.1 Reconfiguration System-Restart Requirements

Any changes to the color depth, display resolution, or refresh rate require the system to be restarted. When the **OK** button is selected after a change is made, the system prompts the user to either “restart the system for the changes to take place” or “abandon the changes at this time”.

6.4.2.2 Changing Active Display with Hot Key

The Cirrus Logic VGA BIOS and Windows NT 3.51, 4.0 Drivers provide the resources necessary for the OEM designer to create Hot-key access for changing the active display. The VGA-BIOS function calls operate within the tolerances defined for the SMI (System Maskable Interrupt). Hot-key switching from a CRT-only dispaly to a flat-panel-only dispaly or SimulSCAN displays can only be done when the flat panel supports the same resolution presently being used by the CRT. When the CRT active resolution is too high for the flat panel, it must be lowered to a compatible value, before the dispaly is switched.

6.4.3 Windows NT 3.51, 4.0 Driver Limitations

1) When using SimulSCAN (simultaneous flat panel and CRT) the refresh rate is the same for both displays, and it is controlled by the flat-panel parameters.

6.4.4 Functional Description of Windows NT 3.51 Drivers for the CL-GD755X

Driver Filename	Function
CIRRUS1	Install disk tag file
CL7555.DLL	Cirrus Logic Video display driver
CL7555.INI	Cirrus Logic driver install file
CL7555.SYS	Cirrus Logic mini-port driver
NTCTRL.CPL	Display switch utility
NTCTRL.HLP	Display switch utilility help file
NTLANG.DLL	Localization monitor detection utility
OEMSETUP.INF	Windows setup reference file
TXTSETUP.OEM	Text-mode setup file
VGA.DLL	MS Windows NT 16-color dispaly driver
VGA256.DLL	MS Windows NT 256-color dispaly driver
VGA64K.DLL	MS Windows NT 64K-color dispaly driver

6.4.5 Functional Description of Windows NT 4.0 Drivers for the CL-GD755X

Driver Filename	Function
CL755X.INF	Installation Script
CL755X.DLL	Cirrus Logic Video diaplay driver
CL755X.SYS	Cirrus Logic mini-port driver
NTCTRL.CPL	Display switch utility
NTCTRL.HLP	Display switch utilility help file
NTLANG.DLL	Localization monitor detection utility

6.5 Windows 3.1x Drivers for the CL-GD755X

The Cirrus Logic Windows 3.1x drivers are designed to work with both the Windows 3.1 and Windows 3.11 for Workgroups operating systems to allow the user to configure the display properties controlled by the CL-GD755X. The following graphics functions can be adjusted by using the WinMode utility:

- Installation of new or updated drivers
- Active display or displays
  - Flat panel only
  - CRT monitor only (when a monitor is connected)
  - Flat panel and CRT monitor (SimulSCAN)- (when a monitor is connected)
  - TV only – NTSC or PAL (when a TV is connected through external hardware)
  - CRT and TV (when both a monitor and a TV are connected)
- Flat-panel and CRT-monitor resolution
  - 648 x 480
  - 800 x 600
  - 1024 x 768
  - 1280 x 1024 (CRT only)
  - Higher than normal resolutions are supported on flat panels by using panning and scrolling to view off-screen data
- Color depth for the chosen resolution (check display memory requirements in Sections 4.9 and 4.10)
  - 16, 256, 32K, 64K or 16M for 648 x 480 and 800 x 600 flat-panel and/or CRT monitor resolutions
  - 16, 256, 32K and 64K for 1024 x 768 flat-panel and/or CRT monitor resolution
  - 16 and 256 for 1280 x 1024 CRT monitor only resolution
- Monitor refresh rates for each resolution
  - The available refresh rates supported for each resolution appear in listing boxes
  - When new refresh rates are selected, the driver access the VGA BIOS program to identify the new mode, and to load all of the appropriate registers in the CL-GD755X
  - When SimulSCAN is selected, the refresh rates are controlled by the flat panel parameters
- Font Cache size can be adjusted from zero to 1000 KBytes
- Screen fonts
  - Small, for lower resolutions, and maximum data display on higher resolutions
  - Large, for bettor readability on higher resolutions

6.5.1 Windows 3.1x Driver Installation

The installation program is contained on the software-release disks supplied by Cirrus Logic. To run the installation program, Windows 3.1 is started with the display configured to use the default VGA drivers. The appropriate software disk is loaded as described in the installation notes that accompany the software release. In the first dialog box that is displayed, the path is chosen for the utility programs location. The Windows 3.1 driver installation utility copies all of the graphics driver and utility files to the hard disk. After the drivers have been copied to the hard disk, the install program executes the WinMode utility.

Software updates are available through the Cirrus Logic Bulletin Board (BBS). A “readme” file of specific instructions accompanies each version of the software.

6.5.2 Reconfiguring Windows 3.1x

When in Windows 3.1x, run WinMode to reconfigure the Windows 3.1x drivers. See the WinMode utility in Chapter 7 of this manual for complete instructions on using that program.

The Windows 3.1 drivers can be reconfigured in DOS by using the CLMode utility (see Chapter 8 of this manual), or proceed as follows:

1) Insure that Windows 3.1 and the extended resolution drivers are already installed on the computer.

2) From the Windows directory, at the DOS prompt, type `SETUP` [Enter] to run the Windows SETUP.EXE program. Follow the instructions on the screen. When the screen comes up which lists the hardware and software components such as display adapter (e.g. VGA, CGA, etc.), keyboard type, mouse type, etc., go to the **Display** selection by using cursor keys to move the highlighted bar and press `[Enter]`.

3) The list of drivers and their associated resolutions are displayed, such as:

```
CIRRUS LOGIC 7555 1.xx, 640x480x64K
CIRRUS LOGIC 7555 1.xx, 640x480x16M
CIRRUS LOGIC 7555 1.xx, 800x600x256
CIRRUS LOGIC 7555 1.xx, 800x600x64K
CIRRUS LOGIC 7555 1.xx, 800x600x16M
CIRRUS LOGIC 7555 1.xx, 1024x768x256
CIRRUS LOGIC 7555 1.xx, 1024x768x64K
```

4) Highlight the desired choice by moving the cursor to the correct display driver, and then press [Enter].

5) Setup prompts that the driver is already in the Windows directory, and gives the option to replace it. Use the existing driver.

6) Continue with the remainder of the setup procedure.

6.5.2.1 Changing Resolutions On-the-fly

Using WinMode, the active resolution of the flat panel or CRT monitor can be changed without rebooting Windows. When SimulSCAN is being used, the resolution is controlled by the flat-panel parameters.

6.5.2.2 Virtual Display Mode

It is possible to choose a Panning and Scrolling option in WinMode that supports a Virtual Display on a flat panel. In the Virtual Display mode, a display area in memory can be estabished that is larger than the screen resolution of the flat panel. The display memory data that is off-screen is viewed by panning and scrolling the viewing area of the panel (that is; by dragging the mouse off the edges of the screen).

6.5.2.3 Changing Displays On-the-fly

Using WinMode, the active display can be changed between the flat panel, CRT monitor, or both (SimulSCAN) without rebooting Windows. Unless the Panning and Scrolling option is selected, when changing from a CRT-only dispaly mode, which is using a higher resolution than the flat panel can support, the resolution of the flat panel (and CRT in SimulSCAN) is changed to the highest resolution supported by the flat panel.

Any time an external TV Encoder is either connected or disconnected using WinMode, the utility prompts the user to “restart Windows” before the change takes affect.

6.5.2.4 Changing Active Display with Hot Key

The Cirrus Logic VGA BIOS and Windows 3.1x Drivers provide the resources necessary for the OEM designer to create Hot-key access for changing the active display. The VGA-BIOS function calls operate within the tolerances defined for the SMI (System Maskable Interrupt). When Hot-key switching from a CRT-only dispaly mode in which the CRT is using a higher resolution than the flat panel can support, the resolution of the flat panel (and CRT in SimulSCAN) is changed to the highest resolution supported by the flat panel.

6.5.3 Windows 3.1x Driver Limitations

1) When using SimulSCAN (simultaneous flat panel and CRT) the resolution and refresh rate is the same for both displays, and it is controlled by the flat-panel parameters. It is possible to select a CRT monitor resolution that is higher than the flat panel (see Section 6.5.2.2, Virtual Display Mode), but both displays use the flat panel timing parameters.

2) When using WinMode to select a new CRT refresh rate when in SimulSCAN mode, the new refresh rate does not take effect until CRT-only is selected.

6.5.4 Functional Description of Windows 3.1 Drivers for the CL-GD755X

Driver File Number	Function
7555VPM.DLL	Video-Port Manager driver
AVGA.3GR	Enhanced mode grabber
CIR7555.DRV	Multi-resolution 256-, 64K- and 16M-color linear driver
CIREXT.DLL	80486 enhanced DLL driver
CTL3D.DLL	Microsoft 3D controls
DCI7555.DRV	DCI Provider
DCIMAN.DLL	DCI Manager
ICCVID.DRV	Cinepak driver
INSTALL.INF	Windows 3.1 driver installation script
INSTALL.EXE	Windows 3.1 driver installation executable
INSTLANG.DLL	Windows 3.1 install localization
OEM7555.INF	Windows setup configuration file
PMSAVER.HLP	Power Management Screen Saver Help file
PMSAVER.SCR	Power Management Screen Saver
TIMINGS.DAT	List of CRT Monitors, their resolutions, and thier timing parameters for WinMode
UDH.DLL	Universal Draw Handler
VDD7555.386	Windows enhanced 256/64K VDD without APM
VGACOLOR.2GC	Standard VGA mode grabber
WINMODE.EXE	WinMode Resolution-configuration utility
WINMODE.HLP	WinMode Help file
WINMODE.INI	WinMode Configuration file
WINMODEL.DLL	WinMode Multi-languages modules
WM7542.DLL	GD7555 WinMode DLL Options file
WM7542.HLP	GD7555 WinMode Help file
WM7542L.DLL	GD7555 Multi-languages modules
*.FON	Miscellaneous Font files

6.6 DCI Provider Specifications

The Display Control Interface (DCI) is a software interface for Windows 3.1 defined by Microsoft and Intel that allows applications and drivers to bypass the Graphics Device Interface (GDI) and transfer data directly into a linear frame buffer. DCI supports acceleration of Video for Windows (VfW) Audio-Video-Interleave (.AVI) files, and MPEG-1 encoded files which are created with DCI-compatible MPEG-1 software encoder/decoders. Acceleration of Quicktime for VfW (.MOV) files are not supported by DCI.

The DCI architecture requires a DCI Client and a DCI Provider. A Client is software that renders data using the DCI capabilities. A Provider is a VGA specific software driver that notifies the Client of the hardware's capabilities and provides those capabilities in a safe manner. Currently, VfW is the main DCI Client, but other clients are being developed such as WinG, MPEG software decoders, and others.

The current DCI interface provides the following two levels of support:

1) **Primary Surface Support:** The ability for the DCI Client to write data into the visible frame buffer. This support does not include zooming, double buffering, or color space conversion

2) **Off-screen Surface Support:** The ability for the DCI Client to write data in non-visible frame buffer memory and later use a BitBLT to transfer it to visible memory. For special hardware (e.g. CL-GD755X), this data can be made immediately visible. This level of support also includes zooming and color-space conversion.

NOTE: The current 1.x DCI specification is intended for use with Windows 3.1. Windows 95 is support by the DirectDraw and DirectVideo drivers.

The Cirrus Logic CL-GD755X LCD/CRT controller chip uses the DCI software to take advantage of integrated MotionVideo™ Acceleration (MVA™) hardware via the DCI Provider driver developed by Cirrus Logic for the CL-GD755X. MVA allows a video window to be displayed on top of a graphics background. The video window can be an entirely different color space format or color depth than the background graphics, and provides hardware zooming capabilities.

6.6.1 Installation

Installation of VfW version 1.1d or later is required, to insure DCI/video-window operation with .AVI files. Contact Microsoft to obtain the correct revision of VfW.

NOTE: VfW can be obtained from a number of sources, including a number of multimedia applications that incorporate video; however, not all provide revision 1.1d which is the first revision to support DCI.

Perform the following steps to install the DCI Provider:

1) Start Windows 3.1x.

2) Insert the CL-GD755X Windows Driver Disk into the floppy disk drive.

3) From the **Program Manager File** menu, choose **Run...**

4) In the **Command Line** box, type the letter designation of the drive holding the Driver Disk followed by `:\INSTALL` (example = `A:\INSTALL`).

5) Follow the directions on the screen.

To verify that all of the components necessary to use the DCI Provider are installed:

1) Insure that the files DCIMAN.DLL, UDH.DLL, and DCI7555.DRV are present in the \WINDOWS\SYSTEM directory.

2) Use a text editor or viewer to insure the following lines are in the \WINDOWS\SYSTEM.INI file:

```
[drivers]
DCI=DCI7555.DRV
VIDS.DRAW=UDH.DLL
```

6.6.2 Video Window (VW) specifications

The DCI Provider notifies the DCI Client of the available color space formats and color depths, and the Client chooses the best one for its use.

The DCI Provider for the CL-GD755X enumerates 16-bit RGB (5:6:5) and 16-bit YUV (4:2:2) color spaces, in that order. MPEG-1 files predominantly support YUV color space format, while .AVI files commonly use RGB formats.

6.6.3 MotionVideo Acceleration (MVA™)Support

Many of the considerations listed below involve situations in which the video window is manipulated by moving the window or manually resizing the video clip. These capabilities are easily accomplished using Media Player, the Windows utility that allows an end user to experiment with video and audio clips. However, Media Player may not be the typical application or the typical interface that end users encounter when running applications that incorporate video.

In fact, many off-the-shelf multimedia applications that incorporate video prohibit the user from resizing or moving the window position at all. Applications often fix the video window position to allow the display of additional text or other graphics around the video window. Similarly, the application itself controls zoom capabilities; sometimes, pre-determining a zoom factor based on the graphics resolution, insuring that any surrounding objects is always displayed. In this way, the application guarantees that the end user can not disrupt the organization of the display, and insures the proper operation of the application.

In summary, “real-world” applications may not take advantage of MVA capabilities in the same way that Media Player types of utilities do. The end user’s capabilities with regard to video window adjustments must be completely defined by the applications provider.

6.6.3.1 DCI Provider Determines Video Window Availability

Hardware and graphics-mode configuration differences affect whether MVA is or is not used in any given instance. The DCI Provider determines the availability of the MVA based on changes in configuration. Listed below are the considerations which determine MVA availability:

Available Frame Buffer Memory

The DCI provider requires available off-screen memory in order to utilize MVA. There must be sufficient frame buffer memory remaining after subtracting the amounts required for:

— displaying the graphic display at the resolution and refresh rate configured for Windows,
— special hardware requirements (i.e. half-frame buffer for driving a DSTN LCD),
— the memory reserved for hardware cursor and hardware icons,
— the memory allocated for miscellaneous caching and buffering to improve performance.

When sufficient memory is left for the Off-screen Memory Manager to provide for MVA, then the DCI provider drops back to Primary Surface support for non-occluded video. When the video is also occluded, then video data is provided via the GDI (Graphics Device Interface).

Screen Depth

The video window is only available when the graphics color depth is 8 bits (Windows is configured to run in 256 color mode in Windows Setup or through WinMode.)

Alignment

The left edge of the video clip must be aligned on a 32 bit (DWORD) boundary in the graphics display memory. For eight bit graphics depth, this means the left edge must be a multiple of four pixels from the left edge of the screen.

— Video for Windows Universal Draw Handler (UDH) attempts to align a window containing video on a four pixel boundary. But, when the window cannot be moved (for example, when the window is full-screen) and the left edge of the video is not already aligned to a four pixel boundary, the DCI Provider must disable the video window and revert to standard DCI.

— The **FULL SCREEN** mode option in the Media Player applet does not make use of DCI in general, so it does not use any special hardware-acceleration features that a DCI Provider makes available. In summary Media Player always uses GDI to display full screen video when that option is selected, regardless of the capabilities of the graphics controller.

Video Width

The video source clip width must be a multiple of 64 bits (two DWORDs) in width. For eight bit graphics depth, this means the width must be a multiple of eight.

Minimum Video Width

The video clip width must be at least 32 pixels wide.

Source Relationship to Destination

The video window has four possible size requirements - it must be the same width and height as the source video, double the width of the source, double the height of the source, or double the width and height of the source.

— The UDH attempts to insure that the height of a video window is a multiple of four (which is not necessary for our video window). When the source video does not already have a height which is a multiple of four, the displayed height and source video height does not match. For this reason, allowance has been made for a slight variation in displayed and source height.

— A 320x240 source size video clip does not zoom to 640x480 on a 640x480 screen due to the title bar of the window. For this reason, allowance has been made for a variation in displayed and source dimensions for a near-double zoom.

Stretching

The video window cannot be used, when hardware graphics expansion is used to fill a display (i.e., when Windows running at 640x480 resolution is expanded to fill an 800x600 LCD.) This should be an uncommon occurrence with Windows, because a user typically configures Windows for 800x600 resolution when running on an 800x600 LCD.

Centering

The video window cannot be used, when the graphics are centered on a larger display (i.e., when a 640x480 screen is centered on an 800x600 LCD). This should be an uncommon occurrence with Windows, because a user typically configures Windows for 800x600 resolution when running on an 800x600 LCD.

Panning and Scrolling

WinMode for Windows 3.1 provides an operating option called Pan and Scroll. This allows a user to run SimulSCAN mode where the CRT resolution is selected as 1024x768 or 800x600, and a lower resolution LCD is accessed by panning or scrolling around within that larger resolution display area loaded in the display memory (i.e. an 800x600 LCD panning on a 1024x768 display area, or a 640x480 LCD panning on an 800x600 display area). Unfortunately, the video window cannot be used when the graphics are larger than the physical display.

— As an alternative to Pan & Scroll, WinMode also offers the option of Change Resolution on the Fly. With this option, Windows resolution automatically changes without restarting, whenever a display type is physically unable to run the Windows resolution that had been configured (i.e., Windows running 1024x768 resolution on a CRT is automatically switched to LCD resolutions of 640x480 or 800x600 when LCD-only or SimulSCAN modes are used.) In this case, the DCI Provider properly senses the video window support capabilities of the new and old configurations.

Occlusion

The video window is a rectangular overlay. Placing any window, menu, icon, etc. over any portion of the video window covers a portion of the video, referred to as occlusion. When this occurs, the DCI Provider reverts to standard DCI playback to accommodate the complex shape.

— Special consideration has been made when occlusion of the right or bottom of a video window results in a rectangular display region. In these cases, MVA can still be used depending on the resulting width of the video (see Video Width above).

Clipping

Moving the video window off any screen edge causes the DCI Provider to revert to standard DCI playback. Clipping can also occur, when another window completely covers the left edge or top of the video window, forcing the DCI Provider to revert to standard DCI. However, clipping of the right or bottom edge of the video window may not force the DCI Provider to revert to standard DCI display parameters, depending on the width of the video window (see Video Width above).

Bandwidth

Certain display configurations prohibit use of MVA due to insufficient clock speeds or memory bandwidth. A bandwidth equation is used to determine if the MCLK speeds and the display memory bandwidth are sufficient to support MVA. If the result is “no”, the default DCI Provider is used to control the video window.

DCI Drivers

The DCI software drivers are included with the Windows 3.1x Software release package.

6.7 OS/2 3.0 Drivers for the CL-GD755X

The Cirrus Logic OS/2 3.0 drivers are designed to work with the OS/2 3.0 operating system to allow the user to configure the display properties controlled by the CL-GD755X. The following graphics functions can be adjusted by the user:

- Installation of new or updated drivers
- Active display or displays
  - Flat panel only
  - CRT monitor only (when a monitor is connected)
  - Flat panel and CRT monitor (SimulSCAN)- (when a monitor is connected)
- Flat-panel and CRT-monitor resolution
  - 648 x 480
  - 800 x 600
  - 1024 x 768
  - 1280 x 1024 (CRT only)
- Color depth for the chosen resolution (check display memory requirements in Sections 4.9 and 4.10)
  - 16, 256, 32K, 64K or 16M for 648 x 480 and 800 x 600 flat-panel and/or CRT monitor resolutions
  - 16, 256, 32K and 64K for 1024 x 768 flat-panel and/or CRT monitor resolution
  - 16 and 256 for 1280 x 1024 CRT monitor only resolution
- Monitor refresh rates for each resolution
  - The available refresh rates supported for each resolution appear in listing boxes
  - When new refresh rates are selected, the driver access the VGA BIOS program to identify the new mode, and to load all of the appropriate registers in the CL-GD755X
  - When SimulSCAN is selected, the refresh rates are controlled by the flat panel parameters

6.7.1 OS/2 3.0 Driver Installation

To run the installation program, OS/2 3.0 is started with the display configured to use the default VGA drivers. The appropriate software disk is loaded as described in the installation notes that accompany the software release. In the first dialog box that is displayed, the path is chosen for the utility programs location. The OS/2 3.0 driver installation utility copies all of the graphics driver and utility files to the hard disk.

6.7.2 Reconfiguring OS/2

After the drivers have been copied to the hard disk, the graphics subsystem can be reconfigured using the OS/2 display utilities or the DOS-based CLMode utility. See the CLMode utility in Chapter 8 of this manual for complete instructions on using that program.

6.7.2.1 Changing Active Display with Hot Key

The Cirrus Logic VGA BIOS and OS/2 Drivers provide the resources necessary for the OEM designer to create Hot-key access for changing the active display. The VGA-BIOS function calls operate within the tolerances defined for the SMI (System Maskable Interrupt). Hot-key switching from a CRT-only dispaly to a flat-panel-only dispaly or SimulSCAN displays can only be done when the flat panel supports the same resolution presently being used by the CRT. When the CRT active resolution is too high for the flat panel, it must be lowered to a compatible value, before the dispaly is switched.

6.7.3 OS/2 Driver Limitations

1) When using SimulSCAN (simultaneous flat panel and CRT) the refresh rate is the same for both displays, and it is controlled by the flat-panel parameters.

2) Any time the resolution is changed, the system must be restarted before the change takes effect.

6.8 Video Port Manager (VPM™)

The Video Port Manager, created by Cirrus Logic, is an API which allows an application (client) to easily manage any video port and its associated display hardware without knowledge of vendor-specific details. The Video Port Manager (VPM) interface is designed to be a non-hardware-specific interface for controlling a video data port tied directly to a graphics frame buffer or display. It provides information about the capabilities and limitations of the hardware in a standard manner to enable a VPM client to utilize the hardware to its maximum extent without directly communicating with the hardware.

The VPM allows the client, either an application or a driver, to abstract live-video features in much the same way that DirectDraw abstracts the BitBLT engine on a VGA device. Since DirectDraw and DCI contain some of the primitives required for live-video support, VPM calls down to these drivers, whenever it needs to perform a task common to all of these drivers (such as frame buffer allocation). This ensures that these tasks are all handled in a standard way, and that no conflicts between drivers will occur. The details of this interaction between DCI and DirectDraw are an implementation detail to be handled by the individual VPM provider. Another benefit of VPM is that the same interface works under both Windows 3.1 and Windows 95 because it abstracts the operating-system specific details.

6.8.1 Video Port Manager Specification

The complete detailed Video Port Manager specification is available from Cirrus Logic. Contact the local Cirrus Logic sales office for details.

6.8.2 Video Port Manager Interface

The Video Port Manager interface shown in Figure 6-1 cooperates with DCI and DirectDraw (under Windows 95). VPM cooperates with these other interfaces where common display hardware must be controlled, while adding the ability to control the video port and its associated features.

NOTE: The video port is assumed to be a one-way, input-only port. The data may be intended for immediate display or acquisition into the frame buffer.

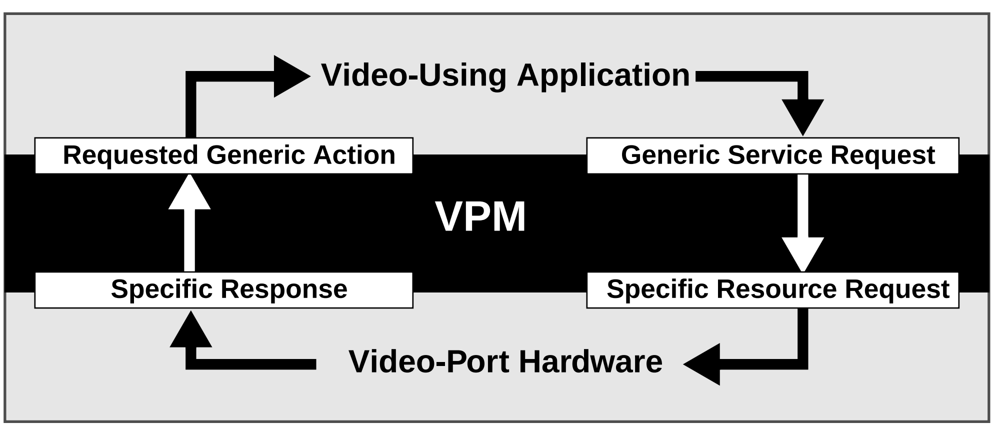


Figure 6-1. VPM Client - Provider Interface

6.8.3 VPM Client Flexibility

Restrictions on VPM clients are minimized. Flexibility was the goal in the design of the specification, so that it serves a wide range of market applications. Wherever required, a bit more complication or additional responsibility of the client was introduced to insure that a particular capability is not artificially limited.

6.8.4 VPM Provider

The VPM provider is the software driver that manipulates the hardware and interfaces with the VPM client. Under Windows 3.1 it coexists with the Display Control Interface (DCI), and under Windows 95 it coexists with the DirectDraw interface. However, VPM and DCI or DirectDraw are completely independent interfaces. Therefore it is not necessary to use or have knowledge of DCI or DirectDraw in order to use VPM.

The VPM provider is not responsible for special requirements of the client. For example, few clients utilize interrupt or Direct Memory Access (DMA) capabilities. Since the implementation of the clients could vary greatly depending on the application, any implementation by VPM would more-than-likely interfere with the goals of the client. Therefore, in the interest of flexibility, the responsibility of supporting such features is left to the client. (Note that an interrupt generated by the video supplier would be preferable to one generated by the VPM device, since the presence of the interrupt on the VPM device cannot be guaranteed, but an interrupt on the video supplier can be relied upon. This also represents why VPM chooses not to address interrupt handling.)

6.8.5 Simplified Client Software Development

VPM simplifies client software development. The self-consistent design is intended to ease implementation, shortening development time for new video/display hardware and reducing time to market. Ideally, correctly-written VPM clients work without executable change on any VPM device. For example, an application or driver written to use VPM work on any VPM device.

6.8.6 Product Differentiation

VPM encourages product differentiation by freeing the designer from the requirement that the new product be hardware-compatible with a previous, possibly-competing, product.

6.8.7 VPM Provider Limitations

- The VPM Provider only runs in native resolutions for flat panels and CRT monitors.

6.9 AutoCAD, Autoshade and 3D-Studio Drivers

Cirrus Logic offers Windows 3.1 and Windows 95 compatible driver support for AutoCAD as part of the standard Windows 3.1 and Windows 95 driver packages. Special DOS-compatible drivers for AutoCAD Versions 10 through 13 are available on special order from Cirrus Logic Portable-Graphics Marketing. These DOS-compatible drivers have been developed by Panacea, Inc. for Cirrus Logic. These drivers allow the referenced application programs to take full advantage of the CL-GD755X graphics capabilities.

Cirrus Logic provides driver support for Autoshade and 3D Studio, which are supplied by AutoDesk, Inc. The versions of these products, which are compatible with the referenced versions of AutoCad, are supported as part of the CL-GD755X AutoCAD and Windows driver packages.

The ADI 4.2 driver provided with your {PRODUCT NAME} VGA is the TurboDLDClassic display list driver from Panacea. It has only two purposes:

- To speed up AutoCAD REDRAWs, PANs, and ZOOMs
- *To provide a more productive, user-friendly, interface to AutoCAD (via the features described later in this manual).

The driver is memory-resident and inserts itself between AutoCAD and the graphics board. It has no other effect on AutoCAD's operation besides speeding up the program; it runs with AutoShade 2 with RenderMan and 3D Studio to provide enhanced rendering support, but does not affect the speed of these programs, since they do not support Display Lists.

Installing TurboDLDClassic does not change any of the AutoCAD program files or alter any of the drawing files stored on disk. TurboDLDClassic was designed to be an easy-to-use, feature-filled product that makes using AutoCAD faster and more productive, not one that makes AutoCAD more complicated and difficult to use.

How does TurboDLDClassic make AutoCAD run faster? There are three things the driver does to speed operation:

- AutoCAD stores drawings in a hierarchical structure, with simple elements intermixed with complex ones. Every time the screen is updated, AutoCAD must decode this structure. TurboDLDClassic works differently. While you are working, it translates the normal hierarchical structure into a Display List, a series of vectors or polygon fills. When you pan or zoom, TurboDLDClassic uses the Display List, then writes the resulting vectors to the video board hardware. Since the hierarchical structure does not have to be decoded, drawing proceeds very quickly.
- TurboDLDClassic also maintains a Drawing Cache. The Drawing Cache is a compressed list which contains the current contents of a viewport. This pre-scaled portion of the Display List allows for even faster pans and zooms and redraws.
- TurboDLDClassic gives you numerous new user-interface features, such as the bird's eye view. (via the features described later in this manual).

By how much does TurboDLDClassic increase the speed of AutoCAD? PANs and ZOOMs, aided by only the Display List, run from two to twelve times faster than a non-display list driver. The Drawing Cache further speeds things up to the point that REDRAWs can be up to twenty-five times faster with TurboDLDClassic, compared to the graphics drivers shipped with AutoCAD.Features

The features provided by TurboDLDClassic include:

- *Bird's eye view.
- *Accelerated redraws, pans, and zooms.
- *Easy to use - no new commands or special menus to learn.

- *Protected-mode ADI 4.2 driver - completely compatible with AutoCAD Release 12, Release 11/386, 3D Studio, Release 13 and AutoShade 2.0 with RenderMan.
- *No memory conflicts. Works with AutoCAD's built-in Virtual Memory Manager
- *Includes CustomColors™, which lets you interactively customize your logical and physical colors from within AutoCAD.
- *Completely compatible with all Autodesk ADI 4.2 compatible third party software.
- *Supports all AutoCAD Release 12 & 13 features, including rendering to viewports and 31-Bit regen space.

TurboDLDClassic requires a '386, '486 or Pentium based PC which supports AutoCAD Release 13, 12, 11/386, AutoShade 2.0 with RenderMan, or 3D Studio v1.x/2.x. Additional extended memory is recommended for optimal performance.

6.9.1 AutoCAD DOS Driver Installation

When installing the DOS compatible display drivers for AutoCAD, Autoshade and 3D Studio, the applications program must be installed prior to installing the device driver. The installation program asks several questions about the system. An ‘Install’ selections menu is displayed. Individual selections are made by using the space bar and ‘enter’ key, or by using “Y” for Yes or “N” for No.

Insert the driver diskette into Drive A: (or B:). Run the installation program (INSTALL.EXE), and choose the option for the Autodesk/AutoCAD drivers and specify the drive and directory where you want them copied to (such as C:\ACAD\DRV).

Since the start-up is a little different for AutoCAD Release 12 & 13 than it is for older versions, please follow the instructions for the version of AutoCAD you will be using.

6.9.2 To Configure AutoCAD 12, 13

Begin AutoCAD with the reconfigure switch by typing

```
ACAD -R[Enter]
```

Choose option 3, ‘Configure Video Display’, from the AutoCAD configuration menu.

Type ‘Y’ at the “Do you want to select...” message to display the available video options for AutoCAD.

Select ‘TurboDLDClassic by Panacea Inc.’ from the list of display options.

NOTE: If you chose to install TurboDLDClassic into a sub-directory other than ACAD\DRV, be sure to modify the ACADDRV environment variable to include that sub-directory. Otherwise, the TurboDLDClassic selection will not appear in the list of available drivers.

7 WINMODE UTILITY

(Version 3.10)

WinMode is a Windows based utility that configures the graphics system for Windows 3.1. It allows the user to configure the system for a CRT monitor only, a flat-panel only, or simultaneous CRT monitor and flat panel. It controls the monitor refresh rates, resolution, number of available colors, large or normal font sizes and font-cache size. It also allows the user to run the flat panel at higher than normal resolutions by panning and scrolling the display to view off-screen data.

When only the display type and resolution are changed in ‘Change Resolution’ mode, the changes take effect immediately without restarting Windows. When any other changes are made, Windows must be restarted before they take effect. The AUTOEXEC.BAT file must be modified to make the changes permanent; this can be done automatically or manually.

7.1 Installation Instructions

WinMode must be installed on the hard disk to get reasonable performance. The WINMODE.INI file is installed in the WINDOWS directory. All other WinMode files, such as the WINMODE.EXE, the WINMODE.HLP help file, the WINMODEL.DLL language utility file, and the TIMINGS.DAT file must be installed in a user defined directory. The device Options files WM7542.DLL. WM7542.HLP, and WM7542L.DLL must be installed in the directory with the WINMODE.EXE file.

WinMode assumes that the Windows drivers have been correctly installed and configured using installation utility provided on the Windows 3.1 Drivers and Utilities Diskette.

For specific installation instructions see the documentation shipped with the software.

7.1.1 Hardware Requirements

The WinMode utility works on any system properly configured to run Microsoft® Windows© 3.1 or 3.1x. It expects to find a Cirrus Logic LCD/CRT VGA Controller.

7.1.2 Software Requirements

WinMode requires Windows 3.1 or 3.1x. A mouse or an equivalent pointing device is recommended.

7.1.3 WinMode-BIOS Compatibility

WinMode v3.1 is designed to work with all previous Cirrus Logic VGA BIOS releases for all products that it supports.

7.1.4 Use and Redistribution of WinMode

A License for Distribution of Executable Software must be acquired from Cirrus Logic to distribute the display drivers and utility software to end users. Such license is required for each Cirrus Logic VGA product type with which the software is shipped.

The software may be used freely by the original owner of the Drivers and Utilities package shipped by Cirrus Logic, but no duplication, distribution or sub-licensing is authorized without a formally executed license.

7.2 Multiple Language Support

Support is available for the following languages - US English, UK English, Japanese, French, Italian, German, Spanish, Latin American Spanish, Dutch, Swedish, Portuguese, Brazilian Portuguese, Danish, Norwegian, Finnish, Korean, Traditional Chinese (Taiwan), and Simplified Chinese.

The language displayed in the WinMode menus and the Help menus can be changed at any time by selecting the Language Selector line at the bottom of the Control menu [–] in the upper left-hand corner of the main menu. A dialog box with all of the available language options is presented. By selecting a new language and clicking the **OK** button, all appropriate menu and help text is changed.

The Japanese, Korean, Traditional Chinese (Taiwan), and Simplified Chinese language options are only displayed in the selection menu when running the applicable version of Windows 3.1x.

7.3 Using WinMode

WinMode is run by selecting it's icon. The icon is in the group specified during the install process. When WinMode is started, the screen may blank for up to two seconds while WinMode attempts to discover the capabilities of the CRT monitor. When the monitor can be identified, only one choice is selected from the other brand and model choices in the Monitor Brand and Monitor Model drop down combo boxes. See Figure 7-1 for WinMode main menu.

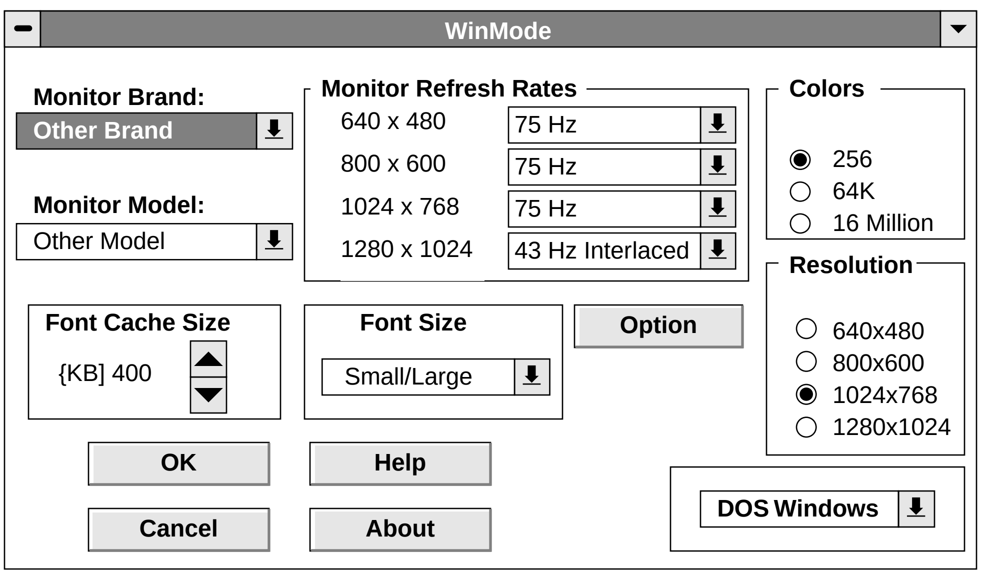


Figure 7-1. WinMode Main Menu

7.3.1 Monitor Selection

The main menu has fields to show and select the attached monitor by Brand and Model. During WinMode initialization, a call is made to inquire for DDC support. When the monitor and the graphics card both support VESA DDC (VESA Display Data Channel), the only Brand and Model choices that are presented are either a DDC compliant monitor or “Other Brand” and “Other Model”. When a DDC compliant monitor is detected, WinMode displays all of the available refresh rates for that monitor. When some other monitor is selected, WinMode updates all of the available refresh-rate lists to comply with the new selection.

When a non-DDC compliant monitor, or a flat-panel only configuration is detected, the Monitor Brand and Monitor Model fields show “Other Brand” and “Other Model” respectively. These fields can be changed by the user to reflect one of the listed brands or models, or the “Other Brand/Model” fields can be retained, and the individual Refresh Rates can be set manually.

7.3.1.1 To Turn On/Off DDC Detection Support

To turn the automatic DDC detection support ‘on’ or ‘off’, the following changes need to be made to the WINMODE.INI file. Locate the line:

`MakeDdcCall=on` or `MakeDdcCall=off`

Change the last word in the line to “on” for automatic detection, and “off” to disable automatic detection.

7.3.2 Monitor Refresh Rates

The drop down list boxes let the monitor refresh rates for each resolution be selected. When “Unavailable” is selected for any screen resolution, any higher resolutions are also “unavailable”. This also turns off the corresponding choices in the ‘Resolution’ box.

The only choices available are the ones for the monitor that was selected in the ‘Monitor Brand’ and ‘Monitor Model’ fields. When “Other Brand” or “Other Model” are selected, then all of the refresh rates available on the graphics adapter are listed. Consult the manual that came with the monitor to determine the best choices.

In general, the higher the refresh rate, the better the display quality and the lower the performance. This is because the graphics system can only do a fixed number of operations per second. The more time it spends redrawing the screen, the less time it has available to perform other operations.

7.3.3 Resolution

These buttons set the resolution that WinMode uses to run Windows 3.1, after Windows is restarted. Some of these buttons may be unavailable, because there is not enough display memory in the system, or because some of the monitor refresh rates have been set to “Unavailable”.

When a flat panel and CRT monitor are both connected, the Options selection for “Change Resolution” and “Panning/Scrolling” can effect the available resolutions. When “Change Resolution” is selected, only the resolutions supported directly by the flat panel are displayed. When “Panning/Scrolling” is selected, larger resolutions are available when supported by the selected CRT monitor. See Section 7.3.8.2 for details.

7.3.4 Colors

These buttons choose the number of colors that are available to Windows 3.1 after Windows is restarted. Some of these choices may be unavailable, because there is not enough display memory in the system, or the selected resolution does not support the amount of colors desired. Generally, 256 color (8-bit) mode is the fastest choice. When more colors are selected, then there can be some slowdown in graphics performance.

7.3.5 Font Size

The small fonts are intended for lower resolutions, and for maximum data availability on higher resolutions monitors. The large fonts are more readable at higher resolutions. This function has the same effect as the *zoom* option in some applications.

7.3.6 Font Cache Size

The Font Cache Size sets the amount of system memory that is available for font caching. Next to the font cache size is an up arrow and a down arrow. Click on the up arrow to increase the cache size; click on the down arrow to decrease the cache size. The minimum cache size is zero, and the maximum is 1000 Kbytes.

Font caching is a technique to increase the performance of Windows by saving the bitmaps of frequently used characters. Normally, when a character is displayed on the screen, it first is created from the True-type outline, then it is copied to the screen. A cached character has already been created and stored, so it is just copied as needed.

WinMode has tried to determine the correct setting for this field, but it may be changed. Just remember, memory set aside for font caching is not available for Windows program and system usage.

7.3.7 DOS-Windows or OS/2-Windows

This selects the operating system on top of which Windows is running.

- When Windows 3.1x is running on top of DOS, select “DOS - Windows”. *(the default)*
- When Windows is running on top of OS/2 2.x (WIN OS/2), select “OS/2 - Windows”.

IMPORTANT: WinMode cannot be used with OS/2 3.0 (Warp) at this time.

OK Button

After clicking **OK**, the computer is reconfigured to use the choices that were made. These changes may need to be added to the AUTOEXEC.BAT file for them to be permanent. When this is necessary, a program prompt asks the user:

1) “make changes for you”

2) “let you make the changes later”

3) “not save the changes”

When WinMode detects that the Windows 3.1 configuration is changed, a dialog asks if Windows should be restarted. When **Yes** is selected, Windows is restarted immediately; when **No** is selected, Windows must be exited and restarted manually for the saved changes to take effect.

Cancel Button

The **Cancel** button closes the dialog box and exits the program without making any changes. Selecting “Close” from the Control menu or double clicking on the control menu box [–] has the same effect.

About Button

The **About** button displays a dialog box with the version number of WinMode and a copyright notice.

Help Button

For help on WinMode, select the **Help** button. The cursor changes to a cursor with a question mark next to it. To select the items for which help is needed, place the new cursor over the field(s) and click the mouse button. The Help Menu for the desired item will appear. Click the mouse button again to make the pop-up text disappear. To change the cursor back to the regular version, put the new cursor plus question mark version over the Help Button and click the mouse button.

7.3.8 Selecting Display Types

The Display Options menu is selected with the **Options** button. It displays a separate dialog box with additional selections pertaining to system configuration and display-driver options. The various sections are shown in Figure 7-2.

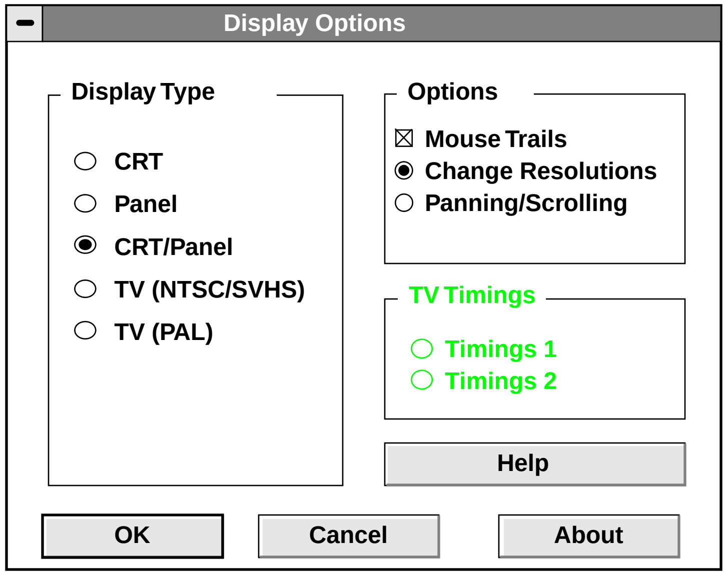


Figure 7-2. Display Options Screen

7.3.8.1 Display Type

When the 'Display Type' dialog box is first displayed, it has the currently active display choice selected. To make a change, select the new display configuration. The NTSC and PAL buttons are presented, only when TV support is enabled at the BIOS level. The actual change takes place after selecting **OK** in the main menu of WinMode.

7.3.8.2 Options

- When the **Mouse Trails** option is selected, mouse trails is immediately enabled for display devices that support them. This is exactly the same as selecting them from the Mouse section of the Control Panel.
- The **Change Resolution** and **Panning/Scrolling** options modify the behavior of the Windows driver.
  - **Change Resolution** makes Windows always run at the exact resolution of the flat panel when in 'Panel' or 'CRT/Panel' modes. When in CRT only mode, Windows runs at the resolution selected in the main screen of WinMode. When the option is changed from **Panning/Scrolling** to **Change Resolution**, and the Resolutions selection in the main menu is higher than the panel can support, the Resolution field automatically falls back to the highest value supported by the panel. When only the resolution and display type (with the exception of TV) are modified while 'Change Resolution' is selected, the changes are made without restarting Windows. (See Notes 1 and 3 below.)
  - **Panning/Scrolling** makes Windows run at the resolution requested in the main screen of WinMode on all displays except the TV. Higher resolutions are made available on the "Resolutions" section of the main menu when supported by the selected CRT monitor. When the selected resolution is higher then that supported by the flat panel, Windows uses a virtual display while in 'Panel' or 'CRT/Panel' modes. This means that the actual display is the resolution of the panel, but when the mouse is moved to an edge of the screen, the display is either scrolled (up or down) or panned (side to side). This gives access to the entire high resolution desktop. Windows must be restarted for changes to take effect when 'Panning/Scrolling' is selected. (See Notes 2 and 3 below.)

NOTES:

1) Changing resolution on-the-fly (without rebooting) can be defeated, if necessary, by changing the WINMODE.INI file. Locate the line "`system.ini,CL_WinAccel,changeres:lin=$winmode.ini,Configuration,changeres,on`", and change the last word from "on" to "`off`".
2) **Panning/Scrolling** option is available for selection only in SimulSCAN or Panel only mode.
3) **Change Resolution** and **Panning/Scrolling** have no effect on far-east Windows. That means Windows always restarts after a selection is made.

7.3.8.3 TV Timings

The TV timings options are <u>not</u> supported with the CL-GD755X, even when a TV encoder and a TV are connected.

OK Button

The **OK** button accepts the choices that have been made and closes the dialog box.

About Button

The **About** button displays a dialog box with the version number of the Display Options utilities and a copyright notice.

Help Button

For help on Display Options in WinMode, select the **Help** button. The cursor changes to a cursor with a question mark next to it. To select the items for which help is needed, place the new cursor over the field(s) and click the mouse button. The Help Menu for the desired item will appear. Click the mouse button again to make the pop-up text disappear. To change the cursor back to the regular version, put the new cursor plus question mark version over the Help Button and click the mouse button.

Cancel Button

The **Cancel** button closes the Display Options dialog box and exits the program without making any changes. Selecting 'Close' from the Control menu or double clicking on the control menu box [–] has the same effect.

7.4 Saving the Changes

Depending on the changes that have been made, either the changes are made immediately, or Windows must be restarted for the changes to take effect. In general, when the only thing that is being changed is the resolution and the display type, and the **Change Resolution** option has been selected, then the changes are made without restarting Windows.

The only exception to this is when changing the display either to or from the TV, assuming that TV is supported. This always causes Windows to restart. WinMode decides whether restarting Windows is necessary. When restarting Windows is necessary, a dialog box is presented asking to "Restart Windows now?" with the options of **OK** or **Cancel**.

- When **OK** is selected, the system does a warm boot and restarts Windows.
- When **Cancel** is selected, the changes made in WinMode are ignored and WinMode is closed.

7.5 TIMINGS.DAT Programming

The list of available monitors and timing parameters that WinMode displays are stored in an ASCII file named TIMINGS.DAT. Based on the information in TIMINGS.DAT, WinMode generates new graphic mode timing parameters and writes them out to the hardware on the controller to take immediate effect for mode preview. Custom timing parameters can be added to TIMINGS.DAT by using the following procedure:

TIMINGS.DAT *Syntax:*

- Any line beginning with '#' or ';' is interpreted as a comment.
- Monitors are grouped in the file by brand name. The brand name is placed in square brackets[ ]. WinMode uses the text inside the brackets when displaying the brand name. The maximum length for a brand name is twenty characters counting spaces and punctuation.
- Monitor models are placed on the lines following the brand name. One model description per line is allowed. A model is described by giving a name, number or short description, followed by available refresh rates for 640 x 480, 800 x 600, 1024 x 768 and 1280 x 1024 resolutions. The resolutions are separated by curly brackets {}. When a particular resolution is not supported, there should be no entries between the brackets. The maximum length for a model name is twenty characters counting spaces and punctuation. All models following a brand will be listed as part of that brand by WinMode.
- Following is the syntax for a brand and monitor entry:

  [*Brand*]
  Model{*ref640*}{*ref800*}{*ref1024*}{*ref1280*} *text*

Brand - monitor brand (max. 20 characters), enclosed in [ ].

Model - monitor model or description (max. 20 characters)

ref640 - List of 640 x 480 integer timing values, separated by commas (up to 8), enclosed in {}.

ref800 - List of 800 x 600 integer timing values, separated by commas (up to 8), enclosed in {}.

ref1024 - List of 1024 x 768 integer timing values, separated by commas (up to 8), enclosed in {}.

ref1280 - List of 1280 x 1024 integer timing values, separated by commas (up to 8), enclosed in {}.

text - optional comment.

Example:

[Brand X]
El Cheapo{60,72}{56,60,72}{60}{}

This is interpreted as a monitor named ‘El Cheapo’ that is made by ‘Brand X’. It supports:

— 640x480 at 60 and 72 Hz refresh.

— 800x600 at 56, 60 and 72 Hz refresh

— 1024x768 at 60 Hz refresh

— It does not support any 1280x1024 configurations

- At a minimum, brand name and model must be specified, along with one refresh value in brackets, or else the entry will be skipped.
- Brand and model name lengths exceeding the maximum will be truncated.
- Brand and model names are case sensitive.
- Do not repeat Brand names, such as [BRAND] and [BRAND].
- White space is allowed for readability.

7.5.1 Replacing EDID Specified Timing

Since the VESA DDC EDID block does not contain bit fields to represent all of the refresh rates supported by Cirrus Logic graphics controllers, TIMINGS.DAT also contains a section for replacing the EDID specified timing parameters for VESA DDC compliant monitors. The **syntax** is similar to the regular monitor timing parameters.

All DDC replacement timing refresh lists are in the section labeled [VESA/DDC Monitors]. This section should come after the last of the standard monitor lists.

Even though [VESA/DDC Monitors] looks like a monitor brand, it will not be listed as a brand by WinMode.

WinMode uses information from the EDID block to identify the monitor in the [VESA/DDC Monitors] section. The syntax for a [VESA/DDC Monitors] entry is:

ABC(*ID*){*ref640*}{*ref800*}{*ref1024*}{*ref1280*} where:

ABC - This is the manufacturers Name exactly as it appears in the EDID block.

ID - ID Product Code from the EDID block. Since some OEMs use multiple IDs for a single brand of monitor, WinMode will parse multiple IDs for a single entry as long as they are separated by commas. Also, when a range of IDs are used for a single monitor model, they can be listed with hyphens. WinMode allows up to four IDs or ranges per monitor. IDs are represented in hexadecimal.

ref640 - List of 640x480 integer timing values, separated by commas (up to 8), enclosed in {}.

ref800 - List of 800x600 integer timing values, separated by commas (up to 8), enclosed in {}.

ref1024 - List of 1024x768 integer timing values, separated by commas (up to 8), enclosed in {}.

ref1280 - List of 1280x1024 integer timing values, separated by commas (up to 8), enclosed in {}.

Examples:

ABC (D, c10, 7, 1C-3C) {60}{60}{60}{}

ABC (B10 - B40, 3001){60,72}{56,60,72}{60}{60}

7.5.2 Custom Refresh Rates

Custom refresh rates can be specified for any of the supported resolutions in WinMode. When the custom timing is selected, WinMode generates the correct timing parameters for the requested refresh rate.

To add a custom refresh rate for a given monitor, add an entry to the refresh list for any of the resolutions, that is a label rather then a numeric refresh rate. A label begins with a letter rather then a number. Labels are defined in the [Custom Timings] section of TIMINGS.DAT. Each custom refresh entry contains all of the data necessary for WinMode to construct a set of new CRTC timing parameters.

The **syntax** for a custom refresh rate is:

label{*ref, hres, vres, pixelclk, hborder, hfporch, hbporch, hsync, hpol, vborder, vfporch, vbporch, vsync, vpol, interlace*} where:

label - The identifier for this refresh rate. Labels always begin with a letter instead of a number. The maximum length is 16 characters.

ref - Refresh rate - decimal integer. When the custom refresh rate is the same as one that is already listed, the custom refresh rate is not listed.

hres - Horizontal display resolution in pixels - decimal integer. When the *hres* field does not match the resolution for the current *hres* list, the custom refresh rate is not listed.

vres - Vertical display resolution in pixels - decimal integer. When the *vres* field does not match the resolution for the current *vres* list, the custom refresh rate is not listed.

pixelclk - Pixel clock in Mhz - decimal floating point number.

hborder - Horizontal border in pixels - decimal integer.

hfporch - Horizontal front porch in pixels - decimal integer.

hbporch - Horizontal back porch in pixels - decimal integer.

hsync - Horizontal sync time in pixels - decimal integer.

hpol - Horizontal polarity - either ‘+’ or ‘–’.

vborder - Vertical border in lines - decimal integer.

vfporch - Vertical back porch in lines - decimal integer.

vbporch - Vertical front porch in lines - decimal integer.

vsync - Vertical sync time in lines - decimal integer.

vpol - Vertical polarity; either ‘+’ or ‘–’.

interlace - is the refresh rate interlaced? ‘0’ = no, ‘1’ = yes.

Example:

c640x480x60{60,640,480,27.5,0,32,96,96,-,0,8,36,6,-,0}

7.5.3 Monitor Selection Field

At the end of the file is the section containing the selected monitor brand and model from the last time WinMode was run. This entry should be blank in any production release.

```
[Select Monitor]
Brand = XXXX
Model = El Cheapo
```

Notes

8 CLMODE UTILITY

(Version 3.10)

CLMode is a DOS based program to configure a Cirrus Logic based graphics controller. It allows the user to select a CRT monitor by brand and model with preferred monitor timing parameters for each available resolution. CLMode also allows the user to preview all graphic modes supported by both the controller and the selected monitor and/or flat panel. The CL-GD755X has panel and/or monitor selection options that can be configured by selecting the **Options** button in the main screen of CLMode. Selections and changes made with this utility can be saved for future use.

8.1 Installation Instructions

CLMode must be installed on the hard disk to get reasonable performance. All external files, such as the individual language data files (CLMODE.EN, CLMODE.FR, CLMODE.JA, etc.) and TIMINGS.DAT must be installed in the same directory as CLMODE.EXE. Specific installation and operating instructions for CLMode v3.1x are supplied with the software.

8.1.1 Hardware Requirements

CLMode requires a 80286 or higher with at least two Mbyte of total RAM memory with at least one Mbyte free, including both standard and extended memory. CLMode needs about one Mbyte of hard-disk space. Also, CLMode needs a graphics card using a supported Cirrus Logic graphics controller.

8.1.2 Software Requirements

CLMode requires MS-DOS 3.0 or PC-DOS 3.0 or higher. A mouse or an equivalent pointing device is recommended.

8.1.3 CLMode-BIOS Compatibility

CLMode v3.1X is designed to work with all previous Cirrus Logic BIOS releases for all products that it supports.

8.1.4 Use and Redistribution of CLMode

This product is designed to be part of the display driver distribution disks shipped by Cirrus Logic. There is no charge for this program. However, a valid binary license agreement is required for redistribution; see Chapter 2 for licensing details.

8.2 Multiple Language Support

Support is available for the following languages - US English, UK English, French, Italian, German, Spanish, Dutch, Swedish, Japanese, Portuguese, Danish, Norwegian, Finnish, Latin American Spanish, Brazilian Portuguese, Korean, Traditional Chinese (Taiwan), and Simplified Chinese. See Section 8.6 on page 10 for details.

Japanese, Korean and Chinese support is provided through unicode versions of CLMode. This a different executables then the standard CLMode, because of the additional size and performance considerations of double-byte support. The unicode version of this program is named UCLMODE.EXE.

The current language is determined at start up from the current codepage. The help file is displayed in the language currently chosen. Each language uses a data file (CLMODE.xx) that is approximately 50 Kbytes per language. If not needed, these files can be deleted.

8.3 User Interface

When CLMODE.EXE is executed, the configuration shown on the CLMode main menu (see Figure 8-1) represents the current graphics configuration. Any choices that are not available are grayed out and not selectable.

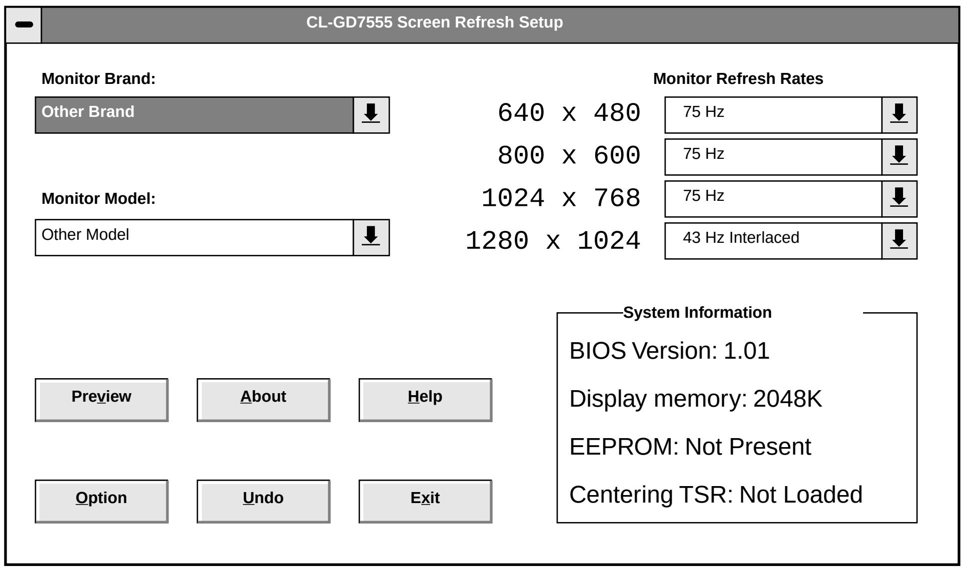


Figure 8-1. CLMode Main Menu

8.3.1 Monitor Selection

The main menu has fields to show and select the attached monitor by brand and model. During initialization, a call is made to inquire for DDC support. If the monitor and the graphics card both support VESA DDC (VESA Display Data Channel), the only choices that are presented are a DDC compliant monitor or ‘Other’ monitor. When a DDC compliant monitor is detected, CLMode displays all of the available refresh rates for that monitor. When some other monitor is selected, CLMode updates all of the available refresh-rate lists to comply with the new selection.

NOTE: During initialization, the screen can be blank for up to three seconds while the system looks for DDC support.

When a non-DDC compliant monitor, or a flat-panel only configuration is detected, the Monitor Brand and Monitor Model fields show ‘Other Brand’ and ‘Other Model’ respectively. These fields can be changed by the user to reflect one of the listed brands or models, or the “Other” fields can be retained, and the individual Refresh Rates can be set manually.

8.3.2 Monitor Refresh Rate Selection

When a DDC compliant monitor is detected, CLMode uses the EDID Established timing-parameters bitmask fields to determine which resolution and refresh rates are available on the current monitor, which are then shown in the applicable ‘Monitor Refresh Rate’ column. The available refresh rates are an ‘ANDed’ list of the rates available on both the monitor and the graphics controller. In the case of the ‘Other’ monitor selection, CLMode provides a list of all of the refresh rates available on the graphics card.

There are a number of refresh rates that are commonly supported on graphics controllers that are not present in the EDID Established timing-parameters fields. To support these capabilities, the TIMINGS.DAT file contains a section of monitor timing parameters that are for DDC monitors. When a DDC monitor is detected, CLMode extracts the brand and model from the EDID. Then TIMINGS.DAT is parsed to see if that monitor is listed in the DDC portion of TIMINGS.DAT. If it is, the rest of the information in the EDID is ignored in favor of the information in TIMINGS.DAT. If there is no entry in TIMINGS.DAT for the monitor, then the EDID is processed normally. The VESA VBE/DDC v1.0 specification documents the BIOS calls that are used to attempt to identify the monitor and available refresh rates.

NOTE: The DDC section of TIMINGS.DAT is never displayed to the end-user with the other monitor brands and models. It is for internal processing only.

8.3.2.1 Custom Refresh Rates

Additional custom refresh rates can be defined in the TIMINGS.DAT file by OEMs. Custom monitor timing parameters are supported through the detailed timing parameters section of the EDID. Any detailed timing parameters for 640 x 480, 800 x 600, 1024 x 768, or 1280 x 1024 are listed in the refresh rates list for the appropriate resolution. If either the monitor or the graphics card does not support VESA DDC, the user is presented with a list of available monitor brands and models. These choices come from the file TIMINGS.DAT. TIMINGS.DAT is an editable ASCII file containing descriptions of monitors.

NOTE: For all resolutions above 640x480, there is an entry labeled ‘none’. If ‘none’ is selected, that resolution, and any higher resolution, are changed to ‘none’. This affects not only the refresh rate combination boxes, but also the resolution radio buttons.

OEM defined refresh rates can be specified in the TIMINGS.DAT file by using a label in the refresh rate for a given monitor instead of a numeric refresh rate. The data for the OEM defined monitor timing parameters are defined in the [Custom Timings] section of TIMINGS.DAT. Whenever a new selection is made in the ‘Monitor Brand’ field, the entries in the ‘Monitor Model’ field are updated to reflect the newly selected brand. When a new monitor model is selected from the ‘Monitor Model’ field, the refresh rates for all resolutions are updated to reflect the highest refresh rates available on both the monitor and the graphics card. See Section 8.5, TIMINGS.DAT Programming for details.

8.3.3 Mode Previews

The **Preview** button allows the user to preview all available graphics modes to verify that the selected CRT monitor timing parameters work properly on the selected monitor. The 'Modes Preview' dialog box is shown in Figure 8-2. Only the graphic modes that are available with the controller, selected monitor timing, and graphic memory are listed. The mode list provides a list of graphic modes sorted in descending order of resolution and palette-depth complexity. For each mode, it presents the resolution, number of colors, type (text or graphic), character-cell size, Cirrus Logic mode number, and VESA mode number (where applicable).

When panel-only or SimulSCAN (CRT and flat panel) options are selected in the Panel Control screen, the parameters in the Modes Preview screen are changed to reflect the modes supported by the flat-panel connected to the graphics controller. The flat panel mode information is provided by the controller VGA BIOS.

Modes Preview

Resolution	Colors	Type	Cell	Mode	VESA Mode
1280 x 1024	256	Graphic	8 x 16	6Dh	107h
1280 x 1024	16	Graphic	8 x 16	6Ch	106h
1024 x 768	64K	Graphic	8 x 16	74h	117h
1024 x 768	32K	Graphic	8 x 16	68h	116h
1024 x 768	256	Graphic	8 x 16	60h	105h
1024 x 768	16	Graphic	8 x 16	5Dh	104h
800 x 600	16M	Graphic	8 x 16	78h	115h
800 x 600	64K	Graphic	8 x 16	65h	114h
800 x 600	32K	Graphic	8 x 16	67h	113h
800 x 600	256	Graphic	8 x 16	5Ch	103h
800 x 600	16	Graphic	8 x 16	6Ah	102h
640 x 480	16M	Graphic	8 x 16	71h	112h

Show   Quit

Figure 8-2. Graphic Mode Preview Screen

Show Button

The **Show** button displays a preview screen of the currently selected graphics mode. This gives the viewer a chance to see what each mode will look like when displayed.

Quit Button

The **Quit** button returns to the main menu.

8.3.4 Panel Control Options

The **Option** button on the main menu selects Panel Control options for the CL-GD755X, which allow switching between CRT, flat panel, simultaneous CRT and flat panel (SimulSCAN), or TV. When the dialog box is first displayed, It will reflect the current state of the graphics controller.

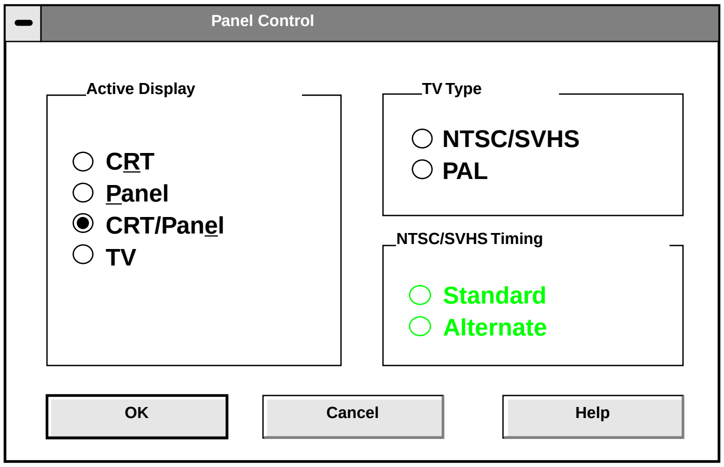


Figure 8-3. Display Options Screen

Active Display

These buttons are used to select whether the CL-GD755X outputs to the CRT monitor, flat panel, simultaneous flat panel and CRT (SimulSCAN), a TV, or a CRT and TV. There are mutiple TV-Out functions available depending on the device type.

- Internal TV encoding is supported with an external analog decoder in the CL-GD7556, but not in the CL-GD7555. The internal TV encoding is the default fro the CL-GD7556. It is selected by choosing the **TV** button. When the internal TV-Out function is selected, no other displays are supported.
- The default in the CL-GD7555 is no TV-Out.
- External TV Encoders are supported in both devices. This function must be enabled by changing the VGA/BIOS. See the VGA/BIOS, Chapter 6, for details. When the VGA/BIOS is changed, this external TV-Out function can support simultaneous TV and CRT operation. It is selected by choosing the **TV** button.

TV Type

When TV encoding is enabled, and a TV is connected, the ‘TV’ and ‘TV/CRT’ Active Display and the TV-Type controls in the Panel Control window are available. In the TV Type selection box:

- ‘NTSC/SVHS’ is the default selection when the current language is U.S. English or Japanese
- ‘PAL’ standard is used almost everywhere except USA, Japan, France and Russia.

These buttons are grayed out and not selectable unless the ‘TV’ or ‘TV/CRT’ are selected in the ‘Active Display’ box.

NTSC/SVHS Timing

These controls are not available with the CL-GD755X.

OK Button

When the **OK** button is selected, CLMode immediately makes all changes, closes the dialog box, and returns the user to the main menu of CLMode.

Cancel Button

When the **Cancel** button is selected, the 'Panel Control' dialog box closes with no changes being made, and returns the user to the main menu of CLMode.

Help Button

Selecting the **Help** button displays the CLMode help text.

8.3.5 Saving the Changes

When the refresh rates have been changed, or custom refresh rates have been added, the new CRT monitor timing parameters are programmed using Cirrus Logic extended BIOS call BIOS 'Set Monitor Type'. These changes take effect on subsequent mode sets. When CLMode terminates, a dialog box is presented asking if the changes should be saved to the AUTOEXEC.BAT file. If the answer is yes, a line is added to the AUTOEXEC.BAT. If the answer is no, the monitor timing parameters are lost after the next cold boot or restart.

Undo Button

The **Undo** button on the main menu cancels all changes that have been made since CLMode was started.

Help Button

The **Help** button on the main menu displays the CLMode help file.

About Button

The **About** button on the main menu displays a dialog box containing the version and copyright information for CLMode.

Exit Button

The **Exit** button on the main menu saves the current settings, updates the configuration, and exits the program.

8.4 Error Handling/Messages

CLMode has very few error messages. Instead, the program tries to make choices available that are currently appropriate.

8.5 TIMINGS.DAT Programming

The list of available monitors and timing parameters that CLMode displays are stored in an ASCII file named TIMINGS.DAT. The same file is also used for WINMODE.EXE, the Cirrus Logic Windows 3.1 based configuration utility.

Based on the information in TIMINGS.DAT, CLMode generates new graphic mode timing parameters and writes them out to the hardware on the controller to take immediate effect for mode preview. Custom timing parameters can be added to TIMINGS.DAT by using the following procedure:

TIMINGS.DAT *Syntax:*

- Any line beginning with '#' or ';' is interpreted as a comment.
- Monitors are grouped in the file by brand name. The brand name is placed in square brackets[ ]. CLMode uses the text inside the brackets when displaying the brand name. The maximum length for a brand name is twenty characters counting spaces and punctuation.
- Monitor models are placed on the lines following the brand name. One model description per line is allowed. A model is described by giving a name, number or short description, followed by available refresh rates for 640 x 480, 800 x 600, 1024 x 768 and 1280 x 1024 resolutions. The resolutions are separated by curly brackets {}. If a particular resolution is not supported, there should be no entries between the brackets. The maximum length for a model name is twenty characters counting spaces and punctuation. All models following a brand will be listed as part of that brand by CLMode.
- Following is the syntax for a brand and monitor entry:

  [*Brand*]
  Model{*ref640*}{*ref800*}{*ref1024*}{*ref1280*} *text*

  Brand - monitor brand (max. 20 characters), enclosed in [ ].

  Model - monitor model or description (max. 20 characters)

  ref640 - List of 640 x 480 integer timing values, separated by commas (up to 8), enclosed in {}.

  ref800 - List of 800 x 600 integer timing values, separated by commas (up to 8), enclosed in {}.

  ref1024 - List of 1024 x 768 integer timing values, separated by commas (up to 8), enclosed in {}.

  ref1280 - List of 1280 x 1024 integer timing values, separated by commas (up to 8), enclosed in {}.

  text - optional comment.

Example:

[Brand X]
El Cheapo{60,72}{56,60,72}{60}{}

This is interpreted as a monitor named ‘El Cheapo’ that is made by ‘Brand X’. It supports:

— 640x480 at 60 and 72 Hz refresh.
— 800x600 at 56, 60 and 72 Hz refresh
— 1024x768 at 60 Hz refresh
— It does not support any 1280x1024 configurations

- At a minimum, brand name and model must be specified, along with one refresh value in brackets, or else the entry will be skipped.
- Brand and model name lengths exceeding the maximum will be truncated.

- Brand and model names are case sensitive.
- Do not repeat Brand names, such as [BRAND] and [BRAND].
- White space is allowed for readability.

8.5.1 Replacing EDID Specified Timing

Since the VESA DDC EDID block does not contain bit fields to represent all of the refresh rates supported by Cirrus Logic graphics controllers, TIMINGS.DAT also contains a section for replacing the EDID specified timing parameters for VESA DDC compliant monitors. The **syntax** is similar to the regular monitor timing parameters.

All DDC replacement timing refresh lists are in the section labeled [VESA/DDC Monitors]. This section should come after the last of the standard monitor lists.

Even though [VESA/DDC Monitors] looks like a monitor brand, it will not be listed as a brand by CLMode.

CLMode uses information from the EDID block to identify the monitor in the [VESA/DDC Monitors] section. The syntax for a [VESA/DDC Monitors] entry is:

ABC(*ID*){*ref640*}{*ref800*}{*ref1024*}{*ref1280*} where:

ABC - This is the manufacturers Name exactly as it appears in the EDID block.

ID - ID Product Code from the EDID block. Since some OEMs use multiple IDs for a single brand of monitor, CLMode will parse multiple IDs for a single entry as long as they are separated by commas. Also, if a range of IDs are used for a single monitor model, they can be listed with hyphens. CLMode allows up to four IDs or ranges per monitor. IDs are represented in hexadecimal.

ref640 - List of 640x480 integer timing values, separated by commas (up to 8), enclosed in {}.

ref800 - List of 800x600 integer timing values, separated by commas (up to 8), enclosed in {}.

ref1024 - List of 1024x768 integer timing values, separated by commas (up to 8), enclosed in {}.

ref1280 - List of 1280x1024 integer timing values, separated by commas (up to 8), enclosed in {}.

Examples:

ABC (D, c10, 7, 1C-3C) {60}{60}{60}{}

ABC (B10 - B40, 3001){60,72}{56,60,72}{60}{60}

8.5.2 Custom Refresh Rates

Custom refresh rates can be specified for any of the supported resolutions in CLMode. If the custom timing is selected, CLMode generates the correct timing parameters for the requested refresh rate.

To add a custom refresh rate for a given monitor, add an entry to the refresh list for any of the resolutions, that is a label rather then a numeric refresh rate. A label begins with a letter rather then a number. Labels are defined in the [Custom Timings] section of TIMINGS.DAT. Each custom refresh entry contains all of the data necessary for CLMode to construct a set of new CRTC timing parameters.

The **syntax** for a custom refresh rate is:

label{*ref, hres, vres, pixelclk, hborder, hfporch, hbproch, hsync, hpol, vborder, vfporch, vbproch, vsync, vpol, interlace*} where:

label - The identifier for this refresh rate. Labels always begin with a letter instead of a number. The maximum length is 16 characters.

ref - Refresh rate - decimal integer. When the custom refresh rate is the same as one that is already listed, the custom refresh rate is not listed.

hres - Horizontal display resolution in pixels - decimal integer. When the *hres* field does not match the resolution for the current *hres* list, the custom refresh rate is not listed.

vres - Vertical display resolution in pixels - decimal integer. When the *vres* field does not match the resolution for the current *vres* list, the custom refresh rate is not listed.

pixelclk - Pixel clock in Mhz - decimal floating point number.

hborder - Horizontal border in pixels - decimal integer.

hfporch - Horizontal front porch in pixels - decimal integer.

hbproch - Horizontal back porch in pixels - decimal integer.

hsync - Horizontal sync time in pixels - decimal integer.

hpol - Horizontal polarity - either ‘+’ or ‘–’.

vborder - Vertical border in lines - decimal integer.

vfporch - Vertical back porch in lines - decimal integer.

vbproch - Vertical front porch in lines - decimal integer.

vsync - Vertical sync time in lines - decimal integer.

vpol - Vertical polarity; either ‘+’ or ‘–’.

interlace - is the refresh rate interlaced? ‘0’ = no, ‘1’ = yes.

Example:

c640x480x60{60,640,480,27.5,0,32,96,96,-,0,8,36,6,-,0}

8.5.3 Monitor Selection Field

At the end of the file is the section containing the selected monitor brand and model from the last time CLMode was run. This entry should be blank in any production release.

```
[Select Monitor]
Brand = XXXX
Model = El Cheapo
```

8.6 Language Files Available for CLMode

CLMODE.BR - The Brazilian Portuguese data file.

CLMODE.DA - The Danish data file.

CLMODE.DE - The German data file.

CLMODE.EN - The US English data file.

CLMODE.ES - The Spanish data file.

CLMODE.FI - The Finnish data file.

CLMODE.FR - The French data file.

CLMODE.IT - The Italian data file.

CLMODE.LS - The Latin-American Spanish data file.

CLMODE.NL - The Dutch data file.

CLMODE.NO - The Norwegian data file.

CLMODE.PT - The Portuguese data file.

CLMODE.SV - The Swedish data file.

CLMODE.UK - The UK English data file.

The unicode version of the program for the following languages is named UCLMODE.EXE.

UCLMODE.CN - The Simplified Chinese (Mainland China) data file.

UCLMODE.JA - The Japanese data file.

UCLMODE.KR - The Korean data file.

UCLMODE.TW - The Traditional Chinese (Taiwan) data file.

9 WINDOWS® 95 UTILITIES

Cirrus Logic supplies a set of software drivers and utilities that works with the Microsoft Windows 95™ operating system to configure and run the CL-GD755X graphics LCD/CRT controller. The user can configure the system for a CRT monitor only, a flat-panel only, simultaneous CRT monitor and flat panel, a TV, or a CRT monitor and TV. The resolution, number of available colors, and large or small font sizes are adjustable for both a CRT monitor and a flat panel. The monitor type and refresh rates are also selectable.

When switching between a CRT monitor and a flat panel, or when using SimulSCAN (CRT and flat panel simultaneously), the available options are usually limited by the color depth and resolution capabilities of the flat panel. Therefore, selections are limited to flat panel resolution and color depth.

Cirrus Logic also supplies the Video Port Manager (VPM) Provider software to control a video window available through the V-Port. The details of the VPM software are explained in a separate specification available from Cirrus Logic. Contact the local Cirrus Logic sales office to acquire a copy of the VPM software and the related specification.

9.1 Installation Instructions

The Cirrus Logic 'Windows 95' software drivers and utilities must be loaded into the Windows\System folder. When Windows 95 is supplied by the computer manufacturer, the appropriate Cirrus Logic software should already be installed on the hard disk. When an upgrade package is provided to the end user for Windows 95, the Cirrus Logic software should be accompanied by an Installation program to assure all the files are loaded in the proper folders. For specific installation instructions see the documentation shipped with the software.

9.1.1 Software Requirements

The Cirrus Logic 'Windows 95' software drivers and utilities require Windows 95. A mouse or an equivalent pointing device is recommended.

9.1.2 Hardware Requirements

The Cirrus Logic 'Windows 95' software drivers and utilities work on any system properly configured to run Microsoft® Windows 95. It expects to find a Cirrus Logic LCD/CRT Controller.

9.1.3 Windows 95 – BIOS Compatibility

The Cirrus Logic 'Windows 95' software drivers and utilities are designed to work with all previous Cirrus Logic VGA BIOS releases for all products that it supports.

9.1.4 Use and Redistribution of Cirrus Logic 'Windows 95' Software Drivers and Utilities

A License for Distribution of Executable Software must be acquired from Cirrus Logic to distribute the display drivers and utility software to end users. Such license is required for each Cirrus Logic VGA product type with which the software is shipped.

The software may be used freely by the original owner of the Drivers and Utilities package shipped by Cirrus Logic, but no duplication, distribution or sub-licensing is authorized without a formally executed license.

9.2 Multiple Language Support

Support is available for the following languages - US English, UK English, French (Standard), Italian (Standard), German (Standard), Spanish (Modern Sort), Spanish (Traditional Sort), Dutch (Standard), Swedish, Danish, Norwegian (Nynorsk), Finnish, Latin American Spanish, Portuguese (Standard), Portuguese (Brazilian), Japanese, Korean, Traditional Chinese (Taiwan), and Simplified Chinese.

The Windows menus and Help menus use the language supplied with the version originally installed. Language changes cannot be made from within the Windows program.

9.3 Using the Cirrus Logic 'Windows 95' Software Drivers and Utilities

The drivers are activated with a set of default parameters when Windows 95 is started. By choosing the 'Display' icon in the 'Control Panel' window, a 'Display Properties' folder is opened which contains a number of tabs which are used to choose and configure the display device(s). By selecting the 'Monitor Refresh' tab as shown in Figure 9-1, the user can configure the system for a CRT monitor only, a flat-panel only, or simultaneous CRT monitor and flat panel (SimulSCAN). The TV selections are shown only when TV-Out is enabled and a TV encoder and TV are connected (see VGA/BIOS Chapter 6 for details).

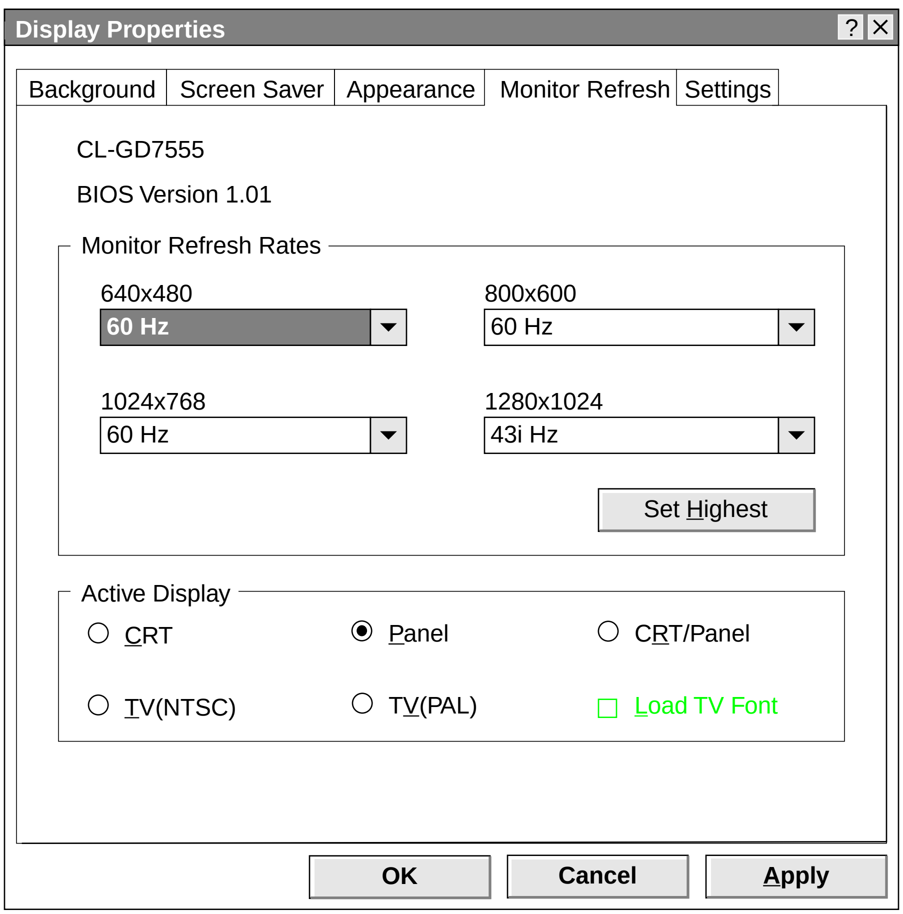


Figure 9-1. Active Display and Monitor Refresh Selection

9.3.1 Monitor Refresh Rates

The drop-down list boxes shown in Figure 9-1 let you select the monitor refresh rates for each resolution. The only choices available are the ones for the monitor that was selected in the 'Monitor Type' field defined in Section 9.3.5. When SimulSCAN is selected, the refresh rate is set by the Cirrus Logic VGA BIOS to make it compatible with the flat-panel timing requirements.

In general, the higher the refresh rate, the better the display quality and the lower the performance. This is because the graphics system can only do a fixed number of operations per second. The more time it spends redrawing the screen, the less time it has available to perform other operations.

9.3.2 Resolution

The CRT and flat panel resolution is selected using the 'Desktop area' adjustment window in the 'Settings' tab as shown in Figure 9-2. The resolution can be changed by sliding the marker back and forth. A preview of the new resolution is shown in the sample monitor window above the adjustment area. The maximum resolution selectable is determined by the capability of the selected CRT monitor in CRT-only mode, or by the capability of the flat panel in all other modes.

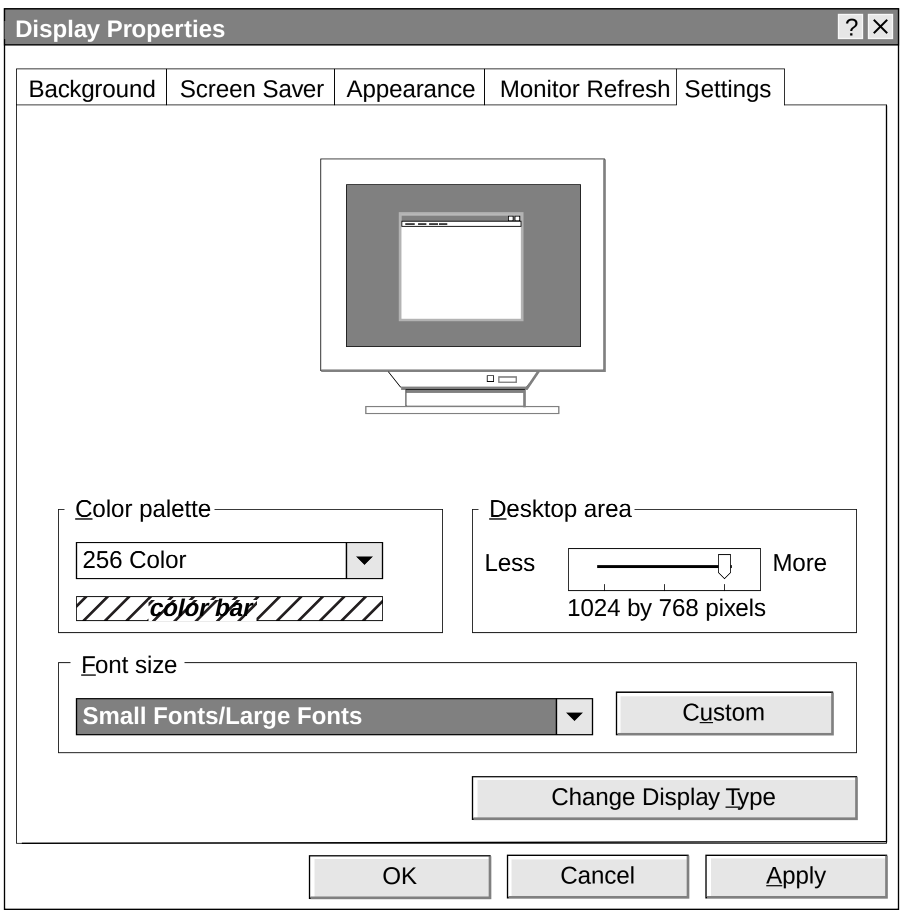


Figure 9-2. Resolution, Color-Depth and Font-Size Adjustments

9.3.3 Color Palette

The color depth supported by the selected CRT monitor and/or panel are shown in the ‘Color palette’ drop-down box in the ‘Settings’ tab as shown in Figure 9-2. The desired color depth can be chosen from the listing box. The available color depth is determined by the VESA modes supported by the CRT/panel controller, and the color capability of the CRT monitor or flat panel itself. Generally, 256 color (8-bit) mode is the fastest choice. If more colors are selected, then there can be some slowdown in graphics performance.

9.3.4 Font Size

The font size used for the basic display items in the menu bars and icons is adjustable in the ‘Font size’ drop-down box in the ‘Settings’ tab as shown in Figure 9-2. The small fonts are intended for lower resolutions, and for maximum data availability on higher resolutions monitors. The large fonts are more readable at higher resolutions. Also, with the larger fonts selected, the windows and icons are enlarged to keep the horizontal separation consistent. This function has the same effect as the *zoom* option in some applications. When changing the font size, the computer must be restarted before the new fonts are shown.

9.3.5 LCD/CRT-Controller Windows 95 Driver Selection

The software drivers used by the CL-GD755X in the Windows 95 environment can be changed or upgraded by selecting the **Change Display Type** button in the ‘Settings’ tab as shown in Figure 9-2. The ‘Adapter Type’ display box shows the current software driver type being used by the computer. By selecting the **Change** button as shown in Figure 9-3, a listing of all compatible Models is shown. Optionally, a listing of all devices provided by the computer manufacturer or software vendor can be selected. The alternate choices are not normally useful in a lap-top computer, where the LCD/CRT Controller chip is not replaceable.

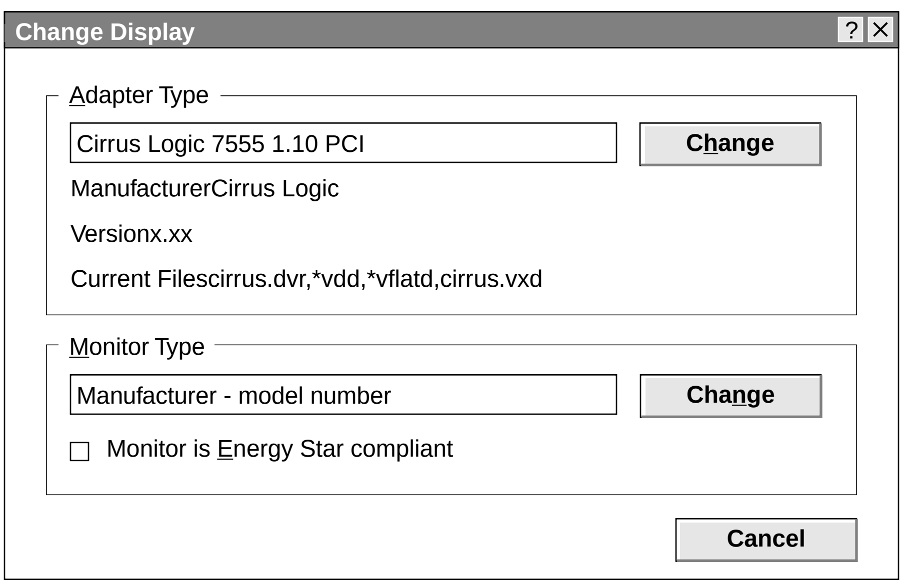


Figure 9-3. LCD/CRT-Controller Windows 95 Driver and Monitor Type Selection

The most likely use for this function is the installation of upgraded software drivers for the existing LCD/CRT Controller chip. To install the new drivers, a disk with the new software is inserted into the computer, and the **Have Disk** button in the ‘Select Device’ window is selected as shown in Figure 9-4. The

Installation software on the disk prompts the user to “add” or “replace” the existing software. The new software shows up in the ‘Models’ listing when the ‘Show compatible devices’ option is selected. Also, the new listing, when selected in the ‘Models’ listing box shown in Figure 9-4, will show in the ‘Adapter Type’ display box shown in Figure 9-3.

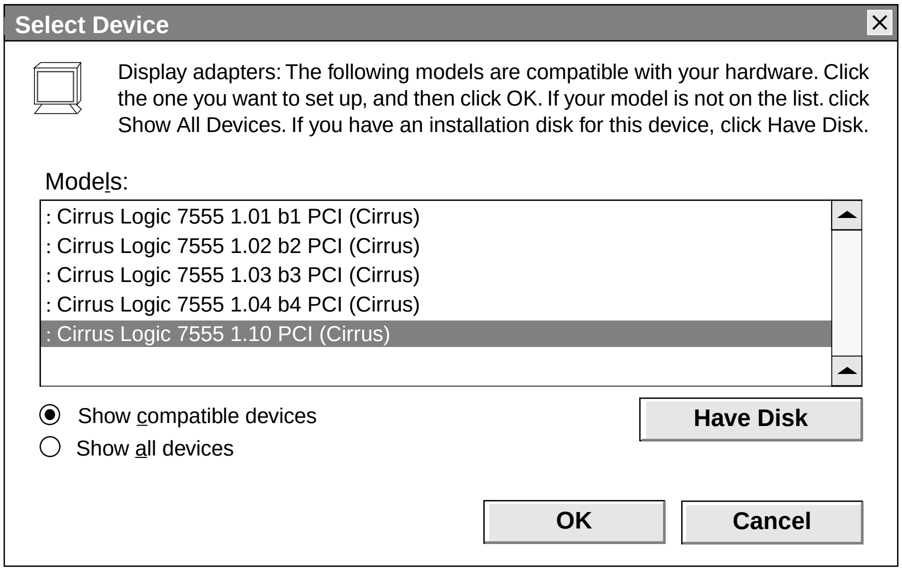


Figure 9-4. Select or Update the Windows 95 Driver

9.3.6 Monitor Type Selection

During Windows 95 initialization, a call is made to identify any attached CRT monitor. If the monitor can be identified, that manufacturers name and model number are displayed in the ‘Monitor Type’ display box as shown in Figure 9-3. When the attached monitor supports VESA DDC (VESA Display Data Channel), the ‘Monitor is Energy Star compliant’ box is checked.

The Monitor type can be changed by selecting the **Change Display Type** button in the ‘Settings’ tab as shown in Figure 9-2, which brings up the ‘Change Display Type’ window shown in Figure 9-3. Selecting the **Change** button next to the ‘Monitor Type’ display box opens a listing of all compatible monitors. When other ‘compatible’ monitors are shown, one of them may be selected from the list, and the OK button confirms the selection.

The ‘Show all devices’ option presents two lists; one for the monitor manufacturers names, and one for their applicable model numbers. Any combination can be selected from these two listings and confirmed with the **OK** button. The new Monitor type will appear in the ‘Monitor Type’ display box.

OK and Apply Buttons

The **OK** and **Apply** buttons in the ‘Settings’ tab confirm any changes made in any the ‘Setting’ fields, and an attempt is made to reconfigured the computer to use the choices that were made. Some changes require a ‘restart’ before they take affect. If this is necessary, you are prompted by the program to “restart your computer now?”. When **Yes** is chosen, the computer restarts. When **No** is chosen, the changes are ignored, as if they were not made.

Cancel Button

The **Cancel** button closes the dialog box without making any changes. Clicking on the control menu box [X] in the upper right-hand corner has the same effect.

9.4 Video Window Size Adjustment

The size of the Video Window can be adjusted by opening the 'Multimedia' icon in the Windows 95 Control Panel. Then, by selecting the 'Video' tab in the 'Multimedia Properties' window, a listing box is presented as shown in Figure 9-5 with the available video window sizes. Video window sizes from 6% of full screen to full screen are available with the CL-GD755X and the Windows 95 software drivers. An optional 'Full screen' button is available which, when selected, expands the video window to cover the entire display. When a new screen size is selected, the video window shown in the center of the sample monitor is changed accordingly to show the relative size relationship to the actual display.

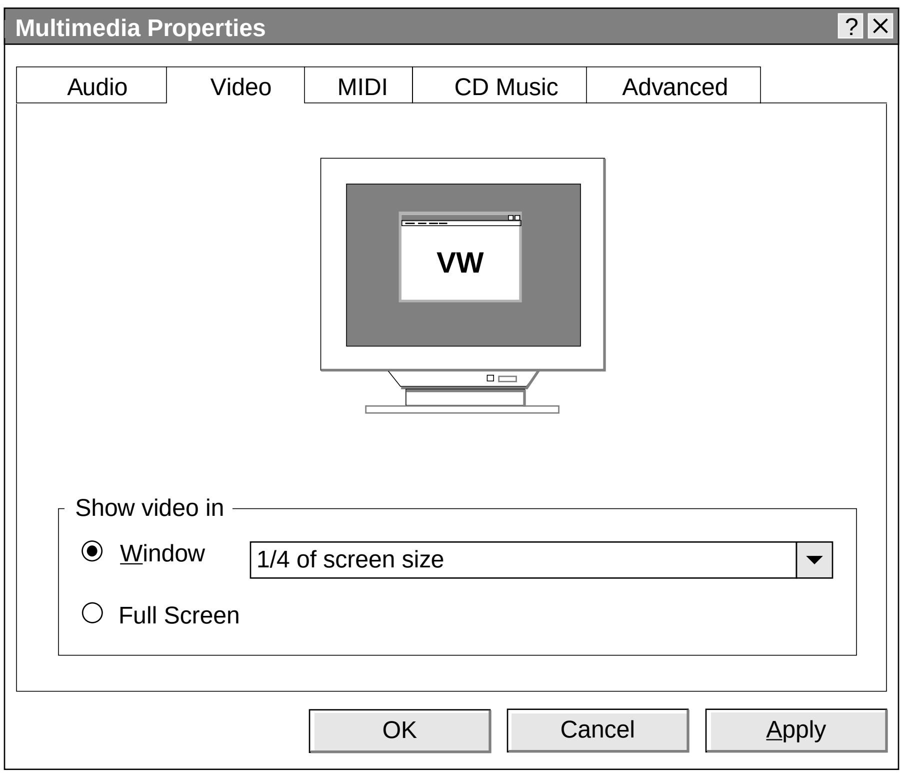


Figure 9-5. Video Window Size Selection

10 Windows NT Utilities

Cirrus Logic does not supply any special utilities for Windows NT 3.51 excepting NTCTRL.CPL which allows a user to change display mode. The utilities included with Windows NT 3.51 support all of the functions necessary to make adjustments to the CL-GD755X.

In Windows NT 4.0, the utility NTCTRL.CPL permits a user to change display mode.

11 REG755X REGISTER EDITOR

11.1 Introduction

REG755X is a Windows™ 3.1x and Windows 95 application program that allows the user to have direct access to all the CL-GD755X’s registers. It also allows users to easily familiarize themselves with the unique video features of the CL-GD755X. This guide briefly explains how to install and use REG755X. It is assumed that the user is familiar with Windows, VGA, and Cirrus Logic’s Extension registers. Those subjects will not be explained. Please refer to the CL-GD755X’s Hardware Reference Manual for more information on the VGA and Extension registers. The MVA™ Registers section is the focus of the discussion that follows.

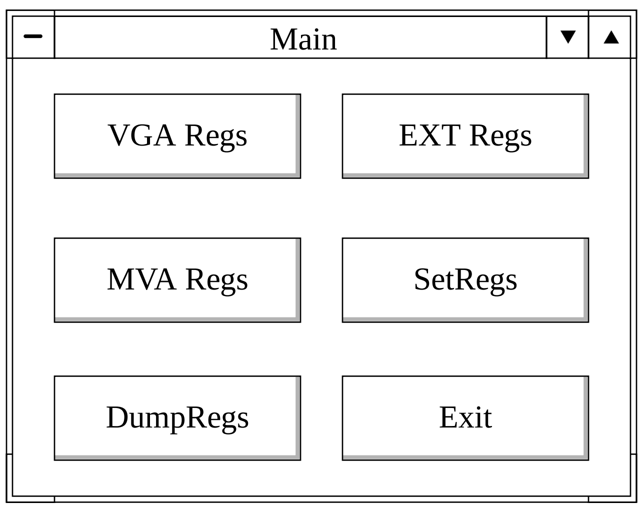
11.2 Installation and Running of REG755X

This program is supplied on two floppy disks. The files occupy about 2.5 meg of disk space when installed.

1) Install the disk labeled DISK 1 and run SETUP.EXE from within program manager. Follow all of the on screen instructions. They are easy and straight forward.

2) After installation, a program group entitled REG755X and an icon called REG755X are constructed.

3) Click on the REG755X icon and a pop-up window entitled MAIN appears with six buttons.



4) The six buttons are called VGA Regs, EXT Regs, MVA Regs, Set Regs, Dump Regs, and Exit.

- **a)** The ***VGA Regs*** button allows access all the standard VGA registers.
- **b)** The ***EXT Regs*** button allows access all the Cirrus Logic CL-GD755X Extension registers.
- **c)** The ***MVA Regs*** button allows access all the MVA and V-Port™ registers and related functions.
- **d)** The ***Dump Regs*** button dumps ALL the registers to a file called DUMPREGS.DAT. This file name cannot be changed within this program. In order to dump register contents of various configurations to multiple file names, the program must first be exited. The DUMPREGS.DAT file must be renamed before trying to save another configuration. If this is not done, any register contents that were previously saved in DUMPREGS.DAT are overwritten.

e) The ***Set Regs*** button allows specific initialization of any of the CL-GD755X's registers. A pop-up open file window appears which allows the selection of specific script files. A default SETREG.DAT file has been provided that will initialize the MVA and V-Port registers. Please use this as an example to create specific script files for specific initialization of any of the registers. Any name can be assigned, but the first two characters of the file must be "#s".

f) The ***Exit*** button terminates the program.

5) When the **VGA Regs** and **EXT Regs** pop-up windows are enlarged to remove the scroll bars, there are buttons on the right side of the window. Most of these buttons are self explanatory. Two buttons used quite often are ***Refresh*** and ***Update***.

a) The ***Refresh*** button writes all the current CL-GD755X register values to the screen.

b) The ***Update*** button moves changes made to specific registers on the screen into the corresponding CL-GD755X registers. Any changes made on the screen do not take affect, unless ***Update*** is used.

6) All changes to register values can be made by manually typing in a value or by using the up/down arrows to increment/decrement values. Any changes made do not take affect unless ***Update*** is used.

7) The pull-down menu of **MENU** will reflect the same functions as the buttons. Either one can be used.

8) Pressing **F1** restores all the register values to their original default values. This is useful when incorrect values have been updated to the CL-GD755X. There is always a way to get back to the original default values.

9) There are on-line register help features. First it is necessary to put the cursor on a specific register (not the register value).

a) By pressing and holding the left button on the mouse, the name of the register will appear on the window caption bar.

b) By double clicking the left button of the mouse, the specific register bit definitions will appear in a pop-up window.

11.3 The MVA and V-Port Registers

All of the above feature and functions are also applicable to the MVA and V-Port Registers. Please refer to the CL-GD755X's Hardware Technical Reference Manual for more information on the MVA and V-Port functions, features, and register descriptions.

11.3.1 *MVA Regs* Button

The ***MVA Regs*** button brings up a pop-up window that allows manipulation of the MVA and V-Port video images. This pop-up window is the window that is most used. This window allows access to the appropriate MVA and V-Port registers by manually writing individual registers, using up/down increment/decrement arrows, or using scroll bars and buttons without any knowledge of the registers. Again all manual changes to the registers do not take effect until the ***Update*** button is touched.

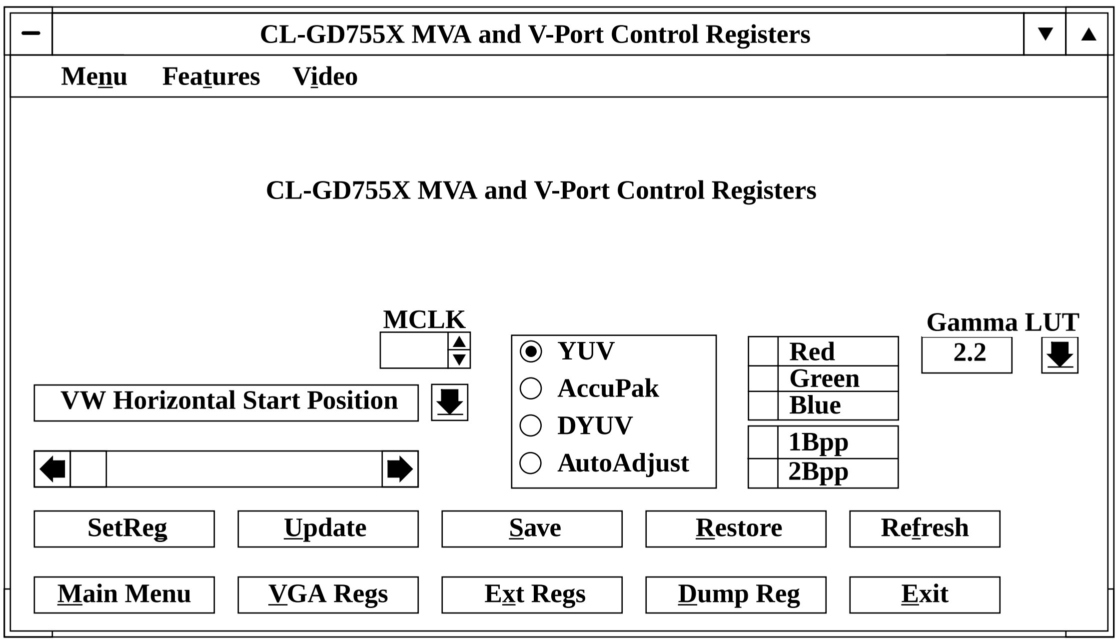


Figure 11-1. MVA and V-Port Control Registers

11.3.2 *Save* and *Restore* Buttons

Most of the buttons on the bottom of the windows are self explanatory and some have already been described above. Two buttons will be explained; they are Save and Restore.

The ***Save*** button allows the saving of the current register contents to memory. This does not save to a file. Therefore, after exiting the program, the saved register contents in memory is lost. Please use **Dump Regs** for saving register contents to a file.

The ***Restore*** button restores the current saved register values in memory to the CL-GD755X. This does not read in a previously saved DUMPREGS.DAT file.

11.4 MVA and V-Port Initialization

The CL-GD755X's MVA and V-Port registers are first initialized to their power-on default values; which is with the V-Port and MVA disabled. The registers can be manually modified to start evaluating the functionality of the MVA and V-Port. A second initialization can be accomplished by loading a script file. This can be accomplished by using Set Regs, as described previously, with the SETREG.DAT default file. Any other script file with the correct format and syntax can also be loaded for specific initialization.

The MVA can be manually enabled by setting CR3C[4] to 1, and the V-Port can be manually enabled by setting CR51[3] to 1. This can also be done by selecting VW Enable in the ***Features*** pull-down menu shown in Figure 11-3.

11.5 Video Source Selection

When the 7110 daughter card is used as a video source, there is a pull-down menu called **VIDEO** that allows initialization of the 7110 chip via the I2C bus. There are two initialization options; one for *S Video* and another for *Composite Video*. One must be selected to choose the type of video source supplied to the V-Port of the CL-GD755X via the 7110.

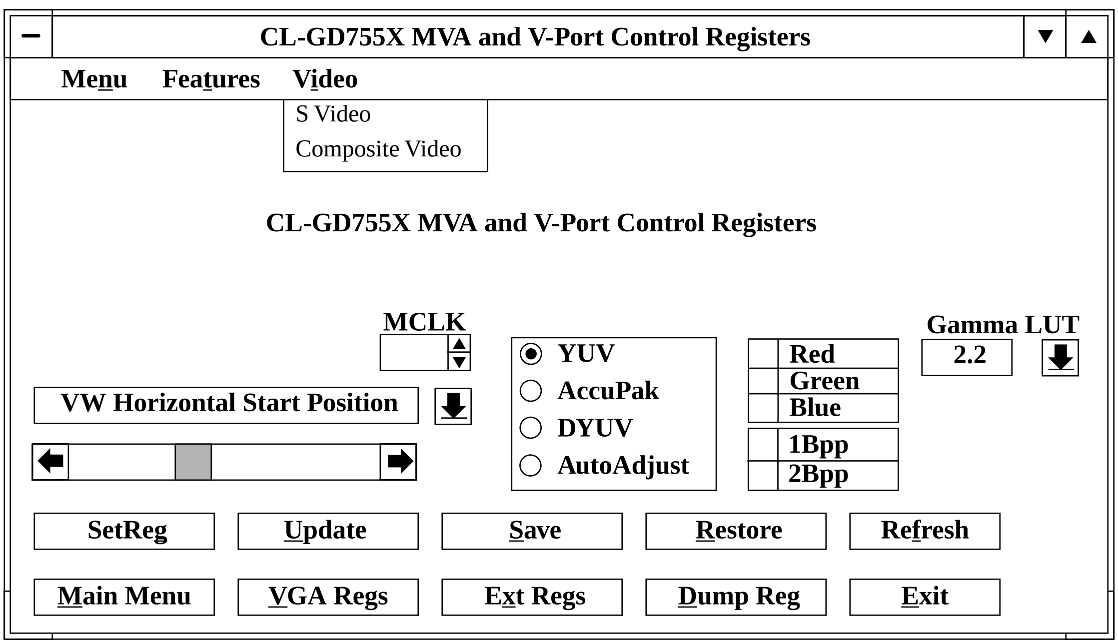


Figure 11-2. Video Source Selection

After selecting either *S-Video* or *Composite Video* from the **VIDEO** pull-down menu, an icon named *I2CTESTW* is created at the bottom left corner of the screen. To complete the initialization of the 7110, open the new icon (double click on the icon) and type <ctrl + c>; the initialization will be completed, and the new window will be closed.

11.6 Features Menu

The ***Features*** pull-down menu enables/disables certain important and most used features without manually writing to a specific register. A check mark means that the feature has been selected. The features are shown in the pull-down box in Figure 11-3.

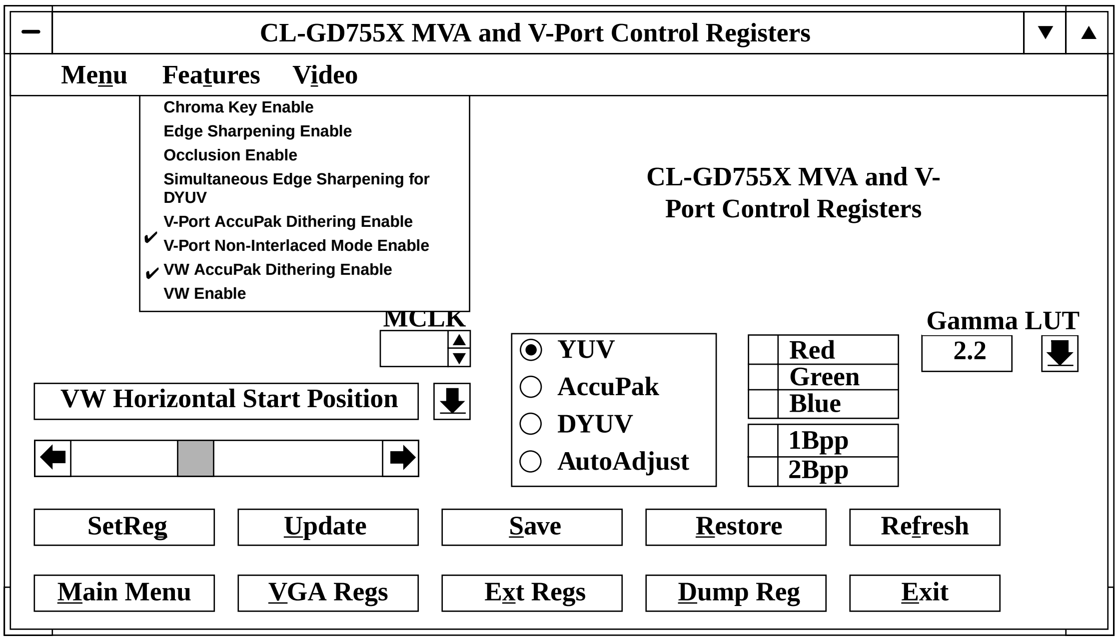


Figure 11-3. REG755X Features Menu

11.6.1 Features Selection Scroll List

There is a selection-box scroll-bar combination that allows you to select 11 Video Window features without having to manually adjust register values. The selection box has the default selection of VW Horizontal Start Position. The down arrow next to it will bring up a pull-down selection menu to choose the 11 different features. They are shown in Figure 11-4.

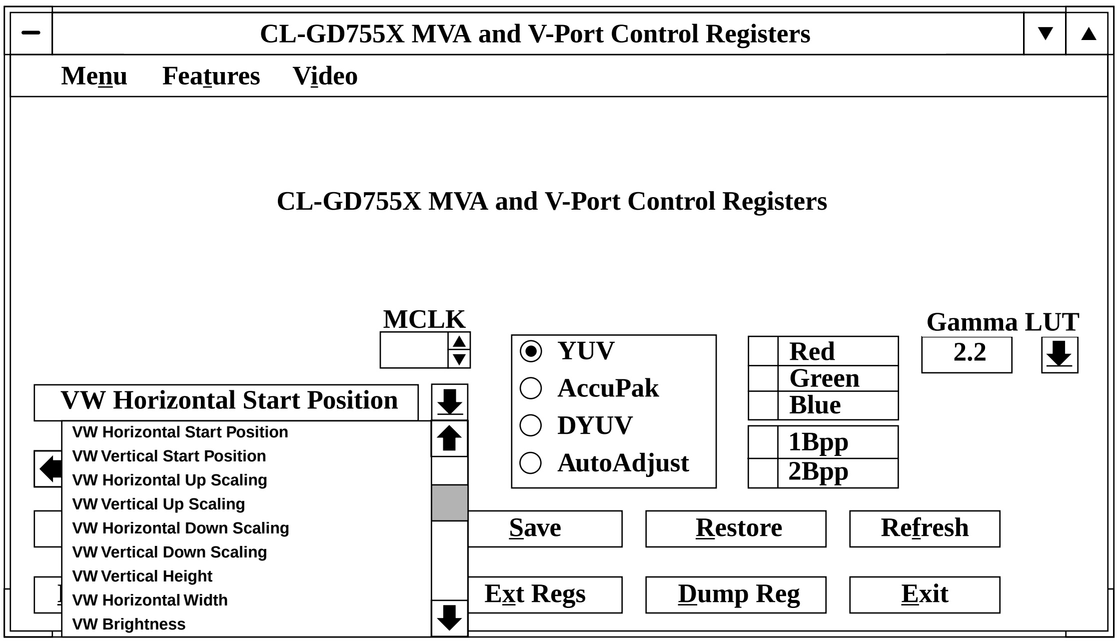


Figure 11-4. Features Selection Scroll List

The scroll bar is located adjacent to the selection box. The scroll bar can be used to modify the appropriate register values automatically. Using this scroll bar changes the video windows in real time; there is no need to use the *Update* button.

11.7 Video Window Gamma Control

The CL-GD755X is the only controller with a VW Gamma Control feature. There are three ways to modify the gamma of the video image to achieve the best display quality. See Figure 11-5.

- One way is to select the ***Gamma*** scroll bar to change the gamma values automatically in real time.
- Another way is to use the ***Gamma LUT*** value selection to change the video gamma values in real time. By selecting the down arrow of the Gamma LUT, a list of gamma values is shown.
- The third way is to set CR36[6] = 1 and press the ***Update*** button or select **VW Live Video Gamma Reduction** in the FEATURES menu. Selecting one, two, all three combinations, or none of the Red, Green, and Blue color buttons, will show the effects on the video image in real time.

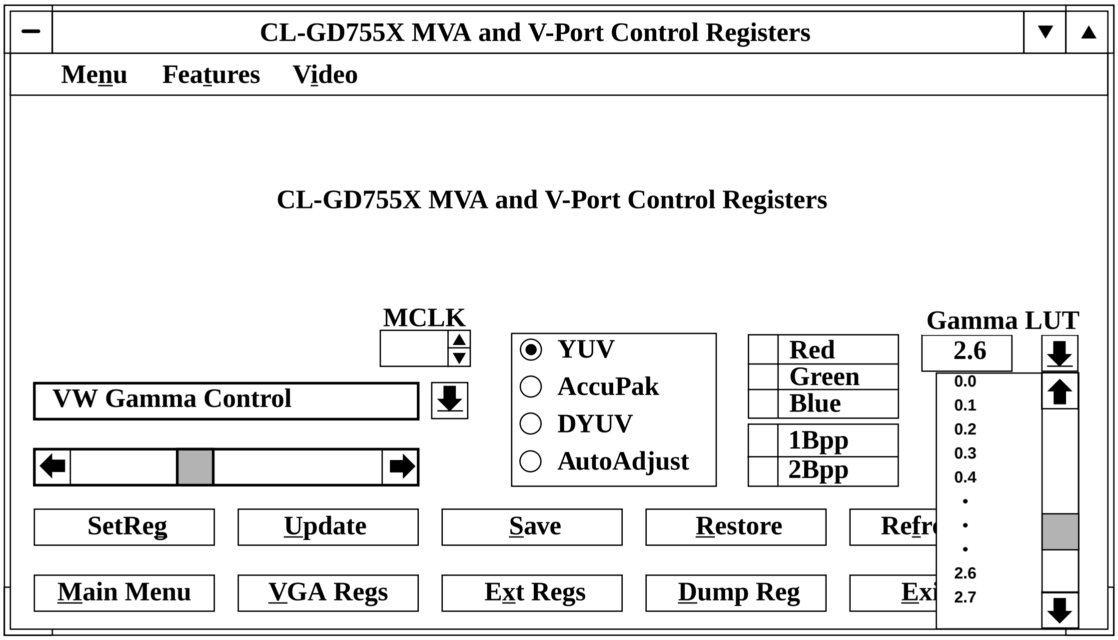


Figure 11-5. VW Gamma Control

11.8 Compression Options

The CL-GD755X has a proprietary compression algorithm that allows the 16-bit 4:2:2 YUV data to be compressed into Cirrus Logic's second generation 8-bit DYUV or first generation 8-bit AccuPak format. The AccuPak format is the same as the industry standard. The user can select the different formats to see the display quality available with each format; see Figure . In certain high-resolution high-color-depth surrounding graphics formats, compression of the 16-bit 4:2:2 YUV format into the 8-bit DYUV or AccuPak format may be necessary to enable display of live video due to memory bandwidth limitations.

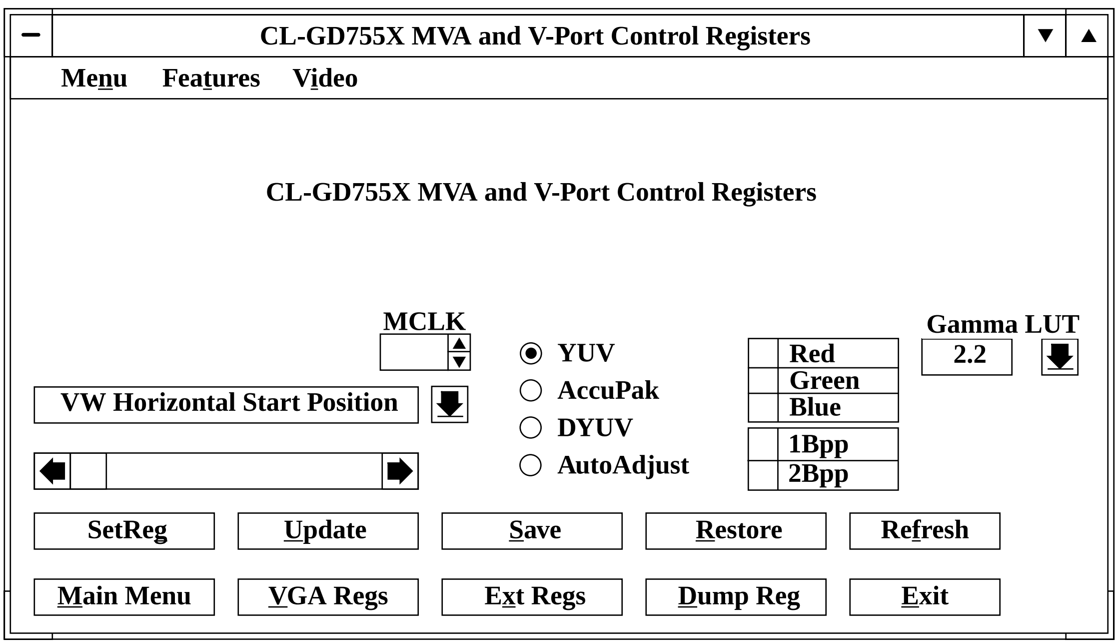


Figure 11-6. Compression and Auto Adjust Options

11.9 Auto Adjust

There is also an ***Auto Adjust*** button that enables the program to make all necessary adjustments of associated registers for a specific function. This button is used with the selection box scroll bar combo, register up/down increment/decrement arrows, and manual updates to registers. The ***Auto Adjust*** button is used in conjunction with the 1-Bpp or 2-Bpp selection box.

- For 256 color surrounding graphics, 1 Bpp must be selected.
- For 64K color surrounding graphics, 2 Bpp must be selected.

This allows the program to adjust all necessary registers for a specific function. Without this, the software may make incorrect adjustments for certain functions.

11.10Program EXIT

When exiting the program to return to Windows program manager, it is advisable to disable the MVA. This can be done by setting CR3C[4] = 0 or by de-selecting VW Enable in the ***Features*** pull-down menu. If this is not done, then the video window remains on the screen.

12 COLOR AND BRIGHTNESS CORRECTION/ADJUSTMENT UTILITY

The purpose of this chapter is to describe the Color and Brightness Correction or Adjustment Utility. The utility has two functions:

- To correct for the brightness and color variations found in flat-panels of different types or from different manufacturers.
- To let the end user modify the color and brightness of the Video Window. This is necessary to compensate for the variations in the video recording format, as well as personal preference in the colors in the video.

12.1 Color and Brightness Correction of the Graphics CLUT

This utility creates a table or equation that loads the graphics CLUT (color lookup table) during the CL-GD755X initialization. This corrected CLUT should be provided by the OEM to compensate for the variations in the flat panel characteristics due to different sizes, types or manufacturers supplied with the portable computer.

Feature Overview

Color Correction Utility — This Windows 95/NT 4.0 utility enables the user to access RGB gamma correction for all color depths, thus allowing the user to adjust for color distortions on the display screen.

Background

While the DAC may generate a linear relationship between the value of a color and the visual appearance of that color, the human eye does not work in the same manner. Without some sort of correction, the color RGB (128,0,0) does not appear to the eye to be half the intensity of the color RGB (255,0,0); in fact it looks much darker. The process of manipulating the meanings of colors to get visual linearity is called Gamma Correction.

Architecture Overview

The color correction software utility is a Win95/NT 4.0 executable that comprises of five slider bars: red, green, blue, brightness and contrast (see Figure 12-1). The setting of the display color adjustment (RGB), brightness and contrast act globally across all tasks. It should be noted that the color adjustment sliders and the brightness slider imply gamma correction. The RGB sliders perform individual gamma correction, and the brightness slider performs gamma correction on all three colors. The contrast effect applies to all three colors also.

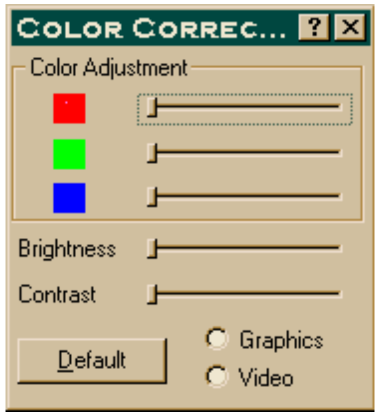


Figure 12-1. Color Adjustment

Figure 12-1 is for CL-GD7555 and CL-GD7556. The two radio buttons, graphics and video, allow the user to do color adjustment on the graphics display or the video window. The default button will display initial settings of the palette lookup table (LUT). By default, these changes apply to all three colors (RGB).

12.2 Initialization

It is assumed that the driver has initialized the RAMDAC (LUT) entries to a predetermined set of values at start-up. In addition, the driver must implement the DDIGammaRamp function, and set the C1_GAMMA_RAMP bit in dpCaps1.

For monitors that support DDC, the gamma factor of the display can be retrieved at start-up from the EDID data structure. This factor will be used by the driver to modify the LUT entries to some default value. The utility will use this number to initialize the slider position. For all other monitors a default gamma value of 0 will be assumed (indicating no change from the default palette).

The GetDeviceGammaRamp API function is called to retrieve the default setting of the LUT. The default entries are then copied into a working structure for user update. The software must determine the color depth of the image. For 16/24bpp colors, the hardware must be directed to make use of the RAMDAC entries by setting the appropriate register bit (SR12[6]). By default, SR12 bit 6 should be set in the driver at start-up.

During the driver installation, new entries will be made in the registry for use in color correction. These entries will be used by the driver to initialize the LUT, and by the utility to initialize the current position of the slider bars and the color radio button. By default, these locations will contain the following values:

Entry	Value (all, blue, green, red)
B Contrast	0x00000080; Binary entry of type DWORD
B Gamma	0x80808080; Binary entry of type DWORD
G 8888Contrast	0x08008800880880; Binary entry of type DWORD
G Gamma	0x80808080; Binary entry of type DWORD

Entry	Value (all, blue, green, red)
ColorFlag	0TBD; Binary entry of type DWORD
V Contrast	0x00000080TBD; Binary entry of type DWORD
V Gamma	0x80808080 TBD; Binary entry of type DWORD

‘G Contrast’ has only a 1 byte entry. The other 3 bytes are reserved. ‘G Gamma’ has 4 byte entries. The most significant byte is used for brightness, followed by blue, green and red gamma respectively. The range for the entries is 0 to 255. When the driver reads the ‘all’ entry for gamma (this is the new brightness value), it will need to add the relative increase or decrease in this value to the RGB entries. This will ensure that all the slider effects are taken into account. The default values must be as shown above. This guaranties a default setting before any changes are made by the user. The first three entries will be used for the Both portion and the last three entries will be used for the Video portion of the color correction utility.

The registry location for these entries will be as follows:

“MyComputer\HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\CIRRUS-Class\DEVICE0Display\Cirrus\ColorCorrection”

The GetDeviceGammaRamp API function is called to retrieve the default setting of the LUT. The default entries are then copied into a working structure for user update. The software must determine the color depth of the image. For 16/24bpp colors, the hardware must be directed to make use of the RAMDAC entries by setting the appropriate register bit (SR12[6]). By default, SR12 bit 6 should be set in the driver at start-up.

12.3 Functionality

Contrast — The following equation is used to determine the color contrast:

New color = (Old color -256128) (contrast factor + 128) / 128 + 128

Brightness — This is really a gamma factor. It is used here to adjust color distortions using the following equation:

((gamma factor - 128) / 128)

10

New color = 256 * (Old color / 256)

Both factors have a range of 0 to 255–128 to +127. If all three primary colors (RGB) are activated, the user will notice a shift in the gray scaling of the colors. For individual color activation, color shift will be emphasized in that particular spectrum.

In the case of 8 bpp/software gamma, the utility will update the brightness (RGB gamma) and contrast factors and inform the driver dynamically, through an ExtEscape call, as the sliders get updated. Then, the API function SetDeviceGammaRamp is used for updating the LUT. All the calculations are performed at the driver level, since the driver needs to initialize the LUT at start-up. The following data format is used to inform the driver of user updates:

Table 12-1. Gamma Correction Data Structure

Cirrus Signature (“CRUS”)	4 bytes
Red Gamma	1 byte
Green Gamma	1 byte
Blue Gamma	1 byte
All Gamma	1 byte
Red Contrast	1 byte
Green Contrast	1 byte
Blue Contrast	1 byte
Reserved Byte	1 byte
Reserved DWord	4 bytes

1- 1- Cirrus signature (9000h) 4 bytes

Cirrus signature (9000h) 4 bytes

2- 2- Red, Green, Blue gamma 4 bytes

Red, Green, Blue gamma 4 bytes

3- 3- Red, Green, Blue, Red, Green, Blue contrast 4 bytes

Red, Green, Blue, Red, Green, Blue contrast 4 bytes

4- 4- reserved 4 bytes

reserved}

There is a reserved byte associated with entry #an2d3 and #3. The fourth entry contains legacy brightness data, from the prior implementation scheme when brightness and gamma were treated separately.

12.4 Driver Interface

The display driver will provide the following escape functions:

```
Color Correction Function
    Function Number:9000
    input: the GammaCorrection Data Structure
    output: none
    return: 1 for success, -1 for failure
Display driver gets the Gamma Correction Data Structure and updates the palette.
```

```
Get_Chip_ID  Function

     Function Number:9001

     input: none

     return: 1 for success, -1 for failure

     output:

          1st byte:   Chip ID

          2nd byte:   0 or 1 for graphics gamma correction support.

 byte:   0 or 1 for Graphics gamma correction support.
          0-   0- chip does not support graphics gamma correction function.

chip does not support graphics gamma correction function.
          2-   2- chip supports graphics gamma correction function.

3rd byte:   0 or 1 for video gamma correction support.

3rd byte:   0 or 1 for video gamma correction support.
          0-   0- chip does not support video gamma correction function.

chip does not support video gamma correction function.
          2-   2- chip supports video gamma correction function.


Write Lookup Table Function

     Function Number:9010

     Input: a pointer to the LUT
```

The driver updates the LUT according to the LUT passed by the utility. The CL-GD7555 driver needs to enable the video gamma correction function.

The help function will be case sensitive, allowing the user to point to a location and get basic information about that particular feature.

Appendix A

BitBLT Engine

A.1 Introduction

BitBLT (bit block transfer) is an operation in which data can be copied from one area of system memory or display memory to another area of display memory.

NOTE: Data cannot be copied from one area of system memory to another area of system memory.

The CL-GD755X BitBLT engine moves data bytes from the source area to the destination area, and it performs 16 logical raster operations on the source bytes as is being transferred to the destination. In addition, it can expand a monochrome image into a color image, and it can replicate a single 8-pixel × 8-pixel pattern to fill a large area. These operations are done with minimum CPU intervention.

A.2 BitBLT Definitions

A.2.1 BitBLT Source Area

The BitBLT source area is the area from which data are copied. The source area is always different than the destination area except in those special cases in which both the source area and the destination area overlap in display memory. The BitBLT source area can have the following characteristics:

- Located either in display memory or system memory, depending on the source area address
- Can be a monochrome image which is expanded into a color image
- Can be a single 8-pixel-by-8-pixel pattern that is replicated to fill a larger area

A.2.2 BitBLT Destination Area

The BitBLT destination area is the area in display memory into which data are written.

A.2.3 BitBLT Width

The BitBLT 'width' refers to the number of bytes (not pixels) in the destination area that are processed before adding BitBLT pitch values to the address values. For an illustration of the BitBLT width, refer to Figure A-1. Special conditions for determining BitBLT width are:

- When the destination area is actually or potentially on the display screen (that is, it is a rasterized area that will be displayed), the BitBLT width is the number of bytes to be written in each scanline.
- When the destination area is off-screen, and the source is a rasterized area (and there is no patter-fill or color expansion), the BitBLT width is the number of bytes per scanline of source data.
- When neither the source nor destination is a rasterized area, the BitBLT width is the number of bytes (not pixels) in the destination area that are processed before adding BitBLT pitch values to the address values.
- The 13-bit BitBLT width is specified in Extension register pair GR20/GR21 and has a maximum value of 8192 bytes. The number written into this register pair is 1 byte less than the actual desired width.

13-Bit BitBLT Width-Register Format.

GR21[4:0]		GR20[7:0]	
12	8	7	0

A.2.4 BitBLT Height

The BitBLT 'height' refers to the number of times the pitch values are added to the address values. For an illustration of the BitBLT height, refer to Figure A-1.

- When the destination area is actually or potentially on the display screen (that is, it is a rasterized area that will be displayed), the BitBLT height is the number of scanlines in that area.
- The 11-bit BitBLT height is specified in the Extension register pair GR22/GR23 and has a maximum height of 2048 scanlines. The number written into this register pair is 1 scanline less than the actual desired height.

BitBLT width and BitBLT height are two numbers multiplied together to define the number of bytes in the destination area.

11-Bit BitBLT Height-Register Format.

GR23[2:0]		GR22[7:0]	
10	8	7	0

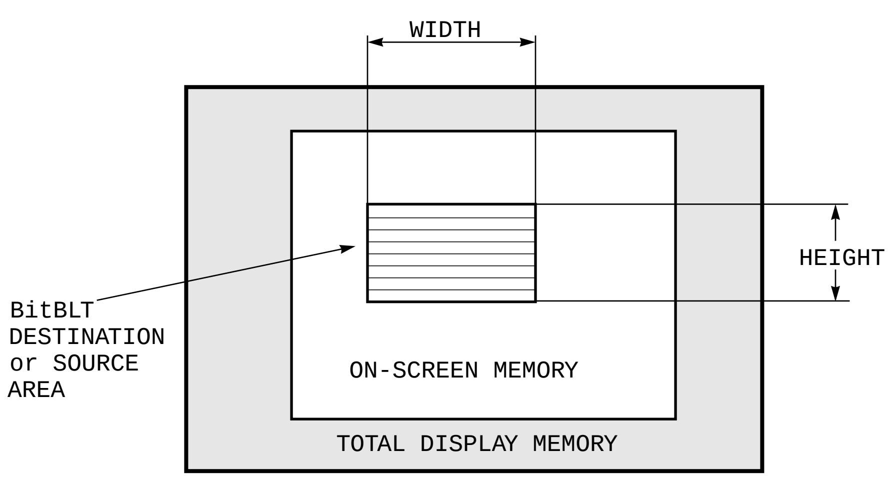


Figure A-1. BitBLT Width and Height

A.2.5 BitBLT Source Pitch and Destination Pitch

The 'source pitch' and 'destination pitch', which are specified separately, are the values added to the respective addresses after each 'width' bytes of destination have been processed.

- The BitBLT 'source pitch' is a value added to the source address.
- The BitBLT 'destination pitch' is a value added to the destination address.

For the case of a rasterized image, refer to Figure A-2.

- When a source area or destination area is a rasterized image, the respective pitch is the number of bytes between vertically adjacent pixels. For example, this value is the number of bytes between the (first) pixel of scanline 'n' and the first pixel of scanline 'n+1', that is, the number that is added to the area address to get from one scanline to the next.
- When a source area or destination area is in off-screen display memory, data pertaining to each of the scan-lines are often stored in contiguous locations, which minimizes fragmentation. In this case, the respective pitch would be set equal to (width + 1).
- When a source area is in system memory, the respective pitch is unused and is a 'don't care'.
- When executing BitBLTs with pattern copy or color expansion, the source area is assumed to be linear, and the source pitch is a 'don't care'.
- The 13-bit BitBLT 'destination pitch' is specified in Extension register pair GR24/GR25 and has a maximum value of 8192 bytes.
- The 13-bit BitBLT 'source pitch' is specified in Extension register pair GR26/GR27 and has a maximum value of 8192 bytes.

13-Bit BitBLT Destination-Pitch-Register Format.

GR25[4:0]		GR24[7:0]	
12	8	7	0

13-Bit BitBLT Source-Pitch-Register Format.

GR27[4:0]		GR26[7:0]	
12	8	7	0

A.2.6 BitBLT Start

The BitBLT 'start' refers to the address of the first byte of a source area or destination area. This first byte is a byte offset from the beginning of display memory. For an illustration of the BitBLT start, refer to Figure A-2.

- The 'source-area start' address is specified in Extension register triplet GR2C/GR2D/GR2E.
- The 'destination-area start' address is specified in Extension register triplet GR28/GR29/GR2A.

Each start address is a 22-bit value, supporting up to 4 Mbytes of display memory.

22-Bit BitBLT Destination-Start-Address Register Format

GR2A[5:0]	GR29[7:0]	GR28[7:0]
21 16	15 8	7 0

22-Bit BitBLT Source-Start-Address Register Format

GR2E[5:0]	GR2D[7:0]	GR2C[7:0]
21 16	15 8	7 0

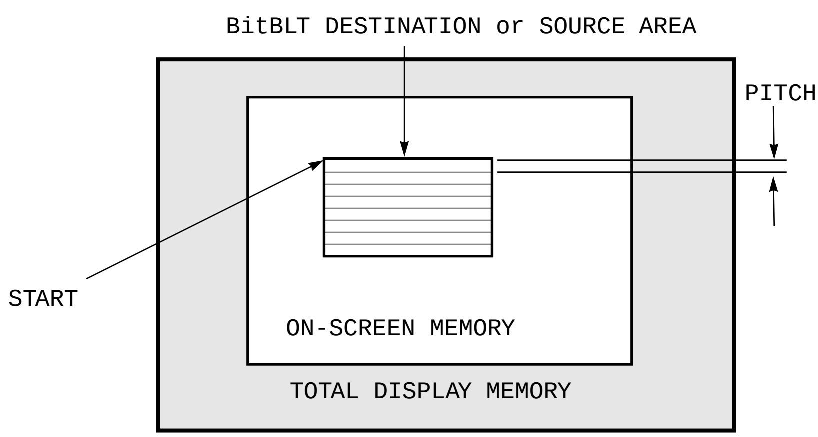


Figure A-2. BitBLT Pitch and Start

A.2.7 Example: BitBLT Display Memory-to-Display Memory

For a BitBLT operation that copies source data of 128 bytes × 64 scanlines from one area of display memory into another part of display memory, Table A-7 shows how the BitBLT registers must be programmed.

In this example, the CL-GD755X is assumed to be programmed in display mode 65h (that is, 800 × 600, 16 bits per pixel). Therefore, each 128-byte scanline represents 64 pixels. The source area of this operation begins at address 0. The destination area begins at 160,200 (that is, 100 scanlines from the top of the screen, and 100 pixels from the left edge).

Table A-1. Typical Register Settings for Source Copy BitBLT Operation

Field	Graphics Controller Registers	Extension Registers	Field Contents	How Field Contents Are Calculated
Background Color	GR0	GR10	'Don't care'	Not applicable
Foreground Color	GR1	GR11	'Don't care'	Not applicable
Width		GR20, GR21	127	(64 pixels × 2 bytes/pixel) – 1 bytes = 127 bytes
Height		GR22, GR23	63	64 scanlines – 1 scanline = 63 scanlines
Destination Pitch		GR24, GR25	1,600	800 pixels × 2 bytes/pixel = 1600 bytes
Source Pitch		GR26, GR27	1,600	800 pixels × 2 bytes/pixel = 1600 bytes
Destination Start		GR28, GR29, GR2A	160,200	(100 scanlines × 1600 bytes/scanline) + (100 pixels × 2 bytes/pixel) = 160,200 bytes
Source Start		GR2C, GR2D, GR2E	0	(From the beginning of display memory)
Write mask		GR2F	0	Only for color expansion
BitBLT Mode		GR30	00 (hex)	From display screen to display screen. No color expansion. No pattern copy.
Start		GR31	02 (hex)	Set GR31[1] to 1.
Raster Operation		GR32	0D (hex)	Source copy
Mode Extensions		GR33	0	No extensions used

This BitBLT is executed as indicated in the following fragment of pseudo-code:

```
sourceAdrs = sourceStartAdrs;
destAdrs = destStartAdrs;
for (i = 0; i < Height; i++)                /*For each scanline*/
    {
    workingSourceAdrs = sourceAdrs
    workingDestAdrs = destAdrs
    for (j =0; j < Width; j++)              /*For each byte*/
         {
         Process one byte of Destination;
         workingDestAdrs++;                 /*to next byte*/
         workingSourceAdrs++;
         }
    sourceAdrs = sourceAdrs + sourcePitch; /*Next scanline*/
    destAdrs = destAdrs + destPitch;
    }
```

The BitBLT engine processes the bytes of a scanline in a destination area, incrementing temporary source and destination addresses after each byte of the scanline. Multiple scanline bytes are processed in parallel.

At the end of each scanline, the source pitch value is added to the source address as it was at the beginning of the scanline, and the destination pitch value is added to the destination address as it was at the beginning of the scanline. The BitBLT engine then moves on to the next scanline, and the process continues. This BitBLT loop iterates these additions (that is, the number of times the pitch values are added to the address values) until the specified height is reached.

All BitBLTs are processed in this general manner. However, some variations include the following:

- For color expansion, the source address is incremented as a bit address rather than as a byte address.
- Pattern fill occurs as explained in Section A.6.
- As discussed in Section A.7, when there is an overlap of the source area and the destination area, the BitBLT direction bit (that is, Extension register bit GR30[0]) must be set to 1 to decrement the source and destination addresses, which reverses the direction in which the BitBLT operation progresses.

A.3 Raster Operations

In addition to moving bytes from the source area to the destination area, the CL-GD755X BitBLT engine can use logical ROPs (Raster OPerations) to combine either source bytes or pattern bytes with destination bytes. Also, the ROPs can be used to ignore bytes in the source area, the destination area, or both.

Table A-2 enumerates the 16 ROPs of Extension register GR32, for logically combining either a source bit (S) or a pattern bit (P) with a destination bit (D). Regardless of whether the ROP uses either a source bit or a pattern bit to combine with a destination bit, the same values of Extension register bits GR32[7:0] are used for the combinations. The selection of the source (S) or pattern (P) is done with GR30[6].

The CL-GD755X BitBLT engine directly implements all 16 ways of logically combining two operands. However, in the unusual cases for which three operands need to be combined, the CL-GD755X BitBLT engine can use the operands of Table A-2 to synthesize an indirect way to logically combine the three operands.

Table A-2. Logical Combinations of Bits Using the BitBLT Engine[a]

GR32[7:0] Setting (hex)	Resulting Source ROP: Source Bit Combined with Destination Bit	Microsoft® Name for Source ROP	Resulting Pattern ROP: Pattern Bit Combined with Destination Bit	Microsoft® Name for Pattern ROP
00	0	BLACKNESS 00000042	0	BLACKNESS 00000042
90	~S • ~D	NOTSRCERASE 001100A6	~P • ~D	– 000500A9
50	~S • D	– 00220326	~P • D	– 000A0329
D0	~S	NOTSRCCOPY 00330008	~P	– 000F0001
09	S • ~D	SRCERASE 00440328	P • ~D	– 00500325
0B	~D	DSTINVERT 00550009	~D	DSTINVERT 00550009
59	S x D	SRCINVERT 00660046	P x D	PATINVERT 005A0049
DA	~S + ~D	– 007700E6	~P + ~D	– 005F00E9
05	S • D	SRCAND 008800C6	P • D	– 00A000C9
95	S = D	– 00990066	P = D	– 00A50065
06	D	– 00AA0029	D	– 00AA0029
D6	~S + D	MERGEPAINT 00BB0226	~P + D	– 00AF0229
0D	S	SRCCOPY 00CC0020	S	PATCOPY 00F00021
AD	S + ~D	– 00DD0228	P + ~D	– 00F50225
6D	S + D	SRCPAINT 00EE0086	P + D	– 00FA0089
0E	1	WHITENESS 00FF0062	1	WHITENESS 00FF0062

[a] Where: S = source, P = pattern, D = destination, + = OR, • = AND, ~ = NOT(inversion), x = Exclusive-OR.

A.4 BitBLT Color Expansion of Graphics Data

When Extension register bit GR30[7] is 1, the source input to the ROPS of Extension register GR32 is not actual data from the source area; instead, it is data that has been color-expanded. Since the source area is a monochrome image that has a single bit per pixel, substantial performance benefits are possible, especially when the source image is expanded to 16 or 24 bits. Color expansion can be used when what appears in a destination area is either of the following:

- A single foreground color by itself
- A single foreground color along with a single background color

The CL-GD755X supports color expansion of data that is in either 8-, 16- or 24-bit-per-pixel display modes. The source can be either an 8 × 8 pattern (display memory only), or monochrome data from either display memory or system memory. When the source data is in display memory:

- The starting address of the data must be located on a 4-byte boundary.
- Except, for a pattern of 8-pixels × 8-pixels, the starting address of the data must be on an 8-byte boundary.

When the data in the source area is a:

- 0, the background color is written into corresponding byte(s) of the destination area, or no change takes place to the destination area (that is, the data written to the destination area is transparent).
- 1, the foreground color is written into corresponding byte(s) of the destination area.

For color expansion, the destination must be the “on-screen” portion of display memory, and the direction must be “increment”. Any ROP can be used; however, SRCCOPY is most common for color expansion.

The sense of monochrome data can be inverted by programming CR33[1] to 1, which allows complete foreground/background 24-bit/pixel color expansion in two passes. Also, for 24-bit/pixel color expansion, transparency must be enabled.

The most-significant bit of the first source byte is expanded into the ROP source data for the first pixel of the destination (however, clipping can keep some data from being wrirtten). Depending on the contents of GR30[5:4], it is expanded to 1, 2 or 3 bytes. Table A-3 shows the expansion registers.

Table A-3. Color Expansion Registers

GR30[5:4]	Width	Background Expansion ('0' in Source Area)		Foreground Expansion ('1' in Source Area)		
		GR0	GR10	GR1	GR11	GR13
00	8-bit	Color	Don’t care	Color	Don’t care	Don’t care
01	16-bit	Low byte	High byte	Low byte	High byte	Don’t care
10	24-bit	*Transparency must be enabled*		Blue	Green	Red
11	--	Reserved		Reserved		

The next bit of source data is processed for the next 1, 2 or 3 bytes of destination, and so on, until the ‘width’ bytes of the scanline in the destination area have been processed. GR33[0] controls the source data granularity for color-expanded system-to-screen BitBLTs. When GR33[0] is 1, unused source data is discarded to the end of the current DWORD. When GR33[0] is 0, unused source data is discarded to the end of the current byte.

The destination address is modified by the destination pitch. The source pitch is ignored, since the source is taken to be a linear string of bytes. The next byte of source is the first 8 pixels for the next scanline.

A.4.1 BitBLT Color Expansion of Graphics Data (with Transparency)

When GR30[3] is 0, the pixels corresponding to ‘0’s in the source area are written with the contents of the background color registers into the destination area. When GR30[3] is 1, a color expansion with transparency is enabled. The pixels corresponding to ‘0’s in the source area are not written into the destination area. When transparency is enabled, the background color registers are ignored, and the foreground color registers are used as shown in Table A-3.

Table A-4. Color Expansion Registers with Transparency

GR30[5:4]	Width	Foreground Expansion (‘1’ in Source Area)		
		GR1	GR11	GR13
00	8-bit	Color	Don’t care	Don’t care
01	16-bit	Low byte	High byte	Don’t care
10	24-bit	Blue	Green	Red
11	--	Reserved		

When color expansion is used with an opaque foreground and a transparent background (analogous to Extended Write mode 4), the transparency feature must be used, and the transparent color must be set to the background color. The registers used for foreground color expansion with transparency are the same as the color expansion registers with ‘1’ in the source.

A.5 Clipping

When the Destination Write mask, GR2F[2:0], is set to any value other than ‘000’, the first *n* pixels of each scanline of the destination are not written. This “clipping” is typically used with “pattern fills” with or without color expansion. The destination address must point to the first pixel that is skipped, and the BitBLT ‘width’ must be programmed to include the pixels that are skipped.

Clipping avoids re-aligning the source, when beginning at other than the first pixel of the color expanded source.

For packed 24-bit modes, the Packed-24 Byte Write mask field, GR2F[4:3], can be added to the Destination Write mask, GR2F[2:0], to get a 5-bit field that controls the scanline clipping. In this case, the value is taken as a byte count rather than a pixel count. This allows skipping up to 31 bytes (seven pixels) of each scanline of the destination area.

The System-to-Screen DWORD Pointer, GR2F[6:5], can be used to skip bytes of the source field for system-to-screen BitBLTs. This allows unaligned source fields to be transferred without the overhead of unaligned bus cycles. This field selects the initial byte of the first DWORD of each scanline. When this field is set to a non-zero value for color-expansion system-to-screen BitBLTs, Source Data Granularity for DWORDs must be enabled, GR33[0] is set to 1.

A.6 BitBLT Pattern Fills

The CL-GD755X BitBLT engine has provisions for filling a destination area with a repeating pattern. The pattern fill size is 8 pixels × 8 pixels, chosen for compatibility with Microsoft Windows. Pattern fill replication can be done either with or without color expansion, as explained in Section A.4.

When Extension register bit GR30[6] is 1, data in the source area is defined as an array of 8 pixels × 8 pixels. This source area data array is copied repeatedly to the destination area, with color expansion if necessary. For the same scanline, the same 8 pixels of source data are used repeatedly. The source pitch is ignored.

As indicated in Table A-5, the operating mode determines the number of data bytes in the source area pattern. The starting address boundary of a source pattern is equal to the number of bytes in the source pattern.

For pattern fill, the source address is incremented until the pattern fill completes and the address returns to the source start address, so that the pattern is used over and over.

Table A-5. Pattern-Fill Data Size

Operating Mode	Number of Data Bytes in Source Area Pattern	Starting Address Boundary
Color Expansion Enabled (That is, Extension registers GR30[7] = 1 *and* GR30[6] = 1.)	8 bytes of monochrome data for 64 pixels	8 bytes
8-bit pixels	64 bytes of color data for 64 pixels	64 bytes
16-bit pixels	128 bytes of color data for 64 pixels	128 bytes
24-bit pixels	256 bytes of color data plus 8 bytes of padding for each scanline, repeated 8 times	256 bytes

A.6.1 Pattern Vertical Preset

The CL-GD755X BitBLT engine supports pattern vertical preset for pattern fills. The low-order three bits of the source-start address, GR2C[2:0], select the scanline used for the first, or only, scanline. This occurs because the pattern must be aligned as shown in Table A-5, which makes it possible to force vertical alignment of the pattern. This vertical alignment is useful for filled rectangles and filled polygons.

A.6.2 Patterned-Polygon Fills

The CL-GD755X has provisions for patterned-polygon fills with color expansion. The polygon to be filled is decomposed into a series of single scanlines, each of which is filled with a single BitBLT. The first operation reads all 8 bytes of monochrome pattern. Then, as long as there are no writes to the source-start address registers or to the BitBLT Mode register, subsequent operations use the previous loaded source data, skipping the read cycle. Additionally, the Y offset, initially set by the three low-order bits of the source-start address, GR2C[2:0], are incremented modulo eight at the end of each operation. This forces each scanline to start one byte further into the pattern than the previous scanline. The polygon can be filled with a series of single-scanline fills that change only the destination-start address, left-edge pixel clipping, and BitBLT width. The normal operation precedes from top to bottom. When the operation precedes from bottom to top, the pattern is vertically flipped.

A.6.3 Solid-Color Fills

When GR33[2] is set to 1, solid-color fills are enabled. However, to accomplish solid-color fills, the following functions must also be set:

- Color Expansion enabled, GR30[7] = 1
- 8 x 8 Source Pattern Copy enabled, GR30[6] = 1
- Transparency disabled, GR30[3] = 0
- BitBLT Source is display memory, GR30[2] = 0

The contents of the foreground registers are written to the destination rectangle. Any expansion width can be used. This functions precisely as though the pattern is all '1's, except the pattern isn't read.

For packed-24 modes, the solid-color fill can be used to implement a full foreground/background color expansion. The destination block can be set to the background color, and then the foreground color can be written using the monochrome pattern (with transparency enabled).

A.7 BitBLT Direction

The direction of the progression of the BitBLT operation must be properly programmed when, as shown in Figure A-3, the source area and destination area overlap in display memory. In this situation, if a BitBLT operation were to start at the upper-left corner of the source and at the upper-left corner of the destination, then the source area data that is within the overlap area would be overwritten before being used.

To ensure that source area data are not overwritten prior to being used, Extension register bit GR30[0] must be set to 1. This setting decrements the source and destination addresses, which reverses the direction in which the BitBLT operation progresses. As a result, processing of data bytes for both the source and the destination starts at the lower-right corner, with data bytes processed right to left and bottom to top. This setting therefore guarantees that data bytes of the source are used prior to being changed. (With this setting, the start addresses in the source and destinations areas are the highest addresses, and not the lowest.)

NOTE: BitBLTs with color expansion must never be programmed for reverse direction.

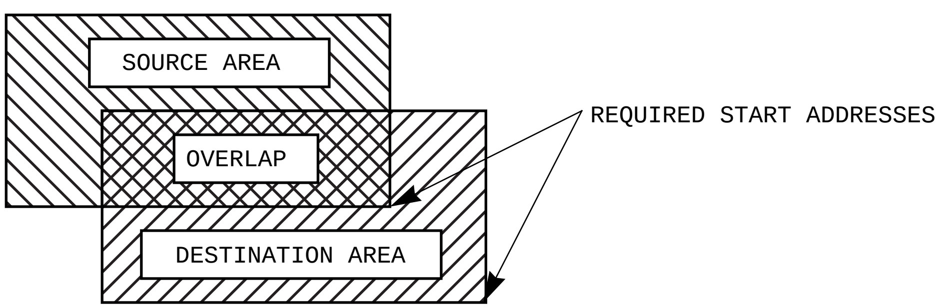


Figure A-3. BitBLT Start Addresses When Source and Destination Area Overlap

A.8 System-to-Screen BitBLTs

When Extension register bit GR30[2] is 1, the source of the BitBLT is system memory. In this case, the CPU must perform the bus transfers, since the CL-GD755X is never a bus master. The CPU must execute DWORD transfers. The address data provided by the CPU is not used, but it must be in the range being decoded as display memory.

For system-to-screen BitBLTs not involving color expansion, up to three bytes of the last transfer for each scanline is ignored (depending on width). The next scanline begins with the next DWORD transfer.

For system-to-screen BitBLTs involving color expansion, up to seven bits of the last (partially used) byte is ignored at the end of each scanline. Unused bytes are used at the beginning of the next scanline.

Extension register bit GR30[0] controls how the skipped data is handled. When GR30[0] is:

0, the unused bits of the current byte are skipped as described above.

1, the unused bits of the current DWORD are skipped at the end of each scanline. That is, up to seven bits of the current byte and the next three whole bytes are discarded. The next scanline begins with the first bit of the next DWORD. This is useful for keeping DWORD transfers aligned, when the monochrome source image must be clipped.

A.9 BitBLT Start, Pause and Reset Controls

Once the parameters have been loaded into the registers, the BitBLT can be started. It is also possible to pause any system-to-screen BitBLT and to unconditionally reset the BitBLT engine. These functions are controlled by bits in Extension register GR31.

A.9.1 BitBLT Start

To start a BitBLT, program Extension register bit GR31[1] to 1. Any required writes to DRAM must occur before the BitBLT registers are programmed. This process allows the write buffers to empty before the BitBLT operation starts. While the BitBLT is in progress, the CL-GD755X display memory and BitBLT registers (except GR31) must not be accessed for read or write.

As long as the BitBLT is in progress, the BitBLT Status bit GR31[0] is 1. When the BitBLT has completed, bit GR31[0] changes to 0. Monitoring BitBLT Status bit GR31[0] is the most straightforward way to synchronize the host CPU operation with the BitBLT engine.

A.9.2 BitBLT Automatic Start

The CL-GD755X has an automatic start capability for the BitBLT engine. By using double-buffering registers, the CL-GD755X allows the parameters for BitBLT ($n + 1$) to be loaded while BitBLT n is taking place. Consequently, when BitBLT n completes, BitBLT ($n + 1$) can begin without host CPU intervention, as long as the BitBLT buffer ($n + 1$) registers have been loaded. As a result, a high degree of parallelism between the host CPU and the CL-GD755X is achieved.

The automatic start is enabled by programming Extension register bit GR31[7] to 1. The automatic start for the BitBLT engine occurs when:

- The BitBLT engine is not busy or another BitBLT has just completed, GR31[0] is 0, **and**
- The BitBLT buffer registers listed in Table A-6 are loaded with new BitBLT parameters. Destination-start address register GR2A[5:0] must be the last buffer register written. Writing to GR2A[5:0] either:
  - — Starts a new BitBLT immediately, when the BitBLT engine is not busy.
  - — Posts a new start, which starts as soon as the previous BitBLT is completed, GR31[0] is set to 0.

To determine when the double-buffered registers are available to be loaded with a new set of parameters for the next BitBLT, the host CPU must monitor bit GR31[4] .

- When bit GR31[4] is 0, the buffered registers listed in Table A-6 can be loaded.
  - — To avoid starting a BitBLT with an incomplete set of parameters, the last address to be written must be the destination-start address of Extension register GR2A[5:0].)
  - — Those registers that have not changed their contents from the previous set of parameters need not be rewritten.
- When bit GR31[4] is 1, the buffered registers contain new BitBLT data that has not yet been used.

A.9.3 BitBLT Pause

A system-to-screen BitBLT can be interrupted without loosing the data. When GR31[5] is set to 1, a BitBLT in progress stops after the next write, and display memory writes from the CPU are redirected to the address specified on the bus. Reads are not permitted, and would return incorrect data.

By resetting GR31[5] to 0, the BitBLT resumes at the place it stopped. Subsequent writes to display memory are directed to the BitBLT engine. While the pause is in effect, Reset status GR31[3] remains a 1.

A.9.4 BitBLT Reset

To unconditionally stop the current operation and reset the BitBLT engine, set Extension register GR31[2] to 1, which forces bits GR31[3] and GR31[0] to 0 and stops the operation after the next write. This operation can result in partial pixels being written. GR31[2] must be reset to 0 before another BitBLT can be programmed.

Setting GR31[2] to 1 does not change the buffered-register status bit, GR31[4]. When GR31[4] is set to 1 by a pending auto-start BitBLT, it can be reset to 0 only be resetting GR31[7] to 0.

NOTE: A BitBLT operation that is reset *cannot* be resumed.

A.10 Memory-Mapped I/O

All the registers listed in Table A-6, except SR2, can be written as memory locations. This is called memory-mapped I/O. This makes it possible to load the registers quickly, because they can be written four-per-access.

Memory-mapped I/O is enabled, when SR17[2] is set to 1. The registers are mapped beginning at address B800:0. The CL-GD755X can be programmed to place the memory-mapped I/O area at the last 256 bytes of the linear address space. GR6[3:2] must be programmed to '01'. When memory-mapped I/O is enabled, only GR31 is accessible as an I/O location.

A.11 Complete BitBLT Register Field Listing

Table A-6 lists every register associated with the BitBLT engine. All registers in Table A-6 are double buffered, except for Extension register GR31, Sequencer register SR2, and the Reserved registers.

Table A-6. Listing of BitBLT Engine Register Fields

Location of BitBLT Register			Memory-Mapped I/O Offset for BitBLT (hex)	How BitBLT Register Is Used	BitBLT Register Size (bits)	Register Modified During BitBLT
Sequencer Register	Graphics Controller Register	Extension Register				
	GR0		00	Background Color byte 0	8	No
	GR1		04	Foreground Color byte 0	8	No
		GR10	01	Background Color byte 1	8	No
		GR11	05	Foreground Color byte 1	8	No
		GR13	06	Foreground Color byte 2	8	No
		GR20	08	Width byte 0	8	Yes
		GR21	09	Width byte 1	5	Yes
		GR22	0A	Height byte 0	8	No
		GR23	0B	Height byte 1	3	No
		GR24	0C	Destination Pitch byte 0	8	No
		GR25	0D	Destination Pitch byte 1	5	No
		GR26	0E	Source Pitch byte 0	8	No
		GR27	0F	Source Pitch byte 1	5	No
		GR28	10	Destination Start byte 0	8	Yes
		GR29	11	Destination Start byte 1	8	Yes
		GR2A	12	Destination Start byte 2	6	Yes
		GR2B	13	Reserved	–	–
		GR2C	14	Source Start byte 0	8	Yes
		GR2D	15	Source Start byte 1	8	Yes
		GR2E	16	Source Start byte 2	6	Yes
		GR2F	17	Write Mask	7	No
		GR30	18	BitBLT mode	8	No
		GR31	40	Start / Status	6	–
		GR32	1A	Raster Operation	8	No
		GR33	1B	BitBLT Mode Extensions	3	–
SR2			–	Plane Mask	8	No

A.12 BitBLT Color Expansion of Text Data in Graphics Write Modes

With non-color-expanded graphics write modes, each individual character that makes up a text string is copied from system memory, and desired color attributes are applied to each individual character. The monochrome image of the text string is arranged in system memory by scanlines.

In contrast, with color-expanded graphic write modes, an entire text string is copied from system memory. By applying the desired color attributes to the entire string, faster text-write operations are achieved.

Example: Assume a destination area is 150 pixels $\times$ 25 scanlines, with 8 bits per pixel, and the destination pitch is 1024 pixels. In this case, the registers must be programmed as indicated in Table A-7.

Table A-7. Typical Register Settings for Color Text Expansion

Field	Graphics Controller Registers	Extension Registers	Field Contents (decimal)	How Field Contents Are Calculated
Background Color	GR0	GR10	'Don't care'	Not applicable
Foreground Color	GR1	GR11, GR13	'Don't care'	Not applicable
Width		GR20, GR21	149	150 pixels – 1 pixel = 149 pixels
Height		GR22, GR23	24	25 scanlines – 1 scanline = 24 scanlines
Destination Pitch		GR24, GR25	1024	Not applicable
Source Pitch		GR26, GR27	Not applicable	(System memory)
Destination Start		GR28, GR29, GR2A	'Don't care'	Not applicable
Source Start		GR2C, GR2D, GR2E	Not applicable	(System memory)
Write Mask		GR2F	0	No Clipping
BitBLT Mode		GR30	84 (hex)	Color Expansion, Source = System memory
Start		GR31	02 (hex)	Set bit GR31[1] to 1
Raster Operation		GR32	0D (hex)	Source copy. (Refer to information on 0Dh in Table A-2)
Mode Extension		GR33	0	Discard partial bytes

NOTES:

1) This example shows register settings for a BitBLT operation that transfers data (which in this case, is a bitmap) from system memory to the display screen. In this case, after registers are programmed, data must be transferred from system memory by the host CPU.

   a) The first double-word write transfers the image for pixels 0 to 31.

   b) The second double-word write transfers the image for pixels 32 to 63.

   c) The third double-word transfers the image for pixels 64 to 95.

   d) The fourth double-word transfers the image for pixels 96 to 127.

   e) The data sent in the fifth double-word transfer is shown in Figure A-4.

2) When background pixels are not to be written, Extension register GR30[3] must be set to 1.

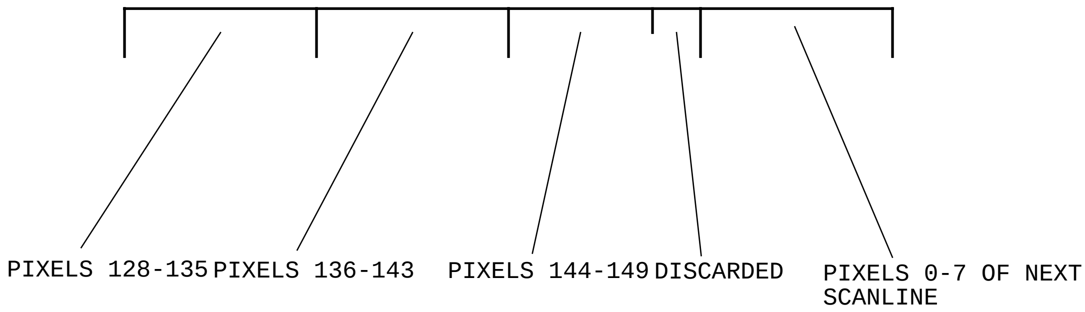


Figure A-4. Color Expand Transfer

A total of 119 DWORD transfers are required. The last byte of the last transfer is discarded.

Appendix B

Hardware Cursor

B.1 Introduction

The CL-GD755X is capable of supporting a 32 × 32 or 64 × 64 hardware-driven cursor in either 16-color planar display modes or in 256, 32K, 64K or 16M color packed-pixel graphics display modes. The hardware cursor (also called a ‘mouse pointer’) replaces a software mouse pointer commonly used by GUI (graphics user interface) applications. The hardware cursor eliminates the need for application software to save and restore the screen data as the on-screen cursor position changes.

After the application software initializes the hardware cursor, the application software must only update the cursor position (X,Y) to move the cursor on the screen. Compared to a software cursor, the hardware cursor offers a smoother-moving mouse pointer with improved overall performance.

B.2 Hardware Cursor Operation

Multiple hardware-cursor patterns can be loaded in the top 8 Kbytes of display-memory area. The number of cursor patterns that can be loaded in the display memory at one time is either 32 (for 32 × 32-bit cursors) or 8 (for 64 × 64-bit cursors). The selection of a 32 × 32 or 64 × 64 hardware cursor is made with Extension register SR12[2]. Of the cursor patterns that are loaded in display memory, application programs select one as the active cursor pattern by programming Extension register SR13[4:0].

Each pixel in the hardware cursor consists of two bits which define Cursor Plane 0 and Cursor Plane 1. The following table shows how the two bits determine the displayed data of each hardware-cursor pixel.

Table B-1. Hardware Cursor Bits

Cursor Plane 0	Cursor Plane 1	Displayed Data of Hardware Cursor Pixel
0	0	Transparent cursor pixel
0	1	VGA display data inverted
1	0	Cursor color 0, background
1	1	Cursor color 1, foreground

Cursor color 0 and Cursor color 1 are supplied by the CL-GD755X’s internal color-palette two extra LUT (Lookup Table) locations, which are mapped into the existing LUT indexes 00h, which selects the background color, and 0Fh, which selects the foreground color. As a result, the hardware cursor colors are independent of the existing LUT indexes 00h through FFh. The extra LUT locations are enabled by setting SR12[1] to 1.

B.2.1 Hardware Cursor Selection

The hardware cursor shares the upper 16 Kbytes of Graphics Memory with the hardware icons. The hardware cursor is enabled when SR12[0] is set to 1. The size of the hardware cursor pattern is determined by SR12[2]; when:

- SR12[2] is 0, the power-on default, a 32 x 32 cursor pattern is selected.
- SR12[2] is 1, a 64 x 64 cursor pattern is selected.

B.2.2 Hardware Cursor Horizontal and Vertical Position

The hardware cursor position is controlled by programming the hardware cursor horizontal (X) position and hardware cursor vertical (Y) position. The SRX[4:0] bits of the Sequencer Index register are the index into the horizontal and vertical position registers. The SRX[7:5] bits of the Sequencer Index register are the least-significant 3 bits of the 11-bit (horizontal or vertical) value. The SR10[7:0] bits are the most-significant 8 bits of the 11-bit horizontal value, and the SR11[7:0] bits are the most-significant 8 bits of the 11-bit vertical value.

NOTE: Either the hardware cursor or hardware icon can be moved at any one time, but not both. When register SR2A[7] is 0, the horizontal and vertical position of the hardware cursor is updated and displayed.

For normal operation, the 11-bit cursor X-location and 11-bit Y-location values are programmed by writing two 16-bit I/O writes each to:

- Sequencer register SRX (3C4h) and Extension register SR10 (3C5h), for the horizontal position
- Sequencer register SRX (3C4h) and Extension register SR11 (3C5h), for the vertical position

Moving the cursor always requires writing to both the horizontal and vertical position registers. The horizontal position is written first, but it does not take effect until the vertical position is updated.

Horizontal Position:

Register SR10[7:0] defines in character clocks the coarse horizontal (X) pixel offset of the graphics hardware cursor. The Sequencer register index bits SRX[7:5] define the fine horizontal position of the hardware cursor. For all 8-bit text modes, non-expanded graphics modes, and text modes which do not require the fourth expansion bit, the entire 11-bit cursor horizontal position can be written in a single 16-bit I/O write as follows:

- Extension register bits SR2A[6] and SR2E[0], which are cleared to 0, are ignored.
- The offset must be placed in AX[15:5].
- AX[4:0] must be ‘10000’.
- DX must be 3C4h.

When 10, 30, 50...F0 is written to 3C4h without writing to 3C5h (a byte write), then a read of 3C4h returns the previously stored three bits of the cursor or icon position.

11-Bit HW-Cursor Horizontal-Position Registers

SR10[7:0]		SRX[7:5]	
10	3	2	0

Horizontal Position – Expanded Graphics Modes:

For expanded graphics display modes (for example, if a 640 × 480 graphics display mode is expanded for use on an 800 × 600 flat-panel display, or if an expanded graphics display mode uses 10-dot character clocks) the hardware cursor requires 12 bits to define the horizontal position:

- SR10[7:0] are the most-significant 8 bits of the 11/12-bit horizontal position.
- Sequencer register bits SRX[7:5] are the least-significant 3 fine-horizontal position bits, defined in dot clocks.
- Extension register bit SR2E[0] extends the hardware-cursor fine-horizontal position by one more-significant bit for horizontally expanded graphics modes. This extra bit is valid only with a 10-dot character clock.

12-Bit HW-Cursor Horizontal-Position Registers

SR10[7:0]		SR2E[0]	SRX[7:5]	
11	4	3	2	0

Vertical Position:

The value in SR11[7:0], together with the Sequencer register index bits SRX[7:5] that are used to access it, defines in scanlines the coarse vertical (Y axis) scanline offset of the graphics hardware cursor. The entire 11-bit cursor vertical position can be written in a single 16-bit I/O write as follows:

- The offset must be placed in AX[15:5].
- AX[4:0] must be ‘10001’.
- DX must be 03C4h.

When 11, 31, 51...F1 is written to 3C4h without writing to 3C5h (a byte write), then a read of 3C4h returns the previously stored three bits of the cursor or icon vertical position.

11-Bit HW-Cursor Vertical-Position Registers

SR11[7:0]		SRX[7:5]	
10	3	2	0

B.2.3 Hardware-Cursor Vertical-Expansion Tracking

Since the hardware cursor is an independently controlled image on the screen, it does not automatically track the position of an expanded image or field. To retain the relationship of the hardware cursor to a displayed image as it is expanded, a special control bit is available. By setting SR2C[7] to 1, the selected hardware cursor automatically tracks hardware vertical expansion.

B.2.4 64-Bit Access for Hardware Cursors in VGA Modes

In standard VGA modes, the host CPU can normally access only 32 bits of display memory. In order to allow the cursors (and icons) to be loaded in to the full 64-bit width of the display memory in standard VGA modes, Extension register bit SR2D[7] must be used. When linear addressing is selected (CR36[7] = 1) and SR2D[7] is set to 1, the host CPU can access the full 64-bit width of display memory in VGA modes. In extended VGA modes, the full 64 bit width of display memory is always available.

B.3 Hardware Cursor Register Map

The following table shows CL-GD755X registers that must be programmed to use hardware cursor:

Table B-2. Registers for Programming Hardware Cursor

Function	Program:		
	Register	Value	I/O Port Address
Select Hardware Cursor (or Icon)[a] Position Update	SR2A[7]	0	3C5h
Enable the Hardware Cursor	SR12[0]	1	3C5h
Enable CPU access to LUT extended colors	SR12[1]	1	3C5h
Select a 32 × 32 Hardware Cursor	SR12[2]	0	3C5h
Select a 64 × 64 Hardware Cursor	SR12[2]	1	3C5h
Select a Hardware Cursor pattern of 32 × 32	SR13[4:0]	31...0	3C5h
Select a Hardware Cursor pattern of 64 × 64	SR13[4:2]	7...0	3C5h
Select the Hardware Cursor horizontal position	SRX[7:5] / SR10[7:0], 30, 50, 70, 90, B0, D0, F0	2047...0	3C4h / 3C5h
Hardware Cursor horizontal-fine-position extension (for expanded graphics and 10-dot character clocks)	SR2E[0]	0 =Non-Expanded 1 = Expanded	3C5h
Select Hardware Cursor vertical position	SRX[7:5] / SR11[7:0], 31, 51, 71, 91, B1, D1, F1	2047...0	3C4h / 3C5h
Hardware Cursor vertical-expansion tracking enable	SR2C[7]	1	3C5h
CPU Access to Display Memory for Hardware Cursor/ Icons Limited to 32-bits wide in VGA Modes	SR2D[7]	0	3C5h
CPU Access to Display Memory for Hardware Cursor/ Icons is Set to 64-bits wide in VGA Modes	SR2D[7]	1	3C5h
Display Memory Access for Hardware Cursor/Icons is 64 bits wide in all Modes	SR2D[6]	0	3C5h
Display Memory Access for Hardware Cursor/Icons is 32 bits wide in VGA Modes and 64 bits wide in Extended VGA Modes	SR2D[6]	1	3C5h

[a] The position of both the hardware cursor or icons can be updated, but not simultaneously. When SR2A[7] = 1, the hardware icon position is updated.

B.4 Hardware Cursor Memory Map

The following table shows the location of the hardware cursor memory map within the display memory.

Table B-3. Hardware Cursor Memory Map in 2-Mbyte Display Memory

Address[a]	32 × 32 Hardware Cursor	64 × 64 Hardware Cursor
3FF80–3FFFF	Cursor Map #28 — Cursor Map #31	Cursor Map #7
3FF00–3FF7F	Cursor Map #24 — Cursor Map #27	Cursor Map #6
3FE80–3FEFF	Cursor Map #20 — Cursor Map #23	Cursor Map #5
3FE00–3FE7F	Cursor Map #16 — Cursor Map #19	Cursor Map #4
3FD80–3FDFF	Cursor Map #12 — Cursor Map #15	Cursor Map #3
3FD00–3FD7F	Cursor Map #8 — Cursor Map #11	Cursor Map #2
3FC80–3FCFF	Cursor Map #4 — Cursor Map #7	Cursor Map #1
3FC00–3FC7F	Cursor Map #0 — Cursor Map #3	Cursor Map #0

[a] Assumes each address is 64 bits wide.

B.4.1 The 32 × 32 Hardware Cursor Pattern

Each 32 × 32 hardware cursor pattern requires 256 bytes (128 bytes per cursor plane × 2 cursor planes), which allows 32 cursor patterns to be loaded into the upper 8 Kbytes of display memory. The hardware cursor pattern that is active is selected by programming Extension register bits SR13[4:0]. The 32 × 32 hardware cursor pattern data for cursor plane 0 and cursor plane 1 is stored in the display memory as shown in the following figure:

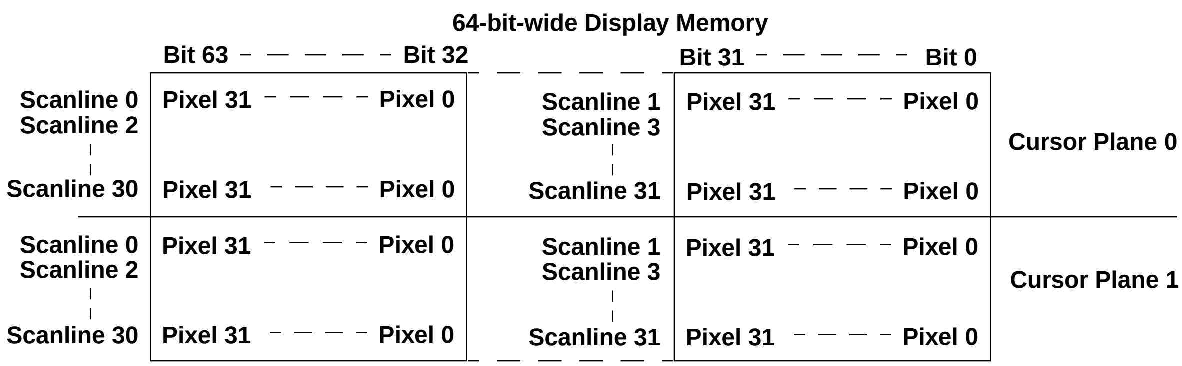


Figure B-1. Mapping of 32 × 32 Hardware Cursor

B.4.2 The 64 × 64 Hardware Cursor Pattern

Each 64 × 64 cursor pattern requires 1024 bytes (512 bytes per cursor plane × 2 cursor planes), which allows 8 cursor patterns to be loaded into the upper 8 Kbytes of the display memory. Extension register bits SR13[4:2] select the active hardware cursor pattern.

The 64 × 64 hardware cursor pattern data from cursor plane 0 and cursor plane 1 is loaded into display memory one cursor scanline at a time; 16 bytes written across the two logical display planes (4-words-per-plane). The same action occurs until all 64 scanlines from cursor plane 0 and cursor plane 1 are loaded into the system memory, as shown in the following figure:

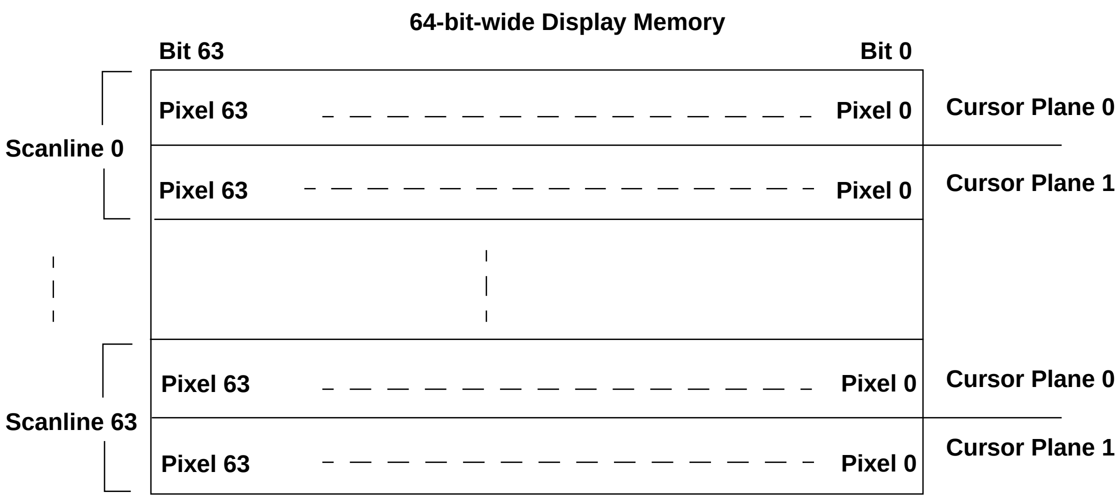


Figure B-2. Mapping of 64 × 64 Hardware Cursor

Appendix C

Hardware Icon

C.1 Introduction

A hardware icon (that is, a hardware-driven icon window), as compared to a software icon, eliminates the need for application software to save and restore the screen data as the icon position changes.

The hardware icon data is located in the second 8 Kbytes at the top of available display memory. The number of icon patterns that can be loaded in the display memory at one time is eight 64 × 64-bit icons (of which four can be simultaneously displayed).

In various operational modes, the CL-GD755X is capable of simultaneously supporting four hardware icons with the following features:

- One to four icons displayed simultaneously on the screen.
- The default icon size of 64 × 64 pixels can be doubled vertically and/or horizontally to the following sizes:
  - 64 × 128 pixels
  - 128 × 64 pixels
  - 128 × 128 pixels
- Up to eight icon maps may be defined in display memory for use and are associated with the displayed icons in two ways:
  - Two icon maps are used for each hardware icon.
  - All eight icon maps are used with the first hardware icon.
- A single location, known as a ‘hot spot’, is used to define the screen position of all of the icons. The hot spot defines the upper left-hand corner of the first icon. All subsequent icons are left-aligned vertically down from the first icon. The hot spot is positioned at a resolution of one pixel.
- Each of the four icons is independently selected and controlled.
- The hardware icons have the second-most display priority in the video sub-system. Only the hardware cursor appears on top of the hardware icon.

C.1.1 64-Bit Access for Hardware Icons in VGA Modes

In standard VGA modes, the host CPU can normally access only 32 bits of display memory. In order to allow the icons (and cursors) to be loaded in to the full 64-bit width of the display memory in standard VGA modes, Extension register bit SR2D[7] must be used. When linear addressing is selected (CR36[7] = 1) and SR2D[7] is set to 1, the host CPU can access the full 64-bit width of display memory in VGA modes. In extended VGA modes, the full 64 bit width of display memory is always available.

C.1.2 Hardware Icon Selection

The hardware icon memory model and programming model are very similar in usage to that of the hardware cursor. The memory used to define the icon maps is located just below the hardware cursor area (which is at the top of the available display memory space), and the icon memory map is fixed by the CL-GD755X. The hardware icons are individually enabled by the following four Extension registers:

- Icon #0 - SR2A[5:0]
- Icon #1 - SR2B[5:0]
- Icon #2 - SR2C[5:0]
- Icon #3 - SR2D[5:0]

The individual bits control the following functions:

- SR2_[0] = 1, Icon Display Enabled
- SR2_[1] = 0, 4-Color Display Mode
- SR2_[1] = 1, 3-Colors-and-Transparency Display Mode
- SR2_[2] = 1, Blink Enable: one-half the text cursor blink rate
- SR2_[3] = 1, Horizontal Pixel Doubling
  - — Each pixel is replicated horizontally for a total of 128 pixels
  - — The Icon is expanded to the right on the display
- SR2_[4] = 1, Vertical Pixel Doubling
  - — Each scanline is replicated vertically for a total of 128 scanlines
  - — The expanded Icon expands down, forcing all lower Icons down
- SR2_[5] = 0, Memory Map 0 is Selected
- SR2_[5] = 1, Memory Map 1 is Selected

C.1.3 Hardware Icon Color

Each pixel in the hardware icon consists of two bits which define Icon Plane 0 and Icon Plane 1. Table C-1 shows how these two bits determine the displayed state of each icon pixel.

Table C-1. Hardware Icon Pixel Bits

Icon Plane 0	Icon Plane 1	Displayed Data of Hardware Icon Pixel
0	0	Icon color 0 (Transparent)
0	1	Icon color 1
1	0	Icon color 2
1	1	Icon color 3

The icon colors 0 through 3 are supplied by the internal color palette's four extra lookup table (LUT) locations, which are mapped into the existing LUT indexes 03h, 04h, 05h, and 06h, respectively. As a result, the hardware icon colors are independent of the existing LUT indexes 00h through FFh.

- When Extension register SR12[1] is set to '1', these four additional palette locations are enabled and are accessible the same way as their standard VGA counterparts.
- Clearing Extension register SR12[1] to '0' restores standard VGA operation to the palette locations.

C.1.4 Hardware Icon Horizontal and Vertical Position

A single location, known as a ‘hot spot’, is used to define the screen position of all of the icons. The hot spot defines the upper left-hand corner of the first selected icon. All subsequent icons are left-aligned vertically down from the first icon. The hot spot is positioned at a resolution of one pixel.

The hardware icon position is controlled by programming the hardware icon horizontal (X) position and hardware icon vertical (Y) position. The SRX[4:0] bits of the Sequencer Index register are the index into the horizontal and vertical position registers. The SRX[7:5] bits of the Sequencer Index register are the least-significant 3 bits of the 11-bit (horizontal or vertical) value. The SR10[7:0] bits are the most-significant 8 bits of the 11-bit horizontal value, and the SR11[7:0] bits are the most-significant 8 bits of the 11-bit vertical value.

For normal operation, the 11-bit icon X-location and 11-bit Y-location values are programmed by writing two 16-bit I/O writes each to:

- Sequencer register SRX (3C4h) and Extension register SR10 (3C5h), for the horizontal position
- Sequencer register SRX (3C4h) and Extension register SR11 (3C5h), for the vertical position

Moving the icon always requires writing to both the horizontal and vertical position registers. The horizontal position is written first, but it does not take effect until the vertical position is updated.

The hardware icons position is updated when Extension register SR2A[7] is set to 1. Since the hardware icons and cursors share the same horizontal and vertical position registers, the position of either one, but not both, can be modified at any one time.

Horizontal Position:

Register SR10[7:0] defines in character clocks the coarse horizontal (X) pixel offset of the graphics hardware icon. The Sequencer register index bits SRX[7:5] define the fine horizontal position of the hardware icon. For all 8-bit text modes, non-expanded graphics modes, and text modes which do not require the fourth expansion bit, the entire 11-bit icon horizontal position can be written in a single 16-bit I/O write as follows:

- Extension register bits SR2A[6] and SR2E[0], which are cleared to 0, are ignored.
- The offset must be placed in AX[15:5].
- AX[4:0] must be ‘10000’.
- DX must be 3C4h.

When 10, 30, 50...F0 is written to 3C4h without writing to 3C5h (a byte write), then a read of 3C4h returns the previously stored three bits of the icon position.

11-Bit HW-Icon Horizontal-Position Registers

SR10[7:0]		SRX[7:5]	
10	3	2	0

Horizontal Position – Expanded Graphics Modes:

For expanded graphics display modes (for example, if a 640 × 480 graphics display mode is expanded for use on an 800 × 600 flat-panel display, or if an expanded graphics display mode uses 10-dot character clocks) the hardware icon requires 12 bits to define the horizontal position:

- SR10[7:0] are the most-significant 8 bits of the 11/12-bit horizontal position.
- Sequencer register bits SRX[7:5] are the least-significant 3 fine-horizontal position bits, defined in dot clocks.
- Extension register bit SR2A[6] extends the hardware-icon fine-horizontal position by one more-significant bit for horizontally expanded graphics modes. This extra bit is valid only with a 10-dot character clock.

12-Bit HW-Icon Horizontal-Position Registers

SR10[7:0]		SR2A[6]	SRX[7:5]	
11	4	3	2	0

To program the horizontal position of the hardware icon:

1. **Place the icon hot spot (top left) at pixel n (with 0 as the first pixel).**
2. Compute the coarse horizontal position as follows:
   CHP = Integer [(n + (k – 1)) ÷ k] (where k = the number of dots in a character clock.)
3. Compute the fine horizontal position value as follows:
   FHP = Remainder [(n + (k – 1)) ÷ k]

Example: To place the icon at pixel 10, which is 11 pixels from the left (that is, n = 10), with an 8-dot character clock, as in graphics mode 12h (that is, k = 8):

1. **Set the coarse position = Integer [(10 + (8 – 1)) ÷ 8] = Integer [17 ÷ 8] = 2h.**
2. Set the fine position = Remainder [17 ÷ 8] = 1h.

Examples of programming for an 8-dot character clock are given in the table below:

Pixel Position	Horizontal Position Programmed (hex)	
	Coarse	Fine
0	0	7
1	1	0
2	1	1
3	1	2
4	1	3
5	1	4
6	1	5
7	1	6
8	1	7
9	2	0
10	2	1

There are only three cases of character clock width:

- 10 dots — used for text and graphics expansion on 800 × 600 LCD
- 9 dots — used for text on 800 × 600 and 1024 x 768 LCDs, when 720 dots are displayed
- 8 dots — used in all other cases

Vertical Position:

The value in SR11[7:0], together with the Sequencer register index bits SRX[7:5] that are used to access it, defines in scanlines the coarse vertical (Y axis) scanline offset of the graphics hardware icon. The entire 11-bit icon vertical position can be written in a single 16-bit I/O write as follows:

- The offset must be placed in AX[15:5].
- AX[4:0] must be '10001'.
- DX must be 03C4h.

When 11, 31, 51...F1 is written to 3C4h without writing to 3C5h (a byte write), then a read of 3C4h returns the previously stored three bits of the icon vertical position.

11-Bit HW-Icon Vertical-Position Registers

SR11[7:0]		SRX[7:5]	
10	3	2	0

Figure C-1 shows two examples the hot spot and vertical alignment of the icons as they appear on the display.

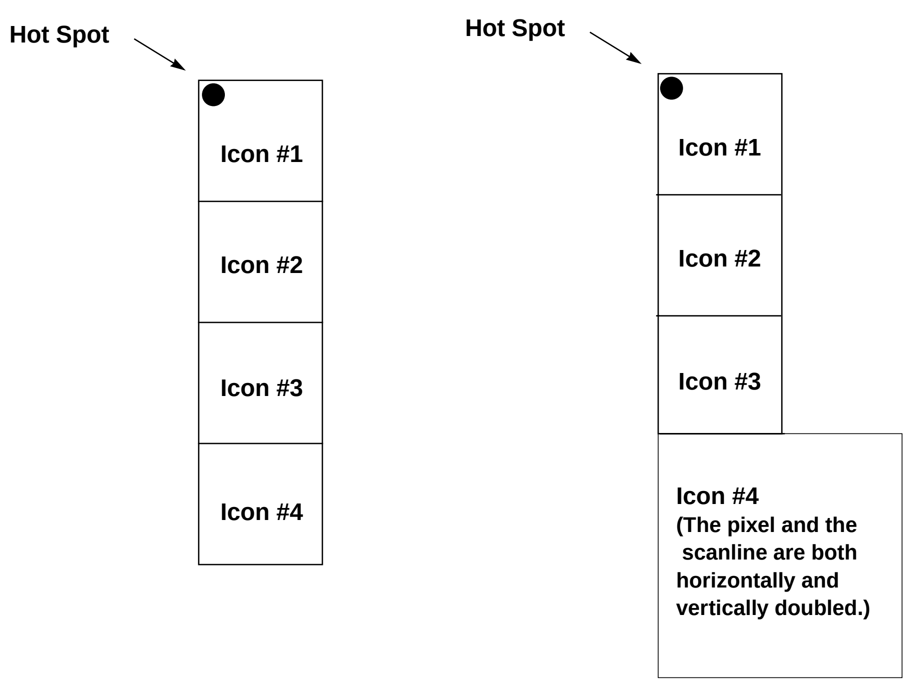


Figure C-1. Hot Spot and Vertical Alignment of the Icons

C.2 Hardware Icon Memory Map and Data Format

Table C-2 shows the location of the hardware icon memory maps within the display memory.

Table C-2. Hardware Icon Memory Map in 2-Mbyte Display Memory

Address	Hardware Icon Memory Map
3FB80–3FBFF	Icon #3 Map #1 or Icon #0 Map #7
3FB00–3FB7F	Icon #3 Map #0 or Icon #0 Map #6
3FA80–3FAFF	Icon #2 Map #1 or Icon #0 Map #5
3FA00–3FA7F	Icon #2 Map #0 or Icon #0 Map #4
3F980–3F9FF	Icon #1 Map #1 or Icon #0 Map #3
3F900–3F97F	Icon #1 Map #0 or Icon #0 Map #2
3F880–3F8FF	Icon #0 Map #1
3F800–3F87F	Icon #0 Map #0

NOTE: The above addresses assume each location is 64-bits wide.

The actual data that is stored in display memory to define the icon is written with two bits of data per pixel, as defined in Section C.1.3. The following figure shows how the 64-pixel-by-64-pixel icon is stored in 64-bit wide display memory.

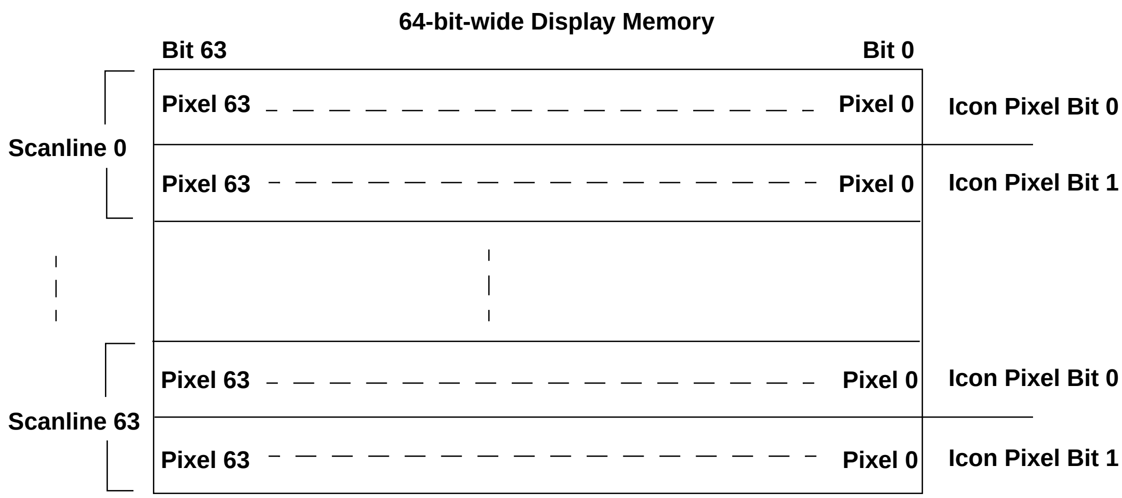


Figure C-2. Icon (64 Pixels × 64 Pixels) Stored in the Display Memory

C.2.1 Memory Map Option

According to the setting of Extension register SR2E[2:1], the eight CL-GD755X icon maps can be configured in two ways:

- Two maps per icon
- All eight maps available to icon #0

Extension register SR2E[2:1] controls this configuration option as follows:

- When this field is ‘00’, the corresponding map pair that is displayed is selected by SR2_[5], (where 2_ = 2A, 2B, 2C and/or 2D).
- When this field is non-zero:
  - — Only icon #0 can be used, and it has accesses to all eight memory maps. The non-zero number points to the corresponding map pair that is displayed as icon #0.
  - — Icons #1, #2, and #3 must be disabled, (SR2_[0] = 0; where 2_ = 2B, 2C and 2D).

The member of the map pair that is actually displayed is selected by the state of SR2A[5]. For example, if SR2E[2:1] = ‘11’ and SR2A[5] = ‘0’, then map #6 is displayed as icon #0. For memory map details, refer to Table C-2.

C.3 Hardware Icon Register Map

Table C-3 shows the CL-GD755X Extension register bits for programming the hardware icon:

Table C-3. Registers for Programming Hardware Icon

Function	Program:		
	Register	Value	I/O Port Address
Select Hardware Icon (or Cursor[a]) Position Update	SR2A[7]	1	3C5h
Enable the Hardware Icon #0,#1,#2,#3	SR2_[b][0]	1	3C5h
Display 4-color display modes	SR2_[b][1]	0	3C5h
Display 3-color-and-transparency mode	SR2_[b][1]	1	3C5h
Enable the Hardware Icon Blinking	SR2_[b][2]	1	3C5h
Enable Horizontal Pixel Doubling	SR2_[b][3]	1	3C5h
Enable Vertical Scanline Doubling	SR2_[b][4]	1	3C5h
Display Memory Map 0 Select	SR2_[b][5]	0	3C5h
Display Memory Map 1Select	SR2_[b][5]	1	3C5h
Enable CPU access to LUT extended colors	SR12[1]	1	3C5h
Select two Memory Maps for All Icons	SR2E[2:1]	00	3C5h
Select Eight Memory Map for Icon #0 Only[c]	SR2E[2:1]	01, 10, 11	3C5h
Select the Hardware Icon Horizontal Position	SRX[7:5] / SR10[7:0], 30, 50, 70, 90, B0, D0, F0	2047...0	3C4h / 3C5h
Hardware Icon Horizontal-fine-position Extension (for expanded graphics and 10-dot character clocks)	SR2A[6]	0 =Non-Expanded 1 = Expanded	3C5h
Select Hardware Icon Vertical Position	SRX[7:5] / SR11[7:0], 31, 51, 71, 91, B1, D1, F1	2047...0	3C4h / 3C5h
CPU Access to Display Memory for Hardware Icons/ Cursor is Limited to 32-bits wide in VGA Modes	SR2D[7]	0	3C5h
CPU Access to Display Memory for Hardware Icons/ Cursor is Set to 64-bits wide in VGA Modes	SR2D[7]	1	3C5h
Display Memory Access for Hardware Icons/Cursor is 64 bits wide in all Modes	SR2D[6]	0	3C5h
Display Memory Access for Hardware Icons/Cursor is 32 bits wide in VGA Modes and 64 bits wide in Extended VGA Modes	SR2D[6]	1	3C5h

[a] The position of both the hardware cursor and hardware icons can be updated, but not simultaneously. When SR2A[7] = 0, the hardware cursor position is updated.

[b] The “2_” represents the four available hardware icon programming registers (2Ah, 2Bh, 2Ch and 2Dh) which correspond to the four Icons #0,#1, #2 and 3# respectively.

[c] The number in SR2E[2:1] points to the pair of 2-icon maps selected by SRS2A[5], but in this case, all maps go to Icon #0 for display.

Appendix D

Color Expansion and Extended Write Modes

D.1 Introduction

The CL-GD755X supports color expansion for faster CPU write performance. This is implemented using two extended write modes and the BitBLT engine. Color expansion enhances data transfer efficiency when the host is sending patterns to the display memory that require one or two colors. Rather than sending the entire color description across the bus for each pixel, the color descriptions are written into registers in the CL-GD755X. Then each pixel is defined with a single bit that specifies one of the two pre-programmed colors.

D.2 Color Expansion

Color expansion is the automatic conversion of a monochrome bitmap, typically defining a character, icon, or pattern, into foreground and background color values that are written into display memory. The foreground and background color values are held in the CL-GD755X, and only the monochrome bit maps need to be transmitted across the bus. Each bit of the monochrome map is converted into an 8-bit or 16-bit pixel value: The bus traffic is reduced accordingly.

In graphics modes, extended write modes can be used for the following operations: faster text write, pattern fill, and block move operations. In 8- or 16-bit-per-pixel graphics modes with packed-pixel addressing, these extended write modes operate on 8 pixels at a time. In addition, the BitBLT engine supports color expansion for 8-, 16- or 24-bit-per-pixel modes.

NOTE: For Application Programmers – Extended Write modes 4 and 5 are no longer recommended for color expansion due to the superior color-expansion capabilities of the BitBLT engine.

D.3 BitBLT Color Expansion of Graphics Data

When Extension register bit GR30[7] is 1, the source input to the ROPS of Extension register GR32 is not actual data from the source area; instead, it is data that has been color-expanded. Since the source area is a monochrome image that has a single bit per pixel, substantial performance benefits are possible, especially when the source image is expanded to 16 or 24 bits. Color expansion can be used when what appears in a destination area is either of the following:

- A single foreground color by itself
- A single foreground color along with a single background color

The CL-GD755X supports color expansion of data that is in either 8-, 16- or 24-bit-per-pixel display modes. The source can be either an 8×8 pattern (display memory only), or monochrome data from either display memory or system memory. When the source data is in display memory:

- The starting address of the data must be located on a 4-byte boundary.
- Except, for a pattern of 8-pixels $\times$ 8-pixels, the starting address of the data must be on an 8-byte boundary.

When the data in the source area is a:

- 0, the result is the background color is written into corresponding byte(s) of the destination area, or no change takes place to the destination area (that is, the data written to the destination area is transparent).
- 1, the result is the foreground color is written into corresponding byte(s) of the destination area.

For color expansion, the destination must be the “on-screen” portion of display memory, and the direction must be “increment”. Any ROP can be used; however, SRCCOPY is most common for color expansion.

The sense of monochrome data can be inverted by programming CR33[1] to 1, which allows complete foreground/background 24-bit/pixel color expansion in two passes. Also, for 24-bit/pixel color expansion, transparency must be enabled.

The most-significant bit of the first source byte is expanded into the ROP source data for the first pixel of the destination. Depending on the contents of GR30[5:4], it is expanded to 1, 2 or 3 bytes. Table D-1 indicates the registers that contain the expansion colors.

Table D-1. Color Expansion Registers

GR30[5:4]	Width	Background Expansion ('0' in Source Area)			Foreground Expansion ('1' in Source Area)		
		GR0	GR10	GR12	GR1	GR11	GR13
00	8-bit	Color	Don't care	Don't care	Color	Don't care	Don't care
01	16-bit	Low byte	High byte	Don't care	Low byte	High byte	Don't care
10	24-bit	*Transparency must be enabled*			Blue	Green	Red
11	--	Reserved			Reserved		

The next bit of source data is processed for the next 1, 2 or 3 bytes of destination, and so on, until the 'width' bytes of the scanline in the destination area have been processed. GR33[0] controls the source data granularity for color-expanded system-to-screen BitBLTs. When GR33[0] is 1, unused source data is discarded to the end of the current DWORD. When GR33[0] 0, unused source data is discarded to the end of the current byte.

The destination address is modified by the destination pitch. The source pitch is ignored, since the source is taken to be a linear string of bytes. The next byte of source is the first 8 pixels for the next scanline.

D.3.1 BitBLT Color Expansion of Graphics Data (with Transparency)

When GR30[3] is 0, the pixels corresponding to ‘0’s in the source area are written with the contents of the background color registers into the destination area. When GR30[3] is 1, a color expansion with transparency is enabled. The pixels corresponding to ‘0’s in the source area are not written into the destination area. When transparency is enabled, the background color registers are ignored, and the foreground color registers are used as shown in Table D-1.

Table D-2. Color Expansion Registers with Transparency

GR30[5:4]	Width	Foreground Expansion (‘1’ in Source Area)		
		GR1	**GR11**	**GR13**
00	8-bit	Color	Don’t care	Don’t care
01	16-bit	Low byte	High byte	Don’t care
10	24-bit	Blue	Green	Red
11	--	Reserved		

When color expansion is used with an opaque foreground and a transparent background (analogous to Extended Write mode 4), the transparency feature must be used, and the transparent color must be set to the background color. The registers used for foreground color expansion with transparency are the same as the color expansion registers with ‘1’ in the source.

D.4 Color Expansion Using Extended Write Modes

The registers shown in Table D-3 are involved in color expansion using Extended Write modes 4 and 5.

Table D-3. Color Expansion Registers Using Extended Write Modes

Register Bit	Function
SR2[7:0]	Enable writing pixels[7:0]
SR7[0]	‘1’ enables packed-pixel modes and GR0/GR1 for color
GRB[1]	‘1’ enables By-8 addressing
GRB[2]	‘1’ enables Extended Write modes
GRB[4]	‘1’ Extended Write modes to 16-bit pixels
GR5[2:0]	‘100’ selects Extended Write mode 4 ‘101’ selects Extended Write mode 5
GR0[7:0]	Extended Write mode 5, background-color low byte
GR1[7:0]	Extended Write mode 4/5, foreground-color low byte
GR10[7:0]	Extended Write mode 5, background-color high byte
GR11[7:0]	Extended Write mode 4/5, foreground-color high byte

D.4.1 Extended Write Modes

The CL-GD755X supports two extended write modes: Extended Write modes 4 and 5. These extended write modes can selectively update up to eight pixels of 8 or 16 bits each. For more information, refer to Section D.5 and Section D.6.

Extension register GRB, the Graphics Controller Mode Extensions register, bits [4,2:1] are used in combination with Graphics Controller register GR5[2:0] to enable the CL-GD755X Extended Write modes.

GRB: GRAPHICS CONTROLLER MODE EXTENSIONS

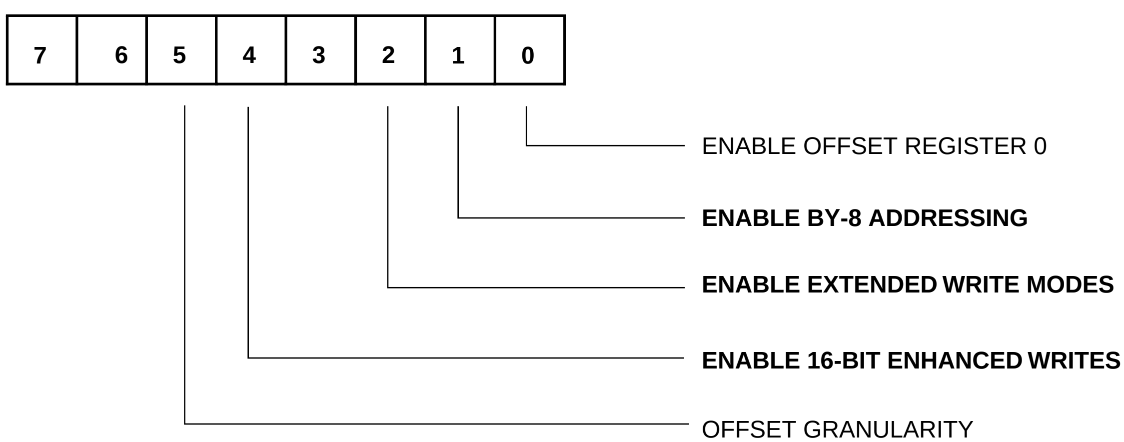


Figure D-1. Extension Register GRB

When Extended Write modes are enabled by setting GRB[2] to ‘1’:

- Up to 8 bytes (that is, 8 pixels) can be transferred to display memory for every CPU byte transfer.
- Graphics Controller register GR0[7:0] contains the background color low byte for Extended Write mode 5.
- Graphics Controller register GR1[7:0] contains the foreground color low byte for Extended Write modes 4/5.
- Graphics Controller register GR5[2] is enabled to select Extended Write modes 4 and 5.
- Graphics Controller register GRB[4] is enabled to select enhanced writes for Write modes 4 and 5.
- Sequencer register SR2[3:0] extends to SR2[7:0] for Write modes 4 and 5.

D.4.2 By-8 Addressing

GRB[1], when set to ‘1’, selects By-8 addressing for 8-bit-per-pixel (256-color) graphics modes. The system address is shifted left by 3 bits relative to packed-pixel addressing, so that each system byte address points to a different 8-pixel (8 byte) block in display memory. When GRB[4] is ‘1’, this bit is a ‘don’t care’.

D.4.3 By-16 Addressing

GRB[4], in combination with GRB[2], is used to select Write mode 4 and 5 for 16-bits-per-pixel graphics modes (RGB 555 or 565). The system address is shifted left by four bits relative to packed-pixel addressing, so each system byte address points to a different 8 pixels (16-byte block) in display memory. GRB[4] and GRB[2], when set to ‘1’, select the following Write mode operation:

- Enables By-16 addressing.
- Enables up to 16 bytes (eight 2-byte pixels) to be transferred for each CPU byte cycle.
- Enables GR10 and GR11 as high-byte data for background and foreground color extensions. (GR0 and GR1 contain the low-byte data.)
- Enables each bit of SR2 to be used as pixel write mask for two bytes.

D.5 Extended Write Mode 4

The CL-GD755X supports Extended Write mode 4, which is used in combination with By-8 or By-16 addressing modes to operate on 8 pixels of data at a time (8 or 16 bytes, depending on 8- or 16-bit-per-pixel mode). This mode is enabled when GRB[2] is 1 and GR5[2:0] is ‘100’.

- In 8-bits-per-pixel mode:
  - — The Foreground Color bits GR1[7:0] are used to update each pixel byte in display memory.
  - — The 8-bit value is written as the foreground color, when the corresponding CPU data is a ‘1’, and the corresponding Map Mask register bit in SR2 is also set to a ‘1’.
  - — When the CPU Data bit is a ‘0’, the corresponding pixel in display memory is not changed by the write.
- In 16-bits-per-pixel mode:
  - — The Foreground Color register bits GR11[7:0] (foreground-color high byte) and GR1[7:0] (foreground-color low byte) are used to update each pixel word in display memory.
  - — When the corresponding CPU data is ‘1’ and the corresponding Map Mask register bit in SR2 is also ‘1’, the 16-bit value is written as the foreground color.
  - — When the CPU Data bit is ‘0’, the corresponding pixel in display memory is not changed by the write.

This mode, for example, can be used to write text to display memory with the foreground color while preserving the background color. This action allows 8 pixels of display data to be updated with a single CPU byte cycle.

D.6 Extended Write Mode 5

The CL-GD755X supports Write mode 5, which is used in combination with By-8 or By-16 addressing mode to operate on 8 pixels of data at a time (8 or 16 bytes, based on 8- or 16-bit-per-pixel mode). This mode is used to write text to display memory with a selected foreground color and a background color. This action allows 8 pixels of display data to be updated with a single CPU byte cycle. This mode is enabled when GRB[2] is 1 and GR5[2:0] is ‘101’.

- In 8-bit-per-pixel mode:
  - — The Foreground Color bits GR1[7:0] or the Background Color bits GR0[7:0] are used to update each pixel byte in display memory.
  - — When the corresponding CPU data bit is ‘1’, the 8-bit value is written as the foreground color.
  - — When the corresponding CPU data bit is ‘0’, the corresponding pixel in display memory is written with the background color.
  - — The corresponding Map-Mask register, SR2[7:0], also inhibits writes to selected pixel in display memory.
- In 16-bit-per-pixel mode:
  - — The Foreground-Color bits GR11[7:0] (foreground-color high byte) and GR1[7:0] (foreground-color low byte), or the Background Color bits GR10[7:0] (background-color high byte) and GR0[7:0] (background-color low byte), are used to update each pixel word in display memory.
  - — When the corresponding CPU data is ‘1’, the 16-bit value is written with the foreground color.
  - — When the corresponding CPU data bit is ‘0’, the corresponding pixel in display memory is written with the background color.
  - — The corresponding Map-Mask register, SR2[7:0], also inhibits the corresponding pixel value (2 bytes) to be written to display memory.

Appendix E

True-Color Modes

E.1 Introduction

The CL-GD755X has a built-in true-color multi-mode palette DAC that supports the following modes:

- 8 bits per pixel, VGA-standard 256-color-palette mode, and RGB 3-3-2 256-color mode
- 8 bits per pixel, grayscale
- 15 bits per pixel, 32K color, the RGB 5-5-5 mode
- 16 bits per pixel, 64K color, the RGB 5-6-5 XGA™ mode
- 24 bits per pixel, 16.8-million color, the RGB 8-8-8 True-Color mode

In the 256-Color Palette mode, the palette DAC is VGA-compatible and provides 256 simultaneous colors on the display screen from a palette of 256K colors.

E.2 Programming for a True-Color Multi-Mode Palette DAC

Extended Color modes are enabled by the Hidden DAC register (Extension register HDR at I/O Port 3C6h). At reset, the HDR is loaded with 00h, which programs the palette DAC in a VGA-compatible mode. In this mode, the palette DAC is functionally equivalent to the industry-standard Brooktree® BT476 RAM-DAC. By writing code to the HDR, the palette DAC can be programmed in one of the modes mentioned above. The following method is used to write to the HDR:

1. **Read from the Pixel-Mask register (3C6h) four times in succession.**

   No other reads/writes must be directed to that address. After the fourth read, the Pixel-Mask register points to the Hidden DAC register.

2. **Program the Hidden DAC register by writing to it at Port 3C6h.**

   Any read from or write to any address (other than 3C6h) resets an internal counter, which 'hides' the HDR again. To continue, repeat Step 1.

Table E-1. Extended Color Mode Selected with the Hidden DAC Register

HDR							Color Modes
[7]	[6]	[5:4]	[3]	[2]	[1]	[0]	
0	X	X	X	X	X	X	Standard VGA-compatible 256-color
1	0	X	X	X	X	X	RGB 5-5-5, 32K colors
1	1	X	0	0	0	1	RGB 5-6-5, XGA, 64K colors
1	1	X	0	1	0	1	RGB 8-8-8 16M true color
1	1	X	0	1	1	X	DAC power-down
1	1	X	1	0	0	0	8-bit grayscale

E.3 CL-GD755X 24-Bit Color-Lookup-Table Palette RAM

The Color-Lookup Table (CLUT) is a 24-bit wide, 256 word deep RAM which is loaded by the BIOS when the device is initialized. The data loaded into the CLUT can be color corrected to compensate for the effects of the gamma coefficient or luminance/brightness variance of the LCD or CRT display. The true-color data, which would normally bypass the CLUT RAM, is used as an address into the CLUT instead. This provides the ability to use the gamma/luminance correction capability of the CL-GD755X. The full 24 bit CLUT is used in all extended color modes to provide 8 bits of corrected color data for each of the red, green and blue colors. This provides a palette of 16.8 million colors for the extended color modes.

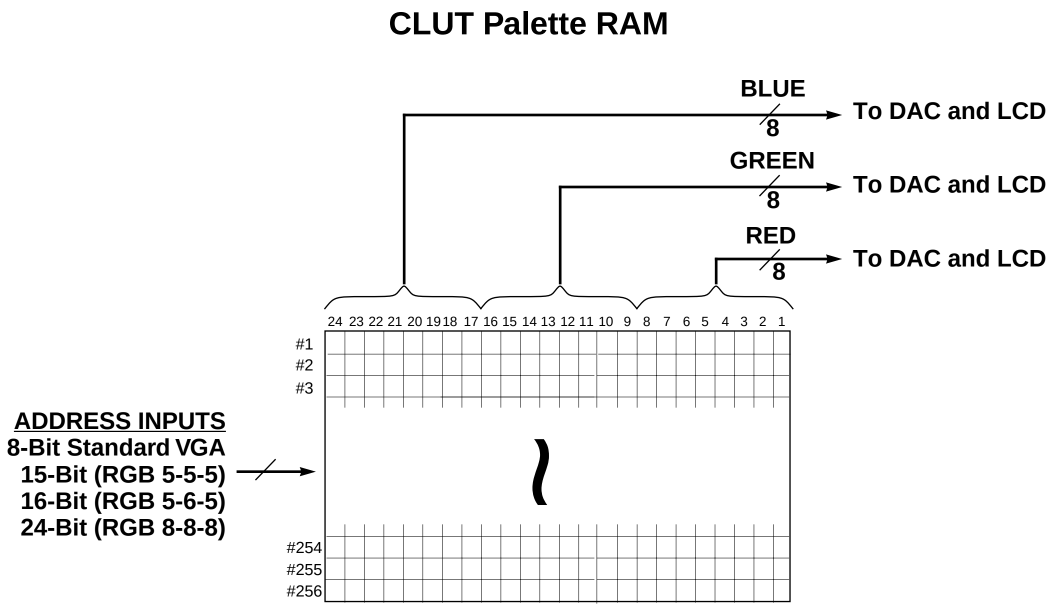


Figure E-1. CL-GD755X Graphics Color Lookup Table

E.3.1 8-Bit VGA Compatible – 256 Color Palettized

This is the power-on default mode for the CL-GD755X. It selects 256 simultaneous colors from a palette of 256K possible colors. The 8-bit input to the CLUT shown in Figure E-1 selects one of 256 18-bit words in the standard VGA-compatible mode. These 18-bit words provide six bits of color data for each of the red, green and blue colors. To send a 24-bit word to the DAC and LCD, the CL-GD755X converts each 6-bit word to 8 bits by shifting the data two bits to the left and adding two least-significant zeros.

E.3.2 8-Bit Grayscale

Each pixel is represented by one byte. The byte is duplicated and sent as an address to each of the red, green and blue sections of the CLUT. The 24-bit color-corrected data is sent from the CLUT to the DAC. The result on the screen is a gray pixel with brightness corresponding to the value of the original byte.

E.3.3 RGB 5-5-5 Mode with 32K Colors

This mode supports the industry-standard RGB 5-5-5 mode with 32,768 colors. Each pixel is represented by 15 bits containing 5 bits each of red, green, and blue color information. By using the CLUT, this data provides 32K colors from a palette of 16.8M. This mode is selected by programming the Hidden DAC register (HDR) to **80h**.

The input sequence for each pixel is the low byte first, followed by the high byte. The first low byte is taken on the first rising edge of clock occurring after BLANK# has gone inactive (high). Each pixel is stored on a 2-byte boundary. When pixels are written into display memory using 16- or 32-bit write operations, they can be treated as shown in Table E-2. When partial pixels are being stored one byte at a time in a little-endian machine like the '86 (80386 or 80486), the byte containing the Blue bits are stored at the first address, and the byte containing the Red bits is stored at the next higher address.

Table E-2. Pixel Data Format in RGB 5-5-5 Mode

MSB HIGH BYTE								LSB LOW BYTE							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
X	4	3	2	1	0	4	3	2	1	0	4	3	2	1	0
X	RED					GREEN					BLUE				

E.3.4 5-6-5 Mode with 64K Colors (XGA™)

This mode supports the XGA™ 5-6-5 RGB mode with 65,536 colors. Each pixel is represented by 16 bits containing 5 bits of red, 6 bits of green, and 5 bits of blue color information. By using the CLUT, this data provides 64K colors from a palette of 16.8M. This mode is selected by programming the HDR to C1h.

The input sequence for each pixel is the low byte first, followed by the high byte. The first low byte is taken on the first rising edge of clock occurring after BLANK# has gone inactive (high). Each pixel is stored as two contiguous bytes on a 2-byte boundary. When pixels are written into display memory using 16- or 32-bit write operations, they can be treated as shown in Table E-3. When partial pixels are being stored, one byte at a time in a little-endian machine like the '86 (80386 or 80486), the byte containing the Blue bits are stored at the first address, and the byte containing the Red bits is stored at the next higher address.

Table E-3. Pixel Data Format in 5-6-5 Mode with 64K Colors

MSB HIGH BYTE								LSB LOW BYTE							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
4	3	2	1	0	5	4	3	2	1	0	4	3	2	1	0
RED					GREEN						BLUE				

E.3.5 RGB 8-8-8 Mode with 16.8 Million Colors (True Color Mode)

This mode supports the industry-standard RGB 8-8-8 mode with 16,777,216 colors. Each pixel is represented by 24 bits containing one byte each of red, green, and blue color information. This mode is selected by programming the HDR to C5h.

Each pixel is stored as three contiguous bytes. The Blue value is stored in the lowest-addressed byte, the Green value is stored in the next higher-addressed byte, and the Red value is stored in the next higher-addressed byte. The input sequence for each pixel is the low byte (Blue) first, followed by the middle byte (Green), followed by the high byte (Red). The first low byte is taken on the first rising edge of clock occurring, after BLANK# has gone inactive (high). All subsequent bytes are clocked in on the rising edge of clock. This true-color data, which would normally bypass the CLUT RAM, is used as an address into the graphics CLUT instead. This provides the ability to use the gamma/brightness correction capability of the CL-GD755X.

Table E-4. Pixel Data Format in RGB 8-8-8 Mode with 16.8 Million Colors

MSB HIGH BYTE								MIDDLE BYTE								LSB LOW BYTE							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
RED								GREEN								BLUE							

Appendix F

Quality Assurance Procedures

F.1 Introduction

The Cirrus Logic Quality Assurance Procedures are intended verify that the software provided by Cirrus Logic is fully compatible with respective system environments, application programs, and the CL-GD755X 64-bit video and graphics LCD/CRT controller. Each piece of software is tested and verified in a known environment before it is integrated with the other CL-GD755X compatible pieces.

The testing is performed first on the VGA BIOS and the chip to verify the compatibility of the hardware and firmware supplied with the product. Subsequent testing is done on the software drivers to verify the functional interface to the system and applications software.

Any problems encountered with the software are fully documented and returned to the software and hardware engineering departments for verification and fixing.

The procedures described in this document constitute the minimum level of software testing performed by the QA lab. In most cases, the actual testing performed is much more rigorous.

F.1.1 Test Equipment

The equipment used by the QA testing personnel consist of:

- pentium based computers with enough memory to handle all of the planned tests
- multi-frequency color monitors
- color analog fixed frequency PS/2 compatible monitors
- CL-GD755X based demonstration board with flat panel connected

F.2 VGA BIOS Test Plan

The purpose of this QA Test Procedure is to ensure that the released Cirrus Logic VGA BIOS is fully compatible with the CL-GD755X chip, software and system standards, and is free of software bugs. It is tested with the latest revision of the CL-GD755X chip. During the early stages of chip development, each revision of the chip may require a different version of the VGA BIOS software. Check with your Cirrus Logic sales office for details.

F.2.1 VGA BIOS Testing Sequence

The testing is done in phases to provide the quickest feedback to the software and chip developers when a problem is detected. The VGA BIOS is released following the successful completion of these tests. The testing sequence is:

- Basic Acceptance Testing – is performed on all new VGA BIOS software to verify basic functionality.
- Phase One Testing – is composed of a suite of test programs that are run to verify basic VGA compatibility before the software is made available outside of the engineering and QA departments.
  - These tests do not check any extended features or capabilities of the CL-GD755X, or extended mode drivers.
- Phase Two Testing – verifies complete functionality of the VGA BIOS in a controlled environment. It tests:
  - the utilities shipped with the VGA BIOIS
  - all special capabilities and features of the product
  - any changes or fixes from previous versions of the VGA BIOS (regression testing)
  - complete hardware and software configuration
- Phase Three Testing – checks compatibility of the VGA BIOS with commercial applications and extended-mode drivers.

F.2.2 VGA BIOS Testing Rules

1) When running applications software, the next tests should be run without rebooting. When a problem occurs on a subsequent test, the computer must be cold booted and the test repeated. In all cases, every effort should be made to identify the cause of the failure; i.e., what the application changed.

2) When running any test program, no device drives or ‘terminate and stay resident’ (TSR) programs should be running, unless they are the application being tested, or are necessary for the application being tested.

3) The computer should be cold booted (cycle the power) both before and after running any compatibility tests.

4) Problems are generally found in one of two ways:

   a) Some programs report problems found during execution. This is especially true in Phase One compatibility testing, and advanced systems applications like MS Windows.

   b) Other problems are only found by close observation of the programs behavior. Even programs that report problems. may not be able to detect all passible failures. Therefore, careful observation is critical.

5) All new reproducible problems are first verified, then documented on a Product Deficiency Report (PDR). After finding a new problem the PDR is completed immediately, before any further testing is attempted.

6) Whenever a problem is detected during phase-one testing, it is retested with the reference display card. For VGA systems, the reference card is an IBM PS/2 display adapter.

7) Every effort is made to remove variables in order to isolate the problem. The PDR that is generated must be as specific as possible.

8) Any PDR forms generated outside of QA are sent immediately to the QA group manager.

9) PDR forms must contain all of the information necessary to reproduce and identify the problem.

10) Completed PDR forms are given to another test engineer who is using the same BIOS, so the problem, and the steps necessary to reproduced it, can be confirmed.

F.2.3 Basic Acceptance Testing

These tests consist of a simple set of checks that verify basic functionality of the BIOS. These checks do not evaluate the quality of the BIOS, but rather determine the basic functionality needed to perform Phase One testing. Any failure of these tests causes the BIOS to be returned to engineering with no further testing.

VGACC

VGACC.EXE is a mode 12 convergence-check program that quickly establishes basic alignment in a 640 x 480 display.

SETMODE

SETMODE.EXE is used to set all standard and extended modes supporting text output. A simple DIR command is used after setting the mode to establish that text scrolling is supported. This test only establishes basic functionality, no display quality assessments are made.

Display Type

The adapter is set for all supported display types.

F.2.4 Phase One Testing

This is a suite of five programs that test basic VGA compatibility. Their primary use is to establish the ability of the BIOS to support basic VGA modes. These tests must be run on any Beta release of the BIOS before it is made available outside of engineering and QA.

RENAISSANCE

Execute `'VGATEST'` and work through all choices on all the menus.

PMINTA

PMINTA must be run manually. The tester is prompted for the appropriate actions.

PC-TECH

PC-TECH can be run in an automatic mode. It writes a disk file of any detected errors. The fast (-f) mode allows the program to complete in just a couple of minutes. The syntax to run the program is:

```
VGATEST errfile -f<CR>
```

where `errfile` is the name of the file containing the list of errors.

Even when running in the ‘automatic’ mode, the testing person must watch the test to be sure that the test appears correct.

DisplayMate

DisplayMate is run by:

1) typing `'dmu<CR>'`

2) selecting technical tests

3) running tests for all modes under ‘Speed Performance Tests’

4) under 'Compatibility Tests, selecting:
   a) Display Modes Test Suite
   b) Configuration Tests
   c) VGA Compatibility Tests

PCMag 9.0

PCMag is run by:

1) typing `'bench<CR>'`
2) running all tests under 'VIDEO' selection of the 'PERFORMANCE' menu
3) running all tests under the 'QUALITY menu

F.2.5 Phase Two Testing

These tests are used to verify the extended functionality of the VGA BIOS and the CL-GD755X chip. These tests cover all display options and display modes supported by the CL-GD755X.

F.2.5.1 Cirrus Logic Utilities

CLMODE.EXE

All the options listed in the of CLMode menus are tested. All of the keywords and switches, that are listed when 'CLMODE /h <CR>' is typed, must be tested to make sure that all the options function properly and the listed 'keywords' actually work.

MODES.EXE

The 'Modes' program provides a quick method of verifying that all the VGA mode states are functioning properly. The Modes program can verify that acceptable extended parameters are used in standard and extended modes.

F.2.5.2 Other Utilities

All the utilities should be tested in each of the display modes whenever possible. Some of the utilities tested are:

- MODE.COM (Supplied with DOS)
- VESATEST v1.06a
- Vidspeed v2.2
- TESTICON
- LPTIME
- IDTEST
- DISPSW
- VBEPM
- PWM
- M1EXP.EXE

F.2.6 Phase Three Testing

These tests verify compatibility of the VGA BIOS with end-user systems and applications in both out-of-the-box configurations, and with extended resolution drivers supplied by Cirrus Logic. All applications should be tested by executing the tutorial included with the program, and by using all possible display modes. The following is a list of programs presently being used to perform these tests:

- Operating Environments
  - MS-DOS 6.22
  - MS-Windows 3.1
  - MS-Windows 95
  - MS-Windows NT 3.51, 4.0
  - MS-Windows 3.11 (Windows for Workgroups)
  - OS/2 3.0 Warp
- MS-DOS Application Programs
  - 3D-Studio
  - AutoCAD (v10 - v13)
  - 7th Guest
  - Generic Cadd v6.0
  - Lotus 123
  - MS-Word v5.5, v6.0
  - QA Plus v5.2
  - Battle Chess
  - DPaint v2.0
  - Word Perfect v5.1, v6.0
  - Wordstar v7.0
- Windows Application Programs
  - AfterDark v3
  - AMIPro v3.1
  - AutoCAD v13 (for Windows)
  - MS-Video for Windows v1.1D
  - Corel Draw v5.0
  - Designer v4.01
  - Excel v5.0
  - Lotus 123 for Windows v4.0
  - PageMaker v5.0
  - PowerPoint v4.0
  - Windows Entertainment Pack 3, 4
  - Word for Windows v7.0
  - WordPerfect for Windows v6.1
- Extended Mode Driver Applications Programs
  - Windows 3.1
  - Windows 95
  - Windows NT 3.51, 4.0
- Benchmark Programs
  - PC-Labs Windows Benchmark v‘96, ‘97
  - WITS

  - WINTUNE
  - WinTACH v1.2
  - WINSTONE ‘97
  - PHOTOSTONE
  - TORQUE
  - SPEEDY
- Games Programs
  - MYSTS
  - BATTLECHESS
  - RAPTOR
  - 7th Guest
  - DOOM 1, 2
  - TRISTAN PINBALL
  - XWING
  - LEMMINGS
  - LINKS 386
  - MS-Flight Simulator v5.0
  - WOLF 3D
  - RASTAN
- Multimedia Applications Programs
  - JFK
  - ENCARTA ‘95
  - CINEMANIA ‘95, 96
  - Compton’s Encyclopedia
  - MS-Bookshelf
- MPEG Player
  - Video for Windows v1.1
  - Mediametic
  - Compcore
  - Xing
- DOS-Based Utilities
  - PCTools v7.1
  - Norton Utilities v6.0
  - Norton Desktop v3.0
  - QA Plus
  - VPIC
  - CSHOW v8.50A
  - Norton Anti-Virus

F.3 Display-Driver Test Plan

The purpose of the Display-Driver QA Procedures is to ensure that any released Cirrus Logic driver software is fully compatible with respective applications/system environments and is free of bugs. The procedures described in this document constitute the minimum level of software testing which the QA lab performs. Often, more rigorous tests are performed on the drivers.

F.3.1 Testing rules

1) Driver testing will not begin until the correct operation of the BIOS has been confirmed according to the BIOS test plan, and the application program has been successfully tested in it's out-of-the-box configuration.
2) Whenever there are different resolution drivers for a particular product, all tests should be repeated for each driver being tested.
3) Tests will be repeated on CRT monitor, flat panel, SimulSCAN (simultaneous CRT and flat panel), and TV.
4) Problems are generally found in two ways:
   a) Some programs report problems that are found during execution. This is especially true of the compatibility test in phase one of the testing. It is also true of some sophisticated applications such as MS Windows.
   b) Other problems are only found by close observation of the program's behavior. Even those programs that report display problems that they find, cannot find every problem. Careful observation is critical.
5) When a problem is found, it is reproduced with the application configured to use the built in VGA driver. If the built-in VGA driver is not applicable, an alternate product’s driver is used.
6) Every effort will be made to remove variables in order to isolate the problem. The PDR (Product Deficiency Report) that is generated should be as specific as possible.
7) Any NEW reproducible problems found will be documented in a PDR.
8) After finding a new problem, the PDR will be completed immediately, before proceeding any further with testing.
9) PDRs should be written in such a way that all of the information required to reproduce and identify the problem is on the form. It should not be necessary to get any additional information to reproduce the problem.
10) Completed PDRs will be given to another tester who is using the same configuration so that the problem and the steps required to reproduce it may be confirmed.

F.3.2 Extended Mode Windows Driver Tests

Each version of the Cirrus Logic Windows drivers must be tested with the applicable version of the Windows operating system. Start by testing both methods of installing the CL-GD755X Windows drivers.

- Install utility - this utility loads the supported drivers and its component files all at one time.
- Setup - this Windows utility allows you to install the supported drivers individually.

For each process install the drivers from a diskette and from a subdirectory on the hard drive. After the installation processes have been verified, perform the following tests on each driver in each resolution.

Windows 3.1x

1) In Windows 3.1x, run the WinMode utility to exercise all the features. This also tests the correct operation of the WinMode program.
2) Run the entire Windows 3.1 DCT (Driver Compatibility Test suites) that is included in the Windows 3.1 DDK. This will test all driver functions and VDD functions.
3) Run all the available display benchmarks from the latest version of the Ziff-Davis’s Windows Benchmark. Run other benchmark programs (such as WinTACH and WITS).

4) Run programs that come with Windows, i.e., programs from the Main, Accessories, Games, and Control Panel groups, etc.

5) Start three or four applications from these groups and do the following tests.

- **a)** Switch between them using the keyboard and mouse.
- **b)** Move the application windows around and off the screen while the application is still active.
- **c)** Make a text scroll using the keyboard and each of the scroll bar operations.
- **d)** Maximize a window and force other windows on top of it. Scroll a graphics window (such as Paint).
- **e)** Switch between display types.

Windows 95

1) Run Windows 95 DCT (Driver Compatibility Test suites) that is included in the Windows 95 DDK.

2) Run all the available display benchmarks from the latest version of the Ziff-Davis's Windows Benchmark. Run other benchmark programs (such as WinTACH and WITS).

3) Run programs that come with Windows, i.e., programs from the Main, Accessories, Games, and Control Panel groups, etc.

4) Start three or four applications from these groups and do the following tests.

- **a)** Switch between them using the keyboard and mouse.
- **b)** Move the application windows around and off the screen while the application is still active.
- **c)** Make a text scroll using the keyboard and each of the scroll bar operations.
- **d)** Maximize a window and force other windows on top of it. Scroll a graphics window (such as Paint).
- **e)** Switch between display types via a hot-key or Cirrus Logic utilities.

Windows NT 4.0

1) Run Windows 95 WHQL (Windows Logo Certification Test) that is included in the Windows 95 DDK.

2) Run all the available display benchmarks from the latest version of the Ziff-Davis's Windows Benchmark. Run other benchmark programs (such as WinTACH and WITS).

3) Run programs that come with Windows, i.e., programs from the Main, Accessories, Games, and Control Panel groups, etc.

4) Start three or four applications from these groups and do the following tests.

- **a)** Switch between them using the keyboard and mouse.
- **b)** Move the application windows around and off the screen while the application is still active.
- **c)** Make a text scroll using the keyboard and each of the scroll bar operations.
- **d)** Maximize a window and force other windows on top of it. Scroll a graphics window (such as Paint).
- **e)** Switch between display typesvia a hot-key or Cirrus Logic utilities.

Notes

Direct Sales Offices

Domestic

N. CALIFORNIA
Fremont
TEL: 510/623-8300
FAX: 510/252-6020

S. CALIFORNIA
Irvine
TEL: 714/453-5961
FAX: 714/453-5962

Westlake Village
TEL: 805/371-5860
FAX: 805/371-5861

SOUTH CENTRAL AREA
Austin, TX
TEL: 512/255-0080
FAX: 512/255-0733

Dallas, TX
TEL: 214/252-6698
FAX: 214/252-5681

Houston, TX
TEL: 281/257-2525
FAX: 281/257-2555

NORTHEASTERN AREA
Andover, MA
TEL: 508/474-9300
FAX: 508/474-9149

SOUTHEASTERN AREA
Duluth, GA
TEL: 770/935-6110
FAX: 770/935-6112

Raleigh, NC
TEL: 919/859-5210
FAX: 919/859-5334

Boca Raton, FL
TEL: 407/241-2364
FAX: 407/241-7990

International

FRANCE
Paris
TEL: 33/1-48-12-2812
FAX: 33/1-48-12-2810

GERMANY
Munich
TEL: 49/81-52-40084
FAX: 49/81-52-40077

HONG KONG
Tsimshatsui
TEL: 852/2376-0801
FAX: 852/2375-1202

ITALY
Milan
TEL: 39/2-3360-5458
FAX: 39/2-3360-5426

JAPAN
Tokyo
TEL: 81/3-3340-9111
FAX: 81/3-3340-9120

KOREA
Seoul
TEL: 82/2-565-8561
FAX: 82/2-565-8565

SINGAPORE
TEL: 65/743-4111
FAX: 65/742-4111

TAIWAN
Taipei
TEL: 886/2-718-4533
FAX: 886/2-718-4526

UNITED KINGDOM
London, England
TEL: 44/1727-872424
FAX: 44/1727-875919

The Company

Headquartered in Fremont, California, Cirrus Logic is a leading manufacturer of advanced integrated circuits for desktop and portable computing, telecommunications, and consumer electronics. The Company applies its system-level expertise in analog and digital design to innovate highly integrated, software-rich solutions.

Cirrus Logic has developed a broad portfolio of products and technologies for applications spanning multimedia, graphics, communications, system logic, mass storage, and data acquisition.

The Cirrus Logic formula combines innovative architectures in silicon with system design expertise. We deliver complete solutions — chips, software, evaluation boards, and manufacturing kits — on-time, to help you win in the marketplace.

Cirrus Logic's manufacturing strategy ensures maximum product quality, availability, and value for our customers.

Talk to our systems and applications specialists; see how you can benefit from a new kind of semiconductor company.

Cirrus Logic Inc.
3100 West Warren Ave., Fremont, CA 94538
TEL: 510/623-8300 FAX: 510/252-6020

Publications Ordering: 800/359-6414 (USA) or 510/249-4200
World Wide Web: http://www.cirrus.com

387559-001