# OC65548

VGA BIOS

OEM Reference Guide
Revision 1.1

October 1995

CHIPS®

*Confidential Information*

**Copyright Notice**

Copyright© 1995 Chips and Technologies, Inc. ALL RIGHTS RESERVED.

This manual is copyrighted by Chips and Technologies, Inc. You may not reproduce, transmit, transcribe, store in a retrieval system, or translate into any language or computer language, in any form or by any means, electronic, mechanical, magnetic, optical, chemical , manual, or otherwise, any part of this publication without the express written permission of Chips and Technologies, Inc.

**Restricted Rights Legend**

Use, duplication, or disclosure by the Government is subject to restrictions set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at 252.277-7013.

**Trademark Acknowledgment**

CHIPS logotype, CHIPSlink, CHIPSPort, ELEAT, LeAPSet, NEAT, NEATsx, PEAK, SCAT, SuperMathDX, SuperState, PRINTGINE, and WINGINE are registered trademarks of Chips and Technologies, Inc.

CHIPSet, SuperMath, WinPC, and XRAM Video Cache are trademarks of Chips and Technologies, Incorporated.

IBM®, is a registered trademarks of International Business Machines Corporation.
IBM® AT, XT, PS/2, Micro Channel, Enhanced Graphics Adapter, Color Graphics Adapter, Video Graphics Adapter, IBM Color Display and IBM Color Display and IBM Monochrome Display are trademarks of International Business Machines Corporation.

TRI-STATE® is a registered trademark of National Semiconductor Corporation.

VESA® is a registered trademark of Video Electronics Standards Association.

VL-Bus is a trademark of Video Electronics Standards Association.

All other trademarks are the property of their respective holders.

**Disclaimer**

This document is provided for the general information of the customer. Chips and Technologies, Inc., reserves the right to modify the information contained herein as necessary and the customer should ensure that they have the most recent revision of the document. CHIPS makes no warranty for the use of its products and bears no responsibility for any errors which may appear in this document. The customer should be on notice that the field of personal computers is the subject of many patents held by different parties. Customers should ensure that they take appropriate action so that their use of the products does not infringe upon any patents. It is the policy of Chips and Technologies, Inc. to respect the valid patent rights of third parties and not to infringe upon or assist others to infringe upon such rights.

# Revision History

| Revision | Date | By | Comment |
|---|---|---|---|
| 0.1 | 4/24/95 | RN | Preliminary Release |
| 1.0 | 7/12/95 | RN/lc | Official Release |
| 1.1 | 8/21/95 | SH/BB/lc | Updated PCI Data Structure Table, removed reference to Emulation Registers<br>Updated Extended Mode Parameter Tables |

# TABLE OF CONTENTS

# FORWARD

This manual provides Original Equipment Manufacturers (OEMs) and end-users with information describing the functions and features of the OC65548 and OC65548-PCI VGA BIOS.

## Terminology And Conventions

| | |
|---|---|
| `COURIER FONT` | This font is used to specify commands typed by the user, as well as output produced by the system. |
| *Italic Type* | This type style is used to denote titles. It is also used to denote computer program names, menu alternatives and other items at their first occurrence. |
| [] | Items enclosed in square brackets are optional and may be ignored. |
| <> | Items enclosed in less than and greater than symbols are keystrokes the user should type. |
| h | An "h" which immediately follows a number indicates the value is hexadecimal. |
| Bordered | Denotes important points. |

## About This Manual

This manual contains five chapters and three appendices. The purpose of each is as follows:

| | |
|---|---|
| Chapter 1 | This chapter summarizes the features of the VGA BIOS and the BIOS kits. |
| Chapter 2 | This chapter describes the features of the VGA BIOS. |
| Chapter 3 | This chapter explains the hardware requirements of the VGA BIOS. |
| Chapter 4 | This chapter describes the standard function calls, extensions to the standard function calls and extended function calls of the VGA BIOS. |
| Chapter 5 | This chapter explains use of the OEM utility programs provided in the BIOS kits. |
| Appendix A | This appendix explains how to create a binary version of the VGA BIOS from source code. |
| Appendix B | This appendix explains how to determine and program extended control registers. Parameter tables are also listed in this appendix. |
| Appendix C | This appendix explains how to implement the Suspend/Resume Procedure in the system BIOS. |

# CHAPTER 1 - INTRODUCTION TO THE VGA BIOS

## VGA BIOS

The OC65548, and OC65548-PCI VGA BIOS (hereafter referred to as 65548 BIOS) is an enhanced, high performance BIOS that is used with the 65548 VGA Flat-Panel/CRT Controller to provide an integrated Flat-Panel VGA solution. The BIOS supports the following features:

- Full compatibility with the IBM VGA BIOS
- Optimized for the 65548 VGA Flat-Panel/CRT controller
- Support for monochrome LCD, 640x480 STN or TFT, 800x600 STN or TFT, and 1024x768 TFT displays. Optional support for other displays.
- Supports VESA BIOS Extensions, including VBE 2.0, VBE/DDC 1.0, and VBE/PM 1.0.
- Supports either VESA local bus or PCI bus
- Extended BIOS functions which offer easy access to 65548 features and capabilities
- Support for simultaneous display

## Customization Support

The 65548 BIOS design to allows the user to customize of values in the binary version of the BIOS. This capability enables an OEM to create a custom version of the BIOS without access to the source code. Customization is accomplished with the BMP548 utility included with the BIOS kits.

## BIOS Kits

The 65548 BIOS is available in three kit formats. These kits and their contents are as follows:

SE65548 VGA BIOS Evaluation Kit:

- Evaluation diskette (Evaluation copy of BIOS and utility programs)
- *OC65548 VGA BIOS OEM Reference Guide*
- Release notes
- Software Incident Report (SIR) forms

SK65548 VGA BIOS Binary Kit:

- Binary diskette (Master copy of BIOS and utility programs)
- *OC65548 VGA BIOS OEM Reference Guide*
- Release notes
- Software Incident Report (SIR) forms

SC65548 VGA BIOS Source Code Kit:

- Source code diskette
- SK65548 VGA BIOS binary kit

## Licensing

Please contact your Chips and Technologies, Inc. representative for information on how to obtain a production copy of the software.

## Customer Support

CHIPS software products are supported by field application engineers located in each sales office. If you encounter a problem, or have any questions regarding a CHIPS software product, please complete a copy of the Software Incident Report form included with your product. Forward the completed form to the local CHIPS sales office or to the following address:

Chips and Technologies, Inc.
2950 Zanker Road
San Jose, CA 95134
Attn.: Software Product Support

FAX SIR forms to:

(408) 894-2086

# CHAPTER 2 - BIOS FEATURES

The 65548 BIOS is fully compatible with the IBM VGA BIOS and contains many additional features that support the 65548 Flat-Panel/CRT VGA controller.

## VGA BIOS Compatibility Target

The 65548 BIOS can be configured to operate in a manner compatible with an IBM PS/2 motherboard VGA BIOS or IBM AT VGA adapter BIOS. The compatibility target can be configured with the BMP548 utility program.

## PS/2 Compatibility

If *PS/2 BIOS compatibility* is selected, direct switching between color and monochrome video modes is supported when the VGA is the only adapter in the system. The 65548 BIOS supports color modes when a secondary monochrome display adapter or Hercules adapter is installed in the system. The BIOS supports monochrome modes when a secondary CGA is installed in the system. This is the default operating mode.

## AT Compatibility

If *AT BIOS compatibility* is selected, direct switching between color and monochrome video modes is NOT supported when the VGA is the only adapter in the system. The equipment byte must be set for color before switching from a monochrome mode to a color mode and vice-versa. The 65548 BIOS supports color modes when a secondary MDA or Hercules adapter is installed in the system. In addition, the BIOS supports monochrome modes when a secondary CGA is installed in the system.

## Multiple Bus Support

The 65548 BIOS supports multiple bus specifications:

- VESA Local Bus
- PCI Local Bus, Revision 2.0

## Monitor Support

The 65548 BIOS supports interlaced and non-interlaced analog monitors (VGA color, VGA monochrome), but the 65548 does not support digital monitors.

**Note:** Multiple frequency monitors are supported as analog monitors.

## Display Boot/Display Type Configurations

The 65548 BIOS can be configured to boot on a CRT, Flat-Panel, or Simultaneous display (CRT and Flat-Panel are both active). The OEM may select the display boot device by changing BIOS parameters with the BMP548 utility program. The parameters which select the boot display device also select the CRT configuration. The available display boot devices are:

- CRT Boot
- Flat-Panel Boot
- Simultaneous Display Boot

## Extended Video Modes

The 65548 BIOS supports all standard modes and the following extended modes for analog monitors. If the total video memory size is less than the memory required by the given mode, the 65548 BIOS will not switch to that mode.

### Table 1: CRT Extended Video Modes

| Mode | VESA Mode | Resolution | Type | Color | Vertical Refresh |
|---|---|---|---|---|---|
| 20h | | 640x480 | Linear PackPixel | 16 | 60, 72, 75 Hz |
| 22h | | 800x600 | Linear PackPixel | 16 | 56, 60, 72, 75 Hz |
| 24h(I) | | 1024x768 | Linear PackPixel | 16 | 43, 60, 70, 75 Hz |
| 28h(I) | | 1280x1024 | Linear PackPixel | 16 | 43 Hz |
| 30h | 101h | 640x480 | Linear PackPixel | 256 | 60, 72, 75 Hz |
| 32h | 103h | 800x600 | Linear PackPixel | 256 | 56, 60, 72, 75 Hz |
| 34h(I) | 105h | 1024x768 | Linear PackPixel | 256 | 43, 60, 70, 75 Hz |
| 40h | 110h | 640x480 | Linear PackPixel | 32K | 60, 72, 75 Hz |
| 41h | 111h | 640x480 | Linear PackPixel | 64K | 60, 72, 75 Hz |
| 42h | 113h | 800x600 | Linear PackPixel | 32K | 56, 60 Hz |
| 43h | 114h | 800x600 | Linear PackPixel | 64K | 56, 60 Hz |
| 50h | 112h | 640x480 | Linear PackPixel | 16M | 60 Hz |
| 60h | | 132x25 | Text (8x16) | 16 | 70 Hz |
| 61h | | 132x50 | Text (8x8) | 16 | 70 Hz |
| 6Ah/70h | 102h | 800x600 | Planar | 16 | 60 Hz |
| 72h/75h(I) | 104h | 1024x768 | Planar | 16 | 43, 60, 70, 75 Hz |
| 76h(I) | 106h | 1280x1024 | Planar | 16 | 43 Hz |
| 79h | 101h | 640x480 | PackPixel | 256 | 60, 72, 75 Hz |
| 7Ch | 103h | 800x600 | PackPixel | 256 | 56, 60, 72, 75 Hz |
| 7Eh(I) | 105h | 1024x768 | PackPixel | 256 | 43, 60, 70, 75 Hz |

### Table 2: Flat-Panel Extended Video Modes

| Mode | VESA Mode | Resolution | Type | Color | Panel CLK |
|---|---|---|---|---|---|
| 20h | | 640x480 | Linear PackPixel | 16 | 25 MHz |
| 22h | | 800x600 | Linear PackPixel | 16 | 25 MHz |
| 24h | | 1024x768 | Linear PackPixel | 16 | 25 MHz |
| 28h | | 1280x1024 | Linear PackPixel | 16 | 25 MHz |
| 30h | 101h | 640x480 | Linear PackPixel | 256 | 25 MHz |
| 32h | 103h | 800x600 | Linear PackPixel | 256 | 25 MHz |
| 34h | 105h | 1024x768 | Linear PackPixel | 256 | 25 MHz |
| 40h | 110h | 640x480 | Linear PackPixel | 32K | 50 MHz |
| 41h | 111h | 640x480 | Linear PackPixel | 64K | 50 MHz |
| 42h | 113h | 800x600 | Linear PackPixel | 32K | 50 MHz |
| 43h | 114h | 800x600 | Linear PackPixel | 64K | 50 MHz |
| 50h | 112h | 640x480 | Linear PackPixel | 16M | 75 MHz |
| 60h | | 132x25 | Text (8x16) | 16 | 25 MHz |
| 61h | | 132x50 | Text (8x8) | 16 | 25 MHz |
| 6Ah/70h | 102h | 800x600 | Planar | 16 | 25 MHz |
| 72h/75h | 104h | 1024x768 | Planar | 16 | 25 MHz |
| 76h | 106h | 1280x1024 | Planar | 16 | 25 MHz |
| 79h | 101h | 640x480 | PackPixel | 256 | 25 MHz |
| 7Ch | 103h | 800x600 | PackPixel | 256 | 25 MHz |
| 7Eh | 105h | 1024x768 | PackPixel | 256 | 25 MHz |

**Note:**
1. The Flat Panel cannot support Interlaced modes. All modes are Non-Interlaced. The default Flat-Panel size is 640x480.
2. The Flat-Panel clock value shown in the preceding table is for a 640x480 DD Panel. It may require different clock values for different Flat-Panel resolutions.
3. VESA modes may use either a linear or a paged frame buffer. See the VBE 2.0 function documentation for more information on specifying the frame buffer format for a mode.
4. (I) = Interlaced

## Audible Signals

If an error is encountered when the 65548 Flat-Panel/CRT VGA controller is initialized, the 65548 BIOS will return a beep pattern consisting of one long beep and several short beeps. (See Table 3.)

### Table 3: Audible Signals

| Beep Pattern | Error Condition |
|---|---|
| 1 Long, 2 Short | Horizontal retrace failed, |
| 1 Long, 3 Short | DAC test failed, |
| 1 Long, 4 Short | Monitor sense failed, |
| 1 Long, 5 Short | Vertical retrace failed, |
| 1 Long, 6 Short | Video memory test failed. |

## Flat-Panel Support

The 65548 BIOS provides support for features used in conjunction with a Flat-Panel display. The type of Flat-Panel display can be changed with the BMP548 utility program. The default Flat-Panel type is a 640x480 monochrome dual-drive LCD.

## Flat-Panel BIOS Operation

The 65548 BIOS operates as if the Flat-Panel is a VGA display independent of the display size. This allows display of all video modes (00h - 13h). Flat-Panels of lower resolution than the video mode will display a partial image that is left and top justified.

## Vertical Compensation

The 65548 BIOS supports the following vertical compensation modes for Flat-Panel operation:

- None — Image is top justified.
- Automatic Centering — Image is automatically centered vertically.
- Non-Automatic Centering — Image display start location is specified.
- Tall Font/Text Compensation — Text is compensated by stretching the font in the hardware.
- Line Replication/Graphic Compensation — Line replication stretches graphics im age to fill the display.

The Vertical Compensation can be set by using function 5F5Eh (Enable/Disable Vertical Compensation). Vertical Compensation status can be read by using function 5F50h, (See F65548 Information).

## Horizontal Compensation

The 65548 BIOS supports the following horizontal compensation modes for Flat-Panel operation:

- None — Image is left justified.
- Automatic Centering — Image is automatically centered horizontally.
- Non-Automatic Centering — Image display start location is specified.
- Text Compression — 720 dot wide Hercules applications can be compressed to fit on 640 horizontal resolution panels by either adding the eighth and ninth pixels or deleting the ninth pixel.
- Automatic Horizontal Doubling — 640 dot wide images can be automatically doubled to fill 1280 dot wide flat panels.

The Horizontal Compensation can be set with function 5F5Fh.

## 8x19 Font Support (Tall Fonts)

The 65548 BIOS supports a special Tall Font in Flat-Panel operation which provides a larger, more readable font without the distortion that can be caused by graphics/text compensation. The Tall Font replaces the standard 8x16 VGA font and can be enabled or disabled with function 5F5Eh (Enable/Disable Text Compensation).

## Inverse Video Switching

The 65548 BIOS supports inverted and non-inverted video display. The BIOS directly initializes the video polarity on boot. The video polarity can be switched with function 5F5Ah (Set Flat-Panel Video Polarity).

## Display Switching

The 65548 BIOS provides support for switching among a Flat-Panel, CRT, CRT Zoom and Simultaneous display at run time. Function 5F51h (Switch Display Device) performs the switching function.

## Simultaneous Display

The 65548 BIOS can be configured to operate an analog CRT monitor and monochrome 640x480 flat panel at the same time. Function 5F51h (Switch Display Device) enables and disables Simultaneous display operation.

## Extended Save and Restore

The 65548 BIOS provides functions to save and restore the state of the 65548 Flat-Panel/CRT VGA controller. This includes all standard and extended registers, the memory latches and the attribute flip/flop state. The functions provided are 5FA0h (Extended BIOS Save/Restore State), 5FA1h (Save Video State), and 5FA2h (Restore Video State).

## SMI and Hot Key Support

An alternate INT 10h entry point (word pointer) is located at 8Bh in the BIOS which will bypass the STI instruction at the beginning of the usual INT 10h handler. STI instructions are not allowed during processing of an SMI, or System Management Interrupt.

The CHIPS 65548 BIOS and CHIPS Flat-Panel Windows drivers are designed to support display switching with hot keys. The following describes how to use the video BIOS to implement hot key display switching. The system BIOS hot key function handler should call the video BIOS switch display function (INT 10h, function 5F51h) when the switch display hot key is pressed.

If the processor is an xSL CPU, the SMI normally handles the hot key. The system BIOS must temporarily patch the INT 10h entry point prior to calling the video BIOS during the SMI. The INT 10h vector should be set to the point to offset 8Bh in the video BIOS, bypassing the STI instruction at the beginning of the usual entry point. After the INT 10h returns to the system BIOS, the system BIOS should restore the original INT 10h vector prior to returning from the hot key interrupt.

The video BIOS switch display function will switch the displays and program a flag in bits 0-3 of extended register XR1F of the 65548 chip. These bits are dedicated to the CHIPS Flat-Panel Windows display drivers.

## Hardware Pop-Up Window Interface

The 65548 VGA Flat-Panel/CRT Controller has the capability of overlaying a 32x32 / 64x64 / 128x128 area of screen with the off-screen data stored in different formats. The off-screen data could be an AND/XOR format cursor (Windows or OS/2), or a monochrome - 2 bit per pixel format bit map. The 65548 can enable both the pop-up or hardware cursor function, but not both at the same time. SMI hot key function controls the switching. Under Windows and OS/2, the display driver has to know when to switch to the software cursor or the hardware cursor. The hot-key usually generates an SMI, and the SMI handler must inform driver about this change. The 65548 BIOS provides an interface between the driver and SMI to support the pop-up through the 5F14H function.

# PCI Support

## Video BIOS

The 65548-PCI Video BIOS is developed for use with PCI Local Bus board configurations as defined in the *PCI Local Bus Specification, Rev 2.0*. The 65548-PCI BIOS has Word Pointers to the PCI Data Structure at offset C000:18h/E000:18h. The PCI Data Structure is defined as follows:

### Table 4: PCI Data Structure

| Offset | Length | Data | Description |
|---|---|---|---|
| 0 | 4 | PCIR | Signature |
| 4 | 2 | 102Ch | Vendor ID |
| 6 | 2 | 00DCh | Device ID |
| 8 | 2 | 0000h | Pointer to vital product data |
| A | 2 | 0018h | PCI data structure length |
| C | 1 | 00h | PCI structure revision |
| D | 3 | 00h,00h,03h | Class code |
| 10 | 2 | ???? | Image Length in 512 Byte units:<br>0040h = 32KB<br>0050h = 40KB<br>0058h = 44KB |
| 12 | 2 | 0000h | Revision level of code/data |
| 14 | 1 | 00h | Code Type |
| 15 | 1 | 80h | Indicator |
| 16 | 2 | 0000h | Reserved |

## System BIOS

The 65548 PCI Flat Panel/CRT controller does not support expansion ROM Base Address at offset 30h in the configuration space. The video BIOS is usually merged with the system BIOS and is located at Address E000:0 in the system. To find the video device during power up, the system BIOS reads Class Code at offset 09h (00h,00h,3h) in the configuration space . The system BIOS then looks for PCIR signatures in the C000/E000 segment (Word Pointer to the PCIR string is at C000:18h/E000:18h) followed by the video device Class Code (00h,00h,03h) at offset 0Dh in the PCI Data Structure of the video BIOS. When the system BIOS finds the video device, it should map the video BIOS at a very high address, then copy the video BIOS at Address C000:0/E000:0.

**Note:** The PCI BIOS should be BMPed for the correct segment.

# CHAPTER 3 - HARDWARE REQUIREMENTS

This chapter describes the external hardware requirements for the 65548 BIOS. For more information regarding hardware implementation, please refer to the *65548 High Performance Flat-Panel/CRT VGA Controllers Data Sheet,* Chips & Technologies, Inc.

## 65548 VGA Flat-Panel/CRT Controller

The 65548 BIOS requires a Chips and Technologies 65548 VGA Flat-Panel/CRT Controller.

## Color Palette RAMDAC

The 65548 BIOS assumes that the 65548 on-chip color palette/DAC is used.

## Monitor Detection Circuitry

The monitor detection circuitry should be implemented in a manner compatible with the IBM AT VGA adapter. The 65548 BIOS assumes that the monitor detection circuitry, if present, is attached to the SENSE pin of the 65548 Flat-Panel/CRT VGA controller.

## Pixel Clocks

The 65548 BIOS requires specific clock frequencies to operate. The 65548 BIOS assumes that the on-chip clock synthesizer generates the clock signals.

Table 5 specifies the values for the Clock Select and Feature Control outputs necessary to use the specified clock source and the frequencies expected by the 65548 BIOS.

### Table 5: Pixel Clock Selection Values and Frequencies

| MSR or XR54 Bits 3,2 | Pixel Clock Selection | Pixel Clock Frequency | Comments |
|---|---|---|---|
| 0, 0 | CLK0 | 25.175 MHz. (Analog) | |
| 0, 1 | CLK1 | 28.322 MHz. (Analog) | |
| 1, X | CLK2 | Programmable | Program XR33,XR30,XR31,XR32 |

CRT Mode:

MSR = Miscellaneous Output Register (Write at 3C2h, Read at 3CCh)

Flat-Panel/Simultaneous Mode:

XR54 = Extended Register (Read/Wri te at 3D6h-3D7h)

## Memory Clock and Dot Clock

The 65548 BIOS allows the user to program the Dot Clock (Pixel Clock) and the Memory Clock using the on-chip clock synthesizer.

# CHAPTER 4 - BIOS FUNCTION CALLS

## Standard VGA BIOS Functions

The 65548 BIOS uses the same function and sub-function calls that are implemented in the IBM VGA BIOS.

## Standard VGA BIOS Function Extensions

The 65548 BIOS provides a set of extended functions that are implemented as extensions to the standard Set Video Mode (00h), Get Video State (0Fh) (see Table 6), and Save/Restore Video State (1Ch) functions. They are fully supported by the other standard VGA BIOS function calls.

### Set Video Mode - 00h

The Set Video Mode function sets the display mode used by the 65548 Flat-Panel/CRT VGA controller.

Calling Registers:
- AH - 00h Set Video Mode
- AL - Display Mode

Return Registers:
- Nothing

### Get Video State - 0Fh

The Get Video State function returns the screen width in character columns, video display mode, and active display page.

Calling Registers:
- AH - 0Fh Get Video State

Return Registers:
- AH - Number of Character Columns
- AL - Display Mode
- BH - Active Display Page

### Save/Restore Video State - 1Ch

This function and its sub-functions save and restore specified video environment parameters (BIOS data area, color palette, and registers). The 65548 BIOS also allows the extended registers and emulation mode registers to be saved. This is specified by CX:Bit 15.

### Get Save/Restore Buffer Size - 00h

Calling Registers:
- AH - 1Ch Save/Restore Video State Function
- AL - 00h Get Save/Restore Buffer Size
- CX - Save/Restore Options:
  - Bit 0 Video Hardware State
  - Bit 1 Video BIOS Data Areas
  - Bit 2 Video DAC State and Color Registers
  - Bit 15 Extended Registers

Return Registers:
- AL - 1Ch Function was successful
- BX - Buffer Size Required in 64 byte blocks

**Save Video State - 01h**

Calling Registers:

| | | | |
|---|---|---|---|
| AH | - | 1Ch | Save/Restore Video State Function |
| AL | - | 01h | Save Video State |
| CX | - | Save/Restore Options: | |
| | | Bit 0 | Video Hardware State |
| | | Bit 1 | Video BIOS Data Areas |
| | | Bit 2 | Video DAC State and Color Registers |
| | | Bit 15 | Extended Registers |
| ES | - | Segment of Save Area | |
| BX | - | Offset of Save Area | |

Return Registers:

| | | | |
|---|---|---|---|
| AL | - | 1Ch | Function was successful |

**Restore Video State - 02h**

Calling Registers:

| | | | |
|---|---|---|---|
| AH | - | 1Ch | Save/Restore Video State Function |
| AL | - | 02h | Save Video State |
| CX | - | Save/Restore Options: | |
| | | Bit 0 | Video Hardware State |
| | | Bit 1 | Video BIOS Data Areas |
| | | Bit 2 | Video DAC state and Color Registers |
| | | Bit 15 | Extended Registers |
| ES | - | Segment of Restore Area | |
| BX | - | Offset of Restore Area | |

Return Registers:

| | | | |
|---|---|---|---|
| AL | - | 1Ch | Function was successful |

## Extended BIOS Functions

The 65548 BIOS provides a set of function calls to control operation of the extended features of the 65548 Flat-Panel/CRT VGA Controller. These function calls are implemented as sub-functions under the extended VGA control function (5Fh).

## Table 6: 65548 BIOS Extended Functions

| AH | AL | Function | AH | AL | Function |
|---|---|---|---|---|---|
| 5Fh | 00h | Get Controller Information | 5Fh | 15h | Notify Video BIOS of 5V mode switch |
| 5Fh | 02h | Set Clock | 5Fh | 50h | Get F65548 Information |
| 5Fh | 04h | Get Refresh Rate Information | 5Fh | 51h | Switch Display Device |
| 5Fh | 05h | Set Refresh Rate Information | 5Fh | 54h | Set Panel ON/OFF |
| 5Fh | 10h | Get Linear Display Memory Information | 5Fh | 55h | Monitor Detect |
| 5Fh | 11h | Get Memory Map I/O Information (PCI) | 5Fh | 56h | Get Panel Type |
| 5Fh | 13h | Set Up Video Memory for Save/Restore | 5Fh | 5Ah | Set Flat-Panel Video Polarity |
| 5Fh | 14h | Set Pop-Up Memory Mode | 5Fh | 5Ch | Set Vertical Compensation |
| 5Fh | 14h | Reset Pop-Up Memory Mode | 5Fh | 5Eh | Enable/Disable Vertical Compensation |
| 5Fh | 14h | Enable Pop-Up | 5Fh | 5Fh | Set Horizontal Compensation |
| 5Fh | 14h | Disable Pop-Up | 5Fh | A0h | Extended BIOS Save/Restore State |
| 5Fh | 14h | Get Pop-up Memory Offset | 5Fh | A1h | Save Video State |
| 5Fh | 15h | Notify Video BIOS of 3.3V/5V mixed | 5Fh | A2h | Restore Video State |

**Get Controller Information - 00h**

This sub-function returns configuration information about the 65548 VGA system.

Calling Registers:
- AH - 5Fh Extended VGA Control Function
- AL - 00h Get controller information

Return Registers:
- AL - 5Fh
- AH - Return Status:
  - 00h Function failed
  - 01h Function was successful
- BL - Chip type and version number
  - D7-D3 = Chip Type
  - 00000b 82C451
  - 00010b 82C452 or 82C452A
  - 00100b 82C455
  - 00110b 82C453
  - 01000b 82C450
  - 01010b 82C456
  - 01100b 82C457
  - 01110b F65520
  - 10000b F65530/525
  - 10010b F65510
  - 10100b F64200
  - 10110b F64300
  - 11000b F65535

11010b F65540
11011b F65545
110111b F65548
D1-D0 = Revision number (65548)
D2-D0 = Revision number (other)

BH Video memory available
0 = 256KB
1 = 512KB
2 = 1MB
3 = Reserved

**Set Clock - 02h**

This sub-function is used to program the internal clock synthesizer with a given frequency. The selected list of frequencies is shown below.

Calling Registers:

| | | | |
|---|---|---|---|
| | AH - | 5Fh | Extended VGA Control Function |
| | AL - | 02h | Set Dot Clock/Memory Clock |
| IF | BH - | 02 | Set Dot Clock |
| | | 03 | Set Memory Clock |
| | BL - | | 12 to 80 MHz (decimal) |
| IF | BH - | FFh | Program Default Dot and Memory Clocks |

Return Registers:

| | | |
|---|---|---|
| AL - | 5Fh | |
| AH - | Return Status: | |
| | 00h | Function failed |
| | 01h | Function was successful |

**Get Refresh Rate Information - 04h (44KB BIOS only)**

This sub-function returns the current setting of the vertical refresh rate for the given mode, and the list of available vertical refresh rates for the mode.

Calling Registers:

| | | |
|---|---|---|
| AH - | 5Fh | Extended VGA BIOS Function |
| AL - | 04h | Get refresh rate |
| BL - | Mode Number | |

Return Registers:

| | | |
|---|---|---|
| BL - | Available refresh rates, indicated by setting one or more bits as follows: | |
| | Bit0 | Interlaced |
| | Bit1 | 56Hz |
| | Bit2 | 60Hz |
| | Bit3 | 70Hz |
| | Bit4 | 72Hz |
| | Bit5 | 75Hz |
| | Bit6 | Reserved |
| | Bit7 | Reserved |
| BH - | Currently set refresh rate, one bit set as above | |
| CX - | Reserved | |
| AL - | 5Fh | Extended VGA BIOS Function Supported |
| AH - | Function Return Status: | |
| | 00h | Function failed |
| | 01h | Function was successful |

**Set Refresh Rate Information - 05h (44KB BIOS only)**

This sub-function sets a new vertical refresh rate for the given mode. If the mode is currently active, the BIOS will program the CRT controller for the new refresh rate. If the mode is not active, the new refresh rate will be programmed the next time this mode is set. The refresh rates are effective in the CRT modes only.

Calling Registers:

| | | |
|---|---|---|
| AH | - | 5Fh Extended VGA BIOS Function |
| AL | - | 05h Set Refresh Rate |
| BL | - | Mode Number |
| BH | - | Set refresh rate, indicated by setting one bit as follows: |
| | | Bit0 Interlaced |
| | | Bit1 56Hz |
| | | Bit2 60Hz |
| | | Bit3 70Hz |
| | | Bit4 72Hz |
| | | Bit5 75Hz |
| | | Bit6 Reserved |
| | | Bit7 Reserved |
| CX | - | Reserved |

Return Registers:

| | | |
|---|---|---|
| AL | - | 5Fh Extended VGA BIOS Function Supported |
| AH | - | Function Return Status: |
| | | 00h Function failed |
| | | 01h Function was successful |

**Get Linear Display Memory Information - 10h**

This sub-function returns information regarding the linear memory starting address, size and width.

Calling Registers:

| | | |
|---|---|---|
| AH | - | 5Fh Extended VGA Control Function |
| AL | - | 10h Get Linear Display Memory Information |

Return Registers:

| | | |
|---|---|---|
| AL | - | 5Fh |
| AH | - | Return Status: |
| | | 00h Function failed |
| | | 01h Function was successful |
| BX | - | Display Memory Base Address (High) |
| CX | - | Display Memory Base Address (Low) |
| SI | - | Display Memory Size (High) |
| DI | - | Display Memory Size (Low) |
| DX | - | Display Width in bytes |

**Get Memory Map I/O Information - 11h (65548-PCI BIOS only)**

This sub-function returns information regarding memory mapped I/O on a PCI configuration.

Calling Registers:

| | | | |
|---|---|---|---|
| AH | - | 5Fh | Extended VGA Control Function |
| AL | - | 11h | Get Memory Mapped I/O Information |

Return Registers:

| | | |
|---|---|---|
| AL | - | 5Fh |
| AH | - | Return Status: |
| | | 00h Function failed |
| | | 01h Function was successful |
| BX | - | Memory Mapped I/O Base Address (High) |
| CX | - | Memory Mapped I/O Base Address (Low) |
| SI | - | Reserved |
| DI | - | Reserved |
| DX | - | Reserved |

**Set Up Video Memory For Save/Restore - 13h (44K BIOS only)**

This sub-function sets up for saving/restoring all of the video memory in 64KB blocks (the appropriate 64KB block is made available at location A000:0000 for reading/writing).

Calling Registers:

| | | |
|---|---|---|
| AX | - | 5F13h |
| BX | - | 0000h |
| CX | - | $n_{th}$ 64K block to save/restore where (n = 0,1,2,....) |

Return Registers:

64K block of video memory will be mapped to A000:0000 for saving/restoring.

**Set Pop-Up Memory Mode - 14h (65548 44KB BIOS - VL-Bus only)**

This sub-function sets the pop-up memory mode. The Video BIOS saves the necessary registers in the 32-byte buffer passed by the SMI, and then sets up registers for dumping the pop-up bit map into the off-screen video memory. If the BLTer is in System to Screen BLT mode, the Video BIOS may return with an error to indicate that video memory cannot be accessed at this time, and that the SMI handler should exit SMI mode in this situation. **Note:** If the function is successful, then the SMI must call the Video BIOS with AX = 5F14h, BL = 01h to restore video controller registers after the SMI loads the pop-up screen into the video memory.

Calling Registers:

| | | |
|---|---|---|
| AX | - | 5F14h Set Pop-Up Function |
| BL | - | 00h Set Pop-Up Memory Mode |
| ES:DX | - | Pointer to 32 byte buffer for Video BIOS in SMI |

Return Registers:

| | | |
|---|---|---|
| ES:DI | - | Pointer to Pop-up screen memory @ A000h |
| BL | - | 00h Function Error (System to Screen BLT mode) |
| | | 01h Function OK |
| AL | - | 5Fh |
| AH | - | Function Return Status: |
| | | 00h Function failed |
| | | 01h Function was successful |

**Reset Pop-Up Memory Mode - 14h (65548 44KB BIOS - VL-Bus only)**

This sub-function resets the pop-up memory mode. The Video BIOS restores all the registers changed by the *Set Pop-Up Memory Function* from the buffer passed by the SMI.

Calling Registers:

| | | |
|---|---|---|
| AX | - | 5F14h Set Pop-Up Function |
| BL | - | 01h Reset Pop-Up Memory Mode |
| ES:DX | - | Pointer to 32 byte buffer for Video BIOS in SMI |

Return Registers:

| | | |
|---|---|---|
| AL | - | 5Fh |
| AH | - | Function Return Status: |
| | | 00h Function failed |
| | | 01h Function was successful |

**Enable Pop-Up - 14h (65548 44KB BIOS - VL-Bus only)**

This sub-function sets and enables the pop-up menu, and notifies the driver to use the software cursor.

Calling Registers:

| | | | |
|---|---|---|---|
| AX | - | 5F14h | Set Pop-Up Function |
| BL | - | 02h | Enable Pop-Up |
| ES:DX | - | Pointer to 32 byte buffer for Video BIOS in SMI Bytes | |
| | | 0-1 | Cursor Color0 |
| | | 2-3 | Cursor Color1 |
| | | 4-5 | Cursor Color2 |
| | | 6-7 | Cursor Color3 |
| | | 8-31 | Video BIOS data area in SMI |
| CX | - | Bits | |
| | | 7-0 | Reserved |
| | | 8 | Pop-up Type |
| | | 0 | 128x128x2 - not implemented |
| | | 1 | 64x64x4 |
| | | 9-11 | Reserved |
| | | 13-12 | Cursor Position |
| | | 00 | Upper Left corner |
| | | 01 | Bottom Left corner |
| | | 10 | Upper Right corner - not implemented |
| | | 11 | Bottom Right corner - not implemented |
| | | 14-15 | Reserved |

Return Registers:

| | | | |
|---|---|---|---|
| AL | - | 5Fh | |
| AH | - | Function Return Status: | |
| | | 00h | Function failed |
| | | 001 | Function was successful |

**Disable Pop-Up - 14h (65548 44KB BIOS - VL-Bus only)**

This sub-function disables the pop-up menu and notifies the driver to use the hardware cursor.

Calling Registers:

| | | | |
|---|---|---|---|
| AX | - | 5F14h | Set Pop-Up Function |
| BL | - | 03h | Disable Pop-Up |
| ES:DX | - | Pointer to 32 byte buffer for Video BIOS in SMI | |

Return Registers:

| | | | |
|---|---|---|---|
| AL | - | 5Fh | |
| AH | - | Function Return Status: | |
| | | 00h | Function failed |
| | | 01h | Function was successful |

**Get Pop-Up Memory Offset - 14h (65548 44KB BIOS - VL-Bus only)**

This sub-function returns pop-up memory offset. This offset should be added to the video memory start address to get the absolute pop-up memory address.

Calling Registers:

| | | | |
|---|---|---|---|
| AX | - | 5F14h | Set Pop-Up function |
| BL | - | 04h | Get Pop-Up memory offset |

Return Registers:

| | | | |
|---|---|---|---|
| BX | - | Pop-up memory address offset (High) | |
| DI | - | Pop-up memory address offset (Low) | |
| AL | - | 5Fh | |
| AH | - | Function Return Status: | |
| | | 00h | Function failed |
| | | 01h | Function was successful |

**Notify Video BIOS for 3.3V/5V mixed mode switch - 15h**

This sub-function notifies the Video BIOS to switch to 3.3V/5V mixed mode. If the Video BIOS cannot switch immediately, then it will set an internal flag to perform the switch later, when switching is possible.

Calling Registers:

| | | | |
|---|---|---|---|
| AX | - | 5F15h | |
| BH | - | 00h | |
| BL | - | 55h | |

Return Registers:

| | | | |
|---|---|---|---|
| BL | - | 00h | Function supported, but cannot change to 3.3V at this time |
| | | 01h | Function supported, switched to 3.3V |
| | | 55h | Function not supported |

**Notify Video BIOS for 5V mode switch - 15h**

This sub-function notifies the Video BIOS to switch to 5V mode. If the Video BIOS cannot switch immediately, then it will set an internal flag to perform the switch later, when switching is possible.

Calling Registers:

| | | | |
|---|---|---|---|
| AX | - | 5F15h | |
| BH | - | 01h | |
| BL | - | 55h | |

Return Registers:

| | | | |
|---|---|---|---|
| BL | - | 01h | Function Supported, switched to 5V |
| | - | 55h | Function not Supported |

**Get F65548 Information - 50h**

This sub-function returns the current CRT/Flat-Panel information.

Calling Registers:

AH - 5Fh Extended VGA Control Function
AL - 50h Get F65548 Information

Return Registers:

AL - 5Fh
AH - Return Status:
- 00h Function failed
- 01h Function was successful

BX - Flat-Panel Horizontal size in pixels
CX - Flat-Panel Vertical size in pixels
DX - F65548 status
- D0 Reserved
- D1 0 = CRT, Check D4 first for CRT Zoom
  1 = Flat-Panel
- D2 0 = Normal Video Polarity
  1 = Inverted Video Polarity
- D3 0 = CRT/Flat-Panel
  1 = Simultaneous
- D4 1 = CRT Zoom Mode, ignore D1 & D3
- D5-D7 Reserved
- D8 0 = Text Compensation Disabled
  1 = Text Compensation Enabled
- D9 0 = Centering Disabled
  1 = Centering Enabled
- D10 0 = Graphics Compensation Disabled
  1 = Graphics Compensation Enabled
- D11 Reserved
- D12 0 = VL/ISA
  1 = PCI
- D13-D15 Reserved

**Switch Display Device - 51h**

This sub-function switches between CRT, Flat-Panel, and Simultaneous displays.

Calling Registers:

| | | | |
|---|---|---|---|
| AH | - | 5Fh | Extended VGA Control Function |
| AL | - | 51h | Switch Display Device |
| BL | - | 00 | Switch to CRT |
| | | 01 | Switch to Flat-Panel |
| | | 02 | Switch to Simultaneous |
| | | 03 | If CRT attached toggle to next disp lay state: LCD → CRT → Simultaneous → LCD |
| | | 04 | Switch to CRT Zoom |

**Note:** If BL = 03 then ES:DX = Pointer to 1K buffer

Return Registers:

| | XR06 D1 | XR51 D2 | XR45 D3 |
|---|---|---|---|
| CRT | 0 | 0 | 0 |
| CRT Zoom | 0 | 0 | 1 |
| Flat-Panel | 1 | 1 | 0 |
| Simultaneous | 0 | 1 | 0 |

AL - 5Fh

AH Return Status:

00h Function failed

01h Function was successful

**Set Panel ON/OFF - 54h**

This sub-function sets the panel ON or OFF. Power consumption is reduced in Panel OFF Mode.

Calling Registers:

AX - 5F54h

BL - Power Down Mode

0 Panel ON

1 Panel OFF

Return Registers:

AL - 5Fh

AH - Return status:

00h Function failed

01h Function was successful

**Monitor Detect - 55h**

This sub-function detects the monitor type and returns the status to the caller.

Calling Registers:

| | | | |
|---|---|---|---|
| AX | - | 5F55h | |
| BX | - | 0001h | |
| CX:DX | - | Pointer to 1K buffer to Save/Restore Color Palette | |

Return Registers:

| | | | |
|---|---|---|---|
| BL | - | 0 | Color CRT detected |
| | | 1 | Monochrome CRT detected |
| | | 2 | No CRT detected |
| AL | - | 5Fh | |
| AH | - | Function Return Status | |
| | | 00h | Function failed |
| | | 01h | Function was successful |

**Get Panel Type - 56h**

This sub-function is used to return panel type information.

Calling Registers:

| | | | |
|---|---|---|---|
| AX | - | 5F56h | |
| BX | - | 0000h | |

Return Registers:

| | | | |
|---|---|---|---|
| BL | - | 0, 1, 2,...7 (Panel #1, 2, 3, ...8) | |
| AL | - | 5Fh | |
| AH | - | Function Return Status | |
| | | 00h | Function failed |
| | | 01h | Function was successful |

**Set Flat-Panel Video Polarity - 5Ah**

This sub-function sets the polarity of the video output to the Flat-Panel in monochrome panels only. In graphics modes, the 65548 BIOS sets/resets XR63 bit 7 to change Video Output Polarity. In Text modes, the 65548 BIOS sets/resets XR61 bit 7 to change Video Output Polarity.

Calling Registers:

| | | | |
|---|---|---|---|
| AH | - | 5Fh | Extended VGA Control Function |
| AL | - | 5Ah | Set Flat-Panel video stream polarity |
| BL | - | 0 | Normal polarity |
| | | 1 | Inverted polarity |
| | | 2 | Toggle polarity |

Return Registers:

| | | | |
|---|---|---|---|
| AL | - | 5Fh | |
| AH | | Function Return Status: | |
| | | 00h | Function failed |
| | | 01h | Function was successful |

**Set Vertical Compensation - 5Ch**

This sub-function sets the type of vertical compensation used.

Calling Registers:

AH - 5Fh Extended VGA Control Function
AL - 5Ch Set Vertical Compensation
BL - 00h No Vertical Compensation
Program:
XR57(bit 0) = 0

01h Automatic Vertical Centering
Program:
XR51(bit 6) = 1
XR57(bit 0) = 1
XR57(bit 1) = 1

02h Set Vertical Centering Register
Program:
XR59(bit 6-5) = DX(bit 9-8)
XR58 = DL

03h Non Automatic Vertical Centering
Program:
XR51(bit 6) = 1
XR57(bit 0) = 1
XR57(bit 1) = 0; Start from Top
(Offset = XR59 bits 6-5 & XR58)

05h Set Alternate Maximum Scanline Register
Program:
XR24(bit 4-0) = DL(bit 4-0)

06h Enable Text Stretching0
Program:
XR51(bit 6) = 1
XR57(bit 0) = 1
XR57(bit 2) = 1
XR57(bit 4-3) = 00 ;DS+LI,DS,LI

07h Enable Text Stretching1
Program:
XR51(bit 6) = 1
XR57(bit 0) = 1
XR57(bit 2) = 1
XR57(bit 4-3) = 01 ;DS+LI,LI,DS

BL - 0Ah Disable Text Stretching
Program:
XR57(bit 2) = 0

0Bh Set Vertical Line Replication Register
Program:
XR5A(bit 3-0) = DL(bit 3-0)

0Ch Enable Graphics Stretching 0
Program:
XR51(bit 6) = 1
XR57(bit 0) = 1
XR57(bit 5) = 1
XR57(bit 6) = 0 ; DS+LR,DS,LR

0Dh Enable Graphics Stretching 1
Program:
XR51(bit 6) = 1
XR57(bit 0) = 1
XR57(bit 5) = 1
XR57(bit 6) = 1 ; DS+LR,LR,DS

0Eh Disable Vertical Graphics Stretching
Program:
XR57(bit 5) = 0

0Fh Disable All Horizontal & Vertical Compensation
Program:
XR51(bit 6) = 0

10h Enable Optimal Compensation

11h Disable Optimal Compensation

Return Registers:

AL - 5Fh

AH - Function Return Status:
00h Function failed
01h Function was successful

**Enable/Disable Vertical Compensation - 5Eh**

This sub-function enables or disables Tall Font. If Text Compensation is enabled (XR0F bit 7 = 1) and in mode 0*, 1*, 2*, 3*, 7, 0+, 1+, 2+, 3+ or 7+, the BIOS will use the hardware Tall Font. Use function 5F50h to receive compensation status.

Calling Registers:

AH - 5Fh Extended VGA Control Function

AL - 5Eh Enable/Disable Vertical Compensation

BL - 0 Enable Text (Tall Font) Compensation
1 Disable Text (Tall Font) Compensation
2 Enable Centering
3 Disable Centering
4 Enable Graphics (Line Replication) Compensation
5 Disable Graphics (Line Replication) Compensation

Return Registers:

AL - 5Fh

AH - Function Return Status:
00h Function failed
01h Function was successful

**Set Horizontal Compensation - 5Fh**

This sub-function sets the type of horizontal compensation used. In Flat-Panel mode, the 65548 BIOS uses an 8-dot font, or a 9-dot font if XR55 bit 2 is programmed to 0. The 65548 BIOS enables Horizontal Text Compression (default) upon power up (see XR55 value) for a 640x480 Flat-Panel so that the 9-dot font will be compressed to 8 dots. The 9-dot font should be used for horizontal panel sizes of 720 pixels or greater.

Calling Registers:

AH - 5Fh Extended VGA Control Function
AL - 5Fh Set Horizontal Compensation
BL - 00h No Horizontal Compensation
Program:
XR55(bit 0) = 0
01h Automatic Horizontal Centering
Program:
XR51(bit 6) = 1
XR55(bit 0) = 1
XR55(bit 1) = 1
02h Set Horizontal Centering Register
Program:
XR56 = DL
03h Non Automatic Horizontal Centering
Program:
XR51(bit 6) = 1
XR55(bit 0) = 1
XR55(bit 1) = 0; Start from Top
(Offset = XR56)
04h Enable Text Compression
Program:
XR51(bit 6) = 1
XR55(bit 0) = 1
XR55(bit 2) = 1; Force 8 dot mode
05h Disable Text Compression
Program:
XR55(bit 2) = 0
06h Enable Horizontal Auto Doubling
Program:
XR51(bit 6) = 1
XR55(bit 0) = 1
XR55(bit 5) = 1; Set Auto Doubling
07h Disable Horizontal Auto Doubling
Program:
XR55(bit 5) = 0
0Fh Disable All Horizontal & Vertical C ompensation
Program:
XR51(bit 6) = 0

Return Registers:

AL - 5Fh
AH - Function Return Status:
00h Function failed
01h Function was successful

**Extended BIOS Save/Restore State - A0h**

This sub-function returns the size of the buffer needed for saving the state of the video system. The user may specify which aspects of the video system are to be saved.

Calling Registers:

| | | |
|---|---|---|
| AH | - | 5Fh Extended VGA Control Function |
| AL | - | A0h Return Save/Restore buffer size |
| CX | - | Requested state: |
| | | Bit 0 Video hardware state |
| | | Bit 1 Video BIOS data state |
| | | Bit 2 DAC state & Color Registers |
| | | Bit 15 Extended registers |

Return Registers:

| | | |
|---|---|---|
| BX | - | Number of 64 Byte blocks required |
| AL | - | 5Fh |
| AH | - | Function Return Status: |
| | | 00h Function failed |
| | | 01h Function was successful |

**Save Video State - A1h**

This sub-function saves information on the current state of the video system.

Calling Registers:

| | | |
|---|---|---|
| AH | - | 5Fh Extended VGA Control Function |
| AL | - | A1h Save Video State |
| CX | - | Save/Restore Options: |
| | | Bit 0 Video Hardware State |
| | | Bit 1 Video BIOS Data Areas |
| | | Bit 2 Video DAC state and Color Registers |
| | | Bit 15 Extended Registers |
| ES | - | Segment of Save Area |
| BX | - | Offset of Save Area |

Return Registers:

| | | |
|---|---|---|
| AL | - | 5Fh Extended VGA Control Function Supported |
| AH | - | Function Return Status: |
| | | 00h Function failed |
| | | 01h Function was successful |

**Restore Video State - A2h**

This sub-function restores a previous state of the video system from stored information.

Calling Registers:

| | | |
|---|---|---|
| AH | - | 5Fh Extended VGA Control Fun ction |
| AL | - | A2h Save Video State |
| CX | - | Save/Restore Options: |
| | | Bit 0 Video Hardware State |
| | | Bit 1 Video BIOS Data Areas |
| | | Bit 2 Video DAC state and Color Registers |
| | | Bit 15 Extended Registers |
| ES | - | Segment of Restore Area |
| BX | - | Offset of Restore Area |

Return Registers:

| | | |
|---|---|---|
| AL | - | 5Fh Extended VGA Control Function Supported |
| AH | - | Function Return Status: |
| | | 00h Function failed |
| | | 01h Function was successful |

## VESA Extended VGA BIOS Functions (44KB BIOS Only)

The 65548 VGA BIOS provides a set of extended function calls as defined by the Video Electronics Standards Association to support Super VGA modes. More information about these functions can be found in the *VESA BIOS Extension Version 2.0* document. These function calls are implemented as sub-functions under the VESA Extended VGA Control Function (4Fh).

### Table 7: VESA Extended VGA BIOS Functions

| AH | AL | Function |
|---|---|---|
| 4Fh | 00h | Return VBE Controller Information |
| 4Fh | 01h | Return VBE Mode Information |
| 4Fh | 02h | Set VBE Mode |
| 4Fh | 03h | Get Current VBE Mode |
| 4Fh | 04h | Save/Restore Video State |
| 4Fh | 05h | Display Window Control |
| 4Fh | 06h | Get/Set Logical Scan Line Length |
| 4Fh | 07h | Get/Set Display Start |
| 4Fh | 09h | Get/Set Palette Data |
| 4Fh | 0Ah | Return VBE Protected Mode Interface |
| 4Fh | 10h | Display Power Management Extensions |

**Return VBE Controller Information - 00h**

This sub-function returns information about capabilities of the 65548 VGA system.

| | | | |
|---|---|---|---|
| Calling Registers: | | | |
| AH | - | 4Fh | VESA Extended VGA Function |
| AL | - | 00h | Return VBE Controller Information |
| ES:DI | - | | Pointer to a 512 byte buffer for the VbeInfoBlock |
| Return Registers: | | | |
| AL | - | 4Fh | VESA extended function supported |
| | - | not 4Fh | Function not supported |
| AH | - | 00h | Function was successful |
| | - | 01h | Function failed |
| | - | 02h | Function is not supported in current hardware configuration |
| | - | 03h | Function is invalid in current video mode |

The information block has the following structure:

```
VbeInfoBlock struc
    VbeSignature        db      'VESA'          ;4 signature bytes
    VbeVersion          dw      200h            ;VESA version number 2.0
    OEMStringPtr        dd      ?               ;pointer to OEM string
    Capabilities        db      4 dup(?)        ;capabilities of the video environment
    VideoModePtr        dd      ?               ;pointer to supported VBE modes
    TotalMemory         dw      ?               ;number of 64KB memory blocks on board
    OemSoftwareRev      dw      ?               ;VBE implementation software revision
    OemVendorNamePtr    dd      ?               ;Pointer to vendor name string
    OemProductNamePtr   dd      ?               ;Pointer to product name string
    OemProductRevPtr    dd      ?               ;Pointer to product revision string
    Reserved            db      222 dup(?)      ;Reserved for VBE implementation scratch  area
    OemData             db      256 dup(?)      ;Data for OEM strings
VbeInfoBlock ends
```

**Return VBE Mode Information - 01h**

This sub-function returns information about a specific VBE mode.

| | | | |
|---|---|---|---|
| Calling Registers: | | | |
| AH | - | 4Fh | VESA Extended Function |
| AL | - | 01h | Return VBE mode information |
| CX | - | | Video mode number |
| ES:DI | - | | Pointer to a 256 byte buffer for the ModeInfoBlock |
| Return Registers: | | | |
| AL | - | 4Fh | VESA extended VGA  function supported |
| | - | not 4Fh | Function not supported |
| AH | - | Status | See function 00 for status codes |

The mode information block has the following structure:

```
ModeInfoBlock   struc
        ModeAttributes          dw      ?               ;mode attributes
        WinAAttributes          db      ?               ;window A attributes
        WinBAttributes          db      ?               ;window B attributes
        WinGranularity          dw      ?               ;window granularity
        WinSize                 dw      ?               ;window size
        WinASegment             dw      ?               ;window A start segment
        WinBSegment             dw      ?               ;window B start segment
        WinFuncPtr              dd      ?               ;pointer to window function
        BytesPerScanLine        dw      ?               ;bytes per scan line
        XResolution             dw      ?               ;horizontal resolution
        YResolution             dw      ?               ;vertical resolution
        XCharSize               db      ?               ;character cell width
        YCharSize               db      ?               ;character cell height
        NumberOfPlanes          db      ?               ;number of memory planes
        BitsPerPixel            db      ?               ;bits per pixel
        NumberOfBanks           db      ?               ;number of banks
        MemoryModel             db      ?               ;memory model type
        BankSize                db      ?               ;bank size in KB
        NumberOfImagePages      db      ?               ;Number of images
        Reserved                db      1               ;reserved for page function
        RedMaskSize             db      ?               ;size if direct color red mask in bits
        RedFieldPosition        db      ?               ;bit position of lsb of red mask
        GreenMaskSize           db      ?               ;size of direct color green mask in bits
        GreenFieldPosition      db      ?               ;bit position of lsb of green mask
        BlueMaskSize            db      ?               ;size of direct color blue mask in bits
        BlueFieldPosition       db      ?               ;bit position of lsb of blue mask
        RsvdMaskSize            db      ?               ;size of direct color reserved mask in bits
        RsvdFieldPosition       db      ?               ;bit position of lsb of reserved mask
        DirectColorModeInfo     db      ?               ;direct color mode attributes
        PhysBasePtr             dd      ?               ;physical address for linear frame buffer
        OffScreenMemOffset      dd      ?               ;pointer to start of offscreen memory
        OffScreenMemSize        dw      ?               ;amount of offscreen memory in 1K units
        Reserved                db      206 dup(?)      ;remainder of ModeInfoBlock
ModeInfoBlock   ends
```

**Set VBE Mode - 02h**

This sub-function sets a given super VGA mode.

Calling Registers:

| | | | |
|---|---|---|---|
| AH | - | 4Fh | VESA Extended BIOS Function |
| AL | - | 02h | Set VBE video mode |
| BX | - | D0-D13 - | video mode number |
| | | D14 - | 0 = use windowed (paged) frame buffer |
| | | | 1 = use linear (flat) frame buffer |
| | | D15 - | 0 = clear video memory |
| | | | 1 = do not clear video memory |

Return Registers:

| | | | |
|---|---|---|---|
| AL | - | 4Fh | VESA extended VGA function supported |
| | - | not 4Fh | Function not supported |
| AH | - | Status | See function 00 for status codes |

**Get Current VBE Mode - 03h**

This sub-function returns the current video mode.

Calling Registers:

| | | | |
|---|---|---|---|
| AH | - | 4Fh | VESA Extended BIOS Function |
| AL | - | 03h | Get current video mode |

Return Registers:

| | | | |
|---|---|---|---|
| AL | - | 4Fh | VESA extended VGA function supported |
| | - | not 4Fh | Function not supported |
| AH | - | Status | See function 00 for status codes |
| BX | - | | Current video mode number |
| | | D0-13 - | mode number |
| | | D14 - | 0 = use windowed (paged) frame buffer |
| | | | 1 = use linear (flat) frame buffer |
| | | D15 - | 0 = clear video memory |
| | | | 1 = do not clear video memory |

**Save/Restore Video State - 04h**

These sub-functions provide a mechanism for saving and restoring the video state. The functions are a superset of the three sub-functions under the standard VGA BIOS function 1Ch.

| | | | |
|---|---|---|---|
| Calling Registers: | | | |
| AH | - | 4Fh | VESA Extended BIOS Function |
| AL | - | 04h | Save/restore video state |
| **DL** | - | **00h** | **Return save/restore state buffer size** |
| CX | - | Requested states | |
| | | D0 - | Save/restore video hardware state |
| | | D1 - | Save/restore video BIOS data state |
| | | D2 - | Save/restore video DAC state |
| | | D3 - | Save/restore extended video state |
| Return Registers: | | | |
| AL | - | 4Fh | VESA extended VGA function supported |
| | - | not 4Fh | Function not supported |
| AH | - | Status | See function 00 for status codes |
| BX | - | | Number of 64 byte blocks needed to hold the state buffer |
| Calling Registers: | | | |
| AH | - | 4Fh | VESA Extended BIOS Function |
| AL | - | 04h | Save/restore video state |
| **DL** | - | **01h** | **Save video state** |
| CX | - | Requested states | |
| | | D0 - | Save/restore video hardware state |
| | | D1 - | Save/restore video BIOS data state |
| | | D2 - | Save/restore video DAC state |
| | | D3 - | Save/restore extended video state |
| ES:BX | - | | Pointer to state buffer |
| Return Registers: | | | |
| AL | - | 4Fh | VESA extended BIOS function supported |
| | - | not 4Fh | Function not supported |
| AH | - | Status | See function 00 for status code s |

Calling Registers:

| | | | |
|---|---|---|---|
| AH | - | 4Fh | VESA Extended BIOSction |
| AL | - | 04h | Save/restore video state |
| **DL** | **-** | **02h** | **Restore video state** |
| CX | - | Requested states | |
| | | D0 - | Save/restore video hardware state |
| | | D1 - | Save/restore video BIOS data state |
| | | D2 - | Save/restore video DAC state |
| | | D3 - | Save/restore extended video state |
| ES:BX | - | | Pointer to state buffer |

Return Registers:

| | | | |
|---|---|---|---|
| AL | - | 4Fh | VESA extended BIOS function supported |
| | - | not 4Fh | Function not supported |
| AH | - | Status | See function 00 for status co des |

**Display Window Control - 05h**

These sub-functions set or get the position of the specified window in the video memory.

Calling Registers:

| | | | |
|---|---|---|---|
| AH | - | 4Fh | VESA Extended BIOS Function |
| AL | - | 05h | Display window control |
| **BH** | **-** | **00h** | **Select display window** |
| BL | - | Window number | |
| | | 0 = Window A | |
| | | 1 = Window B | |
| DX | - | Window position in video memory (in window granularity units) | |

Return Registers:

| | | | |
|---|---|---|---|
| AL | - | 4Fh | VESA extended BIOS function supported |
| | - | not 4Fh | Function not supported |
| AH | - | Status | See function 00 for status codes |

Calling Registers:

| | | | |
|---|---|---|---|
| AH | - | 4Fh | VESA Extended BIOS Function |
| AL | - | 05h | Display window control |
| **BH** | **-** | **01h** | **Return display window** |
| BL | - | Window number | |
| | | 0 = Window A | |
| | | 1 = Window B | |

Return Registers:

| | | | |
|---|---|---|---|
| AL | - | 4Fh | VESA extended VGA function supported |
| | - | not 4Fh | Function not supported |
| AH | - | Status | See function 00 for status codes |
| DX | - | Window position in video memory (in window granularity units) | |

**Get/Set Logical Scan Line Length - 06h**

These sub-functions set or get the length of a logical scan line.

| | | | |
|---|---|---|---|
| Calling Registers: | | | |
| AH | - | 4Fh | VESA Extended BIOS Function |
| AL | - | 06h | Logical scan line length control |
| **BL** | - | **00h** | **Set logical scan line length in pixels** |
| | - | **01h** | **Return logical scan line length** |
| | - | **02h** | **Set logical scan line length in bytes** |
| | - | **03h** | **Get maximum scan line length** |
| CX | - | | If BL=00h, desired scan line length in pixels<br>If BL=02h, desired scan linelength in bytes<br>(ignored for get functions) |
| Return Registers: | | | |
| AL | - | 4Fh | VESA extended BIOS functio n supported |
| | - | not 4Fh | Function not supported |
| AH | - | Status | See function 00 for status codes |
| BX | - | | Bytes per scan line |
| CX | - | | Actual pixels per scan line |
| DX | - | | Maximum number of scan lines |

**Get/Set Display Start - 07h**

These sub-functions set or get the pixel to be displayed in the upper left corner of the display from the logical page.

| | | | |
|---|---|---|---|
| Calling Registers: | | | |
| AH | - | 4Fh | VESA Extended BIOS Function |
| AL | - | 07h | Display start control functions |
| BH | - | 00h | Reserved, must be 0 |
| **BL** | - | **00h** | **Set display start** |
| | - | **01h** | **Get display start** |
| | - | **80h** | **Set display start during vertical retrace** |
| CX | - | | First displayed pixel in the scan line (Set only) |
| DX | - | | First displayed scan line (Set only) |
| Return Registers: | | | |
| AL | - | 4Fh | VESA extended VGA function supported |
| | - | not 4Fh | Function not supported |
| AH | - | Status | See function 00 for status codes |
| CX | - | | First displayed pixel in scan line (Get only) |
| DX | - | | First displayed scan line (Get only) |

**Get/Set Palette Data - 09h**

These functions are used to get or set the palette registers in the RAMDAC.

| | | | |
|---|---|---|---|
| Calling Registers: | | | |
| AH | - | 4Fh | VESA Extended BIOS Function |
| AL | - | 09h | Get/Set Palette Data |
| **BL** | **-** | **00h** | **Set Palette Data** |
| | **-** | **01h** | **Get Palette Data** |
| CX | - | | Number of palette registers to update (max 255) |
| DX | - | | First palette register index to update |
| ES:DI | - | | Table of palette values, in the format: Alignment byte, red byte, green byte, blue byte |
| Return Registers: | | | |
| AL | - | 4Fh | VESA extended BIOS function supported |
| | - | not 4Fh | Function not supported |
| AH | - | Status | See function 00 for the possible return codes |

**Return VBE Protected Mode Interface - 0Ah**

This function returns a pointer to a table that contains code for a 32-bit protected mode interface that can be either copied into local 32-bit memory space or can be executed from ROM providing the calling application sets all required selectors and I/O access correctly.

| | | | |
|---|---|---|---|
| Calling Registers: | | | |
| AH | - | 4Fh | VESA Extended BIOS Function |
| AL | - | 0Ah | Return VBE protected mode information |
| **BL** | **-** | **00h** | **Return VBE protected mode information** |
| Return Registers: | | | |
| AL | - | 4Fh | VESA extended BIOS function supported |
| | - | not 4Fh | Function not supported |
| AH | - | Status | See function 00 for a list of possible return codes |
| ES | - | | Real mode segment of table |
| DI | - | | Offset of table |
| CX | - | | Length of table including protected mode code, in bytes |

Table format:

| | |
|---|---|
| ES:DI | Word offset in table of protected mode code for the Set Window portion of Function 05h. |
| ES:DI+2 | Word offset in table of protected mode code for the Set Displa y Start portion of Function 07h. |
| ES:DI+4 | Word offset in table of protected mode code for the Set Palette Data portion of Function 09h. |
| ES:DI+6 | Word offset in table of a list of ports and memory locations that the calling application may need I/O privilege for. |
| ES:DI+? | Variable length remainder of table, including code. |

**Display Power Management Extensions - 10h (44KB BIOS Only)**

The VESA BIOS Extension sub-function 10h is used to implement power management services.

Calling Registers:

| | | | |
|---|---|---|---|
| AH | - | 4Fh | VESA Extended BIOS Function |
| AL | - | 10h | Power management services |
| **BL** | **-** | **00h** | **Report VBE/Power Management capabilities** |
| ES:DI | - | | Null pointer, must be 0000:0000 in version 1.0, reserved for future use |

Return Registers:

| | | | |
|---|---|---|---|
| AL | - | 4Fh | VESA extended BIOS function supported |
| | - | not 4Fh | Function not supported |
| AH | - | 00h | Function was successful |
| | - | 01h | Function failed |

BH - Power saving state signals supported by the controller.
1 = supported, 0 = not supported

| | |
|---|---|
| bit 0 | STANDBY |
| bit 1 | SUSPEND |
| bit 2 | OFF |
| bit 3 | REDUCED ON |
| bits 4-7 | Reserved for future use. |

BL - VBE/PM version number

| | |
|---|---|
| bits 0-3 | Minor version number |
| bits 4-7 | Major version number |

ES:DI Unchanged

Calling Registers:

| | | | |
|---|---|---|---|
| AH | - | 4Fh | VESA Extended BIOS Function |
| AL | - | 10h | Power management services |
| **BL** | **-** | **01h** | **Set display power state** |
| BH | - | Requested power state | |
| | | 00h | ON |
| | | 01h | STANDBY |
| | | 02h | SUSPEND |
| | | 04h | OFF |
| | | 08h | REDUCED ON |

Return Registers:

| | | | |
|---|---|---|---|
| AL | - | 4Fh | VESA extended BIOS function supported |
| | - | not 4Fh | Function not supported |
| AH | - | 00h | Function was successful |
| | - | 01h | Function failed |
| BH | - | | Unchanged |

Calling Registers:

| | | | |
|---|---|---|---|
| AH | - | 4Fh | VESA Extended BIOS Function |
| AL | - | 10h | Power management services |
| **BL** | **-** | **02h** | **Get display power state** |

Return Registers:

| | | | |
|---|---|---|---|
| AL | - | 4Fh | VESA extended BIOS function supported |
| | - | not 4Fh | Function not supported |
| AH | - | 00h | Function was successful |
| | - | 01h | Function failed |
| BH | - | Power state currently requested by the controller | |
| | | 00h | ON |
| | | 01h | STANDBY |
| | | 02h | SUSPEND |
| | | 04h | OFF |
| | | 08h | REDUCED ON |

## INT 15h/INT 42h Hooks for the System BIOS

The video BIOS performs several types of INT 15h function calls. Each function provides the system BIOS with the opportunity to gain control at specific times to perform any custom processing that may be required during video POST or mode changes. With all the INT 15h/INT 42h functions described here, the system BIOS must return to the video BIOS at the conclusion of the function. Some functions can be BMPed to INT 15h/INT 42h/Disabled. These functions are implemented at the discretion of the system BIOS designer.

### Table 8: INT 15h/INT 42h Hooks for the System BIOS

| AH | AL | Function | Hook (INT) / BMP option |
|---|---|---|---|
| 5Fh | 31h | POST Completion Notification | 15h |
| 5Fh | 33h | Hook After Mode Set | Set INT 15h or INT 42h, AX = 5F33/38/39h |
| 5Fh | 34h | Multiple Panel Support (32KB BIOS) | 15h |
| 5Fh | 34h | Set Panel Type (44KB BIOS) | 15h |
| 5Fh | 40h | Set Panel Type (44KB BIOS) *Alternate Method* | 15h |
| 5Fh | 38h | Hook Before Mode Set | Set INT 15h or INT 42h, AX = 5F33/38/39h |
| 5Fh | 39h | 3.3V/5V Power Switching (32KB & 44KB BIOS) | |
| 5Fh | 42h | 3.3V/5V Power Switching (44KB BIOS) *Alternate Method* | Enable/Disable 5F42h, INT 15h Voltage Switching |

## POST Completion Notification - 5F31h (INT 15h Hook)

This function signals the completion of the video BIOS power-up initialization procedure. The function executes just before the sign-on message is displayed, allowing the OEM's system BIOS to switch to a different display before attempting to display the sign-on message.

Calling Registers:

AX - 5F31h

Return Registers:

None required.

## Hook After Mode Set - 5F33h (BMP option: Set INT 15h or INT 42h, AX = 5F33/38/39h)

This function allows the system BIOS to intercept the video BIOS at the end of a mode set. The OEM has the option of enabling this feature in the BMP to specify whether the interrupt call is INT 15h or INT 42h. Whichever interrupt is chosen, the calling registers remain the same. This function is blocked during power up.

Calling Registers:

| | | |
|---|---|---|
| AX | - | 5F33h |
| BH | - | Number of character columns |
| BL | - | Current mode number |
| CH | - | Active display page |

Return Registers:

None required.

## Multiple Panel Support - 5F34h (INT 15h Hook for 32KB BIOS only)

This function allows the OEM's system BIOS to support multiple flat panel types. During POST, the VGA BIOS calls INT 15h with AX=5F34h. To take advantage of this feature, the OEM determines the flat panel type by special switches, circuitry, CMOS configuration, or other methods. The OEM can then load the correct panel parameters into the video BIOS. To work, the system BIOS must load the video BIOS into shadow RAM and enable writes to the shadow RAM. Enabling the system BIOS to write to a programmable EPROM is another alternative. The BIOS provides the following pointers at the specified offsets into the video BIOS:

(The BMS file for each flat panel contains the appropriate values to be loaded into these tables for that flat panel)

Seven data locations for memory clock & dot clock from 188h to 18Eh:
(All table entries in the BMS file are in Decimal)

Flat panel mode Dot clock freq **FP_Dot** at 188h.
Flat panel mode Mem clock freq **FP_MCLK** at 189h.
Simultaneous mode Dot clock freq **SM_Dot** at 18Ah.
Simultaneous mode Mem clock freq **SM_MCLK** at 18Bh.
Flat panel 15/16 bpp mode Dot clock freq **FP_Dot_15_16Bit** at 18Ch.
Simultaneous 15/16bpp mode Dot clock freq **SM_Dot_15_16Bit** at 18Dh.
15/16bpp mode Mem clock freq **Mem_15_16_Bit** at 18Eh.

Panel Table Pointers: (All table entries in the BMS file are in Hex)

Pointer of **ControlPtr1** table at location 1E7h.
Pointer of **FPSimBootPtr** table at location 1EDh.
Pointer of **SMCommonPtr** table at location 1F3h.
Pointer of **FPCommonPtr** table at location 1F9h.

**NOTE:** (32KB & 44KB BIOS)

1. If the new panel tables are loaded into the video BIOS space, the checksum (in bytes) of the BMP needs to be adjusted at location C000:6Bh/E000:6Bh. The BMP structure starts at location C000:47h/E000:47h and the length (in words) of the BMP structure is available at location C000:68h/E000:68h.

2. The panel table pointer points to a structure that consists of a list of pairs of bytes. The first byte contains the index of the XR register and the second byte contains the value. These table entries are in the .BMS file under corresponding sections.

## Set Panel Type - 5F34h (INT 15h Hook for 44KB BIOS only)

This function call allows the OEM to select one of eight panel types upon power up. The VGA BIOS calls INT 15h with AX = 5F34h. The system BIOS should modify C000:185h/E000:185h for the correct panel type. The system BIOS is also required to adjust checksum. (See Note in *32KB Multiple Panel Support - 5F34h* description).

### Table 9: Panels

| Panel # | Panel Type |
|---|---|
| 1 | 640x480 Dual Scan Monochrome Panel |
| 2 | 640x480 Dual Scan Monochrome Panel No Accel |
| 3 | 640x480 Dual Scan Color Panel |
| 4 | 800x600 Dual Scan Color Panel |
| 5 | 640x480 Sharp TFT Color |
| 6 | 640x480 18-bit TFT Color |
| 7 | 1024x768 TFT Color |
| 8 | 800x600 TFT Color |

## Set Panel Type - 5F40h (Alternate Method - INT 15h Hook for 44KB BIOS only )

This function call allows the system BIOS to select one of eight LCD panel types upon power up (see Table 9). The VGA BIOS calls INT 15h with AX = 5F40h and CL = 55h. Upon exit, the system BIOS will return in CL a panel type value between 0 and 7. A value of 0 in CL corresponds to Panel #1. This function does not require the system BIOS to modify C000:185h/E000:185, unlike 5F34h (44KB BIOS).

Calling Registers:
- AX - 5F40h
- CL - 55h

Return Registers:
- CL - 0 - 7 (panel type)

## Monitor Sensing Hook - 5F35h (INT 15h Hook)

This function call allows the system BIOS to perform monitor sensing and to override the setting in the BMP. The system BIOS will then boot from the display type specified in the return from this call.

Calling Registers:
- AX - 5F35h
- DL - Invalid return code

Return Registers:
- DL - If unchanged, boot according to the BMP setting
- If changed, boot from the following display:
  - 0 - set to CRT mode
  - 1 - set to flat panel mode
  - 2 - set to simultaneous mode

## Hook Before Mode Set - 5F38h (BMP option: Set INT 15h or INT 42h, AX = 5F33/38/39h)

This function call allows the system BIOS to intercept the video BIOS before setting the mode. The OEM has the option of enabling this feature in the BMP, and if enabled, of specifying whether the interrupt call is INT 15h or INT 42h. Whichever interrupt is chosen, the calling registers remain the same. This function is blocked during power up.

Calling Registers:

| | | |
|---|---|---|
| AX | - | 5F38h |
| CL | - | New Video Mode To Be Set |

## 3.3V/5V Power Switching - 5F39h (For dual power supply systems - 32KB and 44KB BIOS) (BMP option: Set INT 15h or INT 42h, AX=5F33/38/39h)

The 65548 supports dual voltage (3.3V/5V) switching. This function allows the system BIOS to switch between a 3.3V and 5V power supply depending on the video mode and display type (CRT, LCD, or simultaneous). The video BIOS calls 5F39h during Mode Set and display type switching. The system BIOS should program XR6C Bit 1 and switch the power supply to the correct voltage (3.3V or 5V). The system BIOS also requires the 3.3V and 5V mode tables to switch to the correct voltages. This function is blocked during power up.

Calling Registers:

| | | |
|---|---|---|
| AX | - | 5F39h |
| CL | - | Current Video Mode |

INT 15h/INT 42h/Disabled (BMP Option)

Return Registers:

The system BIOS should program XR6C Bit 1 and switch the power supply to the correct voltage.
(Example: Run all Extended Modes in 5V and run all Standard Modes 0-13h in 3.3 V)
0 - 3.3V
1 - 5.0V

## 3.3V/5V Power Switching - 5F42h (For dual power supply systems - Alternate Method for 44KB BIOS only) (BMP option: Enable/Disable 5F42h, INT 15h Voltage Switching)

This function call allows the system BIOS to switch between a 3.3V and 5V power supply. This function does not require 3.3V and 5V mode determination support (unlike 5F39h) in the system BIOS.

Calling Registers:

INT 15h

| | | | |
|---|---|---|---|
| AX | - | 5F42h | |
| BH | - | 00h | Switch from 5V to 3.3V |
| | - | 01h | Switch from 3.3V to 5V |
| | - | 02h | Desire to switch to 5V |
| BL | - | 55h | |

Return Registers:

| | | | |
|---|---|---|---|
| BL | - | 01h | Function supported and switch su ccessful |
| | - | 00h | Function supported, but switch unsuccessful |
| | - | 55h | Function not supported |

**Note:** The following steps are required to enable voltage switching:
1. Use the BMP to set up tables in the BIOS for 3.3/5V operation.
2. Enable the mixed mode of operation with INT 15h AX=5F15 BX=0055
3. Monitor 5F42h function call for BH=02h, return BL=01h if the function is supported.
4. Monitor 5F42h function call for BH=00h or 01h. Make appropriate voltage switch and return BL=01h.

## Table 10: Standard Video Display Modes

| Video Mode | Mode Type | Display Adapter | Pixel Resolution | Font Size | Displayed Characters | Colors | Dot Clock (MHz) | Horiz. Freq. (KHz) | Vert Freq (Hz) | Video Memory (KB) |
|---|---|---|---|---|---|---|---|---|---|---|
| 00h | Text | CGA | 320x200 | 8x8 | 40x25 | 16(gray) | 25 | 31.5 | 70 | 256 |
| | | EGA[2] | 320x350 | 8x14 | 40x25 | 16(gray) | 25 | 31.5 | 70 | 256 |
| | | VGA[1] | 360x400 | 9x16 | 40x25 | 16 | 28 | 31.5 | 70 | 256 |
| 01h | Text | CGA | 320x200 | 8x8 | 40x25 | 16 | 25 | 31.5 | 70 | 256 |
| | | EGA[2] | 320x350 | 8x14 | 40x25 | 16 | 25 | 31.5 | 70 | 256 |
| | | VGA[1] | 360x400 | 9x16 | 40x25 | 16 | 28 | 31.5 | 70 | 256 |
| 02h | Text | CGA | 640x200 | 8x8 | 80x25 | 16(gray) | 25 | 31.5 | 70 | 256 |
| | | EGA[2] | 640x350 | 8x14 | 80x25 | 16(gray) | 25 | 31.5 | 70 | 256 |
| | | VGA[1] | 720x400 | 9x16 | 80x25 | 16 | 28 | 31.5 | 70 | 256 |
| 03h | Text | CGA | 640x200 | 8x8 | 80x25 | 16 | 25 | 31.5 | 70 | 256 |
| | | EGA[2] | 640x350 | 8x14 | 80x25 | 16 | 25 | 31.5 | 70 | 256 |
| | | VGA[1] | 720x400 | 9x16 | 80x25 | 16 | 28 | 31.5 | 70 | 256 |
| 04h | Graph | All | 320x200 | 8x8 | 40x25 | 4 | 25 | 31.5 | 70 | 256 |
| 05h | Graph | CGA | 320x200 | 8x8 | 40x25 | 4(gray) | 25 | 31.5 | 70 | 256 |
| | | EGA | 320x200 | 8x8 | 40x25 | 4(gray) | 25 | 31.5 | 70 | 256 |
| | | VGA | 320x200 | 8x8 | 40x25 | 4 | 25 | 31.5 | 70 | 256 |
| 06h | Graph | All | 640x200 | 8x8 | 80x25 | 2 | 25 | 31.5 | 70 | 256 |
| 07h | Text | MDA | 720x350 | 9x14 | 80x25 | Mono | 28 | 31.5 | 70 | 256 |
| | | EGA | 720x350 | 9x14 | 80x25 | Mono | 28 | 31.5 | 70 | 256 |
| | | VGA | 720x400 | 9x16 | 80x25 | Mono | 28 | 31.5 | 70 | 256 |
| 08h - 0Ch | | - | Reserved | | - | | | | | |
| 0Dh | Graph | E/VGA | 320x200 | 8x8 | 40x25 | 16 | 25 | 31.5 | 70 | 256 |
| 0Eh | Graph | E/VGA | 640x200 | 8x8 | 80x25 | 16 | 25 | 31.5 | 70 | 256 |
| 0Fh | Graph | E/VGA | 640x350 | 8x14 | 80x25 | Mono | 25 | 31.5 | 70 | 256 |
| 10h | Graph | E/VGA | 640x350 | 8x14 | 80x25 | 16 | 25 | 31.5 | 70 | 256 |
| 11h | Graph | VGA | 640x480 | 8x16 | 80x30 | 2 | 25 | 31.5 | 60 | 256 |
| 12h | Graph | VGA | 640x480 | 8x16 | 80x30 | 16 | 25 | 31.5 | 60 | 256 |
| 13h | Graph | VGA | 320x200 | 8x8 | 40x25 | 256 | 25 | 31.5 | 70 | 256 |

**Note:** 1. Enhanced VGA mode. Otherwise, the VGA can emulate either the CGA or EGA characteristics of this mode.
2. The availability of these modes is dependent upon hardware & software configuration.

## Table 11: Extended Video Modes

| Video Mode | Mode Type | Display Adapter | Pixel Resolution | Font Size | Displayed Characters | Colors | Dot Clock (MHz) | Horiz. Freq. (KHz) | Vert Freq (Hz) | Video Memory (KB) |
|---|---|---|---|---|---|---|---|---|---|---|
| 20h | Graph(L) | VGA | 640x480 | 8x16 | 80x30 | 16 | 25.175 | 31.5 | 60 | 512 |
| | | | | | | | 31.5 | 37.9 | 72 | 512 |
| | | | | | | | 31.5 | 37.5 | 75 | 512 |
| 22h | Graph(L) | VGA | 800x600 | 8x16 | 100x37 | 16 | 36 | 35.1 | 56 | 512 |
| | | | | | | | 40 | 37.9 | 60 | 512 |
| | | | | | | | 50 | 48.1 | 72 | 512 |
| | | | | | | | 49.5 | 46.9 | 75 | 512 |
| 24h(I) | Graph(L) | VGA | 1024x768 | 8x16 | 128x48 | 16 | 44.9 | 35.5 | 43 | 512 |
| 24h | Graph(L) | VGA | 1024x768 | 8x16 | 128x48 | 16 | 65 | 48.4 | 60 | 512 |
| | | | | | | | 75 | 56.5 | 70 | 512 |
| | | | | | | | 78.75 | 60 | 75 | 512 |
| 30h | Graph(L) | VGA | 640x480 | 8x16 | 80x30 | 256 | 25.175 | 31.5 | 60 | 512 |
| | | | | | | | 31.5 | 37.9 | 72 | 512 |
| | | | | | | | 31.5 | 37.5 | 75 | 256 |
| 32h | Graph(L) | VGA | 800x600 | 8x16 | 100x37 | 256 | 36 | 35.1 | 56 | 512 |
| | | | | | | | 40 | 37.9 | 60 | 512 |
| | | | | | | | 50 | 48.1 | 72 | 512 |
| | | | | | | | 49.5 | 46.9 | 75 | 512 |
| 34h(I) | Graph(L) | VGA | 1024x768 | 8x16 | 128x48 | 256 | 44.9 | 35.5 | 43 | 1024 |
| 34h | Graph(L) | VGA | 1024x768 | 8x16 | 128x48 | 256 | 65 | 48.4 | 60 | 1024 |
| | | | | | | | 75 | 56.5 | 70 | 1024 |
| | | | | | | | 78.75 | 60 | 75 | 1024 |
| 40h | Graph(L) | VGA | 640x480 | 8x16 | 80x30 | 32K | 50.35 | 31.5 | 60 | 1024 |
| | | | | | | | 63 | 37.5 | 72 | 1024 |
| | | | | | | | 63 | 37.9 | 75 | 1024 |
| 41h | Graph(L) | VGA | 640x480 | 8x16 | 80x30 | 64K | 50.35 | 31.5 | 60 | 1024 |
| | | | | | | | 63 | 37.5 | 72 | 1024 |
| | | | | | | | 63 | 37.9 | 75 | 1024 |
| 42h | Graph(L) | VGA | 800x600 | 8x16 | 100x37 | 32K | 72 | 35.1 | 56 | 1024 |
| | | | | | | | 80 | 37.8 | 60 | 1024 |
| 43h | Graph(L) | VGA | 800x600 | 8x16 | 100x37 | 64K | 72 | 35.1 | 56 | 1024 |
| | | | | | | | 80 | 37.8 | 60 | 1024 |
| 50h | Graph(L) | VGA | 640x480 | 8x16 | 80x30 | 16M | 75.525 | 31.5 | 60 | 1024 |
| 60h | Text | VGA | 1056x400 | 8x16 | 132x25 | 16 | 41.539 | 31.5 | 70 | 256 |
| 61h | Text | VGA | 1056x400 | 8x8 | 132x50 | 16 | 41.539 | 31.5 | 70 | 256 |
| 6A/70h | Graph | VGA | 800x600 | 8x16 | 100x37 | 16 | 36 | 35.1 | 56 | 256 |
| | | | | | | | 40 | 37.8 | 60 | 256 |
| | | | | | | | 50 | 48.1 | 72 | 256 |
| | | | | | | | 49.5 | 46.9 | 75 | 256 |
| 72h/75(I) | Graph | VGA | 1024x768 | 8x16 | 128x48 | 16 | 44.9 | 35.5 | 43 | 512 |
| 72h/75 | Graph | VGA | 1024x768 | 8x16 | 128x48 | 16 | 65 | 48.4 | 60 | 512 |
| | | | | | | | 75 | 56.5 | 70 | 512 |
| | | | | | | | 78.75 | 60 | 75 | 512 |
| 79h | Graph | VGA | 640x480 | 8x16 | 80x30 | 256 | 25.175 | 31.5 | 60 | 512 |
| | | | | | | 256 | 31.5 | 37.9 | 72 | 512 |
| | | | | | | 256 | 31.5 | 37.5 | 75 | 512 |
| 7Ch | Graph | VGA | 800x600 | 8x16 | 100x37 | 256 | 36 | 35.5 | 56 | 512 |
| | | | | | | 256 | 40 | 37.9 | 60 | 512 |
| | | | | | | 256 | 50 | 48.1 | 72 | 512 |
| | | | | | | 256 | 49.5 | 46.9 | 75 | 512 |
| 7Eh(I) | Graph | VGA | 1024x768 | 8x16 | 128x48 | 256 | 44.9 | 35.5 | 43 | 1024 |
| 7Eh | Graph | VGA | 1024x768 | 8x16 | 128x48 | 256 | 65 | 48.5 | 60 | 1024 |
| | | | | | | | 75 | 56.5 | 70 | 1024 |
| | | | | | | | 78.75 | 60 | 75 | 1024 |

**Note:** I = Interlaced. L = Linear.

# CHAPTER 5 - OEM UTILITY PROGRAMS

The OEM utility programs allow the OEM to prepare the 65548 VGA BIOS for use. The BMP utility program enables the OEM to prepare a custom version of the BIOS. The ROMUTIL utility program allows the OEM to modify a binary version of the BIOS into the proper form for programming into one or more EPROMs.

**Note:** These programs may not be reproduced or distributed by the OEM.

## BMP

The BMP548 (32KB BIOS) and BMPLARGE (44KB BIOS) enables OEMs to customize the 65548 BIOS for their own specific requirements. The BMP allows the OEM to modify certain parameters of a binary version of the BIOS to be modified. The parameters that the BMP can modify include:

- Sign-on message
- General and Flat-Panel BIOS Features
- Display type determination
- Set FP Dot Clock
- Set FP Memory Clock
- Extended display modes
- 65548 register tables

The BMP may only be used once on a copy of the BIOS. The OEM should make a backup copy of the original diskette(s) before using any of the OEM utilities. Once the BIOS is changed and saved from the BMP, it cannot be modified again.

## Usage

`BMP548 [File]` or
`BMPLARGE [File]`

`[File]` Optional filename of the BIOS file input to the BMP. A default extension of .DAT is assumed if no extension is specified. A default filename of VGA548.DAT (VGALARGE.DAT) or is assumed if no filename is specified.

## Filenames

Default filenames for the standard 32KB 65548 BIOS are:

- `BMP548.EXE`
- `VGA548.DAT`
- `RAM548.DAT`

Default filenames for the extended 44KB 65548 BIOS are:

- `BMPLARGE.EXE`
- `VGALARGE.DAT`
- `RAMLARGE.DAT`

## Examples

ROM Binary:

```
BMP548 [VGA548.DAT]
BMPLARGE [VGALARGE.DAT]
```

Executes BMP548 (BMPLARGE) with the default file VGA548.DAT (VGALARGE.DAT) as the input file.

RAM Executable:

```
BMP548 RAM548.EXE
BMPLARGE RAMLARGE.EXE
```

Executes BMP548 (BMPLARGE) with the RAM548.EXE (RAMLARGE.EXE) or utility program as the input file.

## Commands

BMP548 or (32KB BIOS) and BMPLARGE (44KB BIOS) organizes the modifiable parameters of the 65548 BIOS into several windows. Some values are entered as text or as hexadecimal values within these windows. The following keys are used to change fields or edit values:

| | |
|---|---|
| <Tab> | Go to next window. |
| <Shift Tab> | Go to previous window. |
| <PgUp> | Move up one page within a window. |
| <PgDn> | Move down one page within a window. |
| <↑>,<↓> | Move up or down one line or field. |
| <←>,<→> | Move left or right one character or field. |
| <+>,<-> | Enable/disable parameter. Increment/decrement a value in the field. |
| <F1> | Help. |
| <F5> | Save BMS file. |
| <F6> | Load BMS file. |
| <F10> | Save changes to the BIOS file. |
| <Esc> | Exit program. |

## Help

Help messages can be displayed by moving to the parameter that requires further explanation and pressing the <F1> key. A pop-up window will be displayed describing the parameter.

## BMS Files

BMP548 and BMPLARGE provide the capability to save and load custom files (BMS Files) that contain all of the 65548 BIOS parameters that can be modified. There is no limit to the number of custom files that user may save.

## Error Messages

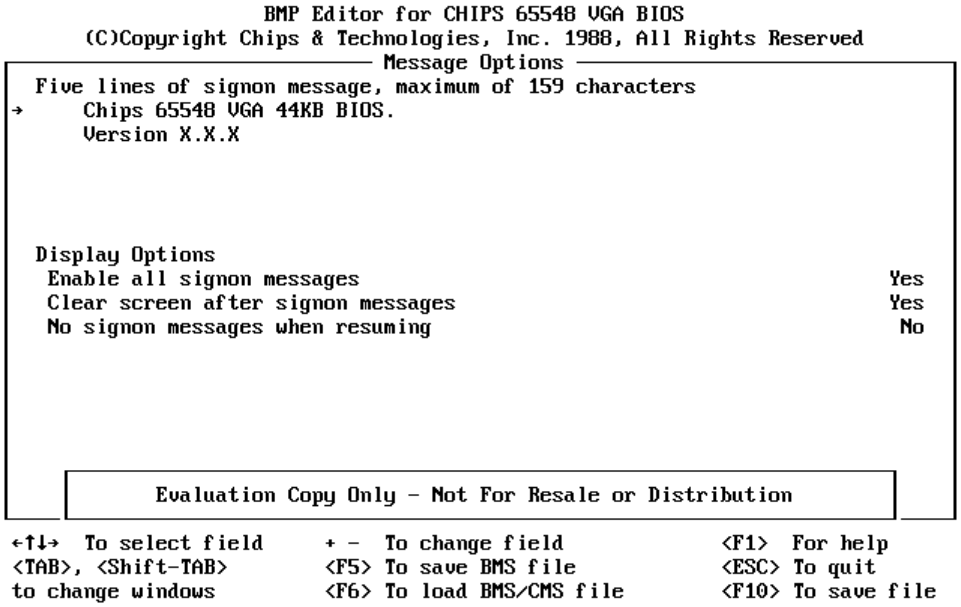
If BMP548 encounters an error during operation, a red window will appear which will contain the error condition. Table 12 lists these errors, the possible cause, and recommended solution.

### Table 11: BMP548 Error Messages.

| Error Message | Problem Description, Recommended Action |
|---|---|
| Use original BIOS file | The file has already been modified and saved. Use the binary file that was supplied on the original disk. If this does not work, contact local CHIPS sales representative. |
| Editable Structure not found | The file can not be modified. This is the wrong file. The binary file that was supplied on the original disk should be used. |
| This program is unable to edit the BMP structure in that file | There is an incompatible version of BMP and binary file. Use the binary file and BMP program supplied on the original disk. |
| Bad BMP structure, Old version was Num, header version was Num | This is an incompatible version of BMP or binary file. Use the binary file and BMP program supplied on original disk. |
| Unable to allocate memory | There is not enough system memory. Remove all unnecessary resident programs and reboot the system. BMP requires approximately 300K of memory. |
| Binary file File not found | BMP could not find the specified file. Verify that the specified file exists. |
| Unable to read binary file File | BMP could not read the specified file. Specified file may be corrupted, use backup copy. |
| Unable to write to File | There was an error during write to specified file. The file may be marked read-only. Try making changes to a file that has read and write access. |
| Unable to reopen File for saving | Unable to re-open binary file. The file may be a read-only file. Try making changes to a file with read and write access. |
| Unable to open my own .EXE file File | Unable to open BMP548.exe for reading. This may be due to insufficient memory, or because the BMP548.exe filename has been changed. Use the BMP and binary files from the original disk. |
| Unable to open BMS file BMSfile | Unable to find or read BMS file. Try specifying a file that does exist. |
| Unable to create file BMSfile | Unable to write a BMS file. There may be insufficient disk space, or an existing file has read-only access. |

**Note:** File Binary filename used.
Num Version number of BMP structure in BMP and/or binary file.
BMSfile BMS filename used.

## Message Options




### Sign-On Message

This sign-on message can be up to 5 lines of 159 characters. Trailing blank lines of the sign-on message are not displayed. Blank lines between lines of text are displayed.

### Enable All Sign-on Messages

If this option is set to *Yes*, the editable message is displayed upon system boot. The message is displayed for approximately three seconds.

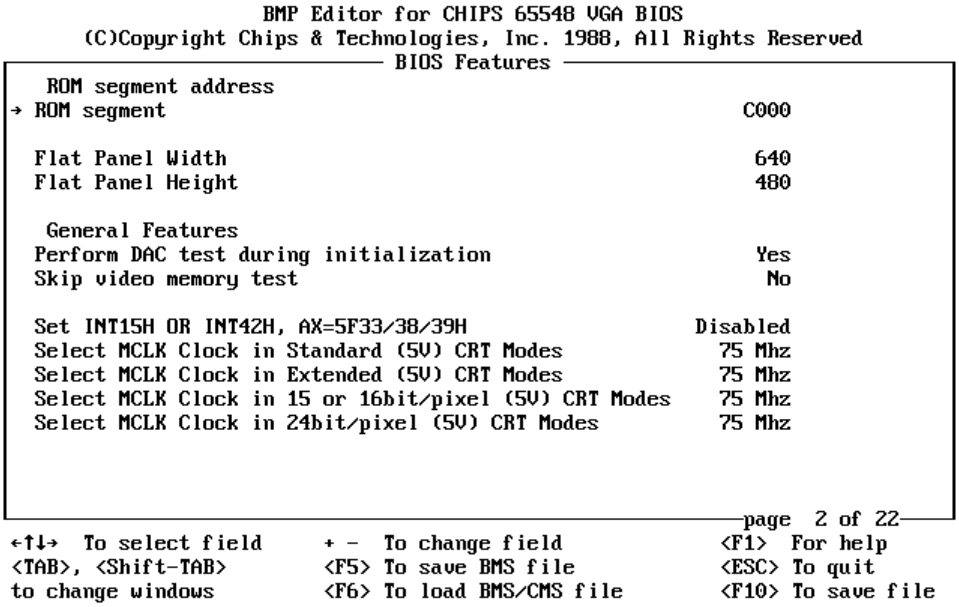
### Clear Screen After Sign-on Message

If this option is set to *Yes*, the Video BIOS clears the screen after the sign-on messages are displayed.

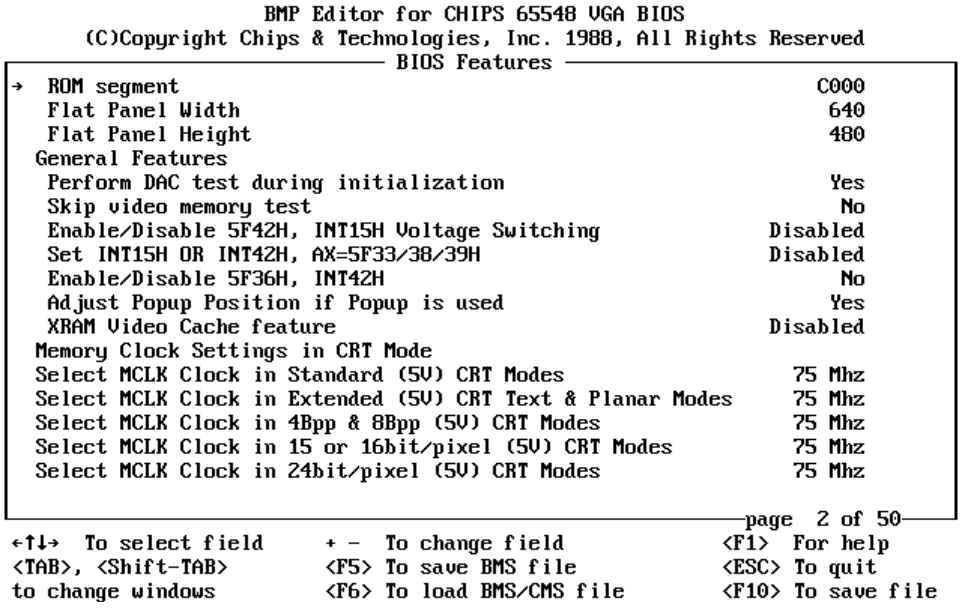
### No Sign-On Messages when Resuming

This option relates to the Laptop System BIOS *Suspend/Resume* feature. Setting this field to *Yes* causes the video BIOS to test during POST whether the BIOS is cold-booting or *resuming* after a *suspend*. This field must be set to *Yes* only when the system BIOS is a Chips and Technologies, Inc. Laptop System BIOS.

## BIOS Features (32KB BIOS)




## BIOS Features (32KB BIOS)



## ROM Segment Address

The 65548 BIOS can be configured to reside at an address other than C000:0h depending on hardware configuration. The default value is C000h, the standard location of a video adapter BIOS.

**Flat Panel Width and Flat Panel Height**

This option configures the BIOS for a panel of a different size.

**Perform DAC Test**

The 65548 BIOS can be set to skip the DAC test during power up initialization.

**Skip Memory Test**

This option allows the BIOS to skip testing of video memory and just clear memory during POST.

**Enable/Disable 5F42h, INT 15h Functions (44KB BIOS)**

This option allows the user to enable or disable voltage switching.

**Set INT 15h or INT 42h, AX = 5F33/38/39h**

This option allows the OEM to enable or disable a call from the BIOS to either INT 15h or INT 42h before mode set, after mode set and before power switching (5F39h). This gives the system BIOS an opportunity to perform any special processing that might be required with a customized system. See the “INT 15h Hooks” section in Chapter 4 for details about the calling parameters.

**Enable/Disable 5F36h, INT42h (44KB BIOS)**

This INT42h hook is reserved for block cursor support in text mode through the system BIOS.

**Adjust Pop Up Position (44KB BIOS)**

This option allows the system BIOS to take control over the position of the pop up window.

**XRAM Video Cache Feature (44KB BIOS)**

This option allows the OEM to enable or disable the use of XRAM Video Cache. Refer to the 65548 Data Sheet for more information on XRAM Video Cache option.

**Select MCLK in Standard CRT Modes**

This option selects the speed of the memory clock in standard CRT modes.

**Select MCLK in Extended CRT Modes**

This option selects the speed of the memory clock in extended CRT modes.

**Select MCLK in 4Bpp & 8Bpp (5V) CRT Modes (44KB BIOS)**

This option selects the speed of the memory clock in 4 or 8 bit per pixel CRT modes.

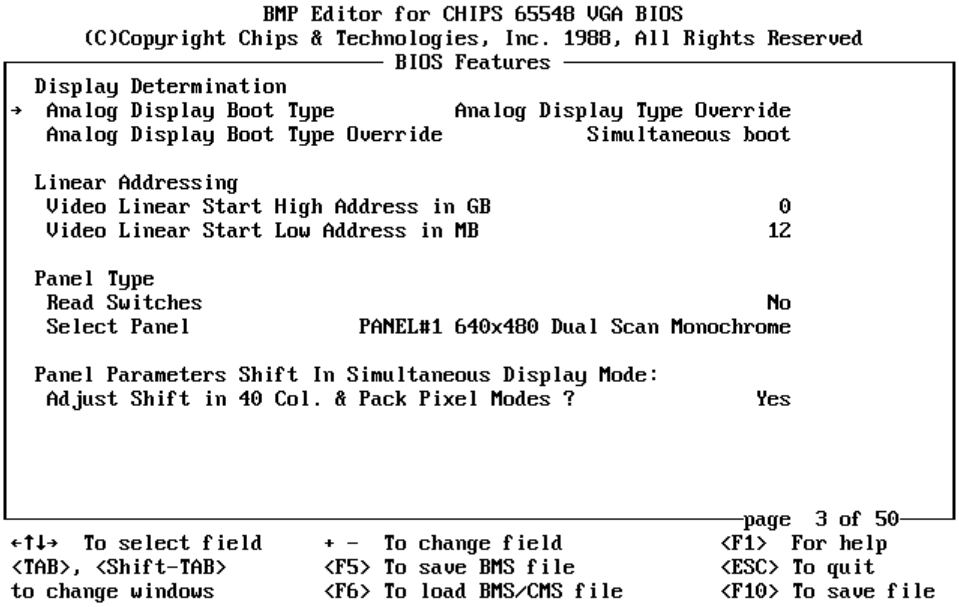
**Select MCLK in 15 or 16 bit/pixel (5V) CRT Modes**

This option selects the speed of the memory clock in 15 or 16 bit per pixel CRT modes.

**Select MCLK in 24 bit/pixel (5V) CRT Modes**

This option selects the speed of the memory clock in 24 bit per pixel CRT modes.

## BIOS Features (44KB BIOS)




**Analog Display Boot Type (32KB BIOS)**

This option selects the boot-up display type: CRT, Flat-Panel or Simultaneous.

**Analog Display Boot Type (44KB BIOS)**

This option allows the user to select whether to boot automatically on flat panel/simultaneous or flat panel/CRT, or to choose an override which forces the BIOS to boot on one of the three display types as specified in the following option.

**Analog Display Boot Type Override (44KB BIOS only)**

This option specifies the display type to boot on when the override option is chosen in the previous line. The type may be flat panel, CRT, or simultaneous.

**Video Linear Start Address**

This option allows the Video Linear Start Address to begin on various 2MB boundaries while in graphics linear modes 20h-50h. The hardware must correlate with the Video Linear Start Address. The 65548 PCI BIOS does not use this option but it calls the system BIOS with AX=B109H, INT 1AH to get Video Linear Start Address.

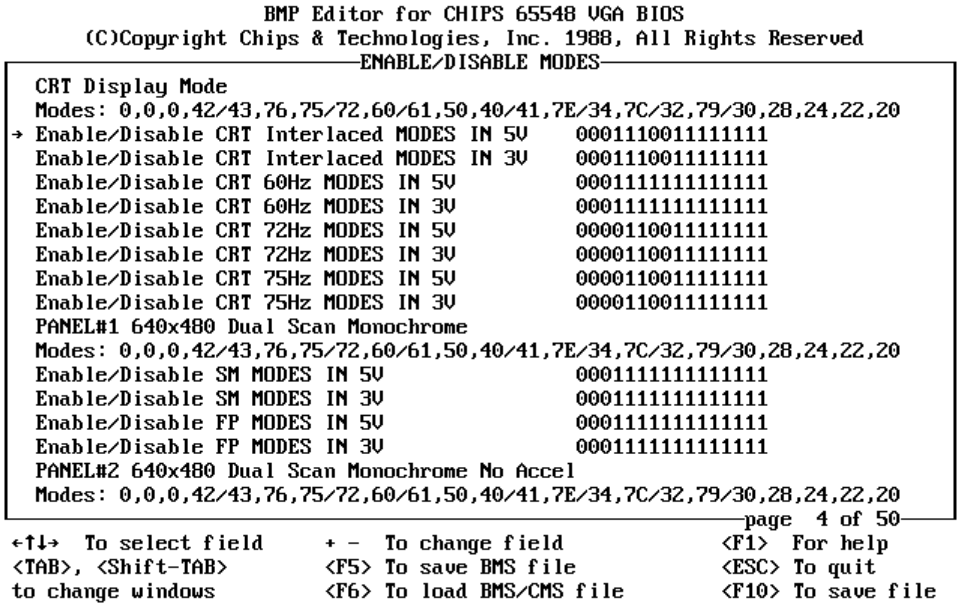
**Panel Type (44KB BIOS only)**

Please refer to Table 9. (page 38) for the different options.

**Adjust Shift in 40 Column and Pack Pixel Modes**

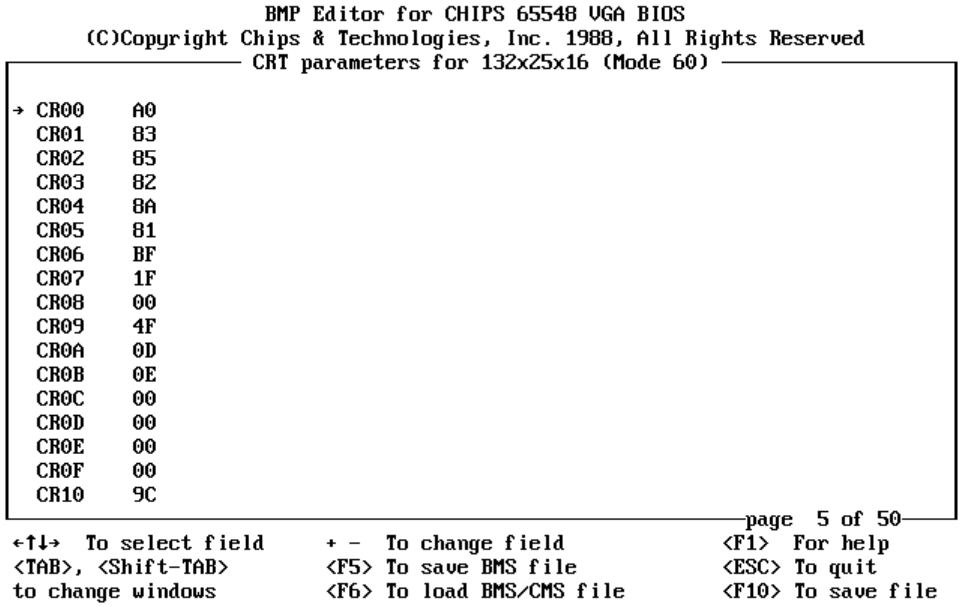
This option allows adjustments of Flat-Panel Horizontal Sync Start and Horizontal Start registers for simultaneous display operation.

## Enable/Disable Modes



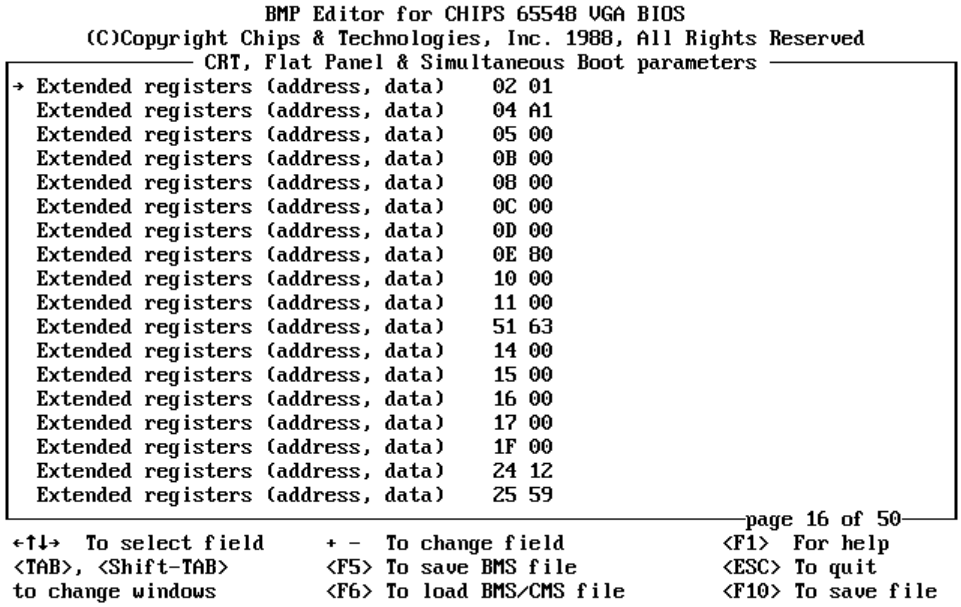

This window allows the OEM to enable or disable the extended display modes which can be accessed by application programs. If the mode is disabled, the BIOS set mode command will return an “unsupported mode” status.

**CRT Registers and Dot Clocks For All Extended Modes**



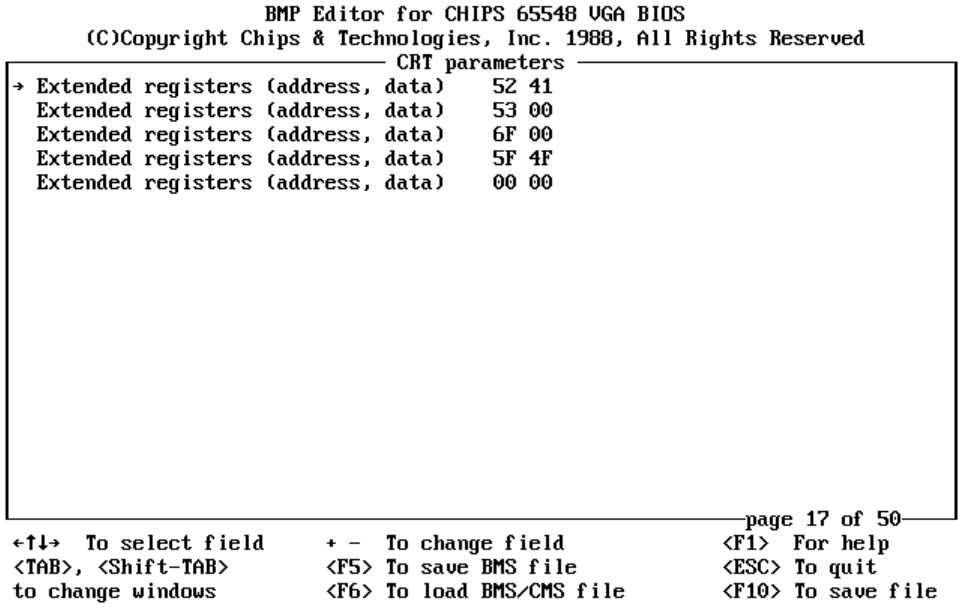

This is an example of the set of windows that allows the OEM to adjust the CRT register values for all extended modes. In addition, the dot clock frequency can be specified for each mode (except modes 60 and 61). There is one screen for each distinct table of CRT register values that the video BIOS uses. Often, more than one mode uses a table, and the dot clock frequencies can be set separately for each of these modes.

## CRT, Flat Panel and Simultaneous Boot Parameters



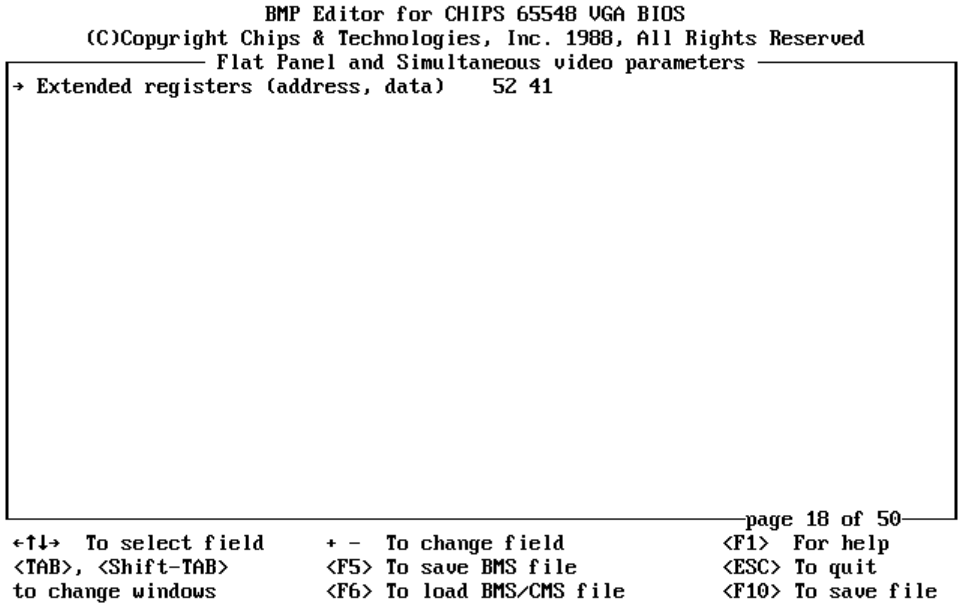

This window is used to change the extended register values for use at boot time.

## CRT Parameters



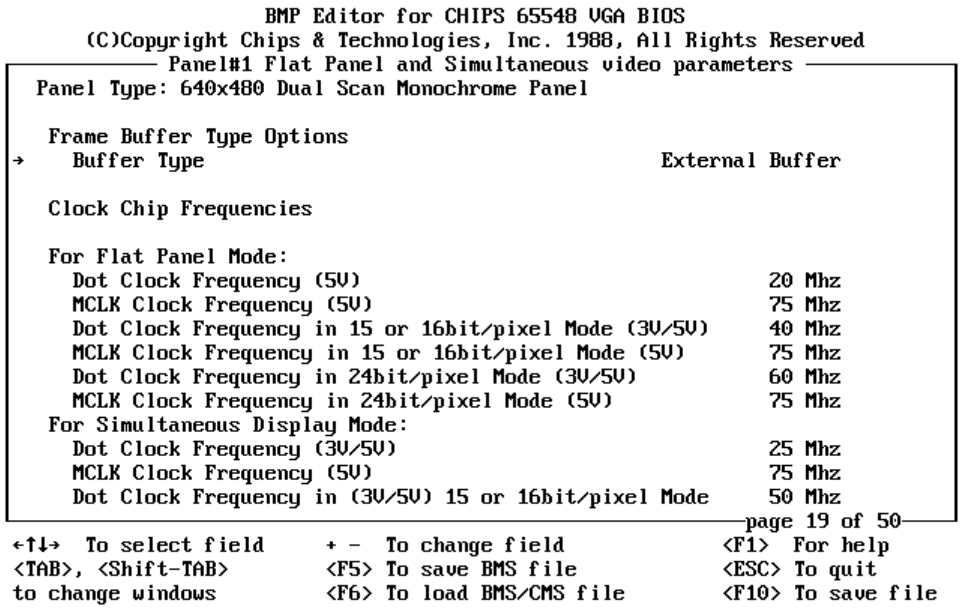

This window is used to change the extended registers for CRT mode.

## Flat Panel and Simultaneous Video Parameters




This window is used to change the extended registers for Flat-Panel and Simultaneous modes.

## Flat Panel and Simultaneous Video Parameters (Panel Type)

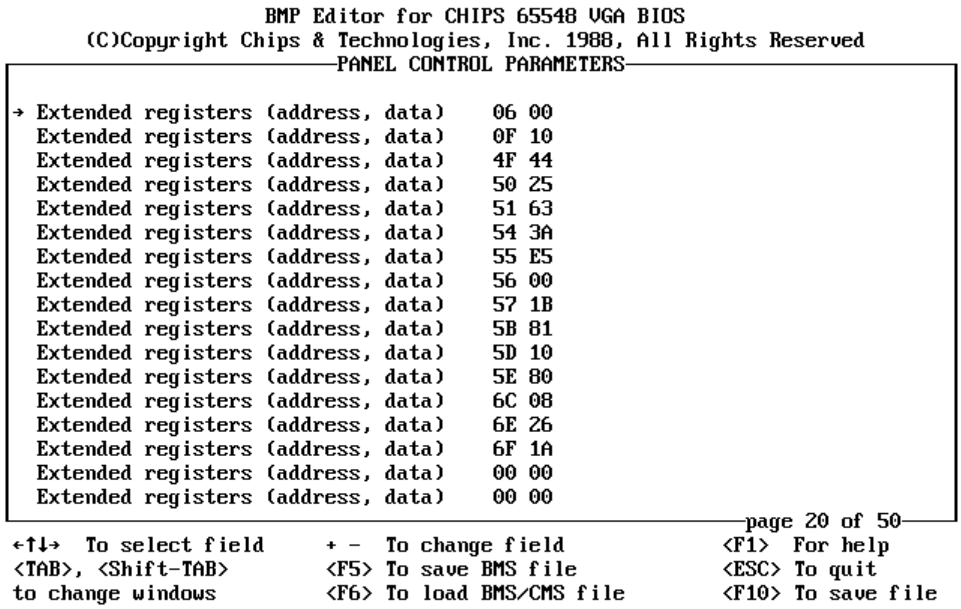


This window is used to change the various Dot and Memory clocks for Flat-Panel and Simultaneous display modes.

For the standard 32KB BIOS, only one type of flat panel is defined in the BIOS at any one time. The type of panel can be changed by adjusting the parameters defining the panel, or by loading a .BMS file containing the parameters of a new panel type. This last method is the easiest, and is discussed in the "BMS Files" section which follows.

The extended 44KB BIOS has 8 of these screens, one for each of the pre-defined flat panel types.

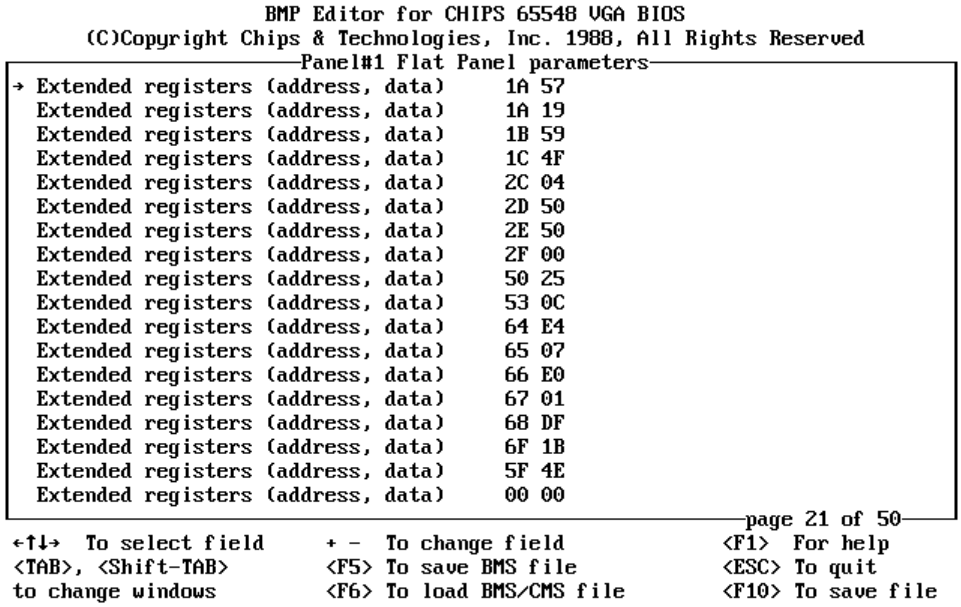
## Panel Control Parameters




This window is used to change the extended registers for the panel selected.

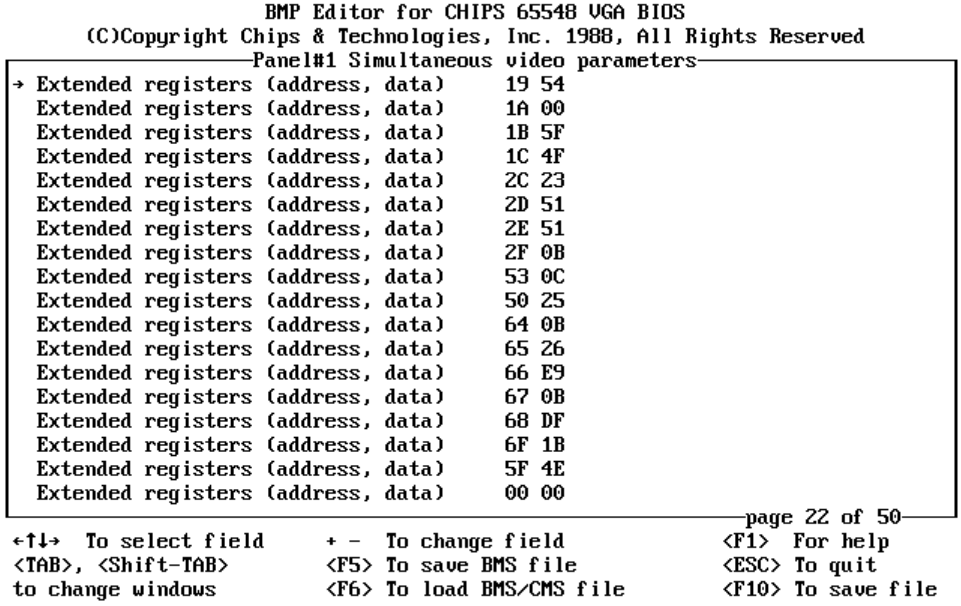
The extended 44KB BIOS has 8 of these screens, one for each of the pre-defined flat panel types.

## Flat Panel Parameters




This window is used to change the extended registers for the panel selected. For the 44KB BIOS, these parameters apply to the pre-defined panel type shown at the top of the screen.

## Simultaneous Video Parameters




This window is used to change the extended registers for the panel selected in simultaneous mode. For the 44KB BIOS, these parameters apply to the pre-defined panel type shown at the top of the screen.

## BMS Files (32 KB BIOS Only)

This manual documents one panel type. There are twelve additional panels located in .BMS files that can be used by executing the Load <F6> command.

The thirteen panels are as follows:

### Table 12: BMS Files

| | |
|---|---|
| **Default BIOS:** | |
| (1) MONODD.BMS | 640x480 Dual Scan Monochrome (Default)<br>(Sharp LM64P80) |
| **BMS Files:** | |
| (2) MONODDEX.BMS | 640x480 Dual Scan Monochrome using external buffer<br>(Sharp LM64P80) |
| (3) MONODDNA.BMS | 640x480 Dual Scan Monochrome using external buffer & no accelerator<br>(Sharp LM64P80) |
| (4) STNDD.BMS | 640x480 Color STN DD -8/16 Bit Interface<br>(Sharp LM64C08) |
| (5) STNDDEX.BMS | 640x480 Color STN DD 8/16 Bit Interface using external buffer<br>(Sharp LM64C08) |
| (6) STNDDNA.BMS | 640x480 Color STN DD -8/16 Bit Interface<br>Using external buffer & no accelerator (Sharp LM64C08) |
| (7) STNDD1DR.BMS | 640x480 Color STN DD -8/16 Bit Interface<br>(Sharp LM64C08)<br>for 1-DRAM Memory Configuration. |
| (8) TFTCLR.BMS | 640x480 Color TFT With Display Enable<br>(Sharp LQ9D011, Toshiba LTM09C015-1) |
| (9) HTCTFT.BMS | 640x480 Color TFT<br>(Hitachi 26DS2, Hitachi TX26D02VC2AA) |
| (10) 1Kx768TFT.BMS | 1024x768 Color TFT<br>(Sharp LQ10dx01) |
| (11) 18BTFT.BMS | 640x480 Color TFT 18-Bit Interface |
| (12) STN4BIT.BMS | 640x480 4-Bit Pack STN Color<br>(Sanyo LCM5327-24NAK, LM-CK53-22NEZ, LCM 5330) |
| (13) STNE4BIT.BMS | 640x480 4-bit Extended pack STN color<br>(Sharp LM64C031) |

## Built in Panel Support (44KB BIOS Only)

The eight panels supported are as follows:

### Table 13: Panel Types

| Panel # | Panel Type |
|---|---|
| 1 | 640x480 Dual Scan Monochrome Panel |
| 2 | 640x480 Dual Scan Monochrome Panel No Accel |
| 3 | 640x480 Dual Scan Color Panel |
| 4 | 800x600 Dual Scan Color Panel |
| 5 | 640x480 Sharp TFT Color |
| 6 | 640x480 18-bit TFT Color |
| 7 | 1024x768 TFT Color |
| 8 | 800x600 TFT Color |

## ROMUTIL

The ROMUTIL program modifies the binary ROM files (.DAT). ROMUTIL can handle from one to four binary files and produce either binary files or an Intel hex format file as output.

## Usage

**Note:** All command line parameters can be omitted, in which case ROMUTIL will run in interactive mode and prompt for action, options, and filenames.

```
ROMUTIL
```

- OR -

```
ROMUTIL [Action][File(s)][Options][Outfile]
```

[Action] This parameter may be one of the following:

| | |
|---|---|
| /A2 | Add 2 binary files to form 1 file (interleave even/odd). |
| /A4 | Add 4 binary files to form 1 file (interl eave every fourth byte). |
| /S2 | Split a binary file into 2 files (every other byte). |
| /S4 | Split a binary file into 4 files (every fourth byte). |
| /C | Convert a binary file into an Intel hex file. |
| /D | Divide a binary file into 2 files (first half/second half). |

`[File(s)]` Filenames to be used as input.

`[Options]` This parameter may be one of the following:

| | |
|---|---|
| `/B<Num.>` | Number of data bytes to generate per line (valid only with Convert). Number must be 16 (default), 32, 48 or 64. |
| `/L<Addr.>` | Set starting address (default is 0000h) Valid only with Convert. |
| `/P<Size>` | Pad with 0xFF up to the end of <Rom size>. The number must be 2, 4, 8, 16, 32, 64 or 128. Only valid with Add and Split. |

`[Outfile]` Filename of destination file.

## Example

```
ROMUTIL /A2 ROM.1 ROM.2 NEWROM.BIN
```

Adds ROM.1 and ROM.2 to form the combined file NEWROM.BIN. If the output file is not specified, ROM.BIN will be created by default.

# APPENDIX A - BUILDING THE VGA BIOS

This appendix describes the process for creating a binary BIOS from source code. Building, modifying, or updating the BIOS Source Code requires the following software utilities:

- A text editor capable of editing ASCII files
- Microsoft Macro Assembler (MASM) version 5.0 or later
- Microsoft Linker (LINK) version 3.60 or later
- Microsoft MAKE utility
- Microsoft C Compiler (CL) version 5.1 or 7.0

The INSTALL.BAT file on the source code diskette will install all the files necessary to create a binary version of the BIOS. Use the following command line to install:

```
A:INSTALL C:\VGA548
```

INSTALL.BAT will create the directory \VGA548 on drive C:. The following subdirectories will also be created:

```
\OBJ
\OBJLG
\LST
```

The BIOS source files will be placed in the \VGA548 or directory, along with batch files for assembling and linking the BIOS. INSTALL.BAT will then copy the binary files into the \OBJ subdirectory. A .LNK file used in linking the BIOS modules is also placed in the \OBJ subdirectory.

To create the binary copy of the BIOS, run the Microsoft Make utility with the following command line:

```
MAKE VGA548.MAK or
MAKE VGALARGE.MAK
```

| | |
|---|---|
| **Note:** | Several “warning” messages may appear while Make is assembling certain source modules. These warning messages should be ignored. However, there should be no “error” messages from the assembler. |

# APPENDIX B - EXTENDED CONTROL REGISTERS AND PARAMETER TABLES

## Memory Control Registers 1 and 2 (XR04 and XR05)

The 65548 BIOS uses Memory Mode Register (XR04) to determine and program how much memory is installed on the board. The F65548 will support up to 1MB of RAM memory configuration.

| D0 | DATA PATH | MEMORY INTERFACE |
|---|---|---|
| 0 | 32-bit | 1 MB DRAM |
| 1 | 16-bit | 512 KB DRAM |

The BIOS uses the Memory Control Register 2 to further specify the memory configuration:

| D4 D3 | CONFIGURATION |
|---|---|
| 0 0 | 9-bit CAS# address, 2 CAS# and 1 WE# configuration (default) |
| 0 1 | 8-bit CAS# address, 2 CAS# and 1 WE# configuration |
| 1 0 | 9-bit CAS# address, 1 CAS# and 2 WE# configuration |
| 1 1 | 8-bit CAS# address, 1 CAS# and 2 WE# configuration |

## Video Interface Register (XR28)

The 65548 BIOS Set Mode function programs bit 5 of this register to value *1* for Interlace and *0* for Non-Interlace modes.

## Mode Register (XR2B)

The BIOS programs the software register with the current mode number. It can be used to find out the existing mode, but cannot change it.

## Software Flag Register (XR1F)

Bits 0 and 1 of this register are used by the BIOS to indicate the current display type. Bit 2 of this register is used by the BIOS to indicate to the driver any pop-up update.

| D2 | STATUS |
|---|---|
| 0 | No pop-up update |
| 1 | Pop-up update |

## Setup Registers

The 65548 BIOS programs register *102* as standard VGA, register *3C3h* to value *01h* and register *XR70* to value *80h* (in this order) upon power-up, and hereafter does not update registers *46E8*, *102*, *3C3* or *XR70*.

## Half-Line Compare Register (XR19)

This register generates the *Half Line Compare* signal that controls the positioning of the VSync for odd frames when interlaced video output is enabled. The 65548 BIOS programs XR19 = 4Ch in Interlace mode. The Flat-Panel supports Non-Interlaced modes only (e.g. Modes 24h, 34h, 72h/75h and 7Eh will be displayed in Non-Interlace/Interlace mode on the CRT, but Non-Interlace mode on the Flat-Panel).

## Higher Page Map (XR10) and Lower Page Map (XR11)

Higher Page Map and Lower Page Map are 8 bits in the 65548 VGA controller. Page map should start on the 1K boundary for Planar modes and the 4K boundary for Packed Pixel modes when the 65548 BIOS accesses extended mode functions.

## 132 Column Mode on 1024x768 Panel

The 65548 VGA BIOS programs CR01 = 7Fh (128 columns) in Flat-Panel Mode or in Simultaneous Display Mode and CR01 = 83h (132 columns) in the CRT Mode.

## Software Flag Register (XR0F)

This register is reserved for the 65548 BIOS flags. Bits *0* and *1* are used to save memory size upon power-up.

| XR0F D1D0 | MEMORY INSTALLED |
|---|---|
| 0 0 | 256KB |
| 0 1 | 512KB |
| 1 x | 1MB |

The BIOS reads XR0F bit 4 to enable/disable accelerator support:

XR0F bit 4 = 1 15/16 bit-per-pixel modes
XR0F bit 4 = 0 All other modes

The BIOS reads XR0F bit 7 to enable/disable 8x19 font:

XR0F bit 7 = 0 8x19 font Disabled (Text Compensation Disabled)
XR0F bit 7 = 1 8x19 font Enabled (Text Compensation Enabled)

## Software Flag Register (XR44)

The BIOS sets bits 2 - 0 for the panel type (1,2,3...8) (44KB BIOS only).

The BIOS reads XR44 bit 4 to enable/disable optimal compensation upon power up only:

XR44 bit 4 = 0 Disabled optimal compensation on power up
XR44 bit 4 = 1 Enabled optimal compensation on power up

If Optimal Compensation is enabled, the BIOS activates Tall Font in modes 0*, 0+, 1*, 1+, 2*, 2+, 3*, 3+, 7, 7+ and 60, and also activates Line Replication in modes 4, 5, 6, D, E, 13 and 78. In other modes, automatic centering is activated if XR57 bits 0 and 1 are set to “1”.

## Software Flag Register (XR45)

The BIOS uses this register for the software flag as follows:

| D0 - D2 | For BIOS Internal Use only |
|---|---|
| D3 | CRT Zoom |
| 0 | CRT Zoom Disabled |
| 1 | CRT Zoom Enabled |
| D5 - D4 | Pop - Up Position |
| 00 | Top Left |
| 01 | Bottom Left |
| 1X | Reserved |
| D6 | Pop - Up Status |
| 0 | Pop - Up Disabled |
| 1 | Pop - Up Enabled |

| | |
|---|---|
| D7 | Set 5V or 3V/5V Mixed Voltage Mode - |
| | **Note:** *5F42h must be Enabled in BMP* |
| 0 | Set 5V mode only |
| 1 | Set 3V/5V mixed voltage mode |

## Extended Linear Mode Parameter Tables

### Table 14: Extended Linear Mode Parameter Tables for F65548 VGA Controller

| Mode No. | 20 60Hz | 20 72Hz | 20 75Hz | 22 56Hz | 22 60Hz | 22 72Hz | 22 75Hz |
|---|---|---|---|---|---|---|---|
| Mode Resolution | **640x480** | **640x480** | **640x480** | **800x600** | **800x600** | **800x600** | **800x600** |
| BitsPerPixel | **4** | **4** | **4** | **4** | **4** | **4** | **4** |
| Interlace/Non-Interlace | NI | NI | NI | NI | NI | NI | NI |
| Graphics/Text | G | G | G | G | G | G | G |
| Planar/PackPix | LI/PP | LI/PP | LI/PP | LI/PP | LI/PP | LI/PP | LI/PP |
| MemoryNeeded | 512KB | 512KB | 512KB | 512KB | 512KB | 512KB | 512KB |
| VideoBuffer | >1MB | >1MB | >1MB | >1MB | >1MB | >1MB | >1MB |
| CPUInterface | SChain | SChain | SChain | SChain | SChain | SChain | SChain |
| HColumns | 80 | 80 | 80 | 100 | 100 | 100 | 100 |
| VRows | 30 | 30 | 30 | 37 | 37 | 37 | 37 |
| FontSize | 8x16 | 8x16 | 8x16 | 8x16 | 8x16 | 8x16 | 8x16 |
| DotClock (MHz) | 25.175 | 31.5 | 31.5 | 36 | 40 | 50 | 49.5 |
| HSweep (KHz) | 31.5 | 37.9 | 37.5 | 31.5 | 37.9 | 48.1 | 46.9 |
| Vrefresh (Hz) | 60 | 72 | 75 | 56 | 60 | 72 | 75 |
| **Clock:** | | | | | | | |
| MiscOutReg | EB | EB | EB | EB | EB | EB | EB |
| Mclock (MHz) | 68 | 68 | 68 | 68 | 68 | 68 | 68 |
| **Sequencer:** | | | | | | | |
| ASyncReset (00) | 03 | 03 | 03 | 03 | 03 | 03 | 03 |
| ClockingMode (01) | 01 | 01 | 01 | 01 | 01 | 01 | 01 |
| MapMask (02) | 0F | 0F | 0F | 0F | 0F | 0F | 0F |
| CharFontSel (03) | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| MemoryMode (04) | 0E | 0E | 0E | 0E | 0E | 0E | 0E |
| **CRT Controller:** | | | | | | | |
| HTotal (00) | 5F | 63 | 64 | 7B | 7F | 7D | 7F |
| HDispEnblEnd (01) | 4F | 4F | 4F | 63 | 63 | 63 | 63 |
| HBlankStart (02) | 50 | 50 | 50 | 64 | 64 | 64 | 64 |
| HBlankEnd (03) | 82 | 86 | 87 | 9E | 82 | 80 | 82 |
| HSyncStart (04) | 53 | 54 | 53 | 68 | 6A | 6C | 67 |
| HSyncEnd (05) | 9F | 99 | 9B | 91 | 1A | 1B | 12 |
| VTotal (06) | 0B | 06 | F2 | 6F | 72 | 98 | 6F |
| VOverflow (07) | 3E | 3E | 0F | F0 | F0 | F0 | F0 |
| PresetRowScan (08) | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| MaxScanLine (09) | 40 | 40 | 40 | 60 | 60 | 60 | 60 |
| CursorStart (0A) | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| CursorEnd (0B) | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| StartAddrH (0C) | FF | FF | FF | FF | FF | FF | FF |
| StartAddrL (0D) | FF | FF | FF | FF | FF | FF | FF |
| CurLocationH (0E) | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| CurLocationL (0F) | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| VSyncStart (10) | EA | E8 | E1 | 58 | 58 | 7C | 58 |
| VSyncEnd (11) | 8C | 8B | 84 | 8A | 8C | 82 | 8B |
| VDispEnblEnd (12) | DF | DF | DF | 57 | 57 | 57 | 57 |
| Offset (13) | 28 | 28 | 28 | 32 | 32 | 32 | 32 |
| UnderLine (14) | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| VBlankStart (15) | E7 | E7 | E0 | 58 | 58 | 58 | 58 |
| VBlankEnd (16) | 04 | FF | F2 | 6F | 72 | 98 | 6F |
| CRTCMode (17) | E3 | E3 | E3 | E3 | E3 | E3 | E3 |
| LineComp (18) | FF | FF | FF | FF | FF | FF | FF |

## Table 15: Extended Linear Mode Parameter Tables for F65548 VGA Controller (continued)

| Mode No. | 20 60 Hz | 20 72Hz | 20 75Hz | 22 56Hz | 22 60Hz | 22 72Hz | 22 75Hz |
|---|---|---|---|---|---|---|---|
| **Attribute Controller:** | | | | | | | |
| Palette0(00) | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| Palette1(01) | 01 | 01 | 01 | 01 | 01 | 01 | 01 |
| Palette2(02) | 02 | 02 | 02 | 02 | 02 | 02 | 02 |
| Palette3(03) | 03 | 03 | 03 | 03 | 03 | 03 | 03 |
| Palette4(04) | 04 | 04 | 04 | 04 | 04 | 04 | 04 |
| Palette5(05) | 05 | 05 | 05 | 05 | 05 | 05 | 05 |
| Palette6(06) | 06 | 06 | 06 | 06 | 06 | 06 | 06 |
| Palette7(07) | 07 | 07 | 07 | 07 | 07 | 07 | 07 |
| Palette8(08) | 08 | 08 | 08 | 08 | 08 | 08 | 08 |
| Palette9(09) | 09 | 09 | 09 | 09 | 09 | 09 | 09 |
| PaletteA(0A) | 0A | 0A | 0A | 0A | 0A | 0A | 0A |
| PaletteB(0B) | 0B | 0B | 0B | 0B | 0B | 0B | 0B |
| PaletteC(0C) | 0C | 0C | 0C | 0C | 0C | 0C | 0C |
| PaletteD(0D) | 0D | 0D | 0D | 0D | 0D | 0D | 0D |
| PaletteE(0E) | 0E | 0E | 0E | 0E | 0E | 0E | 0E |
| PaletteF(0F) | 0F | 0F | 0F | 0F | 0F | 0F | 0F |
| ModeCntl(10) | 01 | 01 | 01 | 01 | 01 | 01 | 01 |
| OverScan(11) | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| CPlaneEnable(12) | 0F | 0F | 0F | 0F | 0F | 0F | 0F |
| PixelPanning(13) | 07 | 07 | 07 | 07 | 07 | 07 | 07 |
| **Graphics Controller:** | | | | | | | |
| Set/Reset(00) | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| EnableSet/Reset(01) | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| ColorCompare(02) | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| DataRotate(03) | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| ReadMapSelect(04) | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| ModeRegister(05) | 40 | 40 | 40 | 40 | 40 | 40 | 40 |
| MisceRegister(06) | 05 | 05 | 05 | 05 | 05 | 05 | 05 |
| ColorDontCare(07) | 0F | 0F | 0F | 0F | 0F | 0F | 0F |
| BitMask(08) | FF | FF | FF | FF | FF | FF | FF |
| **Palette Mask Register:** | | | | | | | |
| 3C6 | 0F | 0F | 0F | 0F | 0F | 0F | 0F |
| **Extended Registers:** | | | | | | | |
| XR0B | 15 | 15 | 15 | 15 | 15 | 15 | 15 |
| XR10 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| XR11 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| XR19 | xx | xx | xx | xx | xx | xx | xx |
| XR28 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| XR04 Bit 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| XR0C | 03 | 03 | 03 | 03 | 03 | 03 | 03 |
| XR06 Bits 3,2 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |

## Table 16: Extended Linear Mode Parameter Tables for F65548 VGA Controller (continued)

| Mode No. | 24I 43Hz | 24 60Hz | 24 70Hz | 24 75Hz | 30 60Hz | 30 72Hz | 30 75Hz |
|---|---|---|---|---|---|---|---|
| Mode Resolution | **1024x768** | **1024x768** | **1024x768** | **1024x768** | **640x480** | **640x480** | **640x480** |
| BitsPerPixel | **4** | **4** | **4** | **4** | **8** | **8** | **8** |
| Interlace/Non-Interlace | I | NI | NI | NI | NI | NI | NI |
| Graphics/Text | G | G | G | G | G | G | G |
| Planar/PackPix | LI/PP | LI/PP | LI/PP | LI/PP | LI/PP | LI/PP | LI/PP |
| MemoryNeeded | 512KB | 512KB | 512KB | 512KB | 512KB | 512KB | 512KB |
| VideoBuffer | >1MB | >1MB | >1MB | >1MB | >1MB | >1MB | >1MB |
| CPUInterface | SChain | SChain | SChain | SChain | SChain | SChain | SChain |
| HColumns | 128 | 128 | 128 | 128 | 80 | 80 | 80 |
| VRows | 48 | 48 | 48 | 48 | 30 | 30 | 30 |
| FontSize | 8x16 | 8x16 | 8x16 | 8x16 | 8x16 | 8x16 | 8x16 |
| DotClock (MHz) | 44.9 | 65 | 75 | 78.75 | 25.175 | 31.5 | 31.5 |
| HSweep (KHz) | 35.5 | 48.4 | 56.5 | 60 | 31.5 | 37.9 | 37.5 |
| Vrefresh (Hz) | 43 | 60 | 70 | 75 | 60 | 72 | 75 |
| **Clock:** | | | | | | | |
| MiscOutReg | 2F | 2F | 2F | 2F | EB | EB | EB |
| Mclock (MHz) | 68 | 68 | 68 | 68 | 68 | 68 | 68 |
| **Sequencer:** | | | | | | | |
| ASyncReset (00) | 03 | 03 | 03 | 03 | 03 | 03 | 03 |
| ClockingMode (01) | 01 | 01 | 01 | 01 | 01 | 01 | 01 |
| MapMask (02) | 0F | 0F | 0F | 0F | 0F | 0F | 0F |
| CharFontSel (03) | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| MemoryMode (04) | 0E | 0E | 0E | 0E | 0E | 0E | 0E |
| **CRT Controller:** | | | | | | | |
| HTotal (00) | 99 | A3 | A1 | 9F | 61 | 65 | 66 |
| HDispEnblEnd (01) | 7F | 7F | 7F | 7F | 4F | 4F | 4F |
| HBlankStart (02) | 80 | 80 | 80 | 80 | 50 | 50 | 50 |
| HBlankEnd (03) | 9C | 86 | 84 | 81 | 82 | 86 | 87 |
| HSyncStart (04) | 82 | 84 | 84 | 83 | 53 | 54 | 53 |
| HSyncEnd (05) | 18 | 95 | 95 | 8F | 9F | 99 | 9B |
| VTotal (06) | 98 | 25 | 24 | 1F | 0B | 06 | F2 |
| VOverflow (07) | 1F | FD | FD | FD | 3E | 3E | 0F |
| PresetRowScan (08) | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| MaxScanLine (09) | 40 | 60 | 60 | 60 | 40 | 40 | 40 |
| CursorStart (0A) | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| CursorEnd (0B) | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| StartAddrH (0C) | FF | FF | FF | FF | 00 | 00 | 00 |
| StartAddrL (0D) | FF | FF | FF | FF | 00 | 00 | 00 |
| CurLocationH (0E) | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| CurLocationL (0F) | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| VSyncStart (10) | 82 | 02 | 02 | 00 | EA | E8 | E0 |
| VSyncEnd (11) | 8C | 88 | 88 | 83 | 8C | 8B | 83 |
| VDispEnblEnd (12) | 7F | FF | FF | FF | DF | DF | DF |
| Offset (13) | 40 | 40 | 40 | 40 | 50 | 50 | 50 |
| UnderLine (14) | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| VBlankStart (15) | 80 | 00 | 00 | 00 | E7 | E7 | E0 |
| VBlankEnd (16) | 98 | 25 | 24 | 1F | 04 | FF | F2 |
| CRTCMode (17) | E3 | E3 | E3 | E3 | E3 | E3 | E3 |
| LineComp (18) | FF | FF | FF | FF | FF | FF | FF |

## Table 17: Extended Linear Mode Parameter Tables for F65548 VGA Controller (continued)

| Mode No. | 24I 43Hz | 24 60Hz | 24 70Hz | 24 75Hz | 30 60Hz | 30 72Hz | 30 75Hz |
|---|---|---|---|---|---|---|---|
| **Attribute Controller:** | | | | | | | |
| Palette0(00) | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| Palette1(01) | 01 | 01 | 01 | 01 | 01 | 01 | 01 |
| Palette2(02) | 02 | 02 | 02 | 02 | 02 | 02 | 02 |
| Palette3(03) | 03 | 03 | 03 | 03 | 03 | 03 | 03 |
| Palette4(04) | 04 | 04 | 04 | 04 | 04 | 04 | 04 |
| Palette5(05) | 05 | 05 | 05 | 05 | 05 | 05 | 05 |
| Palette6(06) | 06 | 06 | 06 | 06 | 06 | 06 | 06 |
| Palette7(07) | 07 | 07 | 07 | 07 | 07 | 07 | 07 |
| Palette8(08) | 08 | 08 | 08 | 08 | 08 | 08 | 08 |
| Palette9(09) | 09 | 09 | 09 | 09 | 09 | 09 | 09 |
| PaletteA(0A) | 0A | 0A | 0A | 0A | 0A | 0A | 0A |
| PaletteB(0B) | 0B | 0B | 0B | 0B | 0B | 0B | 0B |
| PaletteC(0C) | 0C | 0C | 0C | 0C | 0C | 0C | 0C |
| PaletteD(0D) | 0D | 0D | 0D | 0D | 0D | 0D | 0D |
| PaletteE(0E) | 0E | 0E | 0E | 0E | 0E | 0E | 0E |
| PaletteF(0F) | 0F | 0F | 0F | 0F | 0F | 0F | 0F |
| ModeCntl(10) | 01 | 01 | 01 | 01 | 01 | 01 | 01 |
| OverScan(11) | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| CPlaneEnable(12) | 0F | 0F | 0F | 0F | 0F | 0F | 0F |
| PixelPanning(13) | 07 | 07 | 07 | 07 | 00 | 00 | 00 |
| **Graphics Controller:** | | | | | | | |
| Set/Reset(00) | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| EnableSet/Reset(01) | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| ColorCompare(02) | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| DataRotate(03) | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| ReadMapSelect(04) | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| ModeRegister(05) | 40 | 40 | 40 | 40 | 00 | 00 | 00 |
| MisceRegister(06) | 05 | 05 | 05 | 05 | 05 | 05 | 05 |
| ColorDontCare(07) | 0F | 0F | 0F | 0F | 0F | 0F | 0F |
| BitMask(08) | FF | FF | FF | FF | FF | FF | FF |
| **Palette Mask Register:** | | | | | | | |
| 3C6 | 0F | 0F | 0F | 0F | FF | FF | FF |
| **Extended Registers:** | | | | | | | |
| XR0B | 15 | 15 | 15 | 15 | 15 | 15 | 15 |
| XR10 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| XR11 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| XR19 | 4C | xx | xx | xx | xx | xx | xx |
| XR28 | 20 | 00 | 00 | 00 | 10 | 10 | 10 |
| XR04 Bit 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| XR0C | 03 | 03 | 03 | 03 | 00 | 00 | 00 |
| XR06 Bits 3,2 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |

## Table 18: Extended Linear Mode Parameter Tables for F65548 VGA Controller (continued)

| Mode No. | 32 56Hz | 32 60Hz | 32 72Hz | 32 75Hz | 34I 43Hz | 34 60Hz | 34 70Hz | 34 75Hz |
|---|---|---|---|---|---|---|---|---|
| Mode Resolution | **800x600** | **800x600** | **800x600** | **800x600** | **1024x768** | **1024x768** | **1024x768** | **1024x768** |
| BitsPerPixel | **8** | **8** | **8** | **8** | **8** | **8** | **8** | **8** |
| Interlace/Non-Interlace | NI | NI | NI | NI | I | NI | NI | NI |
| Graphics/Text | G | G | G | G | G | G | G | G |
| Planar/PackPix | LI/PP | LI/PP | LI/PP | LI/PP | LI/PP | LI/PP | LI/PP | LI/PP |
| MemoryNeeded | 512KB | 512KB | 512KB | 1MB | 1MB | 1MB | 1MB | 1MB |
| VideoBuffer | >1MB | >1MB | >1MB | >1MB | >1MB | >1MB | >1MB | >1MB |
| CPUInterface | SChain | SChain | SChain | SChain | SChain | SChain | SChain | SChain |
| HColumns | 100 | 100 | 100 | 100 | 128 | 128 | 128 | 128 |
| VRows | 37 | 37 | 37 | 37 | 48 | 48 | 48 | 48 |
| FontSize | 8x16 | 8x16 | 8x16 | 8x16 | 8x16 | 8x16 | 8x16 | 8x16 |
| DotClock (MHz) | 36 | 40 | 50 | 49.5 | 44.9 | 65 | 75 | 78.75 |
| HSweep (KHz) | 35.1 | 37.9 | 48.1 | 46.9 | 35.5 | 48.4 | 56.5 | 60 |
| Vrefresh (Hz) | 56 | 60 | 72 | 75 | 43 | 60 | 70 | 75 |
| **Clock:** | | | | | | | | |
| MiscOutReg | EB | EB | EB | EB | 2F | 2F | 2F | 2F |
| Mclock (MHz) | 68 | 68 | 68 | 68 | 68 | 68 | 68 | 68 |
| **Sequencer:** | | | | | | | | |
| ASyncReset (00) | 03 | 03 | 03 | 03 | 03 | 03 | 03 | 03 |
| ClockingMode (01) | 01 | 01 | 01 | 01 | 01 | 01 | 01 | 01 |
| MapMask (02) | 0F | 0F | 0F | 0F | 0F | 0F | 0F | 0F |
| CharFontSel (03) | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| MemoryMode (04) | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E |
| **CRT Controller:** | | | | | | | | |
| HTotal (00) | 7D | 81 | 7F | 81 | 9B | A5 | A3 | A1 |
| HDispEnblEnd (01) | 63 | 63 | 63 | 63 | 7F | 7F | 7F | 7F |
| HBlankStart (02) | 64 | 64 | 64 | 64 | 80 | 80 | 80 | 80 |
| HBlankEnd (03) | 9E | 82 | 80 | 82 | 9C | 86 | 84 | 81 |
| HSyncStart (04) | 68 | 6A | 6C | 67 | 82 | 84 | 84 | 83 |
| HSyncEnd (05) | 91 | 1A | 1B | 12 | 18 | 95 | 95 | 8F |
| VTotal (06) | 6F | 72 | 98 | 6F | 98 | 25 | 24 | 1F |
| VOverflow (07) | F0 | F0 | F0 | F0 | 1F | FD | FD | FD |
| PresetRowScan (08) | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| MaxScanLine (09) | 60 | 60 | 60 | 60 | 40 | 60 | 69 | 60 |
| CursorStart (0A) | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| CursorEnd (0B) | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| StartAddrH (0C) | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| StartAddrL (0D) | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| CurLocationH (0E) | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| CurLocationL (0F) | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| VSyncStart (10) | 58 | 58 | 7C | 58 | 82 | 02 | 02 | 00 |
| VSyncEnd (11) | 8A | 8C | 82 | 8B | 8C | 88 | 88 | 83 |
| VDispEnblEnd (12) | 57 | 57 | 57 | 57 | 7F | FF | FF | FF |
| Offset (13) | 64 | 64 | 64 | 64 | 80 | 80 | 80 | 80 |
| UnderLine (14) | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| VBlankStart (15) | 58 | 58 | 58 | 58 | 80 | 00 | 00 | 00 |
| VBlankEnd (16) | 6F | 72 | 98 | 6F | 98 | 25 | 24 | 1F |
| CRTCMode (17) | E3 | E3 | E3 | E3 | E3 | E3 | E3 | E3 |
| LineComp (18) | FF | FF | FF | FF | FF | FF | FF | FF |

## Table 19: Extended Linear Mode Parameter Tables for F65548 VGA Controller (continued)

| Mode No. | 32 56Hz | 32 60Hz | 32 72Hz | 32 75Hz | 34I 43Hz | 34 60Hz | 34 70Hz | 34 75Hz |
|---|---|---|---|---|---|---|---|---|
| **Attribute Controller:** | | | | | | | | |
| Palette0(00) | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| Palette1(01) | 01 | 01 | 01 | 01 | 01 | 01 | 01 | 01 |
| Palette2(02) | 02 | 02 | 02 | 02 | 02 | 02 | 02 | 02 |
| Palette3(03) | 03 | 03 | 03 | 03 | 03 | 03 | 03 | 03 |
| Palette4(04) | 04 | 04 | 04 | 04 | 04 | 04 | 04 | 04 |
| Palette5(05) | 05 | 05 | 05 | 05 | 05 | 05 | 05 | 05 |
| Palette6(06) | 06 | 06 | 06 | 06 | 06 | 06 | 06 | 06 |
| Palette7(07) | 07 | 07 | 07 | 07 | 07 | 07 | 07 | 07 |
| Palette8(08) | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 |
| Palette9(09) | 09 | 09 | 09 | 09 | 09 | 09 | 09 | 09 |
| PaletteA(0A) | 0A | 0A | 0A | 0A | 0A | 0A | 0A | 0A |
| PaletteB(0B) | 0B | 0B | 0B | 0B | 0B | 0B | 0B | 0B |
| PaletteC(0C) | 0C | 0C | 0C | 0C | 0C | 0C | 0C | 0C |
| PaletteD(0D) | 0D | 0D | 0D | 0D | 0D | 0D | 0D | 0D |
| PaletteE(0E) | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E |
| PaletteF(0F) | 0F | 0F | 0F | 0F | 0F | 0F | 0F | 0F |
| ModeCntl(10) | 01 | 01 | 01 | 01 | 01 | 01 | 01 | 01 |
| OverScan(11) | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| CPlaneEnable(12) | 0F | 0F | 0F | 0F | 0F | 0F | 0F | 0F |
| PixelPanning(13) | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| **Graphics Controller:** | | | | | | | | |
| Set/Reset(00) | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| EnableSet/Reset(01) | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| ColorCompare(02) | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| DataRotate(03) | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| ReadMapSelect(04) | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| ModeRegister(05) | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| MisceRegister(06) | 05 | 05 | 05 | 05 | 05 | 05 | 05 | 05 |
| ColorDontCare(07) | 0F | 0F | 0F | 0F | 0F | 0F | 0F | 0F |
| BitMask(08) | FF | FF | FF | FF | FF | FF | FF | FF |
| **Palette Mask Register:** | | | | | | | | |
| 3C6 | FF | FF | FF | FF | FF | FF | FF | FF |
| **Extended Registers:** | | | | | | | | |
| XR0B | 15 | 15 | 15 | 15 | 15 | 15 | 15 | 15 |
| XR10 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| XR11 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| XR19 | xx | xx | xx | xx | 4C | xx | xx | xx |
| XR28 | 10 | 10 | 10 | 10 | 30 | 10 | 10 | 10 |
| XR04 Bit 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| XR0C | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| XR06 Bits 3,2 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |

## Table 20: Extended Linear Mode Parameter Tables for F65548 VGA Controller (continued)

| Mode No. | 40 60Hz | 40 72Hz | 40 75Hz | 41 60Hz | 41 72Hz | 41 75Hz | 42 56Hz | 42 60Hz |
|---|---|---|---|---|---|---|---|---|
| Mode Resolution | **640x480** | **640x480** | **640x480** | **640x480** | **640x480** | **640x480** | **800x600** | **800x600** |
| BitsPerPixel | **15** | **15** | **15** | **16** | **16** | **16** | **15** | **15** |
| Interlace/Non-Interlace | NI | NI | NI | NI | NI | NI | NI | NI |
| Graphics/Text | G | G | G | G | G | G | G | G |
| Planar/PackPix | LI/PP | LI/PP | LI/PP | LI/PP | LI/PP | LI/PP | LI/PP | LI/PP |
| MemoryNeeded | 1MB | 1MB | 1MB | 1MB | 1MB | 1MB | 1MB | 1MB |
| VideoBuffer | >1MB | >1MB | >1MB | >1MB | >1MB | >1MB | >1MB | >1MB |
| CPUInterface | SChain | SChain | SChain | SChain | SChain | SChain | SChain | SChain |
| HColumns | 80 | 80 | 80 | 80 | 80 | 80 | 80 | 80 |
| VRows | 30 | 30 | 30 | 30 | 30 | 30 | 30 | 30 |
| FontSize | 8x16 | 8x16 | 8x16 | 8x16 | 8x16 | 8x16 | 8x16 | 8x16 |
| DotClock (MHz) | 50.35 | 63 | 63 | 50.35 | 63 | 63 | 50.35 | 50.35 |
| HSweep (KHz) | 31.5 | 37.9 | 37.5 | 31.5 | 37.9 | 37.5 | 31.5 | 31.5 |
| Vrefresh (Hz) | 60 | 72 | 75 | 60 | 72 | 75 | 60 | 60 |
| **Clock:** | | | | | | | | |
| MiscOutReg | EB | EB | EB | EB | EB | EB | EB | EB |
| Mclock (MHz) | 68 | 68 | 68 | 68 | 68 | 68 | 68 | 68 |
| **Sequencer:** | | | | | | | | |
| ASyncReset (00) | 03 | 03 | 03 | 03 | 03 | 03 | 03 | 03 |
| ClockingMode (01) | 01 | 01 | 01 | 01 | 01 | 01 | 01 | 01 |
| MapMask (02) | 0F | 0F | 0F | 0F | 0F | 0F | 0F | 0F |
| CharFontSel (03) | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| MemoryMode (04) | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E |
| **CRT Controller:** | | | | | | | | |
| HTotal (00) | C5 | CD | CF | C5 | CD | CF | FE | 05 |
| HDispEnblEnd (01) | 9F | 9F | 9F | 9F | 9F | 9F | C7 | C7 |
| HBlankStart (02) | A1 | A1 | A1 | A1 | A1 | A1 | C9 | C9 |
| HBlankEnd (03) | 85 | 8D | 8F | 85 | 8D | 8F | 9E | 85 |
| HSyncStart (04) | A5 | A7 | A5 | A5 | A7 | A5 | CF | D3 |
| HSyncEnd (05) | 1D | 10 | 14 | 1D | 10 | 14 | A2 | 12 |
| VTotal (06) | 0B | 06 | F2 | 0B | 06 | F2 | 6F | 72 |
| VOverflow (07) | 3E | 3E | 0F | 3E | 3E | 0F | F0 | F0 |
| PresetRowScan (08) | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| MaxScanLine (09) | 40 | 40 | 40 | 40 | 40 | 40 | 60 | 60 |
| CursorStart (0A) | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| CursorEnd (0B) | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| StartAddrH (0C) | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| StartAddrL (0D) | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| CurLocationH (0E) | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| CurLocationL (0F) | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| VSyncStart (10) | EA | E8 | E0 | EA | E8 | E0 | 59 | 59 |
| VSyncEnd (11) | 8C | 8B | 83 | 8C | 8B | 83 | 8B | 8D |
| VDispEnblEnd (12) | DF | DF | DF | DF | DF | DF | 57 | 57 |
| Offset (13) | A0 | A0 | A0 | A0 | A0 | A0 | C8 | C8 |
| UnderLine (14) | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| VBlankStart (15) | E7 | E7 | E0 | E7 | E7 | E0 | 58 | 58 |
| VBlankEnd (16) | 04 | FF | F2 | 04 | FF | F2 | 6F | 72 |
| CRTCMode (17) | E3 | E3 | E3 | E3 | E3 | E3 | E3 | E3 |
| LineComp (18) | FF | FF | FF | FF | FF | FF | FF | FF |

## Table 21: Extended Linear Mode Parameter Tables for F65548 VGA Controller (continued)

| Mode No. | 40 60Hz | 40 72Hz | 40 75Hz | 41 60Hz | 41 72Hz | 41 75Hz | 42 56Hz | 42 60Hz |
|---|---|---|---|---|---|---|---|---|
| **Attribute Controller:** | | | | | | | | |
| Palette0(00) | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| Palette1(01) | 01 | 01 | 01 | 01 | 01 | 01 | 01 | 01 |
| Palette2(02) | 02 | 02 | 02 | 02 | 02 | 02 | 02 | 02 |
| Palette3(03) | 03 | 03 | 03 | 03 | 03 | 03 | 03 | 03 |
| Palette4(04) | 04 | 04 | 04 | 04 | 04 | 04 | 04 | 04 |
| Palette5(05) | 05 | 05 | 05 | 05 | 05 | 05 | 05 | 05 |
| Palette6(06) | 06 | 06 | 06 | 06 | 06 | 06 | 06 | 06 |
| Palette7(07) | 07 | 07 | 07 | 07 | 07 | 07 | 07 | 07 |
| Palette8(08) | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 |
| Palette9(09) | 09 | 09 | 09 | 09 | 09 | 09 | 09 | 09 |
| PaletteA(0A) | 0A | 0A | 0A | 0A | 0A | 0A | 0A | 0A |
| PaletteB(0B) | 0B | 0B | 0B | 0B | 0B | 0B | 0B | 0B |
| PaletteC(0C) | 0C | 0C | 0C | 0C | 0C | 0C | 0C | 0C |
| PaletteD(0D) | 0D | 0D | 0D | 0D | 0D | 0D | 0D | 0D |
| PaletteE(0E) | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E |
| PaletteF(0F) | 0F | 0F | 0F | 0F | 0F | 0F | 0F | 0F |
| ModeCntl(10) | 01 | 01 | 01 | 01 | 01 | 01 | 01 | 01 |
| OverScan(11) | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| CPlaneEnable(12) | 0F | 0F | 0F | 0F | 0F | 0F | 0F | 0F |
| PixelPanning(13) | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| **Graphics Controller:** | | | | | | | | |
| Set/Reset(00) | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| EnableSet/Reset(01) | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| ColorCompare(02) | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| DataRotate(03) | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| ReadMapSelect(04) | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| ModeRegister(05) | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| MisceRegister(06) | 05 | 05 | 05 | 05 | 05 | 05 | 05 | 05 |
| ColorDontCare(07) | 0F | 0F | 0F | 0F | 0F | 0F | 0F | 0F |
| BitMask(08) | FF | FF | FF | FF | FF | FF | FF | FF |
| **Palette Mask Register:** | | | | | | | | |
| 3C6 | FF | FF | FF | FF | FF | FF | FF | FF |
| **Extended Registers:** | | | | | | | | |
| XR0B | 15 | 15 | 15 | 15 | 15 | 15 | 15 | 15 |
| XR10 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| XR11 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| XR19 | xx | xx | xx | xx | xx | xx | xx | xx |
| XR28 | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 10 |
| XR04 Bit 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| XR0C | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| XR06 Bits 3,2 | 01 | 01 | 01 | 11 | 11 | 11 | 01 | 01 |

## Extended Mode Parameter Tables

### Table 22: Extended Mode Parameter Tables for F65548 VGA Controller

| Mode No. | 43 56Hz | 43 60Hz | 50 52Hz | 60 70Hz | 61 70Hz | 6A/70 56Hz | 6A/70 60Hz | 6A/70 72Hz | 6A/70 75Hz |
|---|---|---|---|---|---|---|---|---|---|
| Mode Resolution | **800x600** | **800x600** | **640x480** | **1056x400** | **1056x400** | **800x600** | **800x600** | **800x600** | **800x600** |
| BitsPerPixel | **16** | **16** | **24** | | | **4** | **4** | **4** | **4** |
| Interlace/Non-Interlace | NI | NI | NI | NI | NI | NI | NI | NI | NI |
| Graphics/Text | G | G | G | T | T | G | G | G | G |
| Planar/PackPix | LI/PP | LI/PP | LI/PP | | | Pl | Pl | Pl | Pl |
| MemoryNeeded | 1MB | 1MB | 1MB | 256KB | 256KB | 256KB | 256KB | 256KB | 256KB |
| VideoBuffer | >1MB | >1MB | >1MB | B800 | B800 | A000 | A000 | A000 | A000 |
| CPUInterface | SChain | SChain | SChain | Odd/Evn | Odd/Evn | AllP | AllP | AllP | AllP |
| HColumns | 100 | 100 | 80 | 132/128 | 132/128 | 100 | 100 | 100 | 100 |
| VRows | 37 | 37 | 30 | 25 | 50 | 37+ | 37+ | 37+ | 37+ |
| FontSize | 8x16 | 8x16 | 8x16 | 8x16 | 8x8 | 8x16 | 8x16 | 8x16 | 8x16 |
| DotClock (MHz) | 72 | 80 | 75.525 | 41.539 | 41.539 | 36 | 40 | 50 | 49.5 |
| HSweep (KHz) | 35.1 | 37.9 | 31.5 | 31.5 | 31.5 | 35.1 | 37.9 | 48.1 | 46.9 |
| Vrefresh (Hz) | 56 | 60 | 60 | 70 | 70 | 56 | 60 | 72 | 75 |
| **Clock:** | | | | | | | | | |
| MiscOutReg | 2B | 2B | EB | 6B | 6B | EB | EB | EB | EB |
| Mclock (MHz) | 68 | 68 | 68 | 68 | 68 | 68 | 68 | 68 | 68 |
| **Sequencer:** | | | | | | | | | |
| ASyncReset (00) | 03 | 03 | 03 | 03 | 03 | 03 | 03 | 03 | 03 |
| ClockingMode (01) | 01 | 01 | 01 | 01 | 01 | 01 | 01 | 01 | 01 |
| MapMask (02) | 0F | 0F | 0F | 03 | 03 | 0F | 0F | 0F | 0F |
| CharFontSel (03) | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| MemoryMode (04) | 0E | 0E | 0E | 02 | 02 | 06 | 06 | 06 | 06 |
| **CRT Controller:** | | | | | | | | | |
| HTotal (00) | FE | 05 | 2D | A0 | A0 | 7B | 7F | 7D | 7F |
| HDispEnblEnd (01) | C7 | C7 | EF | 83 | 83 | 63 | 63 | 63 | 63 |
| HBlankStart (02) | C9 | C9 | F2 | 85 | 85 | 64 | 64 | 64 | 64 |
| HBlankEnd (03) | 9E | 85 | 8C | 82 | 82 | 9E | 82 | 80 | 82 |
| HSyncStart (04) | CF | D3 | FC | 8A | 8A | 68 | 6A | 6C | 67 |
| HSyncEnd (05) | A2 | 12 | 84 | 81 | 81 | 91 | 1A | 1B | 12 |
| VTotal (06) | 6F | 72 | 0B | BF | BF | 6F | 72 | 98 | 6F |
| VOverflow (07) | F0 | F0 | 3E | 1F | 1F | F0 | F0 | F0 | F0 |
| PresetRowScan (08) | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| MaxScanLine (09) | 60 | 60 | 40 | 4F | 47 | 60 | 60 | 60 | 60 |
| CursorStart (0A) | 00 | 00 | 00 | 0D | 06 | 00 | 00 | 00 | 00 |
| CursorEnd (0B) | 00 | 00 | 00 | 0E | 07 | 00 | 00 | 00 | 00 |
| StartAddrH (0C) | 00 | 00 | 00 | 00 | 00 | FF | FF | FF | FF |
| StartAddrL (0D) | 00 | 00 | 00 | 00 | 00 | FF | FF | FF | FF |
| CurLocationH (0E) | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| CurLocationL (0F) | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| VSyncStart (10) | 59 | 59 | EA | 9C | 9C | 58 | 58 | 7C | 58 |
| VSyncEnd (11) | 8B | 8D | 8C | 8E | 8E | 8A | 8C | 82 | 8B |
| VDispEnblEnd (12) | 57 | 57 | DF | 8F | 8F | 57 | 57 | 57 | 57 |
| Offset (13) | C8 | C8 | F0 | 42 | 42 | 32 | 32 | 32 | 32 |
| UnderLine (14) | 00 | 00 | 00 | 1F | 1F | 00 | 00 | 00 | 00 |
| VBlankStart (15) | 58 | 58 | E7 | 96 | 96 | 58 | 58 | 58 | 58 |
| VBlankEnd (16) | 6F | 72 | 04 | B9 | B9 | 6F | 72 | 98 | 6F |
| CRTCMode (17) | E3 | E3 | E3 | A3 | A3 | E3 | E3 | E3 | E3 |
| LineComp (18) | FF | FF | FF | FF | FF | FF | FF | FF | FF |

## Table 23: Extended Mode Parameter Tables for F65548 VGA Controller (continued)

| Mode No. | 43 56Hz | 43 60Hz | 50 60Hz | 60 70Hz | 61 70Hz | 6A/70 56Hz | 6A/70 60Hz | 6A/70 72Hz | 6A/70 75Hz |
|---|---|---|---|---|---|---|---|---|---|
| **Attribute Controller:** | | | | | | | | | |
| Palette0(00) | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| Palette1(01) | 01 | 01 | 01 | 01 | 01 | 01 | 01 | 01 | 01 |
| Palette2(02) | 02 | 02 | 02 | 02 | 02 | 02 | 02 | 02 | 02 |
| Palette3(03) | 03 | 03 | 03 | 03 | 03 | 03 | 03 | 03 | 03 |
| Palette4(04) | 04 | 04 | 04 | 04 | 04 | 04 | 04 | 04 | 04 |
| Palette5(05) | 05 | 05 | 05 | 05 | 05 | 05 | 05 | 05 | 05 |
| Palette6(06) | 06 | 06 | 06 | 14 | 14 | 14 | 14 | 14 | 14 |
| Palette7(07) | 07 | 07 | 07 | 07 | 07 | 07 | 07 | 07 | 07 |
| Palette8(08) | 08 | 08 | 08 | 38 | 38 | 38- | 38- | 38- | 38- |
| Palette9(09) | 09 | 09 | 09 | 39 | 39 | 39 | 39 | 39 | 39 |
| PaletteA(0A) | 0A | 0A | 0A | 3A | 3A | 3A | 3A | 3A | 3A |
| PaletteB(0B) | 0B | 0B | 0B | 3B | 3B | 3B | 3B | 3B | 3B |
| PaletteC(0C) | 0C | 0C | 0C | 3C | 3C | 3C | 3C | 3C | 3C |
| PaletteD(0D) | 0D | 0D | 0D | 3D | 3D | 3D | 3D | 3D | 3D |
| PaletteE(0E) | 0E | 0E | 0E | 3E | 3E | 3E | 3E | 3E | 3E |
| PaletteF(0F) | 0F | 0F | 0F | 3F | 3F | 3F | 3F | 3F | 3F |
| ModeCntl(10) | 01 | 01 | 01 | 0C | 0C | 01 | 01 | 01 | 01 |
| OverScan(11) | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| CPlaneEnable(12) | 0F | 0F | 0F | 0F | 0F | 0F | 0F | 0F | 0F |
| PixelPanning(13) | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| **Graphics Controller:** | | | | | | | | | |
| Set/Reset(00) | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| EnableSet/Reset(01) | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| ColorCompare(02) | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| DataRotate(03) | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| ReadMapSelect(04) | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| ModeRegister(05) | 00 | 00 | 00 | 10 | 10 | 00 | 00 | 00 | 00 |
| MisceRegister(06) | 05 | 05 | 05 | 0E | 0E | 05 | 05 | 05 | 05 |
| ColorDontCare(07) | 0F | 0F | 0F | 00 | 00 | 0F | 0F | 0F | 0F |
| BitMask(08) | FF | FF | FF | FF | FF | FF | FF | FF | FF |
| **Extended Registers:** | | | | | | | | | |
| XR0B | 15 | 15 | 15 | 00 | 00 | 01 | 01 | 01 | 01 |
| XR10 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| XR11 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| XR19 | xx | xx | xx | xx | xx | xx | xx | xx | xx |
| XR28 | 10 | 10 | 10 | 00 | 00 | 00 | 00 | 00 | 00 |
| XR04 Bit 2 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| XR0C | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| XR06 Bits 3,2 | 11 | 11 | 10 | 00 | 00 | 00 | 00 | 00 | 00 |

## Table 24: Extended Mode Parameter Tables for F65548 VGA Controller

| Mode No. | 72I/75I | 72/75 | 72/75 | 72/75 | 78 | 79 | 79 | 79 |
|---|---|---|---|---|---|---|---|---|
| Mode Resolution | **1024x768** | **1024x768** | **1024x768** | **1024x768** | **640x400** | **640x480** | **640x480** | **640x480** |
| BitsPerPixel | **4** | **4** | **4** | **4** | **8** | **8** | **8** | **8** |
| Interlace/Non-Interlace | I | NI | NI | NI | NI | NI | NI | NI |
| Graphics/Text | G | G | G | G | G | G | G | G |
| Planar/PackPix | Pl | Pl | Pl | Pl | Pp | Pp | Pp | Pp |
| MemoryNeeded | 512KB | 512KB | 512KB | 512KB | 512KB | 512KB | 512KB | 512KB |
| VideoBuffer | A000 | A000 | A000 | A000 | A000 | A000 | A000 | A000 |
| CPUInterface | AllP | AllP | AllP | AllP | SChain | SChain | SChain | SChain |
| Monitor | A,B,C | C | C | C | A,B,C | A,B,C | A,B,C | A,B,C |
| HColumns | 128 | 128 | 128 | 128 | 80 | 80 | 80 | 80 |
| VRows | 48 | 48 | 48 | 48 | 25 | 30 | 30 | 30 |
| FontSize | 8x16 | 8x16 | 8x16 | 8x16 | 8x16 | 8x16 | 8x16 | 8x16 |
| DotClock (MHz) | 44.9 | 65 | 75 | 78.75 | 25.175 | 25.175 | 31.5 | 31.5 |
| HSweep (KHz) | 35.5 | 48.5 | 56.5 | 60 | 31.5 | 31.5 | 37.9 | 37.5 |
| Vrefresh (Hz) | 43 | 60 | 70 | 75 | 70 | 60 | 72 | 75 |
| **Clock:** | | | | | | | | |
| MiscOutReg | 2F | 2F | 2F | 2F | 6B | EB/E3 | EB/E3 | EB/E3 |
| Mclock (MHz) | 68 | 68 | 68 | 68 | 68 | 68 | 68 | 68 |
| **Sequencer:** | | | | | | | | |
| ASyncReset (00) | 03 | 03 | 03 | 03 | 03 | 03 | 03 | 03 |
| ClockingMode (01) | 01 | 01 | 01 | 01 | 01 | 01 | 01 | 01 |
| MapMask (02) | 0F | 0F | 0F | 0F | 0F | 0F | 0F | 0F |
| CharFontSel (03) | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| MemoryMode (04) | 06 | 06 | 06 | 06 | 0E | 0E | 0E | 0E |
| **CRT Controller:** | | | | | | | | |
| HTotal (00) | 99 | A3 | A1 | 9F | 61 | 61 | 65 | 66 |
| HDispEnblEnd (01) | 7F | 7F | 7F | 7F | 4F | 4F | 4F | 4F |
| HBlankStart (02) | 80 | 80 | 80 | 80 | 50 | 50 | 50 | 50 |
| HBlankEnd (03) | 9C | 86 | 84 | 81 | 82 | 82 | 86 | 87 |
| HSyncStart (04) | 83 | 85 | 84 | 83 | 54 | 53 | 54 | 53 |
| HSyncEnd (05) | 19 | 96 | 95 | 8F | 80 | 9F | 99 | 9B |
| VTotal (06) | 98 | 25 | 24 | 1F | BF | 0B | 06 | F2 |
| VOverflow (07) | 1F | FD | FD | FD | 1F | 3E | 3E | 00 |
| PresetRowScan (08) | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| MaxScanLine (09) | 40 | 60 | 60 | 60 | 40 | 40 | 40 | 40 |
| CursorStart (0A) | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| CursorEnd (0B) | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| StartAddrH (0C) | FF | FF | FF | FF | 00 | 00 | 00 | 00 |
| StartAddrL (0D) | FF | FF | FF | FF | 00 | 00 | 00 | 00 |
| CurLocationH (0E) | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| CurLocationL (0F) | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| VSyncStart (10) | 82 | 02 | 02 | 00 | 9C | EA | E8 | E0 |
| VSyncEnd (11) | 8C | 88 | 88 | 83 | 8E | 8C | 8B | 83 |
| VDispEnblEnd (12) | 7F | FF | FF | FF | 8F | DF | DF | DF |
| Offset (13) | 40 | 40 | 40 | 40 | 50 | 50 | 50 | 50 |
| UnderLine (14) | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| VBlankStart (15) | 80 | 00 | 00 | 00 | 96 | E7 | E7 | E0 |
| VBlankEnd (16) | 98 | 25 | 24 | 1F | B9 | 04 | FF | F2 |
| CRTCMode (17) | E3 | E3 | E3 | E3 | E3 | E3 | E3 | E3 |
| LineComp (18) | FF | FF | FF | FF | FF | FF | FF | FF |

## Table 25: Extended Mode Parameter Tables for F65548 VGA Controller (continued)

| Mode No. | 72I/75I | 72/75 | 72/75 | 72/75 | 78 | 79 | 79 | 79 |
|---|---|---|---|---|---|---|---|---|
| **Attribute Controller:** | | | | | | | | |
| Palette0(00) | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| Palette1(01) | 01 | 01 | 01 | 01 | 01 | 01 | 01 | 01 |
| Palette2(02) | 02 | 02 | 02 | 02 | 02 | 02 | 02 | 02 |
| Palette3(03) | 03 | 03 | 03 | 03 | 03 | 03 | 03 | 03 |
| Palette4(04) | 04 | 04 | 04 | 04 | 04 | 04 | 04 | 04 |
| Palette5(05) | 05 | 05 | 05 | 05 | 05 | 05 | 05 | 05 |
| Palette6(06) | 14 | 14 | 14 | 14 | 06 | 06 | 06 | 06 |
| Palette7(07) | 07 | 07 | 07 | 07 | 07 | 07 | 07 | 07 |
| Palette8(08) | 38 | 38 | 38 | 38 | 08 | 08 | 08 | 08 |
| Palette9(09) | 39 | 39 | 39 | 39 | 09 | 09 | 09 | 09 |
| PaletteA(0A) | 3A | 3A | 3A | 3A | 0A | 0A | 0A | 0A |
| PaletteB(0B) | 3B | 3B | 3B | 3B | 0B | 0B | 0B | 0B |
| PaletteC(0C) | 3C | 3C | 3C | 3C | 0C | 0C | 0C | 0C |
| PaletteD(0D) | 3D | 3D | 3D | 3D | 0D | 0D | 0D | 0D |
| PaletteE(0E) | 3E | 3E | 3E | 3E | 0E | 0E | 0E | 0E |
| PaletteF(0F) | 3F | 3F | 3F | 3F | 0F | 0F | 0F | 0F |
| ModeCntl(10) | 01 | 01 | 01 | 01 | 01 | 01 | 01 | 01 |
| OverScan(11) | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| CPlaneEnable(12) | 0F | 0F | 0F | 0F | 0F | 0F | 0F | 0F |
| PixelPanning(13) | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| **Graphics Controller:** | | | | | | | | |
| Set/Reset(00) | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| EnableSet/Reset(01) | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| ColorCompare(02) | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| DataRotate(03) | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| ReadMapSelect(04) | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| ModeRegister(05) | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| MisceRegister(06) | 05 | 05 | 05 | 05 | 05 | 05 | 05 | 05 |
| ColorDontCare(07) | 0F | 0F | 0F | 0F | 0F | 0F | 0F | 0F |
| BitMask(08) | FF | FF | FF | FF | FF | FF | FF | FF |
| **Extended Registers:** | | | | | | | | |
| XR0B | 01 | 01 | 01 | 01 | 05 | 05 | 05 | 05 |
| XR10 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| XR11 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| XR19 | 4C | xx | xx | xx | xx | xx | xx | xx |
| XR28 | 20 | 00 | 00 | 00 | 10 | 10 | 10 | 10 |
| XR04 Bit 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| XR0C | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| XR06 Bits 3,2 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |

## Table 26: Extended Mode Parameter Tables for F65548 VGA Controller

| Mode No. | 7C | 7C | 7C | 7C | 7EI | 7E | 7E | 7E |
|---|---|---|---|---|---|---|---|---|
| Mode Resolution | 800x600 | 800x600 | 800x600 | 800x600 | 1024x768 | 1024x768 | 1024x768 | 1024x768 |
| BitsPerPixel | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 |
| Interlace/Non-Interlace | NI | NI | NI | NI | I | NI | NI | NI |
| Graphics/Text | G | G | G | G | G | G | G | G |
| Planar/PackPix | Pp | Pp | Pp | Pp | Pp | Pp | Pp | Pp |
| MemoryNeeded | 512KB | 512KB | 512KB | 512KB | 1MB | 1MB | 1MB | 1MB |
| VideoBuffer | A000 | A000 | A000 | A000 | A000 | A000 | A000 | A000 |
| CPUInterface | SChain | SChain | SChain | SChain | SChain | SChain | SChain | SChain |
| Monitor | B,C | B,C | B,C | B,C | C | C | C | C |
| HColumns | 100 | 100 | 100 | 100 | 128 | 128 | 128 | 128 |
| VRows | 37 | 37 | 37 | 37 | 48 | 48 | 48 | 48 |
| FontSize | 8x16 | 8x16 | 8x16 | 8x16 | 8x16 | 8x16 | 8x16 | 8x16 |
| DotClock (MHz) | 36 | 40 | 50 | 49.5 | 44.9 | 65 | 75 | 78.75 |
| HSweep (KHz) | 35.1 | 37.9 | 48.1 | 46.9 | 35.5 | 48.4 | 56.5 | 60 |
| Vrefresh (Hz) | 56 | 60 | 72 | 75 | 43 | 60 | 70 | 75 |
| **Clock:** | | | | | | | | |
| MiscOutReg | 2B | 2B | 2B | 2B | 2F | 2F | 2F | 2F |
| Mclock (MHz) | 68 | 68 | 68 | 68 | 68 | 68 | 68 | 68 |
| **Sequencer:** | | | | | | | | |
| ASyncReset (00) | 03 | 03 | 03 | 03 | 03 | 03 | 03 | 03 |
| ClockingMode (01) | 01 | 01 | 01 | 01 | 01 | 01 | 01 | 01 |
| MapMask (02) | 0F | 0F | 0F | 0F | 0F | 0F | 0F | 0F |
| CharFontSel (03) | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| MemoryMode (04) | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E |
| **CRT Controller:** | | | | | | | | |
| HTotal (00) | 7D | 81 | 7F | 81 | 9B | A5 | A3 | A1 |
| HDispEnblEnd (01) | 63 | 63 | 63 | 63 | 7F | 7F | 7F | 7F |
| HBlankStart (02) | 64 | 64 | 64 | 64 | 80 | 80 | 80 | 80 |
| HBlankEnd (03) | 9E | 82 | 80 | 82 | 9C | 86 | 84 | 81 |
| HSyncStart (04) | 68 | 6A | 6C | 67 | 82 | 84 | 84 | 83 |
| HSyncEnd (05) | 91 | 1A | 1B | 12 | 18 | 95 | 95 | 8F |
| VTotal (06) | 6F | 72 | 98 | 6F | 98 | 25 | 24 | 1F |
| VOverflow (07) | F0 | F0 | F0 | F0 | 1F | FD | FD | FD |
| PresetRowScan (08) | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| MaxScanLine (09) | 60 | 60 | 60 | 60 | 40 | 60 | 60 | 60 |
| CursorStart (0A) | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| CursorEnd (0B) | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| StartAddrH (0C) | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| StartAddrL (0D) | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| CurLocationH (0E) | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| CurLocationL (0F) | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| VSyncStart (10) | 58 | 58 | 7C | 58 | 82 | 02 | 02 | 00 |
| VSyncEnd (11) | 8A | 8C | 82 | 8B | 8C | 88 | 88 | 83 |
| VDispEnblEnd (12) | 57 | 57 | 57 | 57 | 7F | FF | FF | FF |
| Offset (13) | 64 | 64 | 64 | 64 | 80 | 80 | 80 | 80 |
| UnderLine (14) | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| VBlankStart (15) | 58 | 58 | 58 | 58 | 80 | 00 | 00 | 00 |
| VBlankEnd (16) | 6F | 72 | 98 | 6F | 98 | 25 | 24 | 1F |
| CRTCMode (17) | E3 | E3 | E3 | E3 | E3 | E3 | E3 | E3 |
| LineComp (18) | FF | FF | FF | FF | FF | FF | FF | FF |

## Table 27: Extended Mode Parameter Tables for F65548 VGA Controller (continued)

| Mode No. | 7C | 7C | 7C | 7C | 7EI | 7E | 7E | 7E |
|---|---|---|---|---|---|---|---|---|
| **Attribute Controller:** | | | | | | | | |
| Palette0(00) | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| Palette1(01) | 01 | 01 | 01 | 01 | 01 | 01 | 01 | 01 |
| Palette2(02) | 02 | 02 | 02 | 02 | 02 | 02 | 02 | 02 |
| Palette3(03) | 03 | 03 | 03 | 03 | 03 | 03 | 03 | 03 |
| Palette4(04) | 04 | 04 | 04 | 04 | 04 | 04 | 04 | 04 |
| Palette5(05) | 05 | 05 | 05 | 05 | 05 | 05 | 05 | 05 |
| Palette6(06) | 06 | 06 | 06 | 06 | 06 | 06 | 06 | 06 |
| Palette7(07) | 07 | 07 | 07 | 07 | 07 | 07 | 07 | 07 |
| Palette8(08) | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 |
| Palette9(09) | 09 | 09 | 09 | 09 | 09 | 09 | 09 | 09 |
| PaletteA(0A) | 0A | 0A | 0A | 0A | 0A | 0A | 0A | 0A |
| PaletteB(0B) | 0B | 0B | 0B | 0B | 0B | 0B | 0B | 0B |
| PaletteC(0C) | 0C | 0C | 0C | 0C | 0C | 0C | 0C | 0C |
| PaletteD(0D) | 0D | 0D | 0D | 0D | 0D | 0D | 0D | 0D |
| PaletteE(0E) | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E |
| PaletteF(0F) | 0F | 0F | 0F | 0F | 0F | 0F | 0F | 0F |
| ModeCntl(10) | 01 | 01 | 01 | 01 | 01 | 01 | 01 | 01 |
| OverScan(11) | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| CPlaneEnable(12) | 0F | 0F | 0F | 0F | 0F | 0F | 0F | 0F |
| PixelPanning(13) | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| **Graphics Controller:** | | | | | | | | |
| Set/Reset(00) | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| EnableSet/Reset(01) | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| ColorCompare(02) | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| DataRotate(03) | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| ReadMapSelect(04) | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| ModeRegister(05) | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| MisceRegister(06) | 05 | 05 | 05 | 05 | 05 | 05 | 05 | 05 |
| ColorDontCare(07) | 0F | 0F | 0F | 0F | 0F | 0F | 0F | 0F |
| BitMask(08) | FF | FF | FF | FF | FF | FF | FF | FF |
| **Extended Registers:** | | | | | | | | |
| XR0B | 05 | 05 | 05 | 05 | 05 | 05 | 05 | 05 |
| XR10 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| XR11 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| XR19 | xx | xx | xx | xx | 4C | xx | xx | xx |
| XR28 | 10 | 10 | 10 | 10 | 30 | 10 | 10 | 10 |
| XR04 Bit 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| XR0C | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| XR06 Bits 3,2 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |

# APPENDIX C - SUSPEND/RESUME PROCEDURE

## Introduction

The following section describes the Suspend/Resume procedure required for the Chips and Technologies 65548 High Performance Flat Panel/CRT VGA Controllers.

Following this procedure will allow the 65548 to perform optimally during Suspend/Resume operations. Ignoring this procedure will result in rare intermittent failures during Suspend/Resume operations. Chips and Technologies cannot be responsible for the operation of the 65548 during Suspend/Resume if this procedure is not properly followed.

This section contains brief description of the procedure followed by an example code. The actual method which implements this Suspend/Resume procedure depends on the system logic chip set as well as the power management software. Please remember that it is only an example code. If you have questions regarding this procedure, please contact your local Chips and Technologies, Inc. sales office.

## Operation

In Standby mode, the 65548 suspends all CPU, memory, and display activities. It places the DRAM(s) in slow- or self-refresh mode (XR52[6]), and may shut off the 14.31818MHz reference clock and/or the 32KHz depending on the configuration of the chip during Standby.

In slow-refresh mode (XR52[6] = 1), using the internal RCLK (XR33[6] = 0) for slow-refresh timing, the 14.31818MHz clock cannot be turned off. The 14.31818MHz clock generates the 37.5KHz RCLK used in the Standby slow-refresh timing. If the chip is using the external 32KHz on pin 154 (AA9) as the slow-refresh timing reference clock, then the 14.31818MHz clock can be shut off.

In self-refresh mode (XR52[6] = 0), the 14.31818MHz clock can be shut off. If the external 32KHz is also used, it can also be shut off during Standby.

The external 32KHz or internal RCLK is used for slow-refresh and panel power sequencing timing (XR33[6]).

If the clock(s) may be shut off, they must be shut off after waiting twice the time programmed in XR5B[3-0] (Panel Power Sequencing Delay Register - Power Down Delay) after the STNDBY# pin is asserted. This will allow the chip to completely finish all housekeeping activities after the STNDBY# pin is asserted.

When exiting Standby mode (Resume), the clocks must be applied (if turned off) to the chip and be stable before the STNDBY# pin may be de-asserted. After the STNDBY# pin has been de-asserted, the chip can be accessed after waiting twice the value time programmed in XR5B[7-4] (Power Up Delay). This will allow the chip to fully come out of Standby.

The VGA subsystem dissipates a minimum amount of power during Standby. Since the 65548 is a fully static device, the contents of the controller's registers and on-chip palette are maintained during Standby. Therefore, Standby mode provides fast Suspend/Resume operations. Standby mode may be activated by asserting the STNDBY# pin low or programming XR52[4] = 1. The only way to come out of Standby is by de-asserting the STNDBY# pin.

The 65548 has been designed to minimize power consumption during Standby in either Panel-only or Simultaneous modes. During these modes, it is assumed that AC power is not available and the system is running on batteries. During CRT-only mode, it is assumed that AC power is available and therefore power consumption does not need to be minimized.

To minimize power consumption during CRT-only mode, then it is recommended that the chip switch to Panel only mode before entering Standby.

The 65548 has also been designed to enter Standby mode only from Normal operation mode, therefore the 65548 cannot enter Standby mode when in the Panel-Off mode (XR52[3] = 1). To enter Standby mode from Panel-Off mode, it is recommended that it first come out of Panel-Off mode (Panel-On mode - XR52[3] = 0) then enter Standby mode.

It must be remembered that after setting XR52[3] = 0 (Panel-On mode), the chip cannot enter Standby mode until waiting twice the value time programmed into XR5B[7-4]. This will allow the chip to fully come out of Panel-Off mode.

The following example procedure and code assumes the chip is in Panel-only or Simultaneous modes and Normal operation before entering Standby.

## Procedure

In order to provide optimal Suspend/Resume operation (Standby mode) with the Chips and Technologies 65548, the following software procedure must be implemented in either the system BIOS or the power management software.

1. Before Entering Suspend Mode

   Software must execute the following procedure before asserting the STNDBY# pin of the 65548:

   a. SAVE the contents of register 3C6h (Color Palette Pixel Mask Register).

   b. PROGRAM register 3C6h to 00 - Disabling access to palette contents.

   c. SAVE all DAC registers (Video DAC State and Color Registers) using the Video BIOS function call 5FA1h.

   d. SAVE the GR (Graphics Register) Index.

   e. SAVE the contents of register GR06 (Mi scellaneous Register).

   f. PROGRAM register GR06 bit 0 to 0 - Setting controller to text mode.

   g. READ double word FFFF FFFFh from Video memory (see Example Code for procedure).

   h. Enter Standby mode by asserting the STNDBY# pin of the 65548.

      Wait a minimum time delay of twice the value programmed into register XR5B[3..0] (Panel Power Sequencing Register - Power Down) in msec. before turning off the external 14.31818MHz oscillator (if applicable). This is to allow the 65548 to completely finish all activities (‘house keeping’) after the STNDBY# pin is asserted .

2. After Exiting Suspend Mode (Resume)

After de-asserting the STNDBY# pin of the 65548, the software must execute the following procedure:

The 14.31818MHz external oscillator must be applied to the 65548 and stable before de-asserting the STNDBY# pin (if applicable).

Wait a minimum time delay of twice the value programmed into register XR5B[7..4] (Power Up) in msec. This is to allow the 65548 to completely come out of Standby after the STNDBY# pin is de-asserted.

a. RESTORE the saved contents of register GR06 (Miscellaneous Register).

b. RESTORE the saved GR (Graphics Register) Index.

c. RESTORE all DAC registers (Video DAC State and Color Registers) using the Video BIOS function call 5FA2h.

d. RESTORE the saved contents of register 3C6h (Color Palette Pixel Mask Register).

Example Code

```
;-------------------------------------------------------------------------
; Module Name           : STANDBY.asm
; Program Name          : STANDBY.com
; Description           : Standby (540/545)
; Date                  : Dec. 20, 1994
; Version               : 1.3
; Programmer            : Chips and Technologies, Inc.
; (C) 1994 Chips and Technologies, Inc.
;-------------------------------------------------------------------------
; Code Segment Starts
        code    segment
        assume  cs:code, ds:code, ss:code, es:code
        org     100h; for making program .COM type
begin:

STANDBY proc near

;*************************************************************************************************
; This delay routine is in case the system is already in Standby.
;*************************************************************************************************
        mov     dx,3d6h                 ; Set to XR Index
        in      al,dx                   ; Read XR Index
        push    ax                      ; Save XR Index
        mov     al,5Bh                  ; Set Index to 5B
        out     dx,al
        in      ax,dx                   ; Read contents of XR5B (Panel Power Sequencing Delay Register)
        mov     CS:XR5B,ax              ; Save XR5B
        pop     ax
        out     dx,al                   ; Restore XR Index
        call    Wdelay                  ; delay
```

```
;PREPARE FOR STANDBY
        mov     dx,3c6h
        in      al,dx                   ; Read Color Palette Pixel Mask Register
        mov     CS:D_3c6,al             ; Save Color Palette Pixel Mask Register
        mov     al,0
        out     dx,al                   ; Disable access to Palette contents
        mov     ax,5fa1h                        ; Video BIOS function call to Save Video State
        mov     cx,04h                  ; Video DAC state
        push    cs
        pop     es
        mov     bx,offset Buff_DAC      ; Set Correct buffer
        int     10h                     ; Save
        mov     dx,3ceh
        in      al,dx                   ; Read GR Index
        mov     cs:GR_Index,al          ; Save GR Index
        mov     al,6
        out     dx,al                   ; Set GR Index to GR06 (Miscellaneous Register)
        in      ax,dx                   ; Read contents of GR06
        mov     cs:GR06_Data,ax                 ; Save GR06 Index, Data
;;      and     ah,not 01h
;;      out     dx,ax                   ; Set for Text Mode
        .386
        push    eax
        push    ecx
        mov     ah,04h
        out     dx,ax                   ; Write 04 to GR06 (Set to Text Mode)
        out     dx,al                   ; Set GR Index to GR05 (Graphics Mode Register)
        in      ax,dx                   ; Read contents of GR05
        mov     CS:GR05,ax              ; Save contents of GR05
        mov     ah,00
        out     dx,ax                   ; Set GR05 to 00 (Write Mode 0)
        mov     dx,3c4h
        in      al,dx                   ; Read SR Index
        mov     CS:SR_Index,al          ; Save SR Index
        mov     al,04h
        out     dx,al                   ; Set SR Index to SR04 (Memory Mode Register)
        in      ax,dx                   ; Read contents of SR04
        mov     CS:SR04,ax              ; Save contents of SR04
        mov     ah,0EH
        out     dx,ax                   ; Set to Packed Pixel Mode
        mov     al,02h
```

```
out     dx,al                   ; Set SR Index to SR02 (Sequencer Plane/Map Mask Register)
in      ax,dx                   ; Read contents of SR02
mov     CS:SR02,ax              ; Save contents of SR02
mov     ah,0FH
out     dx,ax                   ; Enable all color planes
mov     dx,3d6h
in      al,dx                   ; Read XR Index
mov     CS:XR_Index,al          ; Save XR Index
mov     al,0Bh
out     dx,al
in      ax,dx                   ; Read contents of XR0B (CPU Paging Register)
mov     CS:XR0B,ax              ; Save contents of XR0B
mov     ah,05H
out     dx,ax                   ; Set Memory for Extended Packed Pixel Non-Linear Mode
mov     ax,0A000H
mov     es,ax
xor     si,si
mov     ecx,DWord Ptr es:[si]
mov     eax,0FFFFFFFFH
mov     DWord Ptr es:[si],eax   ; Write double word FFFF FFFFh to Video memory
mov     eax,DWord Ptr es:[si]   ; Read double word FFFF FFFFh from Video memory
mov     DWord Ptr es:[si],ecx
pop     ecx
pop     eax
.286
mov     dx,3ceh                 ; Set to GR Index
mov     ax,CS:GR06_Data                 ; Restore GR06
and     ah,not 01h
out     dx,ax                   ; Flip to Text Mode
mov     ax,CS:GR05
out     dx,ax                   ; Restore GR05
mov     dx,3c4h                 ; Set to SR Index
mov     ax,CS:SR04
out     dx,ax                   ; Restore SR04
mov     ax,CS:SR02
out     dx,ax                   ; Restore SR02
mov     al,SR_Index
out     dx,al                   ; Restore SR Index
mov     dx,3d6h                 ; Set to XR Index
mov     ax,CS:XR0B
out     dx,ax                   ; Restore XR0B
mov     al,XR_Index
out     dx,al                   ; Restore XR Index
```

```
; [ A ] STANDBY
;*****************************************************************************************
;
; Code to enter Standby should be placed here.  This code depends on the Standby
; implementation in the System
;
;*****************************************************************************************
        call    Sdelay                          ; delay
;*****************************************************************************************


; [ B] WAKE UP
;*****************************************************************************************
;
; Code to exit Standby should be placed here.  This code depends on the Standby
; implementation in the System.
;
;*****************************************************************************************
        call    Wdelay                          ;delay
; RESTORE STATE AFTER WAKEUP
        mov     dx,3ceh                         ; Set to GR Index
        mov     ax,cs:GR06_Data
        out     dx,ax                           ; Restore GR06
        mov     al,cs:GR_Index
        out     dx,al                           ; Restore GR Index
        mov     ax,5fa2h                                ; Video BIOS function call to Restore Video State
        mov     cx,04h                          ; Video DAC state and Color Registers
        push    cs
        pop     es
        mov     bx,offset Buff_DAC              ; From buffer
        int     10h                             ; Restore
        mov     dx,3c6h                         ; Set to Color Palette Pixel Mask  Register
        mov     al,CS:D_3c6
        out     dx,al                           ; Restore Color Palette Pixel Mask Register
        ret                                     ; Terminate

STANDBY     endp

; Wake-up delay routine
Wdelay  Proc    near
```

```
        mov     ax,CS:XR5B
        and     ah,0f0h                 ; Select Power Up Delay (bits 4 - 7)
        .386
        shr     ah,1                    ; 8 times
        .286
        xor     cx,cx
        mov     cl,ah                   ; CX = delay count in msec
        call    delay                   ; Call User System Specific Delay Routine
        ret

Wdelay  endp

; Standby delay routine
Sdelay  Proc    near
        mov     ax,CS:XR5B
        and     ah,0fh                  ; Select Power Down Delay (bits 3 - 0)
        mov     al,ah
        mov     ah,2*29                 ; msec
        mul     ah
        mov     cx,ax                   ; CX = delay count in msec
        call    delay                   ; Call User Specific Delay Routine
        ret

Sdelay  endp

; [C] User Specific Delay Routine
;*****************************************************************************************
; Code to implement User Specific Delay Routine goes here.
; This is to allow User to implement the delay routine based on the system requirements.
;
; Entry: CX = delay count in msec
;*****************************************************************************************
delay   Proc    near                    ; Delay

        ret
delay   endp
;*****************************************************************************************
```

```
;--------------------------
; Data Declaration
;--------------------------
D_3c6           db      ?
GR_Index        db      ?
GR05            dw      ?
GR06_Data       dw      ?
SR_Index        db      ?
SR02            dw      ?
SR04            dw      ?
XR_Index        db      ?
XR0B            dw      ?
XR5B            dw      ?
Buff_DAC        db      1000h dup(0)
code    ends
        end     begin
```

Note : [A], [B], and [C] should be implemented based on system requirements.

Chips and Technologies, Inc.
2950 Zanker Road
San Jose, California 95134
Phone: 408-434-0600
Telex: 272929 CHIPS UR
FAX: 408-526-2275

Title: OC65548 OEM Reference Guide
Date: 10/2/95
P/N: 1401
Publication No.: UG 154.1
Stock No.: 050154-001
Revision No.: 1.1