

# 65555

HiQVPro™ High Performance  
Flat Panel/CRT GUI Controllers

Data Sheet  
Revision 1.1

September 1997

P R E L I M I N A R Y

**CHIPS**®

**Copyright Notice**

Copyright© 1996-97 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 Logo is a registered trademark of Chips and Technologies, Inc.

HiQVideo and HiQVPro are trademarks of Chips and Technologies, Inc.

Brooktree and RAMDAC are trademarks of Brooktree Corporation.

IBM is a registered trademarks of International Business Machines Corporation.

Microsoft is a registered trademark of Microsoft Corporation. Windows is a trademark of Microsoft Corporation.

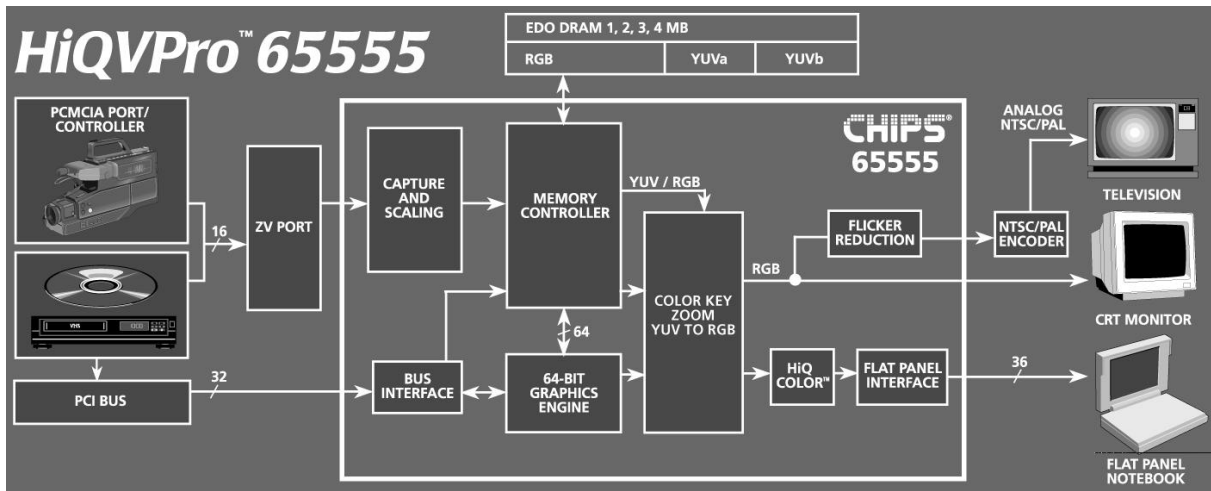
TRI-STATE is a registered trademark of National Semiconductor Corporation.

VESA is a registered trademark of Video Electronics Standards Association.

All other trademarks are the property of their respective holders.

**Disclaimer**

This document provides general information for the customer. Chips and Technologies, Inc., reserves the right to modify the information contained herein as necessary and the customer should ensure that it has 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 many different parties hold patents on products, components, and processes within the personal computer industry. Customers should ensure that their use of the products does not infringe upon any patents. CHIPS respects the patent rights of third parties and shall not participate in direct or indirect patent infringement.



## 65555 (HiQVPro™) High Performance Multimedia Flat Panel/CRT GUI Accelerator

- PC 97 Compliant
- Highly integrated design Flat Panel and CRT GUI Accelerator & Multimedia Engine, Palette/DAC, and Clock Synthesizer
- Hardware Windows Acceleration
  - 64-bit Graphics Engine
    - System-to-Screen and Screen-to-Screen BitBLT
    - 3-Operand Raster-Ops
    - 8/16/24 Color Expansion
  - Transparent BLT
    - Optimized for Windows™ BitBLT format
- PCI Bus with Burst Mode capability and BIOS ROM support
- Flexible Memory Configurations
  - 64-Bit memory interface for EDO
  - Two, four, or eight 256Kx16 DRAMs (1MB, 2MB, 3MB, or 4MB)
  - One or two 512Kx32 DRAMs (2MB or 4MB)
  - Four 256Kx16 plus two 128Kx32 (3MB)
  - Two 128Kx32 DRAMs (1MB)
  - Four 128Kx16 DRAMs (1MB)
- High Performance:
  - Deep write buffers
- CRT Support
  - 110 MHz RAMDAC
- Hardware Multimedia Support
  - Zoom Video port
  - YUV input from System Bus or Video Port
  - YUV-RGB Conversion
  - Capture / Scaling
  - Video Zoom up to 8x
  - Vertical interpolation of video data up to 720 pixels wide.
  - Double Buffered Video
  - Horizontal Interpolation
- Display centering and stretching features for optimal fit of VGA graphics and text on 800x600 and 1024x768 panels
- Simultaneous Hardware Cursor and Pop-up Window
  - 64x64 pixels by 4 colors
  - 128x128 pixels by 2 colors
- Game Acceleration
  - Source Transparent BLT
  - Destination Transparent BLT
  - Double buffer support for YUV and 15/16bpp Overlay Engine
  - Instant Full Screen Page Flip
  - Read back of CRT Scan line counters

- Optimized for High-Performance Flat Panel Display at 3.3V
  - 640x480 x 24bpp
  - 800x600 x 24bpp
  - 1024x768 x 24bpp
  - 1280 x 1024 x 24bpp
- 36-bit direct interface to color and monochrome, single drive (SS), and dual drive (DD), STN & TFT panels
- Flexible On-Chip Activity Timer facilitates ordered shutdown of the display system
- Advanced Power Management feature minimizes power usage in:
  - Normal operation
  - Standby (Sleep) modes
  - Panel-Off Power-Saving Mode
- VESA Standards supported
  - VAFC Port for display of "Live" Video
  - DPMS for CRT power-down (required for support of EPA Energy-Star program)
  - DDC for CRT Plug-Play & Display Control
- Composite NTSC / PAL Support
  - Flicker Reduction Circuitry
- Power Sequencing control outputs regulate application of bias voltage, +5V to the panel and +12V to the inverter for backlight operation
- 3.3V Operation, 5.0V tolerant I/O
- Fully Compatible with IBM<sup>®</sup> VGA

## 65555 (HiQVPro™) Software Support Features

- Drivers Features
  - High Performance Accelerated drivers
  - Compatible across HiQVideo™ family
  - Auto Panning Support
  - LCD/CRT/Simultaneous Mode Support
  - Auto Resolution Change
  - HW Stretching/Scaling
  - Double Buffering
  - Internationalization
  - ChipsCPL (Control Panel Applet)
  - Direct Draw support
  - Games SDK support
  - Dynamic Resolution Switching
  - VGA Graphics applications in Windows
  - VESA DDC extensions
  - VESA DPMS extensions
  - Property Sheet to change Refresh/Display
  - Seamless Windows Support
  - Boot time resolution adjustment
  - DIVE, EnDIVE
  - DCAF
- Multimedia Software
  - Video Port Manager for ZV Port
  - PCVideo DLL plus Tuner with DK Board
- Software Utilities
  - DebugVGA
  - Auto testing of all video modes
  - ChipsVGA
  - ChipsEXT
- Software Documentation
  - BIOS OEM Reference Guide
  - Display Driver User's Guide
  - Utilities User's Guide
  - Release Notes for BIOS, Drivers, and Utilities
- Software Support
  - Dedicated Software Applications Engineer
  - BBS Support for Software Updates
- BIOS Features
  - VGA Compatible BIOS
  - PCI Bus Support
  - PnP Support
  - VESA VBE 2.0 (incl. DPMS)
  - DDC 1, DDC 2AB
  - Text and Graphics Expansion
  - Auto Centering
  - 44 (40) K BIOS
  - CRT, LCD, Simultaneous display modes
  - Auto Resolution Switch
  - Multiple Refresh Rates
  - NTSC/PAL support
  - Extended Modes
  - Extended BIOS Functions
  - 1024x768 TFT, DSTN Color Panels
  - Multiple Panel Support (8 panels built in)
  - Get Panel Type Function
  - HW Popup Interface
  - Monitor Detect
  - Pop Up Support
  - SMI and Hot Key support
- System BIOS Hooks
  - Set Active Display Type
  - Save/Restore Video State
  - Setup Memory for Save/Restore
  - SMI Entry Point
  - Int 15 Calls after POST, Set Mode
- BIOS Modify Program (BMP)
  - Clocks
  - Mode support
  - Panel Tables
  - Voltage Switching
  - Int 15 Hooks
  - Monitor Sensing

**This page intentionally left blank.**

## **REVISION HISTORY**

<b>Revision</b>	<b>Date</b>	<b>By</b>	<b>Comment</b>
0.1	6/13/96	DJ/lc	First Draft- Official Release
0.2	8/27/96	LC	Added "Confidential - NDA Required" markings.
0.3	11/18/96	BB/lc	Added the register information and preliminary AC specifications.
1.0	5/5/97	JN/BB/lc	Updated pin descriptions and register information. Reorganized document chapters and appendices.
1.1	9/16/97	BB/bjb	Updated feature section. Enhanced Electrical Specifications. Enhanced Register Summary Section. Updated DRAM Control Registers Corrected BR04 Register. Fixed FROA description. Added Subsystem Vendor ID Register. Removed 32-bit configurations in Appendix C.

**This page intentionally left blank.**



# **TABLE OF CONTENTS**

## **CHAPTER 1 — Introduction / Overview**

1.1 HiQColor™ Technology.....	1-1
1.2 Flicker Free Output Television.....	1-1
1.3 ZV Port Input.....	1-1
1.4 Hardware Multimedia Support.....	1-1
1.5 Video Acceleration.....	1-1
1.6 Versatile Panel Support.....	1-2
1.7 Low Power Consumption.....	1-2
1.8 Software Compatibility / Flexibility.....	1-2
1.9 Display Memory Size Requirements.....	1-3

## **CHAPTER 2 — Pin Descriptions**

2.0 Introduction.....	2-1
2.1 Top View: BGA Ball Assignments.....	2-2
2.2 Bottom View: BGA Ball Assignments.....	2-3
2.3 PCI Bus Interface.....	2-5
2.4 Display Memory Interface.....	2-8
2.5 Flat Panel Display Interface.....	2-11
2.6 CRT Interface.....	2-14
2.7 Power / Ground and Standby Control.....	2-15
2.8 Video Interface.....	2-16
2.9 Boundary Scan.....	2-17
2.10 Miscellaneous.....	2-18

**CHAPTER 3 — Electrical Specifications**

Absolute Maximum Conditions.....	3-1
Normal Operating Conditions.....	3-1
DAC Characteristics .....	3-1
DC Characteristics.....	3-2
DC Drive Characteristics .....	3-2
AC Test Conditions.....	3-3
AC Timing Characteristics	
Reference Clock.....	3-3
Clock Generator .....	3-4
Reset Timing.....	3-4
PCI Bus Frame (33 MHz).....	3-5
PCI Bus Stop Timing.....	3-6
PCI BIOS ROM Timing.....	3-6
EDO DRAM Read / Write.....	3-7
DRAM Page Mode Read Cycle Timing.....	3-8
DRAM Page Mode Write Cycle Timing.....	3-8
CAS-Before-RAS (CBR) DRAM Refresh Cycle Timing.....	3-9
“Self Refresh DRAM” Refresh Cycle Timing.....	3-9
Video Input Port Timing.....	3-10
Panel Output Timing.....	3-11

**CHAPTER 4 — Mechanical Specifications**

256 + 16 BGA Specifications.....	4-1
----------------------------------	-----

**CHAPTER 5 — I/O and Memory Address Maps**

5.0 Introduction.....	5-1
5.1 I/O Map and Sub-Addressed Register Map.....	5-3
5.2 Sub-Addressing Indexes and Data Ports.....	5-4
5.3 Lower Memory Map.....	5-5
5.4 Upper Memory Map.....	5-5
5.5 PCI Configuration Registers.....	5-6

## CHAPTER 6 — Register Summaries

6.0 Register Summaries.....	6-1
6.1 General Control and Status .....	6-1
6.2 Sequencer.....	6-1
6.3 Graphics Controller .....	6-1
6.4 Attribute Controller .....	6-2
6.5 VGA Color Palette .....	6-2
6.6 CRT Controller .....	6-3
6.7 PCI Configuration .....	6-4
6.8 BitBLT .....	6-5
6.9 Extension .....	6-6
6.10 Multimedia.....	6-8
6.11 Flat Panel .....	6-10

## CHAPTER 7 — PCI Configuration Registers

7.0 Introduction .....	7-1
VENDID Vendor ID.....	7-2
DEVID Device ID.....	7-2
DEVCTL Device Control.....	7-3
DEVSTAT Device Status.....	7-5
REV Revision.....	7-7
PRG Register-Level Programming Interface.....	7-7
SUB Sub-Class Code.....	7-8
BASE Base Class Code.....	7-8
HDR Header Type.....	7-9
MBASE Memory Base Address.....	7-10
RBASE ROM Base Address.....	7-11

## CHAPTER 8 — General Control and Status Registers

8.0 Introduction .....	8-1
ST00 Input Status Register 0.....	8-2
ST01 Input Status Register 1.....	8-3
FCR Feature Control.....	8-4
MSR Miscellaneous Output.....	8-5

**CHAPTER 9 — CRT Controller Registers**

9.0 Introduction..... 9-1

CRX CRT Controller Index ..... 9-2

CR00 Horizontal Total ..... 9-2

CR01 Horizontal Display Enable End ..... 9-3

CR02 Horizontal Blanking Start ..... 9-3

CR03 Horizontal Blanking End ..... 9-4

CR04 Horizontal Sync Start ..... 9-5

CR05 Horizontal Sync End ..... 9-6

CR06 Vertical Total ..... 9-7

CR07 Overflow ..... 9-8

CR08 Preset Row Scan..... 9-12

CR09 Maximum Scan Line ..... 9-13

CR0A Text Cursor Start ..... 9-15

CR0B Text Cursor End ..... 9-16

CR0C Start Address High ..... 9-17

CR0D Start Address Low ..... 9-18

CR0E Text Cursor Location High ..... 9-19

CR0F Text Cursor Location Low ..... 9-19

CR10 Vertical Sync Start ..... 9-20

CR11 Vertical Sync End ..... 9-21

CR12 Vertical Display Enable End ..... 9-22

CR13 Offset ..... 9-22

CR14 Underline Location ..... 9-23

CR15 Vertical Blanking Start ..... 9-24

CR16 Vertical Blanking End ..... 9-24

CR17 CRT Mode Control..... 9-25

CR18 Line Compare ..... 9-28

CR22 Memory Read Latch Data..... 9-29

CR30 Extended Vertical Total ..... 9-29

CR31 Extended Vertical Display End ..... 9-30

CR32 Extended Vertical Sync Start..... 9-31

CR33 Extended Vertical Blanking Start ..... 9-32

CR40 Extended Start Address ..... 9-33

CR41 Extended Offset..... 9-34

CR70 Interlace Control ..... 9-34

CR71 NTSC/PAL Video Output Control..... 9-35

CR72 Horizontal Serration 1 Start ..... 9-36

CR73 Horizontal Serration 2 Start ..... 9-36

CR74 NTSC/PAL Horizontal Pulse Width ..... 9-37

CR75 NTSC/PAL Filtering Burst Read Length..... 9-38

CR76 NTSC/PAL Filtering Burst Read Quantity..... 9-38

CR77 NTSC/PAL Filtering Control ..... 9-39

## CHAPTER 10 — Sequencer Registers

10.0 Introduction.....	10-1
SRX Sequencer Index .....	10-2
SR00 Reset.....	10-3
SR01 Clocking Mode .....	10-4
SR02 Plane Mask .....	10-5
SR03 Character Map Select.....	10-6
SR04 Memory Mode.....	10-7
SR07 Horizontal Character Counter Reset.....	10-8

## CHAPTER 11 — Graphics Controller Registers

11.0 Introduction.....	11-1
GRX Graphics Controller Index .....	11-2
GR00 Set/Reset .....	11-2
GR01 Enable Set/Reset .....	11-3
GR02 Color Compare.....	11-3
GR03 Data Rotate .....	11-4
GR04 Read Plane Select.....	11-5
GR05 Graphics Mode.....	11-6
GR06 Miscellaneous .....	11-9
GR07 Color Don't Care .....	11-10
GR08 Bit Mask .....	11-10

## CHAPTER 12 — Attribute Controller Registers

12.0 Introduction.....	12-1
ARX Attribute Controller Index .....	12-2
AR00-AR0F Palettes Registers 0-F.....	12-3
AR10 Mode Control.....	12-4
AR11 Overscan Color .....	12-6
AR12 Memory Plane Enable .....	12-7
AR13 Horizontal Pixel Panning .....	12-8
AR14 Color Select .....	12-9

## CHAPTER 13 — Palette DAC Registers

13.0 Introduction.....	13-1
DACMASK Pixel Data Mask .....	13-3
DACSTATE DAC State .....	13-3
DACRX Palette Read Index .....	13-4
DACWX Palette Write Index .....	13-4
DACDATA Palette Data .....	13-5

**CHAPTER 14 — Extension Registers**

14.0 Introduction..... 14-1

XR0 Extension Registers Index..... 14-3

XR00 Vendor ID Low..... 14-4

XR01 Vendor ID High..... 14-4

XR02 Device ID Low..... 14-5

XR03 Device ID High..... 14-5

XR04 Revision ID..... 14-6

XR05 Linear Base Address Low..... 14-7

XR06 Linear Base Address High..... 14-7

XR08 Host Bus Configuration..... 14-8

XR09 Access Control..... 14-9

XR0A Frame Buffer Mapping..... 14-10

XR0B PCI Burst Write Support..... 14-12

XR0E Frame Buffer Page Selector..... 14-13

XR10 Subsystem Vendor ID Low..... 14-14

XR11 Subsystem Vendor ID High..... 14-14

XR12 Subsystem Device ID Low..... 14-15

XR13 Subsystem Device ID High..... 14-15

XR20 BitBLT Configuration..... 14-16

XR40 DRAM Access Control..... 14-17

XR41 DRAM Type..... 14-18

XR42 DRAM Configuration..... 14-19

XR43 DRAM Interface..... 14-20

XR44 DRAM Timing..... 14-21

XR60 Video Pin Control..... 14-22

XR61 DPMS Sync Control..... 14-23

XR62 GPIO Pin Control..... 14-24

XR63 GPIO Pin Data..... 14-25

XR67 Pin Tri-State Control..... 14-26

XR70 Configuration Pin 0..... 14-27

XR71 Configuration Pin 1..... 14-29

XR80 Pixel Pipeline Configuration 0..... 14-30

XR81 Pixel Pipeline Configuration 1..... 14-32

XR82 Pixel Pipeline Configuration 2..... 14-33

**CHAPTER 14— Extension Registers (continued)**

XRA0	Cursor 1 Control .....	14-34
XRA1	Cursor 1 Vertical Extension .....	14-36
XRA2	Cursor 1 Base Address Low.....	14-37
XRA3	Cursor 1 Base Address High.....	14-37
XRA4	Cursor 1 X-Position Low.....	14-38
XRA5	Cursor 1 X-Position High.....	14-38
XRA6	Cursor 1 Y-Position Low.....	14-39
XRA7	Cursor 1 Y-Position High.....	14-39
XRA8	Cursor 2 Control .....	14-40
XRA9	Cursor 2 Vertical Extension .....	14-42
XRAA	Cursor 2 Base Address Low.....	14-43
XRAB	Cursor 2 Base Address High.....	14-44
XRAC	Cursor 2 X-Position Low .....	14-44
XRAD	Cursor 2 X-Position High .....	14-45
XRAE	Cursor 2 Y-Position Low .....	14-46
XRAF	Cursor 2 Y-Position High.....	14-46
XRC0	Dot Clock 0 VCO M-Divisor Low.....	14-47
XRC1	Dot Clock 0 VCO N-Divisor Low.....	14-48
XRC2	Dot Clock 0 VCO M/N-Divisor High.....	14-49
XRC3	Dot Clock 0 Divisor Select.....	14-50
XRC4	Dot Clock 1 VCO M-Divisor Low.....	14-52
XRC5	Dot Clock 1 VCO N-Divisor Low.....	14-53
XRC6	Dot Clock 1 VCO M/N-Divisor High.....	14-54
XRC7	Dot Clock 1 Divisor Select.....	14-55
XRC8	Dot Clock 2 VCO M-Divisor Low.....	14-57
XRC9	Dot Clock 2 VCO N-Divisor Low.....	14-58
XRCA	Dot Clock 2 VCO M/N-Divisor High.....	14-59
XRCB	Dot Clock 2 Divisor Select.....	14-60
XRCC	Memory Clock VCO M-Divisor.....	14-62
XRCD	Memory Clock VCO N-Divisor.....	14-63
XRCE	Memory Clock Divisor Select.....	14-64
XRCF	Memory Clock Divisor Select.....	14-66
XRD0	Powerdown Control.....	14-67
XRD1	Power Conservation Control.....	14-68
XRD2	2KHz Down Counter.....	14-68
XRE0–XREB	Software Flags.....	14-69
XRF8-FC	Test Registers.....	14-69

## CHAPTER 15 — Flat Panel Registers

15.0 Introduction .....	15-1
FR00 Feature .....	15-1
FR01 CRT / FP Control.....	15-1
FR02 FP Mode Control.....	15-2
FR03 FP Dot Clock Source.....	15-2
FR04 Panel Power Sequencing Delay.....	15-3
FR05 PowerDown Control 1.....	15-4
FR06 FP PowerDown Control.....	15-5
FR08 FP Pin Polarity.....	15-6
FR0A Programmable Output Drive.....	15-7
FR0B FP Pin Control 1.....	15-8
FR0C Pin Control 2.....	15-9
FR0F Activity Timer Control.....	15-10
FR10 FP Format 0.....	15-11
FR11 FP Format 1.....	15-13
FR12 FP Format 2.....	15-15
FR13 FP Format 3.....	15-17
FR16 FRC Option Select.....	15-18
FR17 Polynomial FRC Control.....	15-19
FR18 FP Text Mode Control.....	15-19
FR19 Blink Rate Control.....	15-20
FR1A Buffering Control.....	15-21
FR1E M (ACDCLK) Control.....	15-22
FR1F Diagnostic.....	15-23
FR20 FP Horizontal Panel Display Size LSB.....	15-24
FR21 FP Horizontal Sync Start LSB.....	15-24
FR22 FP Horizontal Sync End.....	15-25
FR23 FP Horizontal Total LSB.....	15-25
FR24 FP HSync (LP) Delay LSB.....	15-26
FR25 FP Horizontal Overflow 1.....	15-26
FR26 FP Horizontal Overflow 2.....	15-27
FR27 FP HSync (LP) Width and Disable.....	15-27
FR30 FP Vertical Panel Size LSB.....	15-28
FR31 FP Vertical Sync Start LSB.....	15-28
FR32 FP Vertical Sync End.....	15-29
FR33 FP Vertical Total LSB.....	15-29
FR34 FP VSync (FLM) Delay LSB.....	15-30
FR35 FP Vertical Overflow 1.....	15-30
FR36 FP Vertical Overflow 2.....	15-31
FR37 FP VSync (FLM) Disable.....	15-32
FR40 Horizontal Compensation .....	15-33
FR41 Horizontal Stretching .....	15-35
FR48 Vertical Compensation .....	15-36
FR49-4C Text Mode Vertical Stretching 0 MSB.....	15-37
FR4D Vertical Line Replication .....	15-38
FR4E Selective Vertical Stretching Disable.....	15-39
FR70 TMED Red Seed.....	15-40
FR71 TMED Green Seed.....	15-40
FR72 TMED Blue Seed.....	15-41
FR73 TMED Control.....	15-41



## CHAPTER 16 — Multimedia Registers

16.0 Introduction.....	16-1
MR00 Module Capability.....	16-1
MR01 Secondary Capability.....	16-1
16.1 Acquisition Control.....	16-2
MR02 Capture Control 1.....	16-2
MR03 Capture Control 2.....	16-3
MR04 Capture Control 3.....	16-4
MR05 Capture Control 4.....	16-5
MR06-08 Capture Memory Address PTR1.....	16-6
MR09-0B Capture Memory Address PTR2.....	16-7
MR0C Capture Memory Line Storage Width.....	16-8
MR0E Capture Window X-LEFT.....	16-8
MR0F Capture Window X-LEFT.....	16-8
MR10 Capture Window X-RIGHT.....	16-9
MR11 Capture Window X-RIGHT.....	16-9
MR12 Capture Window Y-TOP.....	16-9
MR13 Capture Window Y-TOP.....	16-10
MR14 Capture Window Y-BOTTOM.....	16-10
MR15 Capture Window Y-BOTTOM.....	16-10
MR16 H-SCALE.....	16-11
MR17 V-SCALE.....	16-11
MR18 Capture Frame/Field Count.....	16-11
16.2 Playback Control.....	16-12
MR1E Playback Control 1.....	16-12
MR1F Playback Control 2.....	16-13
MR20 Playback Control 3.....	16-14
MR21 Double Buffer Status.....	16-15
MR22-24 Playback Memory Address PTR1.....	16-16
MR25-27 Playback Memory Address PTR2.....	16-17
MR28 Playback Line Memory Fetch Width.....	16-18
MR2A Playback Window X-LEFT.....	16-18
MR2B Playback Window X-LEFT.....	16-18
MR2C Playback Window X-RIGHT.....	16-19
MR2D Playback Window X-RIGHT.....	16-19
MR2E Playback Window Y-TOP.....	16-19
MR2F Playback Window Y-TOP.....	16-20
MR30 Playback Window Y-BOTTOM.....	16-20
MR31 Playback Window Y-BOTTOM.....	16-20
MR32 H-ZOOM.....	16-21
MR33 V-ZOOM.....	16-22
MR34 Playback Line Display Width.....	16-23
MR3C Color Key Control.....	16-24
MR3D-3F Color Keys.....	16-25
MR40-42 Color Key Masks.....	16-25
MR43 Line Count.....	16-26
MR44 Line Count.....	16-26

**CHAPTER 17 — BitBLT Registers**

17.0 Introduction ..... 17-1  
 BR00 Source and Destination Offset ..... 17-2  
 BR01 Pattern/Source Expansion Background Color ..... 17-3  
 BR02 Pattern/Source Expansion Foreground Color ..... 17-4  
 BR03 Monochrome Source Control ..... 17-5  
 BR04 BitBLT Control ..... 17-7  
 BR05 Pattern Address ..... 17-11  
 BR06 Source Address ..... 17-12  
 BR07 Destination Address ..... 17-13  
 BR08 Destination Width & Height ..... 17-14  
 BR09 Source Expansion Background Color ..... 17-15  
 BR0A Source Expansion Foreground Color ..... 17-16

**APPENDIX A — VGA Standard Modes**

A VGA Standard Mode Table ..... A-1

**APPENDIX B — Clock Generation**

B.1 Clock Synthesizer ..... B-1  
 B.2 Dot Clock (DCLK) ..... B-1  
 B.3 Memory Clock (MCLK) ..... B-1  
 B.4 PLL Parameters ..... B-2  
 B.5 Programming the Clock Synthesizer ..... B-3  
 B.6 DCLK Programming ..... B-3  
 B.7 MCLK Programming ..... B-3  
 B.8 Programming Constraints ..... B-3  
 B.9 Programming Example ..... B-4  
 B.10 Display Memory Bandwidth ..... B-4  
 B.11 STN-DD Panel Buffering ..... B-5  
 B.12 Horizontal and Vertical Clocking ..... B-6

**APPENDIX C Memory Configurations**

C Memory Configurations ..... C-1  
 C.1 General Configuration Details ..... C-1  
 C.2 Memory Configurations ..... C-2

**APPENDIX D — Panel Power Sequencing**

D Panel Power Sequencing ..... D-1

## APPENDIX E — Hardware Cursor and Pop Up Window

E.0 Introduction.....	E-1
E.1 Cursor Configuration.....	E-2
E.1.1 Basic Cursor Configuration.....	E-2
E.1.2 Base Address for Cursor Data .....	E-2
E.1.3 Cursor Vertical Extension.....	E-3
E.1.4 Cursor Colors.....	E-3
E.2 Cursor Modes.....	E-3
E.2.1 32x32x2bpp & 64x64x2bpp ANE/XOR Pixel Plane Modes.....	E-4
E.2.2 64x64x2bpp 4-Color Mode.....	E-5
E.2.3 64x64x2bpp 3-Color and Transparency Mode.....	E-6
E.2.4 128x128x1bpp 2-Color Mode .....	E-7
E.2.5 128x128x1bpp 1-Color and Transparency Mode.....	E-8
E.3 Cursor Positioning.....	E-9

## APPENDIX F BitBLT Engine

F.1 Introduction.....	F-1
F.2 Basic BitBLT Functional Considerations.....	F-2
F.2.1 Color Depth Configuration and Color Expansion.....	F-2
F.2.2 Graphics Data Size Limitation .....	F-3
F.2.3 Bit-Wise Operations.....	F-3
F.2.4 Per-Pixel Write-Masking Operations.....	F-7
F.2.5 When the Source and Destination Locations Overlap.....	F-8
F.3.0 Basic Graphics Data Considerations.....	F-12
F.3.1 Contiguous vs. Discontiguous Graphics Data.....	F-12
F.3.2 Source Data.....	F-14
F.3.3 Monochrome Source Data.....	F-15
F.3.4 Pattern Data .....	F-16
F.3.5 Destination Data .....	F-19
F.4.0 BitBLT Programming Examples.....	F-20
F.4.1 Pattern Fill -- A Very Simple BitBLT.....	F-20
F.4.2 Drawing Characters Using a Font Stored in System Memory.....	F-23

**This page intentionally left blank.**

**LIST OF FIGURES**

Figure 1-1: Display Memory Configurations..... 1-3

Figure 1-2: Display Memory Configurations with an STN-DD Buffer..... 1-3

Figure 2-1: Pin Diagram ..... 2-2

Figure 2-2: Pin Diagram ..... 2-3

Figure 3-1: AC Test Timing..... 3-3

Figure 3-2: Reference Clock Timing..... 3-3

Figure 3-3: Reset Timing..... 3-4

Figure 3-4: PCI Bus Frame Timing..... 3-5

Figure 3-5: PCI Bus Stop Timing..... 3-6

Figure 3-6: PCI BIOS ROM Timing..... 3-6

Figure 3-7: DRAM Page Mode Read Cycle Timing..... 3-8

Figure 3-8: DRAM Page Mode Write Cycle Timing..... 3-8

Figure 3-9: CAS-BeforF-RAS (CBR) DRAM Refresh Cycle Timing..... 3-9

Figure 3-10: “Self Refresh DRAM” Refresh Cycle Timing..... 3-9

Figure 3-11: Video Data Port Timing..... 3-10

Figure 3-12: Panel Output Timing..... 3-11

Figure 4-1: 65555 256+16 Contact Ball Grid Array..... 4-1

Figure B-1: PLL Elements..... B-2

Figure C-1: 1MB, 64-bit Memory Configuration Using Two 128Kx32 EDO DRAMs..... C-2

Figure C-2: 2MB, 64-bit Memory Configuration Using Four 128Kx32 EDO DRAMs..... C-3

Figure C-3: 3MB, 64-bit Memory Configuration Using Four 256Kx16 and Two 128Kx32 EDO DRAMs..... C-4

Figure C-4: 2MB, 64-bit Memory Configuration Using Four 256Kx16 EDO DRAMs..... C-5

Figure C-5: 4MB, 64-bit Memory Configuration Using Eight 256Kx16 EDO DRAMs..... C-6

Figure C-6: 4MB, 64-bit Memory Configuration Using Two 512Kx32 EDO DRAMs..... C-7

Figure F-1: Block Diagram and Data Paths of the BitBLT Engine..... F-1

Figure F-2: Block Diagram and Data Paths of the BitBLT Engine..... F-7

Figure F-3: Source Corruption in BitBLT with Overlapping Source and Destination Locations..... F-8

Figure F-4: Correctly Performed BitBLT with Overlapping Source and Destination Locations..... F-10

Figure F-5: Suggested Starting Points for Possible Source and Destination Overlap Situations..... F-11

Figure F-6: On-Screen Single 6-Pixel Line in the Frame Buffer..... F-12

Figure F-7: On-Screen 6x4 Array of Pixels in the Frame Buffer..... F-13

Figure F-8: Pattern Data..... F-16

Figure F-9: Monochrome Pattern Data -- Occupies a Single Quadword..... F-17

Figure F-10: 8bpp Pattern Data -- Occupies 64 Bytes (8 Quadwords)..... F-17

Figure F-11: 16bpp Pattern Data -- Occupies 128 Bytes (16 Quadwords)..... F-17

Figure F-12: 24bpp Pattern Data -- Occupies 256 Bytes (32 Quadwords)..... F-18

Figure F-13: On-Screen Destination for Example Pattern Fill BitBLT..... F-20

Figure F-14: Pattern Data for Example Pattern Fill BitBLT..... F-21

Figure F-15: Results of Example Pattern Fill BitBLT..... F-22

Figure F-16: On-Screen Destination for Example Character Drawing BitBLT..... F-23

Figure F-17: Source Data in System Memory for Example Character Drawing BitBLT..... F-23

Figure F-18: Results of Example Character Drawing BitBLT..... F-25

**List of Tables**

Table 3-1: Absolute Maximum Conditions..... 3-1

Table 3-2: Normal Operating Conditions..... 3-1

Table 3-3: DAC Characteristics ..... 3-1

Table 3-4: DC Characteristics..... 3-2

Table 3-5: DC Drive Characteristics ..... 3-2

Table 3-6: AC Test Conditions..... 3-3

Table 3-7: AC Timing Characteristics - Reference Clock..... 3-3

Table 3-8: AC Timing Characteristics - Clock Generator ..... 3-4

Table 3-9: AC Timing Characteristics - Reset..... 3-4

Table 3-10: AC Timing Characteristics - PCI Bus Frame (33 MHz)..... 3-5

Table 3-11: AC Timing Characteristics - PCI Bus Stop (33 MHz)..... 3-6

Table 3-12: AC Timing Characteristics - PC BIOS ROM..... 3-6

Table 3-13: AC Timing Characteristics - EDO DRAM Read / Write..... 3-7

Table 3-14: AC Timing Characteristics - CBR Refresh..... 3-9

Table 3-15: AC Timing Characteristics - Self Refresh..... 3-9

Table 3-16: AC Timing Characteristics - Video Input Port..... 3-10

Table 3-17: AC Timing Characteristics - Panel Output..... 3-11

Table A-1 VGA Standard Modes..... A-1

Table C-1: Memory Configurations..... C-2

Table D-1: Memory Organization 32x32x2bpp AND/XOR Pixel Plane Mode..... D-4

Table D-2: Memory Organization 64x64x2bpp AND/XOR Pixel Plane Mode..... D-4

Table D-3: Pixel Data 32x32x2bpp and 64x64x2bpp AND/XOR Pixel Plane Modes..... D-4

Table D-4: Memory Organization 64x64x2bpp 4-Color Mode..... D-5

Table D-5: Pixel Data 64x64x2bpp 4-Color Mode..... D-5

Table D-6: Memory Organization 64x64x2bpp 3-Color & Transparency Mode ..... D-6

Table D-7: Pixel Data 64x64x2bpp 3-Color & Transparency Mode..... D-6

Table D-8: Memory Organization 128x128x1bpp 2-Color Mode..... D-7

Table D-9: Pixel Data 128x128x1bpp 2-Color Mode..... D-7

Table D-10: Memory Organization 128x128x1bpp 1-Color & Transparency Mode ..... D-8

Table D-11: Pixel Bit Definitions 128x128x1bpp 1-Color & Transparency Mode ..... D-8

Table E-1: Bit-Wise Operations and 8-bit Codes (00 - 5F)..... E-4

Table E-2: Bit-Wise Operations and 8-bit Codes (60 - BF)..... E-4

Table E-3: Bit-Wise Operations and 8-bit Codes (C0 - FF)..... E-4

Table E-4: Mem Org 64x64x2bpp 4-Color Mode..... E-5

Table E-5: Pixel Data 64x64x2bpp 4-Color Mode..... E-5

Table E-6: Mem Org 64x64x2bpp 3-Color & Transparency Mode..... E-6

Table E-7: Pixel Data 64x64x2bpp 3-Color Transparency Mode..... E-6

Table E-8: Mem Org 128x128x1bpp 2-Color Mode..... E-7

Table E-9: Pixel Data 128x128x1bpp 2-Color Mode..... E-7

Table E-10: Mem Org 128x128x1bpp 1-Color & Transparency Mode..... E-8

Table E-11: Pixel Data 128x128x1bpp 4-Color & Transparency Mode..... E-8

Table F-1: Memory Interface Differences..... F-1

Table F-2: Memory Configurations..... F-2

# ***CHAPTER 1***

## ***INTRODUCTION/ OVERVIEW***

**This page intentionally left blank.**



# **1. Introduction / Overview**

The HiQVideo™ family of high performance multimedia flat panel/CRT GUI accelerators extend CHIPS' offering of high performance flat panel controllers for full-featured notebooks and sub-notebooks. The HiQVideo family offers 64-bit high performance and new hardware multimedia support features.

## **1.1 HiQColor™ Technology**

The 65555 integrates CHIPS breakthrough HiQColor™ technology. Based on a new proprietary TMED™ (Temporal Modulated Energy Distribution) algorithm, HiQColor technology is a unique process that enables the display of 16.7M colors on STN panels without dithering. TMED reduces the need for panel turning associated with current FRC-based algorithms.

Independent of panel response times, the TMED algorithm eliminates all flaws such as shimmer, Mach banding and crawling currently seen on STN panels. Combined with the new fast response high contrast and low-crosstalk technology found in new STN panels. HiQColor technology enables TFT quality viewing on an STN panel. The 65555 provides the best color fidelity for the widest variety of active and passive panels in the market.

## **1.2 Reduced Flicker Output Television**

The television output circuitry supports both NTSC and PAL television formats. The 65555 provides filtering circuitry to reduce the flicker circuitry to reduce the flicker seen when displaying CRT resolution images on television screens. The television circuitry scales images to fit both PAL and NTSC televisions.

## **1.3 ZV Port Input**

The 65555 supports the ZV port PCMCIA standard for video input. The ZV port video data is fed directly to the graphics memory to reduce traffic on the PCI Bus.

## **1.4 Hardware Multimedia Support**

The HiQVideo™ family uses independent multimedia capture and display systems on-chip. The capture system places data in display memory (usually off screen) and the display system places the data in a window on the screen.

The capture system can receive data from either the system bus or from the ZV enabled video port in either RGB or YUV format. The input data can also be scaled down before storage in display memory. Capture of input data may also be double buffered for smoothing and to prevent image tearing.

The display system can independently place either RGB or YUV data from anywhere in display memory into an on-screen window which can be any size and located at any pixel boundary (YUV data is converted to RGB "on-the-fly" on output). Non-rectangular windows are supported via color keying. The data can be fractionally zoomed on output up to 8x to fit the onscreen window and can be horizontally and vertically interpolated. Interlaced and non-interlaced data are supported in both capture and display systems.

## **1.5 Video Acceleration**

When the system writes to the video YUV memory, the 65555 uses its PCI Bust Mode capabilities to allow for a higher frame rate. Video capture input through the ZV port is scaled and stored into memory allowing frame capture for video conferencing. In addition, the 65555 will use vertical interpolation of video data up to 720 pixels wide to enable smooth zooming to full screen MPEG II video. Double buffering is used to prevent image tearing.

### **1.6 Versatile Panel Support**

The HiQVideo™ family supports a wide variety of monochrome and color Single-Panel, Single-Drive (SS) and Dual-Panel, Dual Drive (DD) standard and high-resolution passive STN and active matrix TFT/MIM LCD, and EL panels. For monochrome panels, up to 64 gray scales are supported. With the help of HiQColor™ Technology, STN panels can afford 256 gray shades per primary resulting in 16M colors for an improved image representation. Additionally, the HiQVPro™ also supports TFT panels up to 36-bit interface. The HiQVideo family offers a variety of programmable features to optimize display quality. Vertical centering and stretching are provided for handling modes with less than 480 lines on 480-line panels. Horizontal and vertical stretching capabilities are also available for both text and graphics modes for optimal display of VGA text and graphics modes on 800x600 and 1024x768 panels. Three selectable color-to-gray scale reduction techniques and SMARTMAP™ are available for improving the ability to view color applications on monochrome panels.

### **1.7 Low Power Consumption**

The HiQVideo™ family uses a variety of advanced power management features to reduce power consumption of the display sub-system and to extend battery life. Although optimized for 3.3V operation, the HiQVideo™ controller's internal logic, memory interface, bus interface, and panel interfaces can be independently configured to operate at either 3.3V or 5V.

### **1.8 Software Compatibility/Flexibility**

The HiQVideo controllers are fully compatible with VGA at the register, and BIOS levels. CHIPS and third-party vendors supply fully VGA-compatible BIOS, end-user utilities and drivers for common application programs such as Microsoft Windows and OS/2.

The 65555 BIOS and drivers are an evolutionary step from the 65554 software. The Windows drivers provided for the 65555 are compliant with both Microsoft WHQL and PC97 standards.

**1.9 Display Memory Size Requirements**

The 65555 supports the following 32-bit wide and 64-bit wide memory configurations show below:

32-bit Memory Bus		64-bit Memory Bus			
1MB	256Kx16   256Kx16	1MB	128Kx32		128Kx32
1MB	128Kx32	2MB	256Kx16   256Kx16	256Kx16	256Kx16
	128Kx32	2MB	128Kx32		128Kx32
1MB	256Kx32	2MB	128Kx32		128Kx32
2MB	256Kx16   256Kx16	2MB	256Kx32		256Kx32
	256Kx16   256Kx16	3MB	256x16   256x16	256x16	256x16
2MB	256Kx32		128Kx32		128Kx32
	256Kx32	3MB	256Kx32		256Kx32
2MB	512Kx32		128Kx32		128Kx32
		4MB	256Kx16   256Kx16	256Kx16	256Kx16
			256Kx16   256Kx16	256Kx16	256Kx16
		4MB	256Kx32		256Kx32
			256Kx32		256Kx32
		4MB	512Kx32		512Kx32

**Figure 1-1: Display Memory Configurations**

**Note:** The 64-bit wide memory configurations have double the memory bandwidth of the 32-bit wide configurations.

The figure below shows the display memory configurations using an external STN-DD buffer:

	32 Bit Bus	16 Bit Bus
1.5 MB	256Kx16   256Kx16	256K x16
1.5 MB	256Kx32	256K x16
2.5 MB	512Kx32	256K x16

**Figure 1-2: Display Memory Configurations with an STN-DD Buffer**

**Notes:**

- Some of the 32-bit configurations allow an additional 256K x 16 device to be used for an external 16-bit wide STN-DD buffer, as shown above.
- The 65555 supports both video capture/playback and an external STN-DD buffer at the same time.

**This page intentionally left blank.**

## ***CHAPTER 2***

### ***PIN DESCRIPTIONS***

**This page intentionally left blank.**

## ***PIN DESCRIPTIONS***

### ***2.0 INTRODUCTION***

The following pages contain the BGA ball assignments and a list of all the pins for the 65555 GUI Accelerator.

The pins are divided into the following groups:

- PCI Bus
- Display Memory Interface
- Flat Panel Display Interface
- CRT Interface
- Power / Ground and Standby Control
- Video Interface; Miscellaneous.

Pin names in parentheses (...) indicate alternate functions.





**2.2 Bottom View: BGA Ball Assignments**

	Y	W	V	U	T	R	P	N	M	L	K	J	H	G	F	E	D	C	B	A																										
20	RSVD	VP10	VP6	VP1	WED	MD0	MD3	MD7	MD9	MD12	MD15	WEC	CASCL	MC3	MC6	MC9	MC11	RSVD	AA2	AA4	20																									
19	VP14	VP11	VP9	VP5	VP2	CASDH	MD1	MD4	MD8	MD11	N/C	CASCH	MC2	MC5	MC8	MC12	MC15	AA1	AA5	AA6	19																									
18	VCLK	VP15	VP13	VP8	VP4	VP0	CASDL	MD2	MD6	MD10	RAS1	MC0	MC4	MC7	MC10	MC14	AA0	AA3	AA7	WEB	18																									
17	P33	HREF	PCLK	VP12	VP7	VP3	GND	MVCC	MD5	MD13	MD14	MC1	MVCC	GND	MC13	RSVD	RSVD	AA8	CASBH	MB2	17																									
16	P31	P34	VREF	RSVD	<table border="1"> <tr> <td>12</td> <td>GND</td> <td>GND</td> <td>GND</td> <td>GND</td> </tr> <tr> <td>11</td> <td>GND</td> <td>GND</td> <td>GND</td> <td>GND</td> </tr> <tr> <td>10</td> <td>GND</td> <td>GND</td> <td>GND</td> <td>GND</td> </tr> <tr> <td>9</td> <td>GND</td> <td>GND</td> <td>GND</td> <td>GND</td> </tr> <tr> <td></td> <td>M</td> <td>L</td> <td>K</td> <td>J</td> </tr> </table>												12	GND	GND	GND	GND	11	GND	GND	GND	GND	10	GND	GND	GND	GND	9	GND	GND	GND	GND		M	L	K	J	AA9	CASBL	MB1	MB4	16
12	GND	GND	GND	GND																																										
11	GND	GND	GND	GND																																										
10	GND	GND	GND	GND																																										
9	GND	GND	GND	GND																																										
	M	L	K	J																																										
15	P28	P30	P32	P35													MB0	MB3	MB5	MB7	15																									
14	P25	P27	P29	GND													GND	MB6	MB8	MB10	14																									
13	P21	P24	P26	WVCC													MVCC	MB9	MB11	MB14	13																									
12	P20	IVCC	P22	P23													MB12	MB13	MB15	N/C	12																									
11	P17	P18	P19	P16	CASAL	RAS0	WEA	CASAH	11																																					
10	P14	P13	P12	P15	MA0	MA3	MA2	MA1	10																																					
9	P11	P10	P8	P7	IVCC	MA7	MA5	MA4	9																																					
8	P9	P6	P4	DVCC	RSVD	MA10	MA8	MA6	8																																					
7	P5	P3	P1	GND	GND	MA13	MA11	MA9	7																																					
6	P2	P0	M	ENABKL	N/C	ROMOE	MA14	MA12	6																																					
5	SHFCLK	FLM	ENAVDD	AVCC	PVCC	N/C	N/C	MA15	5																																					
4	LP	ENAVEE	GREEN	DDC CLK (GPIO3)	GPIO1 (32KHz)	AD2	GND	BVCC	AD13	DEVSEL	TRDY	AD20	BVCC	GND	AD30	STNDBY	RSVD	PVCC	PGND	RSVD	4																									
3	RED	BLUE	DDC DATA (GPIO2)	HSYNC	AD1	AD5	C/BE0	AD10	AD14	SERR	C/BE2	AD19	AD23	AD24	AD27	AD31	TRST	REFCLK	SVCC	PGND	3																									
2	AGND	RSET	VSYNC	AD0	AD3	AD7	AD9	AD12	C/BE1	PERR	FRAME	AD17	AD21	IDSEL	AD25	AD29	BCLK	TDO	TCLK	SGND	2																									
1	RGND	CVCC	GPIO0 (ACT1)	AD4	AD6	AD8	AD11	AD15	PAR	STOP	IRDY	AD16	AD18	AD22	C/BE3	AD26	AD28	RST	TDI	TMS	1																									
	Y	W	V	U	T	R	P	N	M	L	K	J	H	G	F	E	D	C	B	A																										

**Figure 2-2: Pin Diagram**

**This page intentionally left blank.**

### 2.3 PCI BUS INTERFACE

Ball	Pin Name	Type	Active	Description
C1	RST#	In	Low	Reset. This input sets all signals and registers in the chip to a known state. All outputs from the chip are tri-stated or driven to an inactive state. <b>This pin is ignored during Standby mode (STNDBY# pin low). The remainder of the system (therefore the system bus) may be powered down if desired (all bus output pins are tri-stated in Standby mode).</b>
D2	BCLK	In	High	Bus Clock. This input provides the timing reference for all PCI bus transactions. All bus inputs except RESET# are sampled on the rising edge of BCLK. BCLK may be any frequency from DC to 33MHz.
M1	PAR	I/O	High	Parity. This signal is used to maintain even parity across AD0-31 and C/BE0-3#. PAR is stable and valid one clock after the address phase. For data phases PAR is stable and valid one clock after either IRDY# is asserted on a write transaction or TRDY# is asserted on a read transaction. Once PAR is valid, it remains valid until one clock after the completion of the current data phase (i.e., PAR has the same timing as AD0-31 but delayed by one clock). The bus master drives PAR for address and write data phases; the target drives PAR for read data phases.
K2	FRAME#	In	Low	Cycle Frame. Driven by the current master to indicate the beginning and duration of an access. Assertion indicates a bus transaction is beginning (while asserted, data transfers continue); de-assertion indicates the transaction is in the final data phase
K1	IRDY#	In	Low	Initiator Ready. Indicates the bus master's ability to complete the current data phase of the transaction. During a write, IRDY# indicates valid data is present on AD0-31; during a read it indicates the master is prepared to accept data. A data phase is completed on any clock when both IRDY# and TRDY# are sampled then asserted (wait cycles are inserted until this occurs).
K4	TRDY#	S/TS	Low	Target Ready. Indicates the target's ability to complete the current data phase of the transaction. During a read, TRDY# indicates that valid data is present on AD0-31; during a write it indicates the target is prepared to accept data. A data phase is completed on any clock when both IRDY# and TRDY# are sampled then asserted (wait cycles are inserted until this occurs).
L1	STOP#	S/TS	Low	Stop. Indicates the current target is requesting the master to stop the current transaction.
L4	DEVSEL#	S/TS	Low	Device Select. Indicates the current target has decoded its address as the target of the current access

**Note:** S/TS stands for "Sustained Tri-state". These signals are driven by only one device at a time, are driven high for one clock before released, and are not driven for at least one cycle after being released by the previous device. A pull-up provided by the bus controller is used to maintain an inactive level between transactions. All signals listed above are powered by BVCC and GND.

### 2.3 PCI BUS INTERFACE (continued)

Ball	Pin Name	Type	Active	Description
L2	PERR#	S/TS	Low	Parity Error. This signal reports data parity errors (except for Special Cycles where SERR# is used). The PERR# pin is Sustained Tri-state. The receiving agent will drive PERR# active two clocks after detecting a data parity error. PERR# will be driven high for one clock before being tri-stated as with all sustained tri-state signals. PERR# will not report status until the chip has claimed the access by asserting DEVSEL# and completing the data phase.
L3	SERR#	OD	Low	System Error. Used to report system errors where the result will be catastrophic (address parity error, data parity errors for Special Cycle commands, etc.). This output is actively driven for a single PCI clock cycle synchronous to BCLK and meets the same setup and hold time requirements as all other bused signals. SERR# is not driven high by the chip after being asserted, but is pulled high only by a weak pull-up provided by the system. Thus, SERR# on the PCI bus may take two or three clock periods to fully return to an inactive state.

**Note:** S/TS stands for “Sustained Tri-state”. These signals are driven by only one device at a time, are driven high for one clock before released, and are not driven for at least one cycle after being released by the previous device. A pull-up provided by the bus controller is used to maintain an inactive level between transactions.

PERR# and SERR# are powered by BVCC and GND.

ROMOE# is powered by MVCC and GND

**2.3 PCI BUS INTERFACE (continued)**

Ball	Pin Name	Type	Active	Description
U2	AD0	I/O	High	PCI Address/Data Bus
T3	AD1	I/O	High	Address and data are multiplexed on the same pins. A bus transaction consists of an address phase followed by one or more data phases (both read and write bursts are allowed by the bus definition). The address phase is the clock cycle in which FRAME# is asserted (AD0-31 contain a 32-bit physical address). For I/O, the address is a byte address. For memory and configuration, the address is a DWORD address. During data phases AD0-7 contain the LSB and 24-31 contain the MSB. Write data is stable and valid when IRDY# is asserted; read data is stable and valid when TRDY# is asserted. Data transfers only during those clocks when both IRDY# and TRDY# are asserted.
R4	AD2	I/O	High	
T2	AD3	I/O	High	
U1	AD4	I/O	High	
R3	AD5	I/O	High	
T1	AD6	I/O	High	
R2	AD7	I/O	High	
R1	AD8	I/O	High	
P2	AD9	I/O	High	
N3	AD10	I/O	High	
P1	AD11	I/O	High	
N2	AD12	I/O	High	
M4	AD13	I/O	High	
M3	AD14	I/O	High	
N1	AD15	I/O	High	
J1	AD16	I/O	High	
J2	AD17	I/O	High	
H1	AD18	I/O	High	
J3	AD19	I/O	High	
J4	AD20	I/O	High	
H2	AD21	I/O	High	
G1	AD22	I/O	High	
H3	AD23	I/O	High	
G3	AD24	I/O	High	
F2	AD25	I/O	High	
E1	AD26	I/O	High	
F3	AD27	I/O	High	
D1	AD28	I/O	High	
E2	AD29	I/O	High	
F4	AD30	I/O	High	
E3	AD31	I/O	High	

C/BE3-0	Command Type	Supported
0000	Interrupt Acknowledge	
0001	Special Cycle	
0010	I/O Read	Y
0011	I/O Write	Y
0100	-reserved-	
0101	-reserved-	
0110	Memory Read	Y
0111	Memory Write	Y
1000	-reserved-	
1001	-reserved-	
1010	Configuration Read	Y
1011	Configuration Write	Y
1100	Memory Read Multiple	
1101	Dual Address Cycle	
1110	Memory Read Line	
1111	Memory Read & Invalidate	

P3	C/BE0#	In	Low	Bus Command/Byte Enables. During the address phase of a bus transaction, these pins define the bus command (see list above). During the data phase, these pins are byte enables that determine which byte lanes carry meaningful data: byte 0 corresponds to AD0-7, byte 1 to 8-15, byte 2 to 16-23, and byte 3 to 24-31.
M2	C/BE1#	In	Low	
K3	C/BE2#	In	Low	
F1	C/BE3#	In	Low	
G2	IDSEL	In	High	Initialization Device Select. Used as a chip select during configuration read and write transactions

**Note:** All signals listed above are powered by BVCC and GND.

**2.4 DISPLAY MEMORY INTERFACE**

Ball	Pin Name	Type	Active	Description
D18	AA0 (CFG0)	I/O	Both	DRAM address bus for Bank 0 and Bank 1.
C19	AA1 (CFG1)	I/O	Both	
B20	AA2 (CFG2)	I/O	Both	AA0 through AA9 also serve as configuration bits CFG0 through CFG9. Please see the descriptions for registers XR70 and XR71 for complete details on configuration options.
C18	AA3 (CFG3)	I/O	Both	
A20	AA4 (CFG4)	I/O	Both	
B19	AA5 (CFG5)	I/O	Both	
A19	AA6 (CFG6)	I/O	Both	
B18	AA7 (CFG7)	I/O	Both	
C17	AA8 (CFG8)	I/O	Both	
D16	AA9 (CFG9)	I/O	Both	
D10	MA0 (TM0)	I/O	High	DRAM data bits 0-15.
A10	MA1 (TM1)	I/O	High	
B10	MA2 (CFG10)	I/O	High	MA0 is also a test mode signal (Tri-State Enable).
C10	MA3 (CFG11)	I/O	High	
A9	MA4 (CFG12)	I/O	High	MA1 is also a test mode signal (ICT Enable).
B9	MA5 (CFG13)	I/O	High	
A8	MA6 (CFG14)	I/O	High	MA2 through MA7 also serve as configuration bits CFG10 through CFG15. Please see the description for register XR71 for complete details on configuration options.
C9	MA7 (CFG15)	I/O	High	
B8	MA8 (RMD0)	I/O	High	
A7	MA9 (RMD1)	I/O	High	MA8 through MA15 are also serve as the data bus for the BIOS ROM during system startup (i.e., before the system enables the graphics controller memory interface).
C8	MA10 (RMD2)	I/O	High	
B7	MA11 (RMD3)	I/O	High	
A6	MA12 (RMD4)	I/O	High	
C7	MA13 (RMD5)	I/O	High	
B6	MA14 (RMD6)	I/O	High	
A5	MA15 (RMD7)	I/O	High	
D15	MB0 (RMA0)	I/O	High	DRAM data bits 16-31.
B16	MB1 (RMA1)	I/O	High	
A17	MB2 (RMA2)	I/O	High	MB0 through MB15, along with MD11 and MD12, also serve as the address bus for the BIOS ROM during startup (i.e., before the system enables the graphics controller memory interface).
C15	MB3 (RMA3)	I/O	High	
A16	MB4 (RMA4)	I/O	High	
B15	MB5 (RMA5)	I/O	High	
C14	MB6 (RMA6)	I/O	High	
A15	MB7 (RMA7)	I/O	High	Normally, a separate graphics BIOS ROM is not required in portable computer designs, because the graphics BIOS is normally placed in the same ROM devices as the system BIOS. However, this graphics controller provides this BIOS ROM interface capability for use in development systems and add-in cards for flat panel displays. Since the PCI bus specification requires only one load on the PCI bus for each PCI device, this BIOS ROM interface is provided to allow access to the BIOS ROM through the graphics controller chip, itself.
B14	MB8 (RMA8)	I/O	High	
C13	MB9 (RMA9)	I/O	High	
A14	MB10 (RMA10)	I/O	High	
B13	MB11 (RMA11)	I/O	High	
D12	MB12 (RMA12)	I/O	High	
C12	MB13 (RMA13)	I/O	High	
A13	MB14 (RMA14)	I/O	High	
B12	MB15 (RMA15)	I/O	High	

**Note:** All signals listed above are powered by MVCC and GND.

**2.4 DISPLAY MEMORY INTERFACE (continued)**

<b>Ball</b>	<b>Pin Name</b>	<b>Type</b>	<b>Active</b>	<b>Description</b>
J18	MC0	I/O	High	DRAM data bits 32-47.
J17	MC1	I/O	High	
H19	MC2	I/O	High	
G20	MC3	I/O	High	
H18	MC4	I/O	High	
G19	MC5	I/O	High	
F20	MC6	I/O	High	
G18	MC7	I/O	High	
F19	MC8	I/O	High	
E20	MC9	I/O	High	
F18	MC10	I/O	High	
D20	MC11	I/O	High	
E19	MC12	I/O	High	
F17	MC13	I/O	High	
E18	MC14	I/O	High	
D19	MC15	I/O	High	
R20	MD0	I/O	High	DRAM data bits 48-63.
P19	MD1	I/O	High	
N18	MD2	I/O	High	
P20	MD3	I/O	High	
N19	MD4	I/O	High	
M17	MD5	I/O	High	
M18	MD6	I/O	High	
N20	MD7	I/O	High	
M19	MD8	I/O	High	
M20	MD9	I/O	High	
L18	MD10	I/O	High	
L19	MD11 (RMA16)	I/O	High	
L20	MD12 (RMA17)	I/O	High	
L17	MD13	I/O	High	
K17	MD14	I/O	High	
K20	MD15	I/O	High	
C11	RAS0#	Out	Low	RAS for DRAM Bank 0 (128K, 256K, or 512K by 64-bit).
K18	RAS1#	Out	Low	
				RAS for DRAM Bank 1.
C6	ROMOE# (MCLKOUT)	Out	Low	Output Enable for BIOS ROM. May be configured as MCLK output in test mode.

**Note:** All signals listed above are powered by MVCC and GND.

## 2.4 DISPLAY MEMORY INTERFACE (continued)

Ball	Pin Name	Type	Active	Description
D11	CASAL#	Out	High	CAS for dual-CAS EDO DRAM
A11	CASAH#	Out	High	Memory data byte mask signals, one mask signal for each of the eight data bytes in the 64-bit Qword. The masking is performed on a per-byte basis. A given byte is masked when the signal is high, or enabled when the signal is low. Masking is needed on write operations to specify which bytes in the 64-bit word are being written.
C16	CASBL#	Out	High	
B17	CASBH#	Out	High	
H20	CASCL#	Out	High	
J19	CASCH#	Out	High	
P18	CASDL#	Out	High	
R19	CASDH#	Out	High	
B11	WEA#	Out	Low	
A18	WEB#	Out	Low	MB[15:0] write enable for dual-CAS EDO DRAM
J20	WEC#	Out	Low	MC[15:0] write enable for dual-CAS EDO DRAM
T20	WED#	Out	Low	MD[15:0] write enable for dual-CAS EDO DRAM

**Notes:** The 8 bytes comprising each 64-bit Qword are labeled AL, AH, BL, BH, CL, CH, DL, and DH. There is a separate byte mask signal for each byte. Up to two banks can be supported, with RAS0# controlling the first bank and RAS1# controlling the second bank. The address, data and byte mask signals are the same for each bank.

All signals listed above are powered by MVCC and GND.



## 2.5 FLAT PANEL DISPLAY INTERFACE

Ball	Pin Name	Type	Active	Description
W6	P0	Out	High	Flat panel data bus of up to 36-bits.
V7	P1	Out	High	
Y6	P2	Out	High	
W7	P3	Out	High	
V8	P4	Out	High	
Y7	P5	Out	High	
W8	P6	Out	High	
U9	P7	Out	High	
V9	P8	Out	High	
Y8	P9	Out	High	
W9	P10	Out	High	
Y9	P11	Out	High	
V10	P12	Out	High	
W10	P13	Out	High	
Y10	P14	Out	High	
U10	P15	Out	High	
U11	P16	Out	High	
Y11	P17	Out	High	
W11	P18	Out	High	
V11	P19	Out	High	
Y12	P20	Out	High	
Y13	P21	Out	High	
V12	P22	Out	High	
U12	P23	Out	High	
W13	P24	Out	High	
Y14	P25	Out	High	
V13	P26	Out	High	
W14	P27	Out	High	
Y15	P28	Out	High	
V14	P29	Out	High	
W15	P30	Out	High	
Y16	P31	Out	High	
V15	P32	Out	High	
Y17	P33	Out	High	
W16	P34	Out	High	
U15	P35	Out	High	

**Note:** All signals listed above are powered by DVCC and GND.

## 2.5 FLAT PANEL DISPLAY INTERFACE (continued)

Ball	Pin Name	Type	Active	Description
Y5	SHFCLK	Out	High	Shift Clock. Pixel clock for flat panel data.
W5	FLM	Out	High	First Line Marker. Flat Panel equivalent of VSYNC.
Y4	LP (CL1) (DE) (BLANK#)	Out	High	Latch Pulse (may also be called CL1). Flat Panel equivalent of HSYNC. May also be configured as DE (display enable) or BLANK# output.
V6	M (DE) (BLANK#)	Out	High	M signal for panel AC drive control (may also be called ACDCLK). May also be configured as DE (display enable) or BLANK# output.
V5	ENAVDD	I/O	High	Power sequencing control for panel driver electronics voltage VDD.
W4	ENAVEE (ENABKL)	I/O	High	Power sequencing control for panel bias voltage VEE. May also be configured as ENABKL.
U6	ENABKL	I/O	High	Power sequencing control for enabling the backlight.

**Note:** All signals listed above are powered by DVCC and GND.

### Notes for table on following page:

To accommodate a wide variety of panel types, the graphics controller has been designed to output its data in any of a number of formats. These formats include different data widths for the colors belonging to each pixel, and the ability to accommodate different pixel data transfer timing requirements.

For STN-DD panels, pins P0 through P35 are organized into groups corresponding to the upper and lower parts of the panel. The names of the signals for the upper and lower parts follow a naming convention of Uxx and Lxx, respectively.

For panels that require a pair of adjacent pixels be sent with every shift clock, pins P0 through P35 are organized into groups corresponding to the first and second (from right to left) pixels of each pair of pixels being sent. The names of the signals for the first and second pixels of each such pair follow a naming convention of Fxx and Sxx, respectively.

Panels that transfer data on both edges of SHFCLK are also supported. See the description for register FR12 for more details.

**2.5 FLAT PANEL DISPLAY INTERFACE (continued)**

		Mono SS	Mono DD	Mono DD	Color TFT	Color TFT	Color TFT	Color TFT HR	Color STN SS	Color STN SS	Color STN DD	Color STN DD	Color STN DD
Pin#	Pin Name	8-bit	8-bit	16 bit	9/12/16 bit	18/24 bit	36-bit	18/24 bit	8-bit (4bP)	16-bit (4bP)	8-bit (4bP)	16-bit (4bP)	24-bit
W6	P0	P0	UD3	UD7	B0	B0	FB0	FB0	R1	R1	UR1	UR0	UR0
V7	P1	P1	UD2	UD6	B1	B1	FB1	FB1	B1	G1	UG1	UG0	UG0
Y6	P2	P2	UD1	UD5	B2	B2	FB2	FB2	G2	B1	UB1	UB0	UB0
W7	P3	P3	UD0	UD4	B3	B3	FB3	FB3	R3	R2	UR2	UR1	LR0
V8	P4	P4	LD3	UD3	B4	B4	FB4	SB0	B3	G2	LR1	LR0	LG0
Y7	P5	P5	LD2	UD2	G0	B5	FB5	SB1	G4	B2	LG1	LG0	LB0
W8	P6	P6	LD1	UD1	G1	B6	SB0	SB2	R5	R3	LB1	LB0	UR1
U9	P7	P7	LD0	UD0	G2	B7	SB1	SB3	B5	G3	LR2	LR1	UG1
V9	P8	-	-	LD7	G3	G0	SB2	FG0	-	B3	-	UG1	UB1
Y8	P9	-	-	LD6	G4	G1	SB3	FG1	-	R4	-	UB1	LR1
W9	P10	-	-	LD5	G5	G2	SB4	FG2	-	G4	-	UR2	LG1
Y9	P11	-	-	LD4	R0	G3	SB5	FG3	-	B4	-	UG2	LB1
V10	P12	-	-	LD3	R1	G4	FG0	SG0	-	R5	-	LG1	UR2
W10	P13	-	-	LD2	R2	G5	FG1	SG1	-	G5	-	LB1	UG2
Y10	P14	-	-	LD1	R3	G6	FG2	SG2	-	B5	-	LR2	UB2
U10	P15	-	-	LD0	R4	G7	FG3	SG3	-	R6	-	LG2	LR2
U11	P16	-	-	-	-	R0	FG4	FR0	-	-	-	-	LG2
Y11	P17	-	-	-	-	R1	FG5	FR1	-	-	-	-	LB2
W11	P18	-	-	-	-	R2	SG0	FR2	-	-	-	-	UR3
V11	P19	-	-	-	-	R3	SG1	FR3	-	-	-	-	UG3
Y12	P20	-	-	-	-	R4	SG2	SR0	-	-	-	-	UB3
Y13	P21	-	-	-	-	R5	SG3	SR1	-	-	-	-	LR3
V12	P22	-	-	-	-	R6	SG4	SR2	-	-	-	-	LG3
U12	P23	-	-	-	-	R7	SG5	SR3	-	-	-	-	LB3
W13	P24	-	-	-	-	-	FR0	-	-	-	-	-	-
Y14	P25	-	-	-	-	-	FR1	-	-	-	-	-	-
V13	P26	-	-	-	-	-	FR2	-	-	-	-	-	-
W14	P27	-	-	-	-	-	FR3	-	-	-	-	-	-
Y15	P28	-	-	-	-	-	FR4	-	-	-	-	-	-
V14	P29	-	-	-	-	-	FR5	-	-	-	-	-	-
W15	P30	-	-	-	-	-	SR0	-	-	-	-	-	-
Y16	P31	-	-	-	-	-	SR1	-	-	-	-	-	-
V15	P32	-	-	-	-	-	SR2	-	-	-	-	-	-
Y17	P33	-	-	-	-	-	SR3	-	-	-	-	-	-
W16	P34	-	-	-	-	-	SR4	-	-	-	-	-	-
U15	P35	-	-	-	-	-	SR5	-	-	-	-	-	-
Y5	SHFCLK	SHFCLK	SHFCLK	SHFCLK	SHFCLK	SHFCLK	SHFCLK	SHFCLK	SHFCLK	SHFCLK	SHFCLK	SHFCLK	SHFCLK
Pixels/Clock:		8	8	16	1	1	2	2	2-2/3	5-1/3	2-2/3	5-1/3	8

See the notes for this table on the previous page.

## 2.6 CRT INTERFACE

Ball	Pin Name	Type	Active	Description
U3	HYSNC (CSYNC)	Out	Both	CRT Horizontal Sync (polarity is programmable) or "Composite Sync" for support of various external NTSC/PAL encoder chips.
V2	VSNC	Out	Both	CRT Vertical Sync (polarity is programmable).
Y3	RED	Out	Analog	CRT analog video outputs from the internal color palette DAC.
V4	GREEN	Out	Analog	The DAC is designed for a 37.5 $\Omega$ equivalent load on each pin (e.g. 75 $\Omega$ resistor on the board, in parallel with the 75 $\Omega$ CRT load)
W3	BLUE	Out	Analog	
W2	RSET	In	N/A	Set point resistor for the internal color palette DAC. A 560 $\Omega$ 1% resistor is required between RSET and AGND.
V3	DDC DATA (GPIO2)	I/O	High	General purpose I/O, suitable for use as DDC data.
U4	DDC CLK (GPIO3)	I/O	High	General purpose I/O, suitable for use as DDC DATA. These two pins are functionally suitable for a DDC interface between the 65555 and a CRT monitor.

**Note:** HSYNC, VSNC, GPIO2, and GPIO3 are powered by CVCC and GND.  
RED, GREEN, BLUE, and RSET are powered by AVCC and AGND.

**2.7 POWER/ GROUND AND STANDBY CONTROL**

Ball	Pin Name	Type	Active	Description
U5	AVCC	VCC	-	Analog power and ground pins for noise isolation for the internal color palette DAC. AVCC should be isolated from digital VCC as described in the Functional Description of the internal color palette DAC. For proper DAC operation, AVCC should not be greater than IVCC. AGND should be common with digital ground but must be tightly decoupled to AVCC. See the Functional Description of the internal color palette DAC for further information.
Y2	AGND	GND	-	
B3	SVCC	VCC	-	Analog power and ground pins for noise isolation for the internal clock synthesizer (for MCLK). Must be the same as IVCC, 3.3V.
A2	SGND	GND	-	
C4,D5	PVCC	VCC	-	Analog power and ground pins for noise isolation for internal clock synthesizer (for VCLK). Must be the same as IVCC. SVCC/SGND and PVCC/PGND pairs must be carefully decoupled individually. Refer also to the section on clock ground layout in the Functional Description. Power for CRT Interface, 3.3V.
A3,B4	PGND	GND	-	
W1	CVCC	VCC	-	
D9, & W12	IVCC	VCC	-	Power/Ground (Internal Logic), 3.3V. Note that this voltage must be the same as SVCC and PVCC (voltages for internal clock synthesizers).
D14, D7, G17, G4, P17, P4, U14, U7, J9-12, K9-12, L9-12, M9-12	GND	GND	-	
Y1	RGND	GND	-	Internal reference GND, should be tied to GND.
H4,N4	BVCC	VCC	-	Power (Bus Interface), 3.3V.
U8	DVCC	VCC	-	Power (Flat Panel Interface), 3.3V.
D13, H17, N17	MVCC	VCC	-	Power (Memory Interface), 3.3V.
U13	VVCC	VCC	-	Power (Video Interface), 3.3V.

## 2.8 VIDEO INTERFACE

Ball	Pin Name	Type	Active	Description
V16	VREF	I/O	High	Vertical reference input for video data port.
W17	HREF	In	High	Horizontal reference input for video data port.
Y18	VCLK	In	High	Clock input for video data port.
V17	PCLK (VCLKOUT)	Out	High	Outputs DCLK, or DCLK divided by 2. See the description for register XR60 for complete details. Usable with either the video data port or the flat panel interface. May also be configured to output VCLK in test mode.
R18	VP0	In	High	Data bus for video data port.
U20	VP1	In	High	
T19	VP2	In	High	When used as a ZV-Port interface, VP0-7 correspond to Y0-7, and VP8-15 correspond to UV0-7.
R17	VP3	In	High	
T18	VP4	In	High	
U19	VP5	In	High	
V20	VP6	In	High	
T17	VP7	In	High	
U18	VP8	In	High	
V19	VP9	In	High	
W20	VP10	In	High	
W19	VP11	In	High	
U17	VP12	In	High	
V18	VP13	In	High	
Y19	VP14	In	High	
W18	VP15	In	High	

**Note:** All signals listed above are powered by VVCC and GND.

## 2.9 BOUNDARY SCAN

Ball	Pin Name	Type	Active	Description
A1	TMS	In	High	Test mode select for boundary scan.
B2	TCLK (DCLKIN)	In	High	Test clock for boundary scan. Can be configured to be used as an input for an externally provided DCLK through a strapping option. See the descriptions for registers XR70 and XRCF for complete details.
B1	TDI (MCLKIN)	In	High	Test data input for boundary scan. Can be configured to be used as an input for an externally provided MCLK through a strapping option and register programming. See the descriptions for registers XR70 and XRCF for complete details.
C2	TDO	In	High	Test data out for boundary scan.
D3	TRST#	In	High	Test reset for boundary scan

**Note:** TMS, TCLK, TDI, TDO and TRST#, are powered by BVCC and GND.

## 2.10 MISCELLANEOUS

Ball	Pin Name	Type	Active	Description
E4	STNDBY#	In	Low	Standby Control Pin. Pull this pin low to place the chip in Standby Mode. A low to high transition on the pin will cause change to exit standby mode, host standby mode, and panel off mode.
C3	REFCLK (MCLKIN)	In	High	Reference Clock Input. This pin serves as the input for an external reference oscillator (usually 14.31818 MHz). All timings of the 65555 are derived from this primary clock input source. Can be configured to be used as an input for an externally provided MCLK through a strapping option and register programming. For normal operation, TDI should be used as the input for an externally provided MCLK.
V1	GPIO0 (ACTI)	I/O	High	General Purpose I/O pin, or ACTI (Activity Indicator).
T4	GPIO1 (32KHz)	I/O	High	General Purpose I/O pin, or 32KHz input: clock input for refresh of non-self-refresh DRAMs and panel power sequencing.
D6	N/C	n/a	n/a	These pins should be left open.
C5	N/C	n/a	n/a	
A12	N/C	n/a	n/a	
K19	N/C	n/a	n/a	
Y20	Reserved	n/a	n/a	These pins are reserved for future use, and should not be connected.
D8	Reserved	n/a	n/a	
D17	Reserved	n/a	n/a	
A4	Reserved	n/a	n/a	
B5	Reserved	n/a	n/a	
D4	Reserved	n/a	n/a	
U16	Reserved	n/a	n/a	
C20	Reserved	n/a	n/a	
E17	Reserved	n/a	n/a	

**Note:** STANDBY#, RCLK, GPIO0, and GPIO1 are powered by DVCC and GND.



## ***CHAPTER 3***

# ***ELECTRICAL SPECIFICATIONS***

**This page intentionally left blank.**

### 3 ELECTRICAL SPECIFICATIONS

**Table 3-1: 65555 Absolute Maximum Conditions**

Symbol	Parameter	Min	Max	Units
V <sub>CC</sub>	Supply Voltage	-0.5	5.0	V
V <sub>I</sub>	Input Voltage	-0.5	5.5	V
T <sub>STG</sub>	Storage Temp	-40	125	°C

**Note:** Permanent device damage may occur if Absolute Maximum Rating are exceeded.  
Functional operation must be restricted to the conditions under Normal Operating Conditions.

**Table 3-2: 65555 Normal Operating Conditions**

Symbol	Parameter	Min	Typical	Max	Units
V <sub>CC</sub>	Supply Voltage	3.15	3.3	3.6	V
T <sub>A</sub>	Ambient Temperature	0	—	70	°C

**Table 3-3: 65555 DAC Characteristics:  
(Under Normal Operating Conditions Unless Noted Otherwise)**

Symbol	Parameter	Notes	Min	Typical	Max	Units
I <sub>O</sub>	Full Scale Output Current	R <sub>SET</sub> =560Ω and 37.5Ω Load	—	18.6	—	mA
	Full Scale Error		—	—	± 5	%
	DAC to DAC Correlation		—	1.27	—	%
	DAC Linearity		± 2	—	—	LSB

**Table 3-4: 65555 DC Characteristics:**  
(Under Normal Operating Conditions Unless Noted Otherwise)

**Note:** For power configuration data, please refer to application notes.

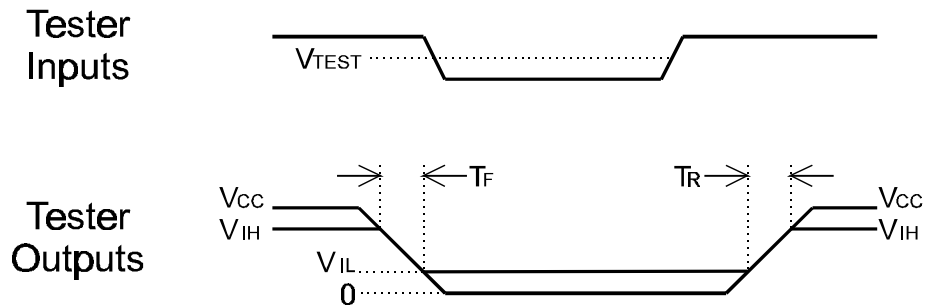
Symbol	Parameter	Notes	Min	Typical	Max	Units
P <sub>D</sub>	Power Dissipation	All VCCs at 3.3V MCLK=50MHz, DCLK=110MHz	-	-	1.7	W
I <sub>IL</sub>	Input Leakage Current		-100	-	+100	μA
I <sub>OZ</sub>	Output Leakage Current	High Impedance	-100	-	+100	μA
V <sub>IL</sub>	Input Low Voltage	All input pins	-0.5	-	0.8	V
V <sub>IH</sub>	Input High Voltage	All input pins	0.6xV <sub>CC</sub>	-	5.5	V
V <sub>OL</sub>	Output Low Voltage	Under max load per table 3-5 (3.3V)	-	-	0.5	V
V <sub>OH</sub>	Output High Voltage	Under max load per table 3-5 (3.3V)	0.7xV <sub>CC</sub>	-	-	V

**Table 3-5: 65555 DC Drive Characteristics:**  
(Under Normal Operating Conditions Unless Noted Otherwise)

Symbol	Parameter	Output Pins	DC Test Conditions	Min	Units
I <sub>OL</sub>	Output Low Current	AA0-AA9	V <sub>OUT</sub> ≤ V <sub>OL</sub> V <sub>CC</sub> = 3.3V	12	mA
		H/VSYNC, P0-P35, SHFCLK, M DEVSEL#, PAR, PERR#, SERR#, STOP#, TRDY#		8	mA
		CASAH/L#, CASBH/L#, CASCH/L#, CASDH/L#			
		ACTI, AD0-AD31, ENABKL, ENAVDD, ENAVEE, FLM, LP			
		COE#, RAS0#, RAS1#, WEA#, WEB#, WEC#, WED#		4	mA
	All other outputs		2	mA	
I <sub>OH</sub>	Output High Current	AA0-AA9	V <sub>OUT</sub> ≥ V <sub>OH</sub> V <sub>CC</sub> = 3.3V	12	mA
		H/VSYNC, P0-P35, SHFCLK, M DEVSEL#, PAR, PERR#, SERR#, STOP#, TRDY#		8	mA
		CASAH/L#, CASBH/L#, CASCH/L#, CASDH/L#			
		ACTI, AD0-AD31, ENABKL, ENAVDD, ENAVEE, FLM, LP			
		COE#, RAS0#, RAS1#, WEA#, WEB#, WEC#, WED#		4	mA
	All other outputs		2	mA	

**Table 3-6: 65555 AC Test Conditions**

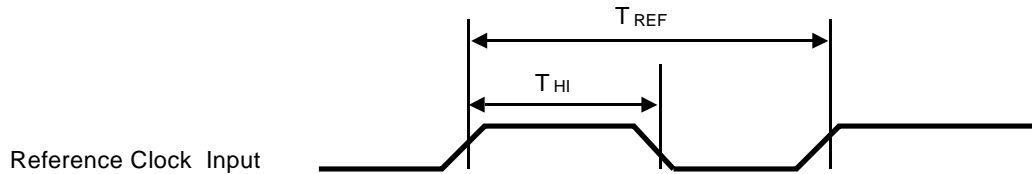
Symbol	Parameter	Minimum	Maximum	Units
$V_{CC}$	Supply Voltage	3.15	3.6	V
$V_{TEST}$	All AC parameters	1.5	1.5	V
$V_{IL}$	Input low voltage (10% of $V_{CC}$ )	$0.1 V_{CC}$	$0.1 V_{CC}$	V
$V_{IH}$	Input high voltage (90% of $V_{CC}$ )	$V_{CC} - 0.1$	$V_{CC} - 0.1$	V
$T_R$	Maximum input rise time	3	3	ns
$T_F$	Maximum input fall time	2	2	ns



**Figure 3-1: AC Test Timing**

**Table 3-7: 65555 AC Timing Characteristics - Reference Clock**

Symbol	Parameter	Notes	Min	Typical	Max	Units
$F_{REF}$	Reference Frequency		1	14.31818	60	MHz
$T_{REF}$	Reference Clock Period		16.6	69.84128	1000	ns
$T_{HI}/T_{REF}$	Reference Clock Duty Cycle		40	-	60	%



**Figure 3-2: Reference Clock Timing**

**Table 3-8: 65555 AC Timing Characteristics - Clock Generator**

Symbol	Parameter	Notes	Min	Typical	Max	Units
F <sub>DCLK</sub>	DCLK Frequency		–	–	110	MHz
F <sub>MCLK</sub>	MCLK Frequency		–	–	50	MHz

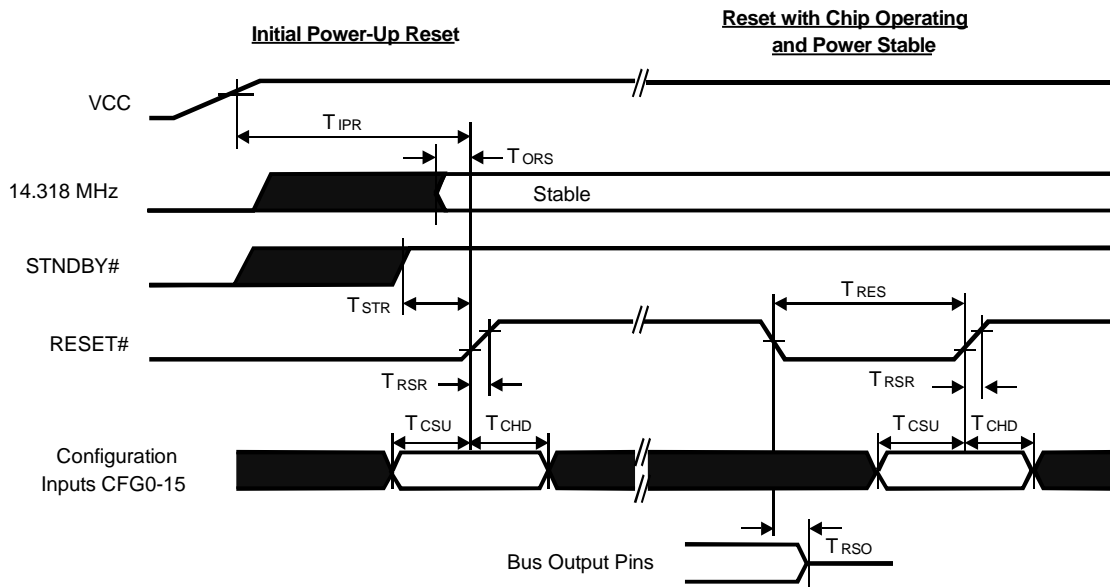
**Table 3-9: 65555 AC Timing Characteristics - Reset**

Symbol	Parameter	Notes	Min	Max	Units
T <sub>IPR</sub>	Reset Inactive from Power Stable	See Note 1	1	–	ms
T <sub>ORS</sub>	Reset Inactive from Ext.Osc. Stable		0	–	ms
T <sub>RES</sub>	Minimum Reset Pulse Width	See Note 2	1	–	ms
T <sub>STR</sub>	Reset Inactive from Standby Inactive	RESET# is ignored in Standby Mode	2	–	ms
T <sub>RSR</sub>	Reset Rise Time	measured 0.1V <sub>cc</sub> to 0.9V <sub>cc</sub>	–	20	ns
T <sub>RSO</sub>	Reset Active to Output Float Delay		–	40	ns
T <sub>CSU</sub>	Configuration Setup Time	See Note 3	20	–	ns
T <sub>CHD</sub>	Configuration Hold Time		5	–	ns

**Note 1:** This parameter includes time for internal voltage stabilization of all sections of the chip, startup and stabilization of the internal clock synthesizer, and setting of all internal logic to a known state.

**Note 2:** This parameter includes time for the internal clock synthesizer to reset to its default frequency and time to set all internal logic to a known state. It assumes power is stable and the internal clock synthesizer is already operating at some stable frequency.

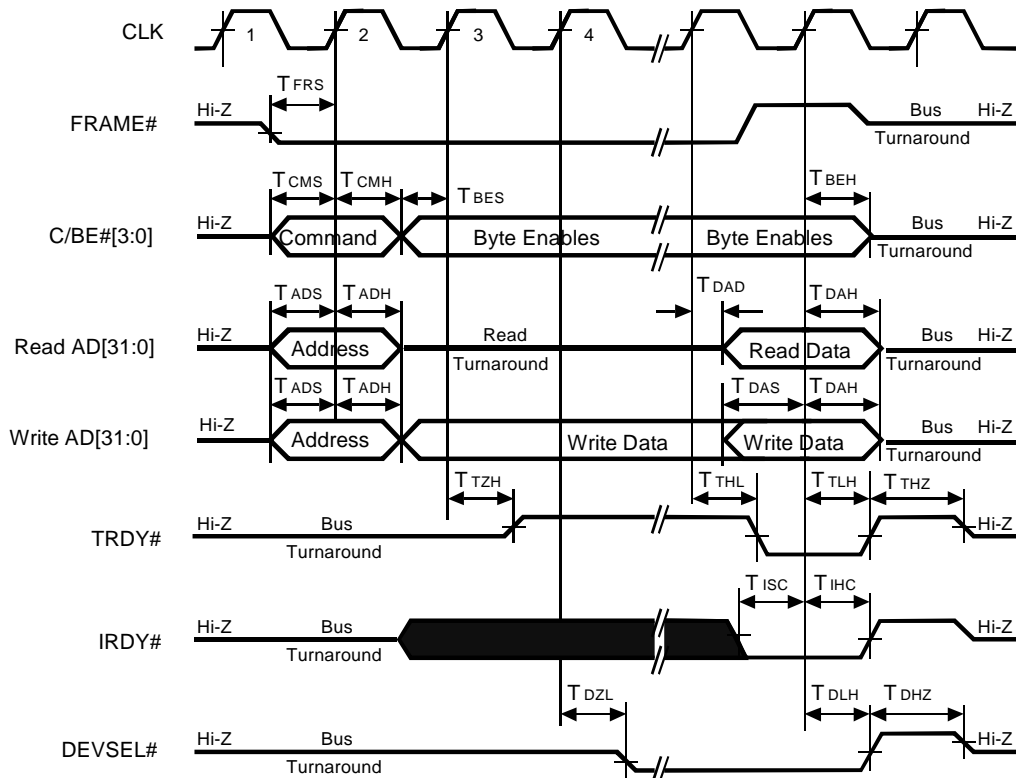
**Note 3:** This parameter specifies the setup time to latch reliably the state of the configuration bits. Changes in some configuration bits may take longer to stabilize inside the chip (such as internal clock synthesizer-related bits 4 and 5). The recommended configuration bit setup time is T<sub>RES</sub> to insure that the chip is in a completely stable state when Reset goes inactive.



**Figure 3-3: Reset Timing**

**Table 3-10: 65555 AC Timing Characteristics - PCI Bus Frame (CLK=33MHz)**

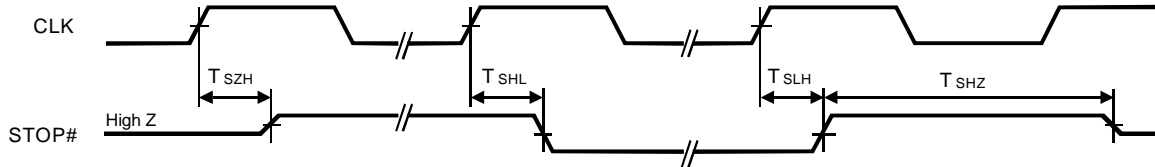
Symbol	Parameter	Notes	Min	Max	Units
T <sub>FRS</sub>	FRAME# Setup to CLK		7	–	ns
T <sub>CMS</sub>	C/BE#[3:0] (Bus CMD) Setup to CLK		7	–	ns
T <sub>CMH</sub>	C/BE#[31:0] (Bus CMD) Hold from CLK		0	–	ns
T <sub>BES</sub>	C/BE#[3:0] (Byte Enable) Setup to CLK		7	–	ns
T <sub>BEH</sub>	C/BE#[3:0] (Byte Enable) Hold from CLK		0	–	ns
T <sub>ADS</sub>	AD[31:0] (Address) Setup to CLK		7	–	ns
T <sub>ADH</sub>	AD[31:0] (Address) Hold from CLK		0	–	ns
T <sub>DAS</sub>	AD[31:0] (Data) Setup to CLK		7	–	ns
T <sub>DAH</sub>	AD[31:0] (Data) Hold from CLK		0	–	ns
T <sub>DAD</sub>	AD[31:0] (Data) Valid from CLK		2	11	ns
T <sub>TZH</sub>	TRDY# High Z to High from CLK		2	11	ns
T <sub>THL</sub>	TRDY# Active from CLK		2	11	ns
T <sub>TLH</sub>	TRDY# Inactive from CLK		2	11	ns
T <sub>THZ</sub>	TRDY# High before High Z		1	–	CLK
T <sub>DZL</sub>	DEVSEL# Active from CLK		2	11	ns
T <sub>DLH</sub>	DEVSEL# Inactive from CLK		2	11	ns
T <sub>DHZ</sub>	DEVSEL# High before High Z		1	–	CLK
T <sub>ISC</sub>	IRDY# Setup to CLK		7	–	ns
T <sub>IHC</sub>	IRDY# Hold from CLK		0	–	ns



**Figure 3-4: PCI Bus Frame Timing**

**Table 3-11: 65555 AC Timing Characteristics - PCI Bus Stop (CLK=33MHz)**

Symbol	Parameter	Notes	Min	Max	Units
T <sub>SZH</sub>	STOP# High Z to High from CLK		2	11	ns
T <sub>SHL</sub>	STOP# Active from CLK		2	11	ns
T <sub>SLH</sub>	STOP# Inactive from CLK		2	11	ns
T <sub>SHZ</sub>	STOP# High before High Z		1	—	CLK

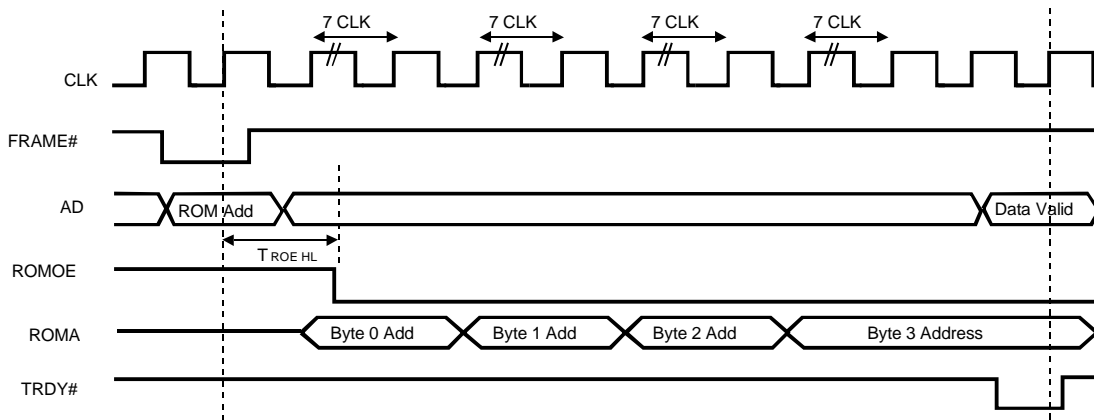


**Figure 3-5: PCI Bus Stop Timing**

**Table 3-12: 65555 AC Timing Characteristics PC BIOS ROM**

Symbol	Parameter	Notes	Min	Max	Units
T <sub>ROE</sub>	ROMOE# Active from CLK		—	40	ns
T <sub>ROM</sub>	Slowest Permissible BIOS ROM Access Speed		—	150	ns

**Note:** PCI BIOS ROM timing is derived from the PCI bus clock. Timing sequences are fixed assuming the use of widely-available, low-cost, typical industry-standard EPROMs. Timing specifications and performance of BIOS ROM memory accesses are non-critical since PCI BIOS ROM data is always shadowed into high-speed system memory prior to execution of BIOS code.



**Figure 3-6: PCI BIOS ROM Timing**



**Table 3-13: 65555 AC Timing Characteristics - EDO DRAM Read / Write**

Symbol	Parameter	Notes	Min	Max	Units
$T_{PC}$	CAS Cycle Time		$T_m$		ns
$T_{RC}$	Read/Write Cycle Time		$6T_m - 5$	–	ns
$T_{RAS}$	RAS# Pulse Width		$4T_m - 5$	–	ns
$T_{RP}$	RAS# Precharge		$2T_m - 5$	–	ns
$T_{CRP}$	CAS# to RAS#Precharge		$2T_m - 15$	–	ns
$T_{CSH}$	CAS# Hold from RAS#		$3T_m - 5$	–	ns
$T_{RCD}$	RAS# to CAS# Delay		$2.5T_m - 10$	–	ns
$T_{RSH}$	RAS# Hold from CAS#		$1.5T_m - 15$	–	ns
$T_{CP}$	CAS# Precharge		$0.5T_m - 2$	–	ns
$T_{CAS}$	CAS# Pulse Width		$0.5T_m - 2$	–	ns
$T_{ASR}$	Row Address Setup to RAS#		$0.5T_m - 5$	–	ns
$T_{ASC}$	Column Address Setup to CAS#		$0.5T_m - 5$	–	ns
$T_{RAH}$	Row Address Hold from RAS#		$2T_m - 5$	–	ns
$T_{CAH}$	Column Address Hold from CAS#		$0.5T_m - 2$	–	ns
$T_{CAC}$	Data Access Time from CAS#		–	$T_m - 7$	ns
$T_{RAC}$	Data Access Time from RAS#		–	$4T_m - 30$	ns
$T_{DS}$	Write Data Setup to CAS#		0	–	ns
$T_{DH}$	Write Data Hold from CAS#		10	–	ns
$T_{COH}$	Read data hold from CAS# fall		5	–	ns
$T_{WS}$	WE# Setup to CAS#		$2T_m - 10$	–	ns
$T_{WH}$	WE# Hold from CAS#		$2T_m - 5$	–	ns

**Note:**  $T_m$  = period of MCLK.

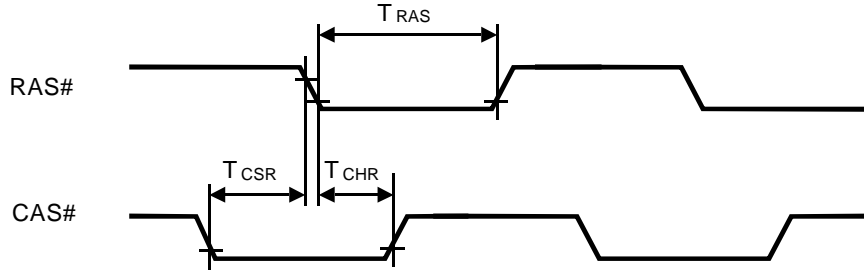
**Test Conditions:** Loading per signal: 80pf.

**Note:** The 65555 does not perform mixed read and write (or read modify write) cycles during the same CAS low interval.



**Table 3-14: 65555 AC Timing Characteristics - CBR Refresh**

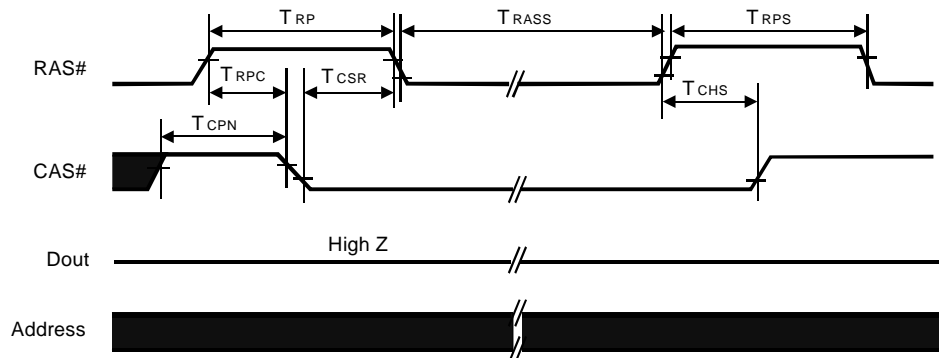
Symbol	Parameter	Notes	Min	Typical	Max	Units
T <sub>CHR</sub>	RAS# to CAS# Delay	Normal Operation	5T <sub>m</sub> - 5	-	-	ns
		Standby Mode	90	-	-	ns
T <sub>CSR</sub>	CAS# to RAS# Delay	Normal Operation	T <sub>m</sub> - 5	-	-	ns
		Standby Mode	8	-	-	ns
T <sub>RAS</sub>	RAS# Pulse Width	Normal Operation	5T <sub>m</sub> - 5	-	-	ns
		Standby Mode	100	-	-	ns



**Figure 3-9: CAS-Before-RAS (CBR) DRAM Refresh Cycle Timing**

**Table 3-15: 65555 AC Timing Characteristics - Self Refresh**

Symbol	Parameter	Notes	Min	Typical	Max	Units
T <sub>RASS</sub>	RAS# Pulse Width for Self-Refresh		100	-	-	μs
T <sub>RP</sub>	RAS# Precharge		4T <sub>m</sub> - 3	-	-	ns
T <sub>RPS</sub>	RAS# Precharge for Self-Refresh		10T <sub>m</sub>	-	-	ns
T <sub>RPC</sub>	RAS# to CAS# Delay		3T <sub>m</sub> - 5	-	-	ns
T <sub>CSR</sub>	CAS# to RAS# Delay		T <sub>m</sub> - 5	-	-	ns
T <sub>CHS</sub>	CAS# Hold Time		0	-	-	ns
T <sub>CPN</sub>	CAS# Precharge		T <sub>m</sub> - 5	-	-	ns



**Figure 3-10: "Self Refresh DRAM" Refresh Cycle Timing**

Table 3-16: 65555 AC Timing Characteristics - Video Input Port

Symbol	Parameter	Notes	Min	Max	Units
$T_{VDS}$	VP (Incoming Data) Setup	ZV-Port Mode	5	–	ns
$T_{VDH}$	VP (Incoming Data) Hold		3	–	ns
$T_{HRS}$	HREF (Incoming HS) Setup		5	–	ns
$T_{HRH}$	HREF (Incoming HS) Hold		3	–	ns
$T_{VRS}$	VREF (Incoming VS) Setup		5	–	ns
$T_{VRH}$	VREF (Incoming VS) Hold		3	–	ns
	VCLK Frequency ( $T_{VCLK}$ is VCLK period)		10	33	MHz
	VCLK Duty Cycle		40	60	%

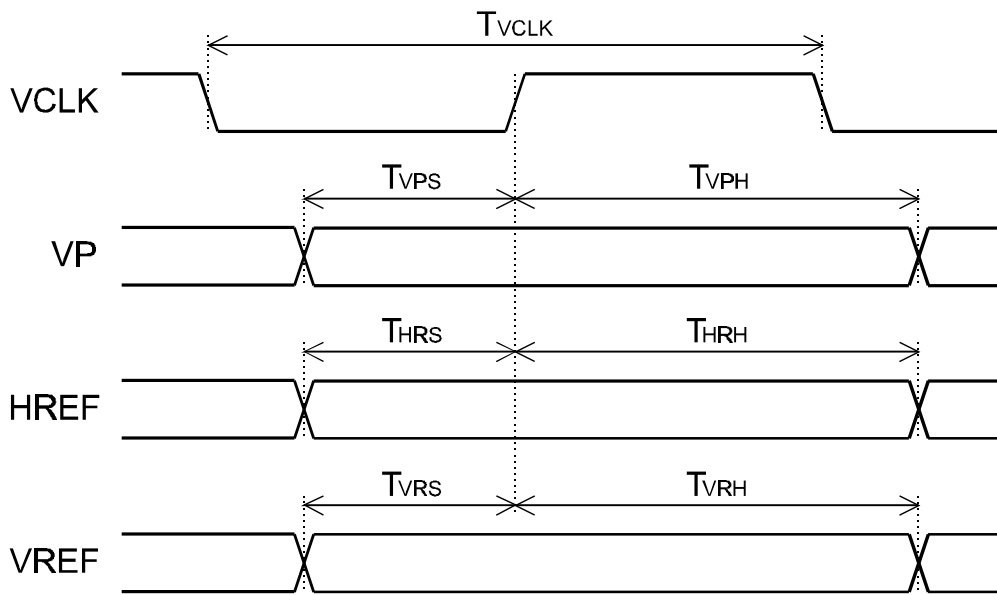
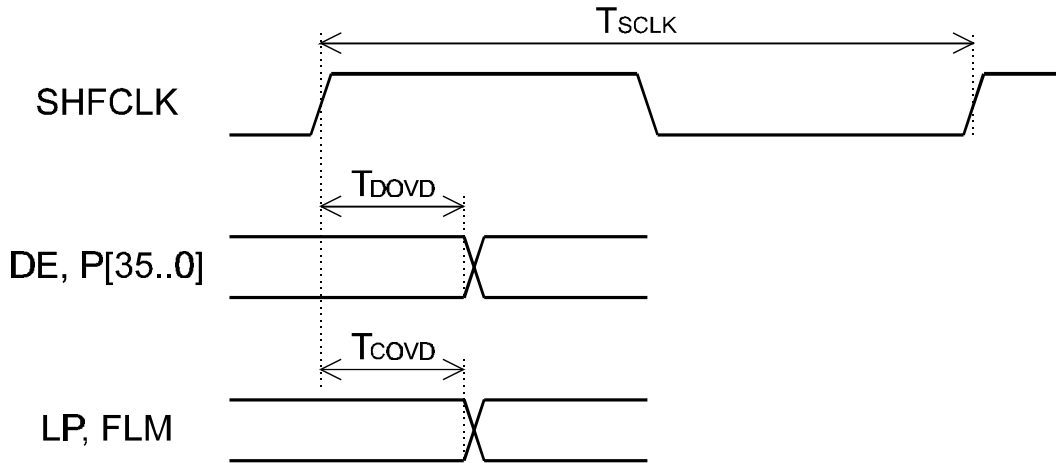


Figure 3-11: Video Data Port Timing

**Table 3-17: 65555 AC Timing Characteristics - Panel Output Timing**

Symbol	Parameter	Signaling	Min	Max	Units
T <sub>SCLK</sub>	SHFCLK cycle time	Measured at 0.4V <sub>CC</sub>	15	–	ns
T <sub>DOVD</sub>	DE and P[35..0] Output Valid Delay		-3	4	ns
T <sub>COVD</sub>	LP and FLM Output Valid Delay		-3	3	ns
	SHFCLK Duty Cycle		40	60	%

**Note:** AC Timing is valid when max output loading=25pF.



**Figure 3-12: Panel Output Timing**

**This page intentionally left blank.**

# ***CHAPTER 4***

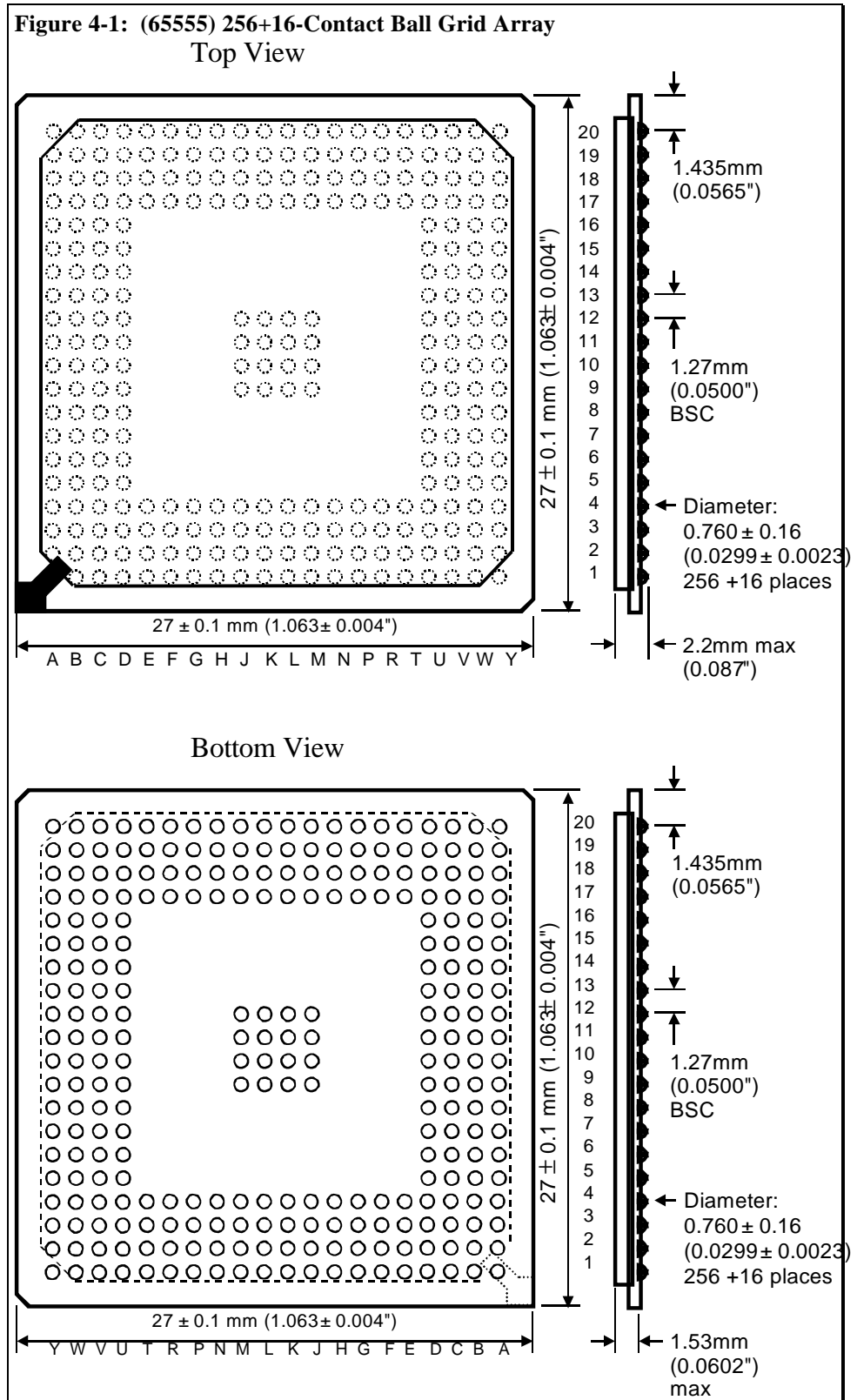
# ***MECHANICAL SPECIFICATIONS***

**This page intentionally left blank.**



# 4 MECHANICAL SPECIFICATIONS

## 256+16-Contact Ball Grid Array



**This page intentionally left blank.**

## ***CHAPTER 5***

# ***I/O AND MEMORY ADDRESS MAPS***

**This page intentionally left blank.**

## **5.0 INTRODUCTION**

An extensive set of registers normally controls the graphics system. These registers are a combination of registers defined by IBM when the Video Graphics Array (VGA) was first introduced, and others that CHIPS has added to support graphics modes that have color depths, resolutions, and hardware acceleration features that go well beyond the original VGA standard. This chapter provides an overview of the address locations and sub-addressing mechanisms used to access the various registers and the frame buffer of the 65555.

Some of the registers are directly accessible at various I/O addresses. They may be read-only or write-only, and some must be read from and written to at different I/O addresses. Most of the other registers are accessed through a sub-addressing arrangement. The index of the desired register is written to an index register, and then the desired register may be read from or written to through a data port. Almost all of these sub-addressed registers are both readable and writable. Still other registers are directly accessible at various memory addresses, and here too, almost all of these registers are both readable and writable.

Part of the VGA standard requires the VGA graphics system to take the place of either the IBM Monochrome Display and Printer Adapter (either MDPA or MDA) or the IBM Color Graphics Adapter (CGA). This was also the case with the IBM Enhanced Graphics Adapter (EGA), VGA's predecessor. The MDA has registers at I/O addresses 3B4-3B5 and 3BA, and a character buffer (not a frame buffer -- the MDA is a text-only device) within the memory address range of B0000-B7FFF. The CGA has registers within I/O addresses 3D4-3D5 and 3DA-3DC, and a frame buffer (for either text or graphics) within the memory address range of B8000-BFFFF.

If a PC with a VGA graphics system does not have either an MDA display system or a CGA graphics system, the VGA BIOS will initialize the VGA graphics system to take the place of either an MDA if a monochrome display is attached to the VGA, or of a CGA if a color display is attached. However, if a PC with a VGA graphics system also has an MDA display system, the VGA is initialized to take the place of a CGA, regardless of the type of monitor attached to the VGA in order to avoid conflicts with the MDA. Likewise, if a PC with a VGA graphics system also has a CGA graphics system, the VGA is initialized to take the place of an MDA, regardless of the type of monitor attached to the VGA. The VGA standard does not allow a system to have both an MDA display system and a CGA graphics system in the same PC along with a VGA graphics system.

The VGA standard has introduced numerous modes with features that go beyond those originally provided by either MDA or CGA. To do this, the VGA standard introduced many additional registers at locations in the 3C0-3CF I/O address range, and an additional frame buffer memory space in the A0000-AFFFF memory address range through which the frame buffer could be accessed. This additional memory address region is a 64KB "port-hole" by which the standard 256KB VGA frame buffer is accessed. Either different 64KB portions of this frame buffer are swapped or "paged" in and out of this port-hole as a way of gaining access to all of it, or this frame buffer can be reorganized into "planes" that can be made selectively or even simultaneously accessible though this port-hole as part of a mechanism to enable bit-wise graphics color manipulation. This was done as part of the VGA standard partly because of the shortage of available addresses in the first 1MB of memory address space in PC-standard systems.

The 65555 improves upon VGA by providing additional features that are used through numerous additional registers. Many of these additional registers are simply added to the sub-addressing schemes already defined in the VGA standard, while others are added through sub-addressing schemes using additional I/O address locations 3D0-3D3 and 3D6-3D7. The 65555 also supports up to 4MB of frame buffer memory -- far larger than VGA's standard complement of 256KB. As an improvement upon the VGA standard frame buffer port-hole, the 65555 also maps the entire frame buffer into part of a single contiguous memory space at a programmable location, providing what is called "linear" access to the frame buffer. The size of this memory space is 16MB, and the base address is set through a PCI configuration register.

Most aspects of the host interface of the 65555 are configured through a set of built-in PCI-compliant setup registers. The system logic accesses these registers through standard PCI configuration read and write cycles. Therefore, the exact location of the PCI configuration registers for the 65555, as well as any other PCI device in the system I/O or memory address space depends on the system logic design.

**5.1 I/O and Sub-Addressed Register Map**

<b>I/O Address</b>	<b>Memory Offset</b>	<b>Read</b>	<b>Write</b>
3B0-3B3			
3B4	0x400768	CRTC Index (MDA Emulation)	
3B5	0x400769	CRTC Data Port (MDA Emulation)	
3B6-3B9			
3BA	0x400774	Input Status Register 1 (ST01) (MDA Emulation)	Feature Control Register (FCR) (MDA Emulation)
3BB-3BF			
3C0	0x400780	Attribute Controller Index	Attribute Controller Index and Data Port
3C1	0x400781	Attribute Controller Data Port	Alternate Attribute Controller Data Port
3C2	0x400784	Input Status Register 0 (ST00)	Misc. Output Register (MSR)
3C3			
3C4	0x400788	Sequencer Index	
3C5	0x400789	Sequencer Data Port	
3C6	0x40078C	Color Palette Mask	
3C7	0x40078D	Color Palette State	Color Palette Read Mode Index
3C8	0x400790	Color Palette Write Mode Index	
3C9	0x400791	Color Palette Data Port	
3CA	0x400794	Feature Control Register (FCR)	
3CB			
3CC	0x400798	Misc. Output Register (MSR)	
3CD			
3CE	0x40079C	Graphics Controller Index	
3CF	0x40079D	Graphics Controller Data Port	
3D0	0x4007A0	Flat Panel Extensions Index	
3D1	0x4007A1	Flat Panel Extensions Data Port	
3D2	0x4007A4	Multimedia Extensions Index	
3D3	0x4007A5	Multimedia Extensions Data Port	
3D4	0x4007A8	CRTC Index (CGA Emulation)	
3D5	0x4007A9	CRTC Data Port (CGA Emulation)	
3D6	0x4007AC	Configuration Extensions Index	
3D7	0x4007AD	Configuration Extensions Data Port	
3D8-3D9			
3DA	0x4007B4	Input Status Register 1 (ST01) (CGA Emulation)	Feature Control Register (FCR) (CGA Emulation)
3DB-3DF			

## 5.2 Sub-Addressing Indexes and Data Ports

Index Port Addresses	Data Port Addresses	Register Group	Name	Function
I/O 3C0 Mem 0x400780	I/O 3C0/3C1 Mem 0x400780/781	Attribute Controller	AR0-14	VGA Attributes Control
I/O 3C4 Mem 0x400788	I/O 3C5 Mem 0x400789	Sequencer	SR0-7	VGA Sequencer Control
I/O 3CE Mem 0x40079C	I/O 3CF Mem 0x40079D	Graphics Controller	GR0-8	VGA Data Path Control
I/O 3D0 Mem 0x4007A0	I/O 3D1 Mem 0x4007A1	Flat Panel	FR00-1F FR20-2F FR30-3F FR40-4F FR48-4F FR50-5F FR60-6F FR70-7F	General Panel Control Horizontal Panel Timing Vertical Panel Timing Horizontal Compensation Vertical Compensation — — —
I/O 3D2 Mem 0x4007A4	I/O 3D3 Mem 0x4007A5	Multimedia	MR0-1F MR20-3F MR40-5F MR60-7F	Acquisition/Capture Playback Window Display Color Key —
I/O 3B4/3D4 Mem 0x400768/7A8	I/O 3B5/3D5 Mem 0x400769/7A9	CRTC	CR0-2F CR30-3F CR40-4F CR50-5F CR60-6F CR70-7F CR80-FF	Basic Display Control Timing Extension Bits Address Extension Bits Display Overlay — Interlace Control —
I/O 3D6 Mem 0x4007AC	I/O 3D7 Mem 0x4007AD	Extension Registers	XR0-F XR10-1F XR20-2F XR30-3F XR40-4F XR50-5F XR60-6F XR70-7F XR80-8F XR90-9F XRA0-AF XRB0-BF XRC0-CF XRD0-DF XRE0-EF XRF0-FF	General Configuration — Graphics Engine Configuration — Memory Configuration — Pin Control Configuration Pins Pixel Pipeline — Hardware Cursor — Clock Control Power Management Software Flags Hardware Testing



### 5.3 Lower Memory Map

Address Range	Function	Size in Bytes
A0000-AFFFF	VGA Frame Buffer	64KB
B0000-B7FFF	MDA Emulation Character Buffer	32KB
B8000-BFFFF	CGA Emulation Frame Buffer	32KB
C0000-C7FFF or C0000 up to CFFFF	VGA BIOS ROM	32KB or Larger (up to 64KB)

### 5.4 Upper Memory Map

		Size	Memory Offset	Function	
16MB	8MB	4MB	0x000000 to 0x3FFFFFF	Linear Frame Buffer (Little-Endian)	
		4MB	64 Bytes	0x400000 to 0x40003F	BitBLT Registers (Little-Endian)
			1728 Bytes	0x400040 to 0x4006FF	
			256 Bytes	0x400700 to 0x4007FF	VGA and Sub-Addressed Registers (Little-Endian)
			62KB	0x400800 to 0x40FFFF	
			64KB	0x410000 to 0x41FFFF	BitBLT Data Port (Little-Endian)
			3968KB	0x420000 to 0x7FFFFFF	
	8MB	4MB	0x800000 to 0xBFFFFFF	Linear Frame Buffer (Big-Endian)	
		4MB	64KB	0xC00000 to 0xC0FFFF	
			64KB	0xC10000 to 0xC1FFFF	BitBLT Data Port (Big-Endian)
			3968KB	0xC20000 to 0xFFFFFFF	

### 5.5 PCI Configuration Registers

Configuration Space Offset	Name	Function	Access	Bits
00	VENDID	Vendor ID	Read-Only	16
02	DEVID	Device ID	Read-Only	16
04	DEVCTL	Device Control	Read/Clear	16
06	DEVSTAT	Device Status	Read-Only	16
08	REV	Revision ID	Read-Only	8
09	PRG	Programming Interface	Read-Only	8
0A	SUB	Sub-Class Code	Read-Only	8
0B	BASE	Base Class Code	Read-Only	8
0C		Reserved (Cache Line Size)	—	8
0D		Reserved (Latency Timer)	—	8
0E		Reserved (Header Type)	—	8
0F		Reserved (Built-In-Self-Test)	—	8
10	MBASE	Memory Base Address	Read/Write	32
14		Reserved (Base Address)	—	32
18		Reserved (Base Address)	—	32
1C		Reserved (Base Address)	—	32
20		Reserved (Base Address)	—	32
24		Reserved (Base Address)	—	32
28		Reserved	—	32
2C	SUBVENDID	Subsystem Vendor ID	Read-Only	16
2E	SUBDEVID	Subsystem Device ID	Read-Only	16
30	RBASE	ROM Base Address	Read/Write	32
34		Reserved	—	32
38		Reserved	—	32
3C		Reserved (Interrupt Line)	—	8
3D		Reserved (Interrupt Pin)	—	8
3E		Reserved (Minimum Grant)	—	8
3F		Reserved (Maximum Latency)	—	8
40 to 6B				
6C	SUBVENDSET	Subsystem Vendor ID Set (Writable)	Read/Write	16
6E	SUBDEVSET	Subsystem Device ID Set (Writable)	Read/Write	16
6F to FF				

## ***CHAPTER 6***

# ***REGISTER SUMMARIES***

**This page intentionally left blank.**

## 6.0 REGISTER SUMMARIES

### 6.1 General Control & Status Registers

Name	Function	Read	Write
ST00	VGA Input Status Register 0	3C2	—
ST01	VGA Input Status Register 1	3BA/3DA	—
FCR	VGA Feature Control Register	3CA	3BA/3DA
MSR	VGA Miscellaneous Output Register	3CC	3C2

### 6.2 Sequencer Registers

Name	Function	Access (via 3C5)	Index Value In 3C4 (SRX)
SR00	Reset Register	Read/Write	00
SR01	Clocking Mode Register	Read/Write	01
SR02	Plane Mask Register	Read/Write	02
SR03	Character Map Select Register	Read/Write	03
SR04	Memory Mode Register	Read/Write	04
SR07	Horizontal Character Counter Reset	Read/Write	07

### 6.3 Graphics Controller Registers

Name	Function	Access (via 3CF)	Index Value In 3CE (GRX)
GR00	Set/Reset Register	Read/Write	00h
GR01	Enable Set/Reset Register	Read/Write	01h
GR02	Color Compare Register	Read/Write	02h
GR03	Data Rotate Register	Read/Write	03h
GR04	Read Map Select Register	Read/Write	04h
GR05	Graphics Mode Register	Read/Write	05h
GR06	Miscellaneous Register	Read/Write	06h
GR07	Color Don't Care Register	Read/Write	07h
GR08	Bit Mask Register	Read/Write	08h

## 6.4 Attribute Controller Registers

Name	Function	Access	Index
AR00-AR0F	Color Data Registers	Read/Write	00-0F
AR10	Mode Control Register	Read/Write	10
AR11	Overscan Color Register	Read/Write	11
AR12	Memory Plane Enable Register	Read/Write	12
AR13	Horizontal Pixel Panning Register	Read/Write	13
AR14	Color Select Register	Read/Write	14

## 6.5 VGA Color Palette Registers

Name	Function	Access	I/O Address
DACMASK	Pixel Data Mask Register	Read/Write	3C6h
DACSTATE	DAC State Register	Read-Only	3C7h
DACRX	Palette Read Index Register	Write-Only	3C7h
DACWX	Palette Write Index Register	Read/Write	3C8h
DACDATA	Palette Data Register	Read/Write	3C9h

## 6.6 CRT Controller Registers

Name	Function	Access 3B5/3D5	Index Value 3B4/3D4 (CRX)
CR00	Horizontal Total Register	Read/Write	00h
CR01	Horizontal Display Enable End Register	Read/Write	01h
CR02	Horizontal Blanking Start Register	Read/Write	02h
CR03	Horizontal Blanking End Register	Read/Write	03h
CR04	Horizontal Sync Start Register	Read/Write	04h
CR05	Horizontal Sync End Register	Read/Write	05h
CR06	Vertical Total Register	Read/Write	06h
CR07	Overflow Register	Read/Write	07h
CR08	Preset Row Scan Register	Read/Write	08h
CR09	Maximum Scan Line Register	Read/Write	09h
CR0A	Text Cursor Start Scan Line Register	Read/Write	0Ah
CR0B	Text Cursor End Scan Line Register	Read/Write	0Bh
CR0C	Start Address High Register	Read/Write	0Ch
CR0D	Start Address Low Register	Read/Write	0Dh
CR0E	Text Cursor Location High Register	Read/Write	0Eh
CR0F	Text Cursor Location Low Register	Read/Write	0Fh
CR10	Vertical Sync Start Register	Read/Write	10h
CR11	Vertical Sync End Register	Read/Write	11h
CR12	Vertical Display Enable End Register	Read/Write	12h
CR13	Offset Register	Read/Write	13h
CR14	Underline Row Register	Read/Write	14h
CR15	Vertical Blanking Start Register	Read/Write	15h
CR16	Vertical Blanking End Register	Read/Write	16h
CR17	CRT Mode Control Register	Read/Write	17h
CR18	Line Compare Register	Read/Write	18h
CR22	Memory Read Latches Register	Read-Only	22h
CR30	Extended Vertical Total Register	Read/Write	30h
CR31	Extended Vertical Display Enable End Reg	Read/Write	31h
CR32	Extended Vertical Sync Start Register	Read/Write	32h
CR33	Extended Vertical Blanking Start Register	Read/Write	33h
CR40	Extended Start Address Register	Read/Write	40h
CR41	Extended Offset Register	Read/Write	41h
CR70	Interlace Control Register	Read/Write	70h
CR71	NTSC/PAL Video Output Control Register	Read/Write	71h
CR72	NTSC/PAL Horizontal Serration 1 Start Reg	Read/Write	72h
CR73	NTSC/PAL Horizontal Serration 2 Start Reg	Read/Write	73h
CR74	NTSC/PAL Horizontal Pulse Width Register	Read/Write	74h
CR75	NTSC/PAL Filtering Burst Read Length Register	Read/Write	75h
CR76	NTSC/PAL Filtering Burst Read Quantity Register	Read/Write	76h
CR77	NTSC/PAL Filtering Control Register	Read/Write	77h

**Note:** CR00-CR22 are standard VGA registers -- CR30-CR77 are CHIPS extensions.

## 6.7 PCI Configuration Registers

Name	Function	Access	Offset
VENDID	Vendor ID	Read-Only	00
DEVID	Device ID	Read-Only	02
DEVCTL	Device Control	Read/Write	04
DEVSTAT	Device Status	Read-Only	06
REV	Revision ID	Read-Only	08
PRG	Programming Interface	Read-Only	09
SUB	Sub-Class Code	Read-Only	0A
BASE	Base Class Code	Read-Only	0B
	Reserved (Cache Line Size)	—	0C
	Reserved (Latency Timer)	—	0D
HDR	Reserved (Header Type)	Read-Only	0E
	Reserved (Built-In-Self-Test)	—	0F
MBASE	Memory Base Address	Read/Write	10
	Reserved (Base Address)	—	14
	Reserved (Base Address)	—	18
	Reserved (Base Address)	—	1C
	Reserved (Base Address)	—	20
	Reserved (Base Address)	—	24
	Reserved	—	28
SUBVENDID	Subsystem Vendor ID	Read-Only	2C
SUBDEVID	Subsystem Device ID	Read-Only	2E
RBASE	ROM Base Address	Read/Write	30
	Reserved	—	34
	Reserved	—	38
	Reserved (Interrupt Line)	—	3C
	Reserved (Interrupt Pin)	—	3D
	Reserved (Minimum Grant)	—	3E
	Reserved (Maximum Latency)	—	3F
	Reserved	—	40 to FF



## 6.8 BitBLT Registers

Name	Function	Access	Offset
BR00	Source and Destination Offset Register	Read/Write	0x400000
BR01	Pat/Src Expansion Background Color Reg.	Read/Write	0x400004
BR02	Pat/Src Expansion Foreground Color Reg.	Read/Write	0x400008
BR03	Monochrome Source Control Register	Read/Write	0x40000C
BR04	BitBLT Control Register	Read/Write	0x400010
BR05	Pattern Address Register	Read/Write	0x400014
BR06	Source Address Register	Read/Write	0x400018
BR07	Destination Address Register	Read/Write	0x40001C
BR08	Destination Width & Height Register	Read/Write	0x400020
BR09	Source Expansion Background Color Reg.	Read/Write	0x400024
BR0A	Source Expansion Foreground Color Reg.	Read/Write	0x400028

**6.9 Extension Registers**

<b>Name</b>	<b>Function</b>	<b>Access Port 3D7</b>	<b>Index Value Port 3D6 (XRX)</b>
XR00	Vendor ID Low	Read-Only	00h
XR01	Vendor ID High	Read-Only	01h
XR02	Device ID Low	Read-Only	02h
XR03	Device ID High	Read-Only	03h
XR04	Revision ID	Read-Only	04h
XR05	Linear Base Address Low Register	Read-Only	05h
XR06	Linear Base Address High Register	Read-Only	06h
XR08	Host Bus Configuration Register	Read-Only	08h
XR09	I/O Control Register	Read/Write	09h
XR0A	Frame Buffer Mapping Register	Read/Write	0Ah
XR0B	PCI Burst Write Support Register	Read/Write	0Bh
XR0E	Frame Buffer Page Select Register	Read/Write	0Eh
XR10	Subsystem Vendor ID Low	Read/Write	10h
XR11	Subsystem Vendor ID High	Read/Write	11h
XR12	Subsystem Vendor Device ID Low	Read/Write	12h
XR13	Subsystem Vendor Device ID High	Read/Write	13h
XR20	BitBLT Configuration Register	Read/Write	20h
XR40	DRAM Access Control Register	Read/Write	40h
XR41	DRAM Type Register	Read/Write	41h
XR42	EDO DRAM Configuration Register	Read/Write	42h
XR43	DRAM Interface Register	Read/Write	43h
XR44	EDO DRAM Timing Register	Read/Write	44h
XR60	Video Pin Control Register	Read/Write	60h
XR61	DPMS Sync Control Register	Read/write	61h
XR62	GPIO Pin Control Register	Read/Write	62h
XR63	GPIO Pin Data Register	Read/Write	63h
XR67	Pin Tri-State Control Register	Read/Write	67h
XR70	Configuration Pins 0 Register	Read-Only	70h
XR71	Configuration Pins 1 Register	Read-Only	71h
XR80	Pixel Pipeline Configuration 0 Register	Read/Write	80h
XR81	Pixel Pipeline Configuration 1 Register	Read/Write	81h
XR82	Pixel Pipeline Configuration 2 Register	Read/Write	82h

**6.9 Extension Registers (continued)**

Name	Function	Access Port 3D7	Index Value Port 3D6 (XRX)
XRA0	Cursor 1 Control Register	Read/Write	A0h
XRA1	Cursor 1 Vertical Extension Register	Read/Write	A1h
XRA2	Cursor 1 Base Address Low Register	Read/Write	A2h
XRA3	Cursor 1 Base Address High Register	Read/Write	A3h
XRA4	Cursor 1 X-Position Low Register	Read/Write	A4h
XRA5	Cursor 1 X-Position High Register	Read/Write	A5h
XRA6	Cursor 1 Y-Position Low Register	Read/Write	A6h
XRA7	Cursor 1 Y-Position High Register	Read/Write	A7h
XRA8	Cursor 2 Control Register	Read/Write	A8h
XRA9	Cursor 2 Vertical Extension Register	Read/Write	A9h
XRAA	Cursor 2 Base Address Low Register	Read/Write	AAh
XRAB	Cursor 2 Base Address High Register	Read/Write	ABh
XRAC	Cursor 2 X-Position Low Register	Read/Write	ACH
XRAD	Cursor 2 X-Position High Register	Read/Write	ADh
XRAE	Cursor 2 Y-Position Low Register	Read/Write	Aeh
XRAF	Cursor 2 Y-Position High Register	Read/Write	Afh
XRC0	Dot Clock 0 VCO M-Divisor Low Register	Read/Write	C0h
XRC1	Dot Clock 0 VCO N-Divisor Low Register	Read/Write	C1h
XRC2	Dot Clock 0 VCO M/N-Divisor High Register	Read/Write	C2h
XRC3	Dot Clock 0 Divisor Select Register	Read/Write	C3h
XRC4	Dot Clock 1 VCO M-Divisor Low Register	Read/Write	C4h
XRC5	Dot Clock 1 VCO N-Divisor Low Register	Read/Write	C5h
XRC6	Dot Clock 1 VCO M/N-Divisor High Register	Read/Write	C6h
XRC7	Dot Clock 1 Divisor Select Register	Read/Write	C7h
XRC8	Dot Clock 2 VCO M-Divisor Low Register	Read/Write	C8h
XRC9	Dot Clock 2 VCO N-Divisor Low Register	Read/Write	C9h
XRCA	Dot Clock 2 VCO M/N-Divisor High Register	Read/Write	CAh
XRCB	Dot Clock 2 Divisor Select Register	Read/Write	CBh
XRCC	Memory Clock VCO M-Divisor Register	Read/Write	CCh
XRCD	Memory Clock VCO N-Divisor Register	Read/Write	CDh
XRCE	Memory Clock VCO Divisor Select Register	Read/Write	CEh
XRCF	Clock Configuration Register	Read/Write	CFh
XRD0	Powerdown Control Register	Read/Write	D0h
XRD1	Power Conservation Control Register	Read/Write	D1h
XRD2	2KHz Down Counter Register	Read-Only	D2h
XRE0-XREB	Software Flag Registers	Read/Write	E0h-E9h
XRF8-XRFC	Test Registers	Read/Write	F8h-FCh

**6.10 Multimedia Registers**

<b>Name</b>	<b>Function</b>	<b>Access</b>	<b>Offset</b>
MR00	Module Capability	Read Only	
MR01	Secondary Capability	Read Only	
MR02	Capture Control 1	Read / Write	
MR03	Capture Control 2	Read / Write	
MR04	Capture Control 3	Read / Write	
MR05	Capture Control 4	Read / Write	
MR06	Capture Memory Address PTR1	Read / Write	
MR07	Capture Memory Address PTR1	Read / Write	
MR08	Capture Memory Address PTR1	Read / Write	
MR09	Capture Memory Address PTR2	Read / Write	
MR0A	Capture Memory Address PTR2	Read / Write	
MR0B	Capture Memory Address PTR2	Read / Write	
MR0C	Capture Memory Line Storage Width	Read / Write	
MR0E	Capture Window X-LEFT	Read / Write	
MR0F	Capture Window X-LEFT	Read / Write	
MR10	Capture Window X-RIGHT	Read / Write	
MR11	Capture Window X-RIGHT	Read / Write	
MR12	Capture Window Y-TOP	Read / Write	
MR13	Capture Window Y-TOP	Read / Write	
MR14	Capture Window Y-BOTTOM	Read / Write	
MR15	Capture Window Y-BOTTOM	Read / Write	
MR16	H-SCALE	Read / Write	
MR17	V-SCALE	Read / Write	
MR18	Capture Frame/Field Drop Count	Read / Write	
MR1E	Playback Control 1	Read / Write	
MR1F	Playback Control 2	Read / Write	
MR20	Playback Control 3	Read / Write	
MR21	Double Buffer Status	Read Only	
MR22	Playback Window Address PTR1	Read / Write	
MR23	Playback Window Address PTR1	Read / Write	
MR24	Playback Window Address PTR1	Read / Write	
MR25	Playback Window Address PTR2	Read / Write	
MR26	Playback Window Address PTR2	Read / Write	
MR27	Playback Window Address PTR2	Read / Write	
MR28	Playback Line Memory Fetch Width	Read / Write	

## 6.10 Multimedia Registers (continued)

Name	Function	Access	Offset
MR2A	Playback Window X-LEFT	Read / Write	
MR2B	Playback Window X-LEFT	Read / Write	
MR2C	Playback Window X-RIGHT	Read / Write	
MR2D	Playback Window X-RIGHT	Read / Write	
MR2E	Playback Window Y-TOP	Read / Write	
MR2F	Playback Window Y-TOP	Read / Write	
MR30	Playback Window Y-BOTTOM	Read / Write	
MR31	Playback Window Y-BOTTOM	Read / Write	
MR32	H-ZOOM	Read / Write	
MR33	V-ZOOM	Read / Write	
MR34	Playback Line Display Width	Read / Write	
MR3C	Color Key Control	Read / Write	
MR3D	Red Color Keys	Read / Write	
MR3E	Green Color Key	Read / Write	
MR3F	<b>Blue</b> Color Key	Read / Write	
MR40	Red Color Key Mask	Read / Write	
MR41	Green Color Key Mask	Read / Write	
MR42:	Blue Color Key Mask	Read / Write	
MR43	Line Count	Read Only	
MR44	Line Count	Read Only	

## 6.11 Flat Panel Registers

Name	Function	Access	Offset
FR00	Feature Register	3D1	R/W
FR01	CRT / FP Control	3D1	R/W
FR02	Mode Control	3D1	R/W
FR03	Dot Clock Source	3D1	R/W
FR04	Panel Power Sequencing Delay	3D1	R/W
FR05	Power Down Control 1	3D1	R/W
FR06	FP Power Down Control	3D1	R/W
FR08	Pin Polarity	3D1	R/W
FR0A	Programmable Output Drive	3D1	R/W
FR0B	Pin Control 1	3D1	R/W
FR0C	Pin Control 2	3D1	R/W
FR0F	Activity Timer Control	3D1	R/W
FR10	Panel Format 0	3D1	R/W
FR11	Panel Format 1	3D1	R/W
FR12	Panel Format 2	3D1	R/W
FR13	Panel Format 3	3D1	R/W
FR16	FRC Option Select	3D1	R/W
FR17	Polynomial FRC Control	3D1	R/W
FR18	Text Mode Control	3D1	R/W
FR19	Blink Rate Control	3D1	R/W
FR1A	Frame Buffer Control	3D1	R/W
FR1E	M (ACDCLK) Control	3D1	R/W
FR1F	Diagnostic	3D1	R/W
FR20	Horizontal Panel Size (LSB)	3D1	R/W
FR21	Horizontal Sync Start (LSB)	3D1	R/W
FR22	Horizontal Sync End	3D1	R/W
FR23	Horizontal Total (LSB)	3D1	R/W
FR24	FP Hsync (LP) Delay (LSB)	3D1	R/W
FR25	Horizontal Overflow 1	3D1	R/W
FR26	Horizontal Overflow 2	3D1	R/W
FR27	FP Hsync (LP) Width and Disable	3D1	R/W
FR30	Vertical Panel Size (LSB)	3D1	R/W

**6.11 Flat Panel Registers (continued)**

Name	Function	Access	Offset
FR31	Vertical Sync Start (LSB)	3D1	R/W
FR32	Vertical Sync End	3D1	R/W
FR33	Vertical Total (LSB)	3D1	R/W
FR34	FP Vsync (FLM) Delay (LSB)	3D1	R/W
FR35	Vertical Overflow 1	3D1	R/W
FR36	Vertical Overflow 2	3D1	R/W
FR37	FP Vsync (FLM) Disable	3D1	R/W
FR40	Horizontal Compensation Register	3D1	R/W
FR41	Vertical Compensation	3D1	R/W
FR48	Vertical Compensation	3D1	R/W
FR49	Text Mode Vertical Stretching 0 MSB	3D1	R/W
FR4A	Text Mode Vertical Stretching 0 LSB	3D1	R/W
FR4B	Text Mode Vertical Stretching 1 MSB	3D1	R/W
FR4C	Text Mode Vertical Stretching 1 LSB	3D1	R/W
FR4D	Vertical Line Replication	3D1	R/W
FR4E	Selective Vertical Stretching Disable	3D1	R/W
FR70	TMED Red Seed Register	3D1	R/W
FR71	TMED Green Seed Register	3D1	R/W
FR72	TMED Blue Seed Register	3D1	R/W
FR73	TMED Control Register	3D1	R/W

**This page intentionally left blank.**



## ***CHAPTER 7***

# ***PCI CONFIGURATION REGISTERS***

**This page intentionally left blank.**

## 7.0 PCI CONFIGURATION REGISTERS

Name	Function	Access	Offset
VENDID	Vendor ID	Read-Only	00
DEVID	Device ID	Read-Only	02
DEVCTL	Device Control	Read/Write	04
DEVSTAT	Device Status	Read-Only	06
REV	Revision ID	Read-Only	08
PRG	Programming Interface	Read-Only	09
SUB	Sub-Class Code	Read-Only	0A
BASE	Base Class Code	Read-Only	0B
	Reserved (Cache Line Size)	—	0C
	Reserved (Latency Timer)	—	0D
HDR	Reserved (Header Type)	Read-Only	0E
	Reserved (Built-In-Self-Test)	—	0F
MBASE	Memory Base Address	Read/Write	10
	Reserved (Base Address)	—	14
	Reserved (Base Address)	—	18
	Reserved (Base Address)	—	1C
	Reserved (Base Address)	—	20
	Reserved (Base Address)	—	24
	Reserved	—	28
SUBVENDID	Subsystem Vendor ID	Read-Only	2C
SUBDEVID	Subsystem Device ID	Read-Only	2E
RBASE	ROM Base Address	Read/Write	30
	Reserved	—	34
	Reserved	—	38
	Reserved (Interrupt Line)	—	3C
	Reserved (Interrupt Pin)	—	3D
	Reserved (Minimum Grant)	—	3E
	Reserved (Maximum Latency)	—	3F
	Reserved	—	40 to FF

**Note:** The mechanism used to generate the PCI configuration read and configuration write cycles used to access the configuration registers of PCI devices is system-dependent.

## VENDID Vendor ID

Read-Only at PCI configuration offset 00h

Byte or word accessible

Accessible only via PCI configuration cycles

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Vendor ID (102Ch)															

### 15-0 Vendor ID

This is the vendor ID assigned to CHIPS by the PCI Special Interest group. This register always returns the 16-bit value 102Ch (4140 decimal).

## DEVID Device ID

Read-Only at PCI configuration offset 02h

Byte or word accessible

Accessible only via PCI configuration cycles

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Device ID (00E5h)															

### 15-0 Device ID

This is the device ID assigned to the 65555 by CHIPS. This register always returns the 16-bit value 00E5h.

## DEVCTL Device Control

Read/Write at PCI configuration offset 04h

Byte or word accessible

Accessible only via PCI configuration cycles

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						Fast Bk-Bk	SERR Enbl	Wait Cycl Ctl	PERR Enbl	VGA Pal Snoop	Mem Wrt / Inval.	Spec Cycl	Bus Mstr	Mem Acc	I/O Acc
(0000:00)						(0)	(0)	(1)	(0)	(0)	(0)	(0)	(0)	(0)	(0)

### 15-10 Reserved

Each of these bits always return a value of 0 when read.

### 9 Fast Back-to-Back Enable for Masters

This bit applies only to PCI Bus masters. Since the 65555 never functions as a PCI Bus master, this bit always returns a value of 0 when read.

### 8 SERR# Enable

- 0 Disables the use of SERR# and the setting of bit 14 (Signaled System Error bit) in the Device Status register (DEVSTAT) to 1 as a response to an address parity error. This is the default after reset.
- 1 Enables the use of SERR# and the setting of bit 14 (Signaled System Error bit) in the Device Status register (DEVSTAT) to 1 as a response to an address parity error.

### 7 Wait Cycle Control

This bit controls enables and disables address stepping. Since the 65555 always supports address stepping, this bit always returns a value of 1 when read.

### 6 Parity Error Response

- 0 Disables the use of PERR# as a response to detecting either data or address parity errors. Disables the setting of bit 14 (Signaled System Error bit) in the Device Status register (DEVSTAT) to 1 as a response to an address parity error. This is the default after reset.
- 1 Enables the use of PERR# as a response to detecting either data or address parity errors. Enables the setting of bit 14 (Signaled System Error bit) in the Device Status register (DEVSTAT) to 1 as a response to an address parity error.

**Note:** Bit 8 (SERR# Enable) of this register must also be set to 1 to enable the use of SERR# and the setting of bit 14 (Signaled System Error bit) in the Device Status register (DEVSTAT) to 1 as a response to an address parity error.

**DEVCTL Device Control (continued)****5 VGA Palette Snoop**

- 0 Accesses to all VGA I/O locations, including those for the palette, will be claimed. All read and write accesses to the palette will be performed, normally. This is the default after reset.
- 1 Accesses to all VGA I/O locations, except for those for the palette, will be claimed. All reads will be entirely ignored, but all writes will still update the palette. This permits accesses to the palette I/O addresses to be answered by other devices that need to be able to snoop accesses to the palette.

**4 Memory Write & Invalidate**

This bit applies only to PCI Bus masters. Since the 65555 never functions as a PCI Bus master, this bit always returns a value of 0 when read.

**3 Special Cycles**

The 65555 always ignores all special cycles, therefore, this bit always returns the value of 0 when read.

**2 Bus Master**

The 65555 never functions as a PCI Bus master, therefore, this bit always returns a value of 0 when read.

**1 Memory Access Enable**

- 0 Disables access to the frame buffer memory locations within the range specified by the MBASE register. This is the default after reset.
- 1 Enables access to the frame buffer memory locations within the range specified by the MBASE register.

**Note:** Accesses with only adjacent active byte enables are supported.

**0 I/O Access Enable**

- 0 Disables I/O port accesses. This is the default after reset.
- 1 Enables I/O port accesses.

**Note:** Accesses with only adjacent active byte enables are supported.

## DEVSTAT Device Status

Read/Clear at PCI configuration offset 06h

Byte or word accessible

Accessible only via PCI configuration cycles

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Det Parity Error (0)	Signal Sys. Error (0)	Rcvd Mstr Abort (0)	Rcvd Target Abort (0)	Signal Target Abort (0)	DEVSEL# Timing (01)		Data Parity Error (0)	Fast Back- Back (1)	UDF (0)	66 MHz (0)					
											Reserved (0:0000)				

**Important:** Read accesses to this register behave normally. Writes, however, behave differently in that bits can be reset to 0, but not set to 1. A bit in this register is reset to 0 whenever it is written with the value of 1. Bits written with a value of 0 are entirely unaffected.

### 15 Detected Parity Error

- 0 No address or data parity error detected.
- 1 An address or data parity error was detected.

**Note:** This bit is set in response to a parity error regardless of the settings of either bit 6 (Parity Error Response bit) and 8 (SERR# Enable) of the Device Control register (DEVCTL).

### 14 Signaled System Error

- 0 SERR# has not been asserted.
- 1 SERR# has been asserted.

**Note:** Both bits 6 (Parity Error Response bit) and 8 (SERR# Enable) of the Device Control register (DEVCTL) must both be set to 1 to enable the use of SERR# and the setting of this bit to 1 in response to an address parity error.

### 13 Received Master Abort

This bit applies only to PCI Bus masters. Since the 65555 never functions as a PCI Bus master, this bit always returns a value of 0 when read.

### 12 Received Target Abort

This bit applies only to PCI Bus masters. Since the 65555 never functions as a PCI Bus master, this bit always returns a value of 0 when read.

### 11 Signaled Target Abort

- 0 A target abort was not generated.
- 1 A target abort was generated.

A target abort can be generated by the 65555 on I/O cycles with non-adjacent active byte enables.

**DEVSTAT Device Status (continued)****10-9 DEVSEL# Timing**

These two bits specify the longest-possible amount of time that the 65555 will take in decoding an address and asserting DEVSEL#. These two bits always return a value of 01, indicating a medium-length timing.

**8 Data Parity Error Detected**

This bit applies only to PCI Bus masters. Since the 65555 never functions as a PCI Bus master, this bit always returns a value of 0 when read.

**7 Fast Back-to-Back Capable**

This bit always returns a value of 1 when read, indicating that the 65555 is capable of fast back-to-back transactions that are not in the same segment.

**6 UDF Supported**

This bit always returns a value of 0 when read, indicating that the 65555 does not provide features that are definable by the end-user.

**5 66MHz Capable**

This bit always returns a value of 0 when read, indicating that the 65555 can support a maximum PCI Bus speed of 33MHz, not 66MHz.

**4-0 Reserved**

Each of these bits always return a value of 0 when read.



## REV Revision

Read-Only at PCI configuration offset 08h

Byte accessible

Accessible only via PCI configuration cycles

7	6	5	4	3	2	1	0
Chip Manufacturing Code (xxxx)				Chip Revision Code (xxxx)			

**Note:** This register is identical to the Revision ID Register (XR04).

### 7-4 Chip Manufacturing Code

These four bits carry the fabrication code.

### 3-0 Chip Revision Code

These four bits carry the revision code. Revision codes start at 0 and are incremented for each new silicon revision.

## PRG Register-Level Programming Interface

Read-Only at PCI configuration offset 09h

Byte accessible

Accessible only via PCI configuration cycles

7	6	5	4	3	2	1	0
Register-Level Programming Interface (00h)							

### 7-0 Register-Level Programming Interface

This register always returns a value of 00h to identify this PCI device as a display controller with a VGA-compatible programming interface (as opposed to 01h, which would indicate a display controller with a 8514/A-compatible programming interface).

## **SUB Sub-Class Code**

Read-Only at PCI configuration offset 0Ah

Byte accessible

Accessible only via PCI configuration cycles

7	6	5	4	3	2	1	0
Sub-Class Code (00h)							

### **7-0 Sub-Class Code**

This register always returns a value of 00h to identify this PCI device as a display controller of the VGA or 8514/A type.

## **BASE Base Class Code**

Read-Only at PCI configuration offset 0Bh

Byte accessible

Accessible only via PCI configuration cycles

7	6	5	4	3	2	1	0
Base Class Code (03h)							

### **7-0 Base-Class Code**

This register always returns a value of 03h to identify this PCI device as a display controller.

## HDR Header Type

Read-Only at PCI configuration offset 0Eh

Byte accessible

Accessible only via PCI configuration cycles

7	6	5	4	3	2	1	0
Single/Multi Function Dev (0)	Reserved (000:0000)						

### 7 Single/Multiple Function Device

This bit always returns a value of 0 when read, indicating that this PCI device is a single-function device, not a multi-function device.

### 6-0 Reserved

Each of these bits always return a value of 0 when read.

## MBASE Memory Base Address

Read/Write at PCI configuration offset 10h

Byte, word, or doubleword accessible

Accessible only via PCI configuration cycles

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Memory Space Base Address								Memory Space Size							
(0000:0000)								(0000:0000)							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Memory Space Size												Pref.	Memory Type	M or I/O	
(0000:0000:0000)												(0)	(00)	(0)	

### 31-24 Memory Space Base Address

These 8 bits select the base address for this 16MB memory space used by the 65555 for the memory mapped registers and linear accesses to the frame buffer.

### 23-4 Memory Space Size

These 20 bits always return 0 to indicate that the size of this memory space is 16MB.

### 3 Prefetchable

This bit always returns a value of 0 when read, indicating that the data in this 16MB memory space should not be prefetched by the CPU.

### 2-1 Memory Type

These 2 bits always return values of 0 when read, indicating that this 16MB memory space may be placed anywhere in the system's 32-bit address space by the system's PCI configuration software.

### 0 Memory/IO Space Indicator

This bit always returns a value of 0 when read, indicating that this is a memory space, not an I/O space.

## RBASE ROM Base Address

Read/Write at PCI configuration offset 30h

Byte, word, or doubleword accessible

Accessible only via PCI configuration cycles

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ROM Space Base Address														ROM Space Size	
(0000:0000:0000:00)														(00)	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ROM Space Size															Addr Enbl
(0000:0000:0000:000)															(0)

### 31-18 ROM Space Base Address

These 14 bits select the base address for this 256KB ROM space used by the 65555 for the video BIOS ROM.

### 17-1 ROM Space Size

These 17 bits always return 0 to indicate that the size of this ROM space is 256KB.

### 0 Address Decode Enable

- 0 Disable access to the video BIOS ROM. This is the default after reset.
- 1 Enable access to the video BIOS ROM.

**Note:** Bit 1 (the Memory Access Enable bit) of the Device Control register (DEVCTL) must also be set to 1 for the video BIOS ROM to be accessible. Also, the ROM address space must not be programmed to a range that overlaps the area specified by the MBASE register.

This page intentionally left blank

## ***CHAPTER 8***

# ***GENERAL CONTROL AND STATUS REGISTER DESCRIPTIONS***

**This page intentionally left blank.**



## 8.0 GENERAL CONTROL AND STATUS REGISTERS

The setup, enable and general registers are all directly accessible by the CPU. A sub indexing scheme is not used to read from and write to these registers.

Name	Function	Read	Write
ST00	VGA Input Status Register 0	3C2	—
ST01	VGA Input Status Register 1	3BA/3DA	—
FCR	VGA Feature Control Register	3CA	3BA/3DA
MSR	VGA Miscellaneous Output Register	3CC	3C2

Various bits in these registers provide control over the real-time status of the horizontal sync signal, the horizontal retrace interval, the vertical sync signal, and the vertical retrace interval.

The horizontal retrace interval is the period during the drawing of each scan line containing active video data, when the active video data is not being displayed. This period includes the horizontal front and back porches, and the horizontal sync pulse. The horizontal retrace interval is always longer than the horizontal sync pulse.

The vertical retrace interval is the period during which the scan lines not containing active video data are drawn. It is the period that includes the vertical front and back porches, and the vertical sync pulse. The vertical retrace interval is always longer than the vertical sync pulse.

Display Enable is a status bit (bit 0) in VGA Input Status Register 1 (ST01) that indicates when either a horizontal retrace interval or a vertical retrace interval is taking place. In the IBM EGA graphics system (and the ones that preceded it, including MDA and CGA), it was important to check the status of this bit to ensure that one or the other of the retrace intervals was taking place before reading from or writing to the frame buffer. In these earlier systems, reading from or writing to frame buffer at times outside the retrace intervals meant that the CRT controller would be denied access to the frame buffer in while accessing pixel data needed to draw pixels on the display. This resulted in either “snow” or a flickering display.

## ST00 Input Status Register 0

Read-only at I/O Address 3C2h

7	6	5	4	3	2	1	0
CRT Interrupt	Reserved		DAC Sense	Reserved			

### 7 CRT Interrupt

0 Indicates that a CRT (vertical retrace interval) interrupt is not pending.

1 Indicates that a CRT (vertical retrace interval) interrupt is pending.

**Note:** Bits 4 and 5 of the Vertical Sync End Register (CR11) can be programmed to “generate” an “interrupt” at the start of the vertical retrace interval. However, no actual interrupt is output to the host system. This is a “phantom” interrupt in that there are bits in various registers to enable/disable it, and to monitor when it has been triggered, but no actual interrupt is ever output to the host as a result. These bits exist only to provide EGA compatibility for older software.

### 6-5 Reserved

### 4 DAC Comparator Sense

Indicates the state of the output of the DAC analog output comparator(s). The BIOS uses this bit to determine whether the display is a color or monochrome CRT. BIOS will blank the screen or clear the frame buffer to display only black. Next, the BIOS will configure the D-to-A converters and the comparators to test for the presence of a color display. Finally, if the BIOS does not detect any colors, it will test for the presence of a display. The result of each such test is read via this bit.

### 3-0 Reserved

## ST01 Input Status Register 1

Read-only at I/O Address 3BAh/3DAh

7	6	5	4	3	2	1	0
VSYNC Output	Reserved	Video Feedback 1,0		Vertical Retrace	Reserved		Display Enable

### 7 VSYNC Output

0 The VSYNC output pin is currently inactive.

1 The VSYNC output pin is currently active.

**Note:** This bit is largely unused by current software.

### 6 Reserved

### 5-4 Video Feedback 1, 0

These are diagnostic video bits that are programmably connected to 2 of the 8 color bits sent to the palette. Bits 4 and 5 of the Memory Plane Enable Register (AR12) selects which two of the 8 possible color bits become connected to these 2 bits of this register. The current software normally does not use these 2 bits. They exist for EGA compatibility.

### 3 Vertical Retrace

0 Indicates that a vertical retrace interval is not taking place.

1 Indicates that a vertical retrace interval is taking place.

**Note:** Bits 4 and 5 of the Vertical Sync End Register (CR11) can be programmed to “generate” an “interrupt” at the start of the vertical retrace interval. However, no actual interrupt is output to the host system. This is a “phantom” interrupt in that there are bits in various registers to enable/disable it, and to monitor when it has been triggered, but no actual interrupt is ever output to the host as a result. These bits exist only to provide EGA compatibility for older software.

### 2-1 Reserved

### 0 Display Enable

0 Active display area data is being drawn on the display. Neither a horizontal retrace interval nor a vertical retrace interval is currently taking place.

1 Either a horizontal retrace interval or a vertical retrace interval is currently taking place.

## FCR Feature Control Register

Write at I/O Address 3BAh/3DAh

Read at I/O Address 3CAh

7	6	5	4	3	2	1	0
Reserved				VSYNC Control	Reserved		

### 7-4 Reserved

### 3 VSYNC Control

- 0 VSYNC output pin simply provides the vertical sync signal.
- 1 VSYNC output pin provides a signal that is the logical OR of the vertical sync signal and the value of the Display Enable bit (bit 0) of Input Status Register 1 (ST01).

**Note:** This feature is largely unused by current software.

This bit is provided for VGA compatibility.

### 2-0 Reserved

## MSR Miscellaneous Output Register

Write at I/O Address 3C2h

Read at I/O Address 3CCh

This register is cleared to 00h by reset.

7	6	5	4	3	2	1	0
VSYNC Polarity	HSYNC Polarity	Page Select	Reserved	Clock Select		RAM Enable	I/O Address

### 7 VSYNC Polarity

0 Selects positive VSYNC polarity.

1 Selects negative VSYNC polarity.

See the note at the end of this register description.

### 6 HSYNC Polarity

0 Selects positive HSYNC polarity.

1 Selects negative HSYNC polarity.

See the note at the end of this register description.

### 5 Odd/Even Page Select

0 Selects the lower 64KB page.

1 Selects the upper 64KB page.

Selects between two 64KB pages of frame buffer memory during standard VGA odd/even modes (modes 0h through 5h). Bit 1 of register GR06 can also program this bit in other modes.

### 4 Reserved

### 3-2 Clock Select

These two bits select the dot clock in standard VGA modes.

Bit 3	Bit 2	Selected Clock
0	0	CLK0, 25MHz (for standard VGA modes with 640 pixel horizontal resolution)
0	1	CLK1, 28MHz (for standard VGA modes with 720 pixel horizontal resolution)
1	0	CLK2 (left "reserved" in standard VGA)
1	1	CLK3 (left "reserved" in standard VGA)

See the note at the end of this register description.

### MSR Miscellaneous Output Register (continued)

#### 1 RAM Access Enable

- 0 Disables CPU access to frame buffer.
- 1 Enables CPU access to frame buffer.

#### 0 I/O Address Select

- 0 Sets the I/O address decode for ST01, FCR, and all CR registers to the 3Bx I/O address range (MDA emulation).
- 1 Sets the I/O address decode for ST01, FCR, and all CR registers to the 3Dx I/O address range (CGA emulation).

**Note:** In standard VGA modes, bits 7 and 6 indicate which of the three standard VGA vertical resolutions the standard VGA display should be used. All extended modes, including those with a vertical resolution of 480 scan lines, use a setting of 0 for both of these bits. This setting was “reserved” in the VGA standard.

Bit 7	Bit 6	Vertical Resolution
0	0	All other vertical resolutions
0	1	400 scan lines
1	0	350 scan lines
1	1	480 scan lines

## ***CHAPTER 9***

# ***CRT CONTROLLER REGISTERS***

**This page intentionally left blank.**



## 9 CRT Controller Registers

Name	Register Function	Access 3B5/3D5	Index Value 3B4/3D4 (CRX)
CR00	Horizontal Total Register	Read/Write	00h
CR01	Horizontal Display Enable End Register	Read/Write	01h
CR02	Horizontal Blanking Start Register	Read/Write	02h
CR03	Horizontal Blanking End Register	Read/Write	03h
CR04	Horizontal Sync Start Register	Read/Write	04h
CR05	Horizontal Sync End Register	Read/Write	05h
CR06	Vertical Total Register	Read/Write	06h
CR07	Overflow Register	Read/Write	07h
CR08	Preset Row Scan Register	Read/Write	08h
CR09	Maximum Scan Line Register	Read/Write	09h
CR0A	Text Cursor Start Scan Line Register	Read/Write	0Ah
CR0B	Text Cursor End Scan Line Register	Read/Write	0Bh
CR0C	Start Address High Register	Read/Write	0Ch
CR0D	Start Address Low Register	Read/Write	0Dh
CR0E	Text Cursor Location High Register	Read/Write	0Eh
CR0F	Text Cursor Location Low Register	Read/Write	0Fh
CR10	Vertical Sync Start Register	Read/Write	10h
CR11	Vertical Sync End Register	Read/Write	11h
CR12	Vertical Display Enable End Register	Read/Write	12h
CR13	Offset Register	Read/Write	13h
CR14	Underline Row Register	Read/Write	14h
CR15	Vertical Blanking Start Register	Read/Write	15h
CR16	Vertical Blanking End Register	Read/Write	16h
CR17	CRT Mode Control Register	Read/Write	17h
CR18	Line Compare Register	Read/Write	18h
CR22	Memory Read Latches Register	Read-Only	22h
CR30	Extended Vertical Total Register	Read/Write	30h
CR31	Extended Vertical Display Enable End Reg	Read/Write	31h
CR32	Extended Vertical Sync Start Register	Read/Write	32h
CR33	Extended Vertical Blanking Start Register	Read/Write	33h
CR40	Extended Start Address Register	Read/Write	40h
CR41	Extended Offset Register	Read/Write	41h
CR70	Interlace Control Register	Read/Write	70h
CR71	NTSC/PAL Video Output Control Register	Read/Write	71h
CR72	NTSC/PAL Horizontal Serration 1 Start Reg	Read/Write	72h
CR73	NTSC/PAL Horizontal Serration 2 Start Reg	Read/Write	73h
CR74	NTSC/PAL Horizontal Pulse Width Register	Read/Write	74h
CR75	NTSC/PAL Filtering Burst Read Length Register	Read/Write	75h
CR76	NTSC/PAL Filtering Burst Read Quantity Register	Read/Write	76h
CR77	NTSC/PAL Filtering Control Register	Read/Write	77h

The CRT controller registers are accessed by writing the index of the desired register into the CRT Controller Index Register at I/O address 3B4h or 3D4h (depending upon whether the graphics system is configured for MDA or CGA emulation), and then accessing the desired register through the data port for the CRT controller registers located at I/O address 3B5h or 3D5h (again depending upon the choice of MDA or CGA emulation).

## **CRX CRT Controller Index Register**

Read/Write at I/O address 3B4h/3D4h

This register is cleared to 00h by reset.

7	6	5	4	3	2	1	0
CRT Controller Register Index							

### **7-0 CRT Controller Register Index**

These 8 bits are used to select any one of the CRT controller registers to be accessed via the data port at I/O location 3B5h or 3D5h (depending upon whether the graphics system is configured for MDA or CGA emulation).

## **CR00 Horizontal Total Register**

Read/Write at I/O address 3B5h/3D5h with 3B4h/3D4h set to index 00h

7	6	5	4	3	2	1	0
Horizontal Total							

### **7-0 Horizontal Total**

This register is used to specify the total length of each scan line. This includes both the part of the scan line that is within the active display area and the part that is outside of it.

This register should be programmed with a value equal to the total number of character clocks within the entire length of a scan line, minus 5.

## CR01 Horizontal Display Enable End Register

Read/Write at I/O address 3B5h/3D5h with 3B4h/3D4h set to index 01h

7	6	5	4	3	2	1	0
Horizontal Display Enable End							

### 7-0 Horizontal Display Enable End

This register is used to specify the end of the part of the scan line that is within the active display area relative to its beginning. In other words, this is the horizontal width of the active display area.

This register should be programmed with a value equal to the number of character clocks that occur within the part of a scan line that is within the active display area, minus 1.

## CR02 Horizontal Blanking Start Register

Read/Write at I/O address 3B5h/3D5h with 3B4h/3D4h set to index 02h

7	6	5	4	3	2	1	0
Horizontal Blanking Start							

### 7-0 Horizontal Blanking Start

This register is used to specify the beginning of the horizontal blanking period relative to the beginning of the active display area of a scan line.

This register should be programmed with a value equal to the number of character clocks that occur on a scan line from the beginning of the active display area to the beginning of the horizontal blanking.

## **CR03 Horizontal Blanking End Register**

Read/Write at I/O address 3B5h/3D5h with 3B4h/3D4h set to index 03h

7	6	5	4	3	2	1	0
Reserved	Display Enable Skew Control		Horizontal Blanking End Bits 4-0				

### **7 Reserved**

Values written to this bit are ignored. To maintain consistency with the VGA standard, a value of 1 is returned whenever this bit is read. At one time, this bit was used to enable access to certain light pen registers. At that time, setting this bit to 0 provided this access, but setting this bit to 1 was necessary for normal operation.

### **6-5 Display Enable Skew Control**

Defines the degree to which the start and end of the active display area are delayed along the length of a scan line to compensate for internal pipeline delays.

These 2 bits describe the delay in terms of a number character clocks.

Bits 6 5	Amount of Delay
0 0	no delay
0 1	delayed by 1 character clock
1 0	delayed by 2 character clocks
1 1	delayed by 3 character clocks

### **4-0 Horizontal Blanking End Bits 4-0**

These 5 bits provide the 5 least significant bits of a 6-bit value that specifies the end of the blanking period relative to its beginning on a single scan line. Bit 7 of the Horizontal Sync End Register (CR05) supplies the most significant bit.

This 6-bit value should be programmed to be equal to the least significant 6 bits of the result of adding the length of the blanking period in terms of character clocks to the value specified in the Horizontal Blanking Start Register (CR02).

## **CR04 Horizontal Sync Start Register**

Read/Write at I/O address 3B5h/3D5h with 3B4h/3D4h set to index 04h

7	6	5	4	3	2	1	0
Horizontal Sync Start							

### **7-0 Horizontal Sync Start**

This register is used to specify the beginning of the horizontal sync pulse relative to the beginning of the active display area on a scan line.

This register should be set to be equal to the number of character clocks that occur from the beginning of the active display area to the beginning of the horizontal sync pulse on a single scan line.

## CR05 Horizontal Sync End Register

Read/Write at I/O address 3B5h/3D5h with 3B4h/3D4h set to index 05h

7	6	5	4	3	2	1	0
Horiz Blnk End Bit 5	Horizontal Sync Delay		Horizontal Sync End				

### 7 Horizontal Blanking End Bit 5

This bit provides the most significant bit of a 6-bit value that specifies the end of the horizontal blanking period relative to its beginning. Bits 4-0 of Horizontal Blanking End Register (CR03) supplies the 5 least significant bits.

This 6-bit value should be set to the least significant 6 bits of the result of adding the length of the blanking period in terms of character clocks to the value specified in the Horizontal Blanking Start Register (CR02).

### 6-5 Horizontal Sync Delay

These bits define the degree to which the start and end of the horizontal sync pulse are delayed to compensate for internal pipeline delays.

These 2 bits describe the delay in terms of a number of character clocks.

Bits	Amount of Delay
<b>6 5</b>	
0 0	no delay
0 1	delayed by 1 character clock
1 0	delayed by 2 character clocks
1 1	delayed by 3 character clocks

### 4-0 Horizontal Sync End

This 5-bit value specifies the end of the horizontal sync pulse relative to its beginning. In other words, this 5-bit value specifies the width of the horizontal sync pulse.

This 5-bit value should be set to the least significant 5 bits of the result of adding the width of the sync pulse in terms of character clocks to the value specified in the Horizontal Sync Start Register (CR04).

## **CR06 Vertical Total Register**

Read/Write at I/O address 3B5h/3D5h with 3B4h/3D4h set to index 06h

7	6	5	4	3	2	1	0
Vertical Total Bits 7-0							

### **7-0 Vertical Total Bits**

These bits provide the 8 least significant bits of either a 10-bit or 12-bit value that specifies the total number of scan lines. This includes the scan lines both inside and outside of the active display area.

In standard VGA modes, where bit 0 of the I/O Control Register (XR09) is set to 0, the vertical total is specified with a 10-bit value. The 8 least significant bits of the vertical total are supplied by these 8 bits of this register, and the 2 most significant bits are supplied by bits 5 and 0 of the Overflow Register (CR07).

In extended modes, where bit 0 of the I/O Control Register (XR09) is set to 1, the vertical total is specified with a 12-bit value. The 8 least significant bits of the vertical total are supplied by the 8 bits of this register (CR06). The 4 most significant bits are supplied by bits 3-0 of the Extended Vertical Total Register (CR30).

This 10-bit or 12-bit value should be programmed to equal the total number of scan lines, minus 2.

## CR07 Overflow Register

Read/Write at I/O address 3B5h/3D5h with 3B4h/3D4h set to index 07h

7	6	5	4	3	2	1	0
Vert Sync Start Bit 9	Vert Disp En Bit 9	Vert Total Bit 9	Line Cmp Bit 8	Vert Blnk Start Bit 8	Vert Sync Start Bit 8	Vert Disp En Bit 8	Vert Total Bit 8

### 7 Vertical Sync Start Bit 9

The vertical sync start is a 10-bit or 12-bit value that specifies the beginning of the vertical sync pulse relative to the beginning of the active display area.

In standard VGA modes, where bit 0 of the I/O Control Register (XR09) is set to 0, the vertical sync start is specified with a 10-bit value. The 8 least significant bits of the vertical sync start are supplied by bits 7-0 of the Vertical Sync Start Register (CR10), and the most and second-most significant bits are supplied by bit 7 and bit 2 of this register (CR07), respectively.

In extended modes, where bit 0 of the I/O Control Register (XR09) is set to 1, the vertical display end is specified with a 12-bit value. The 8 least significant bits of the vertical display end are supplied by bits 7-0 of the Vertical Sync Start Register (CR10), and the 4 most significant bits are supplied by bits 3-0 of the Extended Vertical Sync Start Register (CR32) register. In extended modes, neither bit 7 nor bit 2 of this register are used.

This 10-bit or 12-bit value should be programmed to be equal to the number of scan lines from the beginning of the active display area to the start of the vertical sync pulse. Since the active display area always starts on the 0th scan line, this number should be equal to the number of the scan line on which the vertical sync pulse begins.

### 6 Vertical Display Enable End Bit 9

The vertical display enable end is a 10-bit or 12-bit value that specifies the number of the last scan line within the active display area.

In standard VGA modes, where bit 0 of the I/O Control Register (XR09) is set to 0, the vertical display enable end is specified with a 10-bit value. The 8 least significant bits of the vertical display enable are supplied by bits 7-0 of the Vertical Display Enable End Register (CR12), and the most and second-most significant bits are supplied by bit 6 and bit 1 of this register (CR07), respectively.

In extended modes, where bit 0 of the I/O Control Register (XR09) is set to 1, the vertical display enable end is specified with a 12-bit value. The 8 least significant bits of the vertical display enable are supplied by bits 7-0 of the Vertical Display Enable End Register (CR12), and the 4 most significant bits are supplied by bits 3-0 of the Extended Vertical Display End Enable Register (CR31). In extended modes, neither bit 6 nor bit 1 of this register are used.

This 10-bit or 12-bit value should be programmed to be equal to the number of the last scan line within the active display area. Since the active display area always starts on the 0th scan line, this number should be equal to the total number of scan lines within the active display area, minus 1.



## **CR07 Overflow Register (continued)**

### **5 Vertical Total Bit 9**

The vertical total is a 10-bit or 12-bit value that specifies the total number of scan lines. This includes the scan lines both inside and outside of the active display area.

In standard VGA modes, where bit 0 of the I/O Control Register (XR09) is set to 0, the vertical total is specified with a 10-bit value. The 8 least significant bits of the vertical total are supplied by bits 7-0 of the Vertical Total Register (CR06), and the most and second-most significant bits are supplied by bit 5 and bit 0 of this register (CR07), respectively.

In extended modes, where bit 0 of the I/O Control Register (XR09) is set to 1, the vertical total is specified with a 12-bit value. The 8 least significant bits of the vertical total are supplied by bits 7-0 of the Vertical Total Register (CR06), and the 4 most significant bits are supplied by 3-0 bits of the Extended Vertical Total Register (CR30). In extended modes, neither bit 5 nor bit 0 of this register are used.

This 10-bit or 12-bit value should be programmed to be equal to the total number of scan lines, minus by 2.

### **4 Line Compare Bit 8**

This bit provides the second most significant bit of a 10-bit value that specifies the scan line at which the memory address counter restarts at the value of 0. Bit 6 of the Maximum Scan Line Register (CR09) supplies the most significant bit, and bits 7-0 of the Line Compare Register (CR18) supply the 8 least significant bits.

Normally, this 10-bit value is set to specify a scan line after the last scan line of the active display area. When this 10-bit value is set to specify a scan line within the active display area, it causes that scan line and all subsequent scan lines in the active display area to display video data starting at the very first byte of the frame buffer. The result is what appears to be a screen split into a top and bottom part, with the image in the top part being repeated in the bottom part.

When used in cooperation with the Start Address High Register (CR0C) and the Start Address Low Register (CR0D), it is possible to create a split display, as described earlier, but with the top and bottom parts displaying different data. The top part will display whatever data exists in the frame buffer starting at the address specified in the two start address registers (CR0C and CR0D), while the bottom part will display whatever data exists in the frame buffer starting at the first byte of the frame buffer.

## CR07 Overflow Register (continued)

### 3 Vertical Blanking Start Bit 8

The vertical blanking start is a 10-bit or 12-bit value that specifies the beginning of the vertical blanking period relative to the beginning of the active display area.

In standard VGA modes, where bit 0 of the I/O Control Register (XR09) is set to 0, the vertical blanking start is specified with a 10-bit value. The 8 least significant bits of the vertical blanking start are supplied by bits 7-0 of the Vertical Blanking Start Register (CR15), and the most and second-most significant bits are supplied by bit 5 of the Maximum Scan Line Register (CR09) and bit 3 of this register (CR07), respectively.

In extended modes, where bit 0 of the I/O Control Register (XR09) is set to 1, the vertical blanking start is specified with a 12-bit value. The 8 least significant bits of the vertical blanking start are supplied by bits 7-0 of the Vertical Blanking Start Register (CR15), and the 4 most significant bits are supplied by bits 3-0 of the Extended Vertical Blanking Start Register (CR33). In extended modes, neither bit 3 of CR07 nor bit 5 of the Maximum Scan Line Register (CR09) are used.

This 10-bit or 12-bit value should be programmed to be equal to the number of scan line from the beginning of the active display area to the beginning of the blanking period. Since the active display area always starts on the 0th scan line, this number should be equal to the number of the scan line on which the vertical blanking period begins.

### 2 Vertical Sync Start Bit 8

The vertical sync start is a 10-bit or 12-bit value that specifies the beginning of the vertical sync pulse relative to the beginning of the active display area.

In standard VGA modes, where bit 0 of the I/O Control Register (XR09) is set to 0, the vertical sync start is specified with a 10-bit value. The 8 least significant bits of the vertical sync start are supplied by bits 7-0 of the Vertical Sync Start Register (CR10), and the most and second-most significant bits are supplied by bit 7 and bit 2 of this register (CR07), respectively.

In extended modes, where bit 0 of the I/O Control Register (XR09) is set to 1, the vertical display end is specified with a 12-bit value. The 8 least significant bits of the vertical display are supplied by bits 7-0 of the Vertical Sync Start Register (CR10), and the 4 most significant bits are supplied by bits 3-0 of the Extended Vertical Sync Start Register (CR32) register. In extended modes, neither bit 7 nor bit 2 of this register (CR07) are used.

This 10-bit or 12-bit value should be programmed to be equal to the number of scan lines from the beginning of the active display area to the start of the vertical sync pulse. Since the active display area always starts on the 0th scan line, this number should be equal to the number of the scan line on which the vertical sync pulse begins.

## CR07 Overflow Register (continued)

### 1 Vertical Display Enable End Bit 8

The vertical display enable end is a 10-bit or 12-bit value that specifies the number of the last scan line within the active display area.

In standard VGA modes, where bit 0 of the I/O Control Register (XR09) is set to 0, the vertical display enable end is specified with a 10-bit value. The 8 least significant bits of the vertical display enable are supplied by bits 7-0 of the Vertical Display Enable End Register (CR12), and the most and second-most significant bits are supplied by bit 6 and bit 1 of this register (CR07), respectively.

In extended modes, where bit 0 of the I/O Control Register (XR09) is set to 1, the vertical display enable end is specified with a 12-bit value. The 8 least significant bits of the vertical display enable are supplied by bits 7-0 of the Vertical Display Enable End Register (CR12), and the 4 most significant bits are supplied by bits 3-0 of the Extended Vertical Display End Enable Register (CR31). In extended modes, neither bit 6 nor bit 1 of this register (CR07) are used.

This 10-bit or 12-bit value should be programmed to be equal to the number of the last scan line within in the active display area. Since the active display area always starts on the 0th scan line, this number should be equal to the total number of scan lines within the active display area, subtract by 1.

### 0 Vertical Total Bit 8

The vertical total is a 10-bit or 12-bit value that specifies the total number of scan lines. This includes the scan lines both inside and outside of the active display area.

In standard VGA modes, where bit 0 of the I/O Control Register (XR09) is set to 0, the vertical total is specified with a 10-bit value. The 8 least significant bits of the vertical total are supplied by bits 7-0 of the Vertical Total Register (CR06), and the most and second-most significant bits are supplied by bit 5 and bit 0 of this register (CR07), respectively.

In extended modes, where bit 0 of the I/O Control Register (XR09) is set to 1, the vertical total is specified with a 12-bit value. The 8 least significant bits of the vertical total are supplied by bits 7-0 of the Vertical Total Register (CR06), and the 4 most significant bits are supplied by 3-0 bits of the Extended Vertical Total Register (CR30). In extended modes, neither bit 5 nor bit 0 of this register (CR07) are used.

This 10-bit or 12-bit value should be programmed to be equal to the total number of scan lines, minus 2 .

## CR08 Preset Row Scan Register

Read/Write at I/O address 3B5h/3D5h with 3B4h/3D4h set to index 08h

7	6	5	4	3	2	1	0
Reserved	Leftward Horizontal Pixel Shift		Starting Row Scan Count				

### 7 Reserved

### 6-5 Leftward Horizontal Pixel Shift

Bits 6 and 5 of this register hold a 2-bit value that selects number of bytes (up to 3) by which the image is shifted horizontally to the left on the screen. This function is available in both text and graphics modes.

In text modes with a 9-pixel wide character box, the image can be shifted up to 27 pixels to the left, in increments of 9 pixels.

In text modes with an 8-pixel wide character box, and in all standard VGA graphics modes, the image can be shifted up to 24 pixels to the left, in increments of 8 pixels.

The image can be shifted still further, in increments of individual pixels, through the use of bits 3-0 of the Horizontal Pixel Panning Register (AR13).

Bits 6 5	Number of Pixels by Which Image is Shifted to the Left	
	9-Pixel Wide Text Mode	8-Pixel Wide Text & Graphics Modes
0 0	0	0
0 1	9	8
1 0	18	16
1 1	27	24

**Note:** In the VGA standard, this is called the “Byte Panning” bit.

### 4-0 Starting Row Scan Count

These 5 bits specify which horizontal line of pixels within the character boxes of the characters used on the top-most row of text on the display will be used as the top-most scan line. The horizontal lines of pixels of a character box are numbered from top to bottom, with the top-most line of pixels being number 0. If a horizontal line of these character boxes other than the top-most line is specified, then the horizontal lines of the character box above the specified line of the character box will not be displayed as part of the top-most row of text characters on the display. Normally, the value specified by these 5 bits should be 0, so that all of the horizontal lines of pixels within these character boxes will be displayed in the top-most row of text, ensuring that the characters in the top-most row of text do not look as though they have been cut off at the top.

## **CR09 Maximum Scan Line Register**

Read/Write at I/O address 3B5h/3D5h with 3B4h/3D4h set to index 09h

7	6	5	4	3	2	1	0
Double Scanning	Line Cmp Bit 9	Vert Blk Start Bit 9	Maximum Scan Line				

### **7 Double Scanning**

- 0 Disables double scanning. The clock to the row scan counter is equal to the horizontal scan rate. This is the normal setting for many of the standard VGA modes and all of the extended modes.
- 1 Enables double scanning. The clock to the row scan counter is divided by 2. This is normally used to allow CGA-compatible modes that have only 200 scan lines of active video data to be displayed as 400 scan lines (each scan line is displayed twice).

### **6 Line Compare Bit 9**

This bit provides the most significant bit of a 10-bit value that specifies the scan line at which the memory address counter restarts at the value of 0. Bit 4 of the Overflow Register (CR07) supplies the second most significant bit, and bits 7-0 of the Line Compare Register (CR18) supply the 8 least significant bits.

Normally, this 10-bit value is set to specify a scan line after the last scan line of the active display area. When this 10-bit value is set to specify a scan line within the active display area, it causes that scan line and all subsequent scan lines in the active display area to display video data starting at the very first byte of the frame buffer. The result is what appears to be a screen split into a top and bottom part, with the image in the top part being repeated in the bottom part.

When used in cooperation with the Start Address High Register (CR0C) and the Start Address Low Register (CR0D), it is possible to create a split display, as described earlier, but with the top and bottom parts displaying different data. The top part will display whatever data exists in the frame buffer starting at the address specified in the two start address registers (CR0C and CR0D), while the bottom part will display whatever data exists in the frame buffer starting at the first byte of the frame buffer.

## **CR09 Maximum Scan Line Register (continued)**

### **5 Vertical Blanking Start Bit 9**

The vertical blanking start is a 10-bit or 12-bit value that specifies the beginning of the vertical blanking period relative to the beginning of the active display area.

In standard VGA modes, where bit 0 of the I/O Control Register (XR09) is set to 0, the vertical blanking start is specified with a 10-bit value. The 8 least significant bits of the vertical blanking start are supplied by bits 7-0 of the Vertical Blanking Start Register (CR15), and the most and second-most significant bits are supplied by bit 5 of this register (CR09) and bit 3 of the Overflow Register (CR07), respectively.

In extended modes, where bit 0 of the I/O Control Register (XR09) is set to 1, the vertical blanking start is specified with a 12-bit value. The 8 least significant bits of this value are supplied by bits 7-0 of the Vertical Blanking Start Register (CR15), and the 4 most significant bits are supplied by bits 3-0 of the Extended Vertical Blanking Start Register (CR33). In extended modes, neither bit 5 of CR09 nor bit 3 of the Overflow Register (CR07) are used.

This 10-bit or 12-bit value should be programmed to be equal to the number of scan line from the beginning of the active display area to the beginning of the blanking period. Since the active display area always starts on the 0th scan line, this number should be equal to the number of the scan line on which the vertical blanking period begins.

### **4-0 Starting Row Scan Count**

These bits provide all 5 bits of a 5-bit value that specifies the number of scan lines in a horizontal row of text.

This value should be programmed to be equal to the number of scan lines in a Horizontal row of text, minus 1.

## **CR0A Text Cursor Start Register**

Read/Write at I/O address 3B5h/3D5h with 3B4h/3D4h set to index 0Ah

7	6	5	4	3	2	1	0
Reserved		Text Cursor Off	Text Cursor Start				

This cursor is the text cursor that is part of the VGA standard, and should not be confused with the hardware cursor and popup (cursor 1 and cursor 2), which are intended to be used in graphics modes. This text cursor exists only in text modes, and so this register is entirely ignored in graphics modes.

### **7-6 Reserved**

### **5 Text Cursor Off**

- 0 Enables the text cursor.
- 1 Disables the text cursor.

### **4-0 Text Cursor Start**

These 5 bits specify which horizontal line of pixels within a character box is to be used to display the first horizontal line of the cursor in text mode. The horizontal lines of pixels within a character box are numbered from top to bottom, with the top-most line being number 0. The value specified by these 5 bits should be the number of the first horizontal line of pixels on which the cursor is to be shown.

## CR0B Text Cursor End Register

Read/Write at I/O address 3B5h/3D5h with 3B4h/3D4h set to index 0Bh

7	6	5	4	3	2	1	0
Reserved	Text Cursor Skew		Text Cursor End				

This cursor is the text cursor that is part of the VGA standard, and should not be confused with the hardware cursor and popup (cursor 1 and cursor 2), which are intended to be used in graphics modes. This text cursor exists only in text modes, so this register is entirely ignored in graphics modes.

### 7 Reserved

### 6-5 Text Cursor Skew

Specifies the degree to which the start and end of each horizontal line of pixels making up the cursor is delayed to compensate for internal pipeline delays.

These 2 bits describe the delay in terms of a number of character clocks.

Bits 6 5	Amount of Delay
0 0	no delay
0 1	delayed by 1 character clock
1 0	delayed by 2 character clocks
1 1	delayed by 3 character clocks

### 4-0 Text Cursor End

These 5 bits specify which horizontal line of pixels within a character box is to be used to display the last horizontal line of the cursor in text mode. The horizontal lines of pixels within a character box are numbered from top to bottom, with the top-most line being number 0. The value specified by these 5 bits should be the number of the last horizontal line of pixels on which the cursor is to be shown.



## CR0C Start Address High Register

Read/Write at I/O address 3B5h/3D5h with 3B4h/3D4h set to index 0Ch

7	6	5	4	3	2	1	0
Start Address Bits 15-8							

### 7-0 Start Address Bits 15-8

This register provides bits 15 through 8 of either a 16-bit or 20-bit value that specifies the memory address offset from the beginning of the frame buffer at which the data to be shown in the active display area begins.

In standard VGA modes, where bit 0 of the I/O Control Register (XR09) is set to 0, the start address is specified with a 16-bit value. The eight bits of this register provide the eight most significant bits of this value, while the eight bits of the Start Address Low Register (CR0D) provide the eight least significant bits.

In extended modes, where bit 0 of the I/O Control Register (XR09) is set to 1, the start address is specified with a 20-bit value. The four most significant bits are provided by bits 3-0 of the Extended Start Address Register (CR40), bits 15 through 8 of this value are provided by this register, and the eight least significant bits are provided by the Start Address Low Register (CR0D). Note that in extended modes, these 20 bits are double-buffered and synchronized to VSYNC to ensure that changes occurring on the screen as a result of changes in the start address always have a smooth or instantaneous appearance. To change the start address in extended modes, all three registers must be set for the new value, and then bit 7 of this register must be set to 1. Only if this is done, will the hardware update the start address on the next VSYNC. When this update has been performed, the hardware will set bit 7 of this register back to 0.

## **CR0D Start Address Low Register**

Read/Write at I/O address 3B5h/3D5h with 3B4h/3D4h set to index 0Dh

7	6	5	4	3	2	1	0
Start Address Bits 7-0							

### **7-0 Start Address Bits 7-0**

This register provides the eight least significant bits of either a 16-bit or 20-bit value that specifies the memory address offset from the beginning of the frame buffer at which the data to be shown in the active display area begins.

In standard VGA modes, where bit 0 of the I/O Control Register (XR09) is set to 0, the start address is specified with a 16-bit value. The eight bits of the Start Address High Register (CR0C) provide the eight most significant bits of this value, while the eight bits of this register provide the eight least significant bits.

In extended modes, where bit 0 of the I/O Control Register (XR09) is set to 1, the start address is specified with a 20-bit value. The four most significant bits are provided by bits 3-0 of the Extended Start Address Register (CR40), bits 15 through 8 of this value are provided by the Start Address High Register (CR0C), and the eight least significant bits are provided by this register. Note that in extended modes, these 20 bits are double-buffered and synchronized to VSYNC to ensure that changes occurring on the screen as a result of changes in the start address always have a smooth or instantaneous appearance. To change the start address in extended modes, all three registers must be set for the new value, and then bit 7 of this register must be set to 1. Only if this is done, will the hardware update the start address on the next VSYNC. When this update has been performed, the hardware will set bit 7 of this register back to 0.

## **CR0E Text Cursor Location High Register**

Read/Write at I/O address 3B5h/3D5h with 3B4h/3D4h set to index 0Eh

7	6	5	4	3	2	1	0
Text Cursor Location Bits 15-8							

This cursor is the text cursor that is part of the VGA standard, and should not be confused with the hardware cursor and popup (cursor 1 and cursor 2), which are intended to be used in graphics modes. This text cursor exists only in text modes, so this register is entirely ignored in graphics modes.

### **7-0 Text Cursor Location Bits 15-8**

This register provides the 8 most significant bits of a 16-bit value that specifies the address offset from the beginning of the frame buffer at which the text cursor is located. Bit 7-0 of the Text Cursor Location Low Register (CR0F) provide the 8 least significant bits.

## **CR0F Text Cursor Location Low Register**

Read/Write at I/O address 3B5h/3D5h with 3B4h/3D4h set to index 0Fh

7	6	5	4	3	2	1	0
Text Cursor Location Bits 7-0							

This cursor is the text cursor that is part of the VGA standard, and should not be confused with the hardware cursor and popup (cursor 1 and cursor 2), which are intended to be used in graphics modes. This text cursor exists only in text modes, so this register is entirely ignored in graphics modes.

### **7-0 Text Cursor Location Bits 7-0**

This register provides the 8 least significant bits of a 16-bit value that specifies the address offset from the beginning of the frame buffer at which the text cursor is located. Bits 7-0 of the Text Cursor Location High Register (CR0D) provide the 8 most significant bits.

## **CR10 Vertical Sync Start Register**

Read/Write at I/O address 3B5h/3D5h with 3B4h/3D4h set to index 10h

7	6	5	4	3	2	1	0
Vertical Sync Start Bits 7-0							

### **7-0 Vertical Sync Start Bits 7-0**

This register provides the 8 least significant bits of either a 10-bit or 12-bit value that specifies the beginning of the vertical sync pulse relative to the beginning of the active display area of a screen.

In standard VGA modes, where bit 0 of the I/O Control Register (XR09) is set to 0, this value is described in 10 bits with bits 7 and 2 of the Overflow Register (CR07) supplying the 2 most significant bits.

In extended modes, where bit 0 of the I/O Control Register (XR09) is set to 1, this value is described in 12 bits with bits 3-0 of the Extended Vertical Sync Start Register (CR32) supplying the 4 most significant bits.

This 10-bit or 12-bit value should equal the vertical sync start in terms of the number of scan lines from the beginning of the active display area to the beginning of the vertical sync pulse. Since the active display area always starts on the 0th scan line, this number should be equal to the number of the scan line on which the vertical sync pulse begins.

## CR11 Vertical Sync End Register

Read/Write at I/O address 3B5h/3D5h with 3B4h/3D4h set to index 11h

7	6	5	4	3	2	1	0
Protect Regs 0-7	Reserved	Vert Int Enable	Vert Int Clear	Vertical Sync End			

### 7 Protect Registers 0-7

- 0 Enable writes to registers CR00-CR07.
- 1 Disable writes to registers CR00-CR07.

**Note:** The ability to write to bit 4 of the Overflow Register (CR07) is not affected by this bit. Bit 4 of the Overflow Register is always writable.

### 6 Reserved

Writes to this bit are ignored. In the VGA standard, this bit was used to switch between 3 and 5 frame buffer refresh cycles during the time required to draw each horizontal line.

### 5 Vertical Interrupt Enable

- 0 Enable the generation of an interrupt at the beginning of each vertical retrace period.
- 1 Disable the generation of an interrupt at the beginning of each vertical retrace period.

**Note:** The hardware does not actually provide an interrupt signal which would be connected to an input of the system's interrupt controller. Bit 7 of Input Status Register 0 (ST00) indicates the status of the vertical retrace interrupt, and can be polled by software to determine if a vertical retrace interrupt has taken place. Bit 4 of this register can be used to clear a pending vertical retrace interrupt.

### 4 Vertical Interrupt Clear

Setting this bit to 0 clears a pending vertical retrace interrupt. This bit must be set back to 1 to enable the generation of another vertical retrace interrupt.

**Note:** The hardware does not actually provide an interrupt signal which would be connected to an input of the system's interrupt controller. Bit 7 of Input Status Register 0 (ST00) indicates the status of the vertical retrace interrupt, and can be polled by software to determine if a vertical retrace interrupt has taken place. Bit 5 of this register can be used to enable or disable the generation of vertical retrace interrupts.

### 3-0 Vertical Sync End

These 4 bits provide a 4-bit value that specifies the end of the vertical sync pulse relative to its beginning.

This 4-bit value should be set to the least significant 4 bits of the result of adding the length of the vertical sync pulse in terms of the number of scan lines that occur within the length of the vertical sync pulse to the value that specifies the beginning of the vertical sync pulse. See the description of the Vertical Sync Start Register (CR10) for more details.

## CR12 Vertical Display Enable End Register

Read/Write at I/O address 3B5h/3D5h with 3B4h/3D4h set to index 12h

7	6	5	4	3	2	1	0
Vertical Display Enable End Bits 7-0							

### 7-0 Vertical Display Enable End Bits 7-0

This register provides the 8 least significant bits of either a 10-bit or 12-bit value that specifies the number of the last scan line within the active display area.

In standard VGA modes, where bit 0 of the I/O Control Register (XR09) is set to 0, this value is described in 10 bits with bits 6 and 1 of the Overflow Register (CR07) supplying the 2 most significant bits.

In extended modes, where bit 0 of the I/O Control Register (XR09) is set to 1, this value is described in 12 bits with bits 3-0 of the Extended Vertical Display Enable End Register (CR31) supplying the 4 most significant bits.

This 10-bit or 12-bit value should be programmed to be equal to the number of the last scan line within the active display area. Since the active display area always starts on the 0th scan line, this number should be equal to the total number of scan lines within the active display area, minus 1.

## CR13 Offset Register

Read/Write at I/O address 3B5h/3D5h with 3B4h/3D4h set to index 13h

7	6	5	4	3	2	1	0
Offset Bits 7-0							

### 7-0 Offset Bits 7-0

This register provides either all 8 bits of an 8-bit value or the 8 least significant bits of a 12-bit value that specifies the number of words or doublewords of frame buffer memory occupied by each horizontal row of characters. Whether this value is interpreted as the number of words or doublewords is determined by the settings of the bits in the Clocking Mode Register (SR01).

In standard VGA modes, where bit 0 of the I/O Control Register (XR09) is set to 0, the offset is described with an 8-bit value, with all the bits provided by this register (CR13).

In extended modes, where bit 0 of the I/O Control Register (XR09) is set to 1, the offset is described with a 12-bit value. The four most significant bits of this value are provided by bits 3-0 of the Extended Offset Register (CR41), and the eight least significant bits are provided by this register (CR13).

This 8-bit or 12-bit value should be programmed to be equal to either the number of words or doublewords (depending on the setting of the bits in the Clocking Mode Register, SR01) of frame buffer memory that is occupied by each horizontal row of characters.

## CR14 Underline Location Register

Read/Write at I/O address 3B5h/3D5h with 3B4h/3D4h set to index 14h

7	6	5	4	3	2	1	0
Reserved	Dword Mode	Count By 4	Underline Location				

### 7 Reserved

### 6 Doubleword Mode

- 0 Frame buffer addresses are interpreted by the frame buffer address decoder as being either byte addresses or word addresses, depending upon the setting of bit 6 of the CRT Mode Control Register (CR17).
- 1 Frame buffer addresses are interpreted by the frame buffer address decoder as being doubleword addresses regardless of the setting of bit 6 of the CRT Mode Control Register (CR17).

**Note:** This bit is used in conjunction with bits 6 and 5 of the CRT Mode Control Register (CR17) to select how frame buffer addresses from the CPU are interpreted by the frame buffer address decoder as shown below:

CR14 Bit 6	CR17 Bit 6	Addressing Mode
0	0	Word Mode
0	1	Byte Mode
1	0	Doubleword Mode
1	1	Doubleword Mode

### 5 Count By 4

- 0 The memory address counter is incremented either every character clock or every other character clock, depending upon the setting of bit 3 of the CRT Mode Control Register.
- 1 The memory address counter is incremented either every 4 character clocks or every 2 character clocks, depending upon the setting of bit 3 of the CRT Mode Control Register.

**Note:** This bit is used in conjunction with bit 3 of the CRT Mode Control Register (CR17) to select the number of character clocks are required to cause the memory address counter to be incremented as shown, below:

CR14 Bit 5	CR17 Bit 3	Address Incrementing Interval
0	0	every character clock
0	1	every 2 character clocks
1	0	every 4 character clocks
1	1	every 2 character clocks

### 4-0 Underline Location

These 5 bits specify which horizontal line of pixels in a character box is to be used to display a character underline in text mode. The horizontal lines of pixels within a character box are numbered from top to bottom, with the top-most line being number 0. The value specified by these 5 bits should be the number of the horizontal line on which the character underline mark is to be shown.

## CR15 Vertical Blanking Start Register

Read/Write at I/O address 3B5h/3D5h with 3B4h/3D4h set to index 15h

7	6	5	4	3	2	1	0
Vertical Blanking Start Bits 7-0							

### 7-0 Vertical Blanking Start Bits 7-0

This register provides the 8 least significant bits of either a 10-bit or 12-bit value that specifies the beginning of the vertical blanking period relative to the beginning of the active display area of the screen. Whether this value is described in 10 or 12 bits depends on the setting of bit 0 of the I/O Control Register (XR09).

In standard VGA modes, where bit 0 of the I/O Control Register (XR09) is set to 0, the vertical blanking start is specified with a 10-bit value. The most and second-most significant bits of this value are supplied by bit 5 of the Maximum Scan Line Register (CR09) and bit 3 of the Overflow Register (CR07), respectively.

In extended modes, where bit 0 of the I/O Control Register (XR09) is set to 1, the vertical blanking start is specified with a 12-bit value. The 4 most significant bits of this value are supplied by bits 3-0 of the Extended Vertical Blanking Start Register (CR33).

This 10-bit or 12-bit value should be programmed to be equal the number of scan lines from the beginning of the active display area to the beginning of the vertical blanking period. Since the active display area always starts on the 0th scan line, this number should be equal to the number of the scan line on which vertical blanking begins.

## CR16 Vertical Blanking End Register

Read/Write at I/O address 3B5h/3D5h with 3B4h/3D4h set to index 16h

7	6	5	4	3	2	1	0
Vertical Blanking End Bits 7-0							

### 7-0 Vertical Blanking End Bits 7-0

This register provides a 8-bit value that specifies the end of the vertical blanking period relative to its beginning.

This 8-bit value should be set equal to the least significant 8 bits of the result of adding the length of the vertical blanking period in terms of the number of scan lines that occur within the length of the vertical blanking period to the value that specifies the beginning of the vertical blanking period (see the description of the Vertical Blanking Start Register for details).



## CR17 CRT Mode Control

Read/Write at I/O address 3B5h/3D5h with 3B4h/3D4h set to index 17h

7	6	5	4	3	2	1	0
CRT Ctrl Reset	Word or Byte Mode	Address Wrap	Reserved	Count By 2	Horiz Retrace Sel	Select Row Scan Cntr	Compat Mode Supp.

### 7 CRT Controller Reset

- 0 Forces horizontal and vertical sync signals to be inactive. No other registers or outputs are affected.
- 1 Permits normal operation.

### 6 Word Mode or Byte Mode

- 0 The memory address counter's output bits are shifted by 1 bit position before being passed on to the frame buffer address decoder such that they are made into word-aligned addresses when bit 6 of the Underline Location Register (CR17) is set to 0.
- 1 The memory address counter's output bits remain unshifted before being passed on to the frame buffer address decoder such that they remain byte-aligned addresses when bit 6 of the Underline Location Register (CR17) is set to 0.

**Note:** This bit is used in conjunction with bits 6 and 5 of the CRT Mode Control Register (CR17) to control how frame buffer addresses from the memory address counter are interpreted by the frame buffer address decoder as shown below:

CR14 Bit 6	CR17 Bit 6	Addressing Mode
0	0	Word Mode -- addresses from the memory address counter are shifted once to become word-aligned
0	1	Byte Mode -- addresses from the memory address counter are not shifted
1	0	Doubleword Mode -- addresses from the memory address counter are shifted twice to become doubleword-aligned
1	1	Doubleword Mode -- addresses from the memory address counter are shifted twice to become doubleword-aligned

\*See the note at the end of this register description.

### 5 Address Wrap

- 0 Wrap frame buffer address at 16KB. This is used in CGA-compatible modes.
- 1 No wrapping of frame buffer addresses.

**Note:** This bit is only effective when word mode is made active by setting bit 6 in both the Underline Location Register and this register to 0.

**\*See the note at the end of this register description.**

### 4 Reserved

## CR17 CRT Mode Control (continued)

### 3 Count By 2

- 0 The memory address counter is incremented either every character clock or every 4 character clocks, depending upon the setting of bit 5 of the Underline Location Register.
- 1 The memory address counter is incremented either every other clock.

This bit is used in conjunction with bit 5 of the Underline Location Register (CR14) to select the number of character clocks are required to cause the memory address counter to be incremented as shown, below:

CR14 Bit 5	CR17 Bit 3	Address Incrementing Interval
0	0	every character clock
0	1	every 2 character clocks
1	0	every 4 character clocks
1	1	every 2 character clocks

### 2 Horizontal Retrace Select

This bit provides a method to effectively double the vertical resolution by allowing the vertical timing counter to be clocked by the horizontal retrace clock divided by 2 (usually, it would be undivided).

- 0 The vertical timing counter is clocked by the horizontal retrace clock.
- 1 The vertical timing counter is clocked by the horizontal retrace clock divided by 2.

### 1 Select Row Scan Counter

- 0 A substitution takes place, whereby bit 14 of the 16-bit memory address generated by the memory address counter (after the stage at which these 16 bits may have already been shifted to accommodate word or doubleword addressing) is replaced with bit 1 of the row scan counter at a stage just before this address is presented to the frame buffer address decoder.
- 1 No substitution takes place.

**\*See the note at the end of this register description.**

### 0 Compatibility Mode Support

- 0 A substitution takes place, whereby bit 13 of the 16-bit memory address generated by the memory address counter (after the stage at which these 16 bits may have already been shifted to accommodate word or doubleword addressing) is replaced with bit 0 of the row scan counter at a stage just before this address is presented to the frame buffer address decoder.
- 1 No substitution takes place.

**\*See the note at the end of this register description.**

**CR17 CRT Mode Control (continued)**

**Note:**

The two tables that follow show the possible ways in which the address bits from the memory address counter can be shifted and/or reorganized before being presented to the frame buffer address decoder. First, the address bits generated by the memory address counter (MAOut0 to MAOut15) are reorganized, if needed, to accommodate byte, word, or doubleword modes. The resulting reorganized outputs (Reorg0 to Reorg15) may then also be further manipulated with the substitution of bits from the row scan counter (RSOut0 and RSOut1) before finally being presented to the input bits of the frame buffer address decoder (FBIn15-FBIn0).

Bits Generated by the Memory Address Counter (MAOut0 to MAOut15)							Resulting Reorganized Bits
Byte Mode CR14 bit 6=0 CR17 bit 6=1 CR17 bit 5=X	Word Mode CR14 bit 6=0 CR17 bit 6=0 CR17 bit 5=1	Word Mode CR14 bit 6=0 CR17 bit 6=0 CR17 bit 5=0	Word Mode CR14 bit 6=0 CR17 bit 6=0 CR17 bit 5=0	Word Mode CR14 bit 6=0 CR17 bit 6=0 CR17 bit 5=0	Doubleword Mode CR14 bit 6=1 CR17 bit 6=X CR17 bit 5=X		
MAOut0	OR	MAOut15	OR	MAOut13	OR	MAOut12	Reorg0
MAOut1		MAOut0		MAOut0		MAOut13	Reorg1
MAOut2		MAOut1		MAOut1		MAOut0	Reorg2
MAOut3		MAOut2		MAOut2		MAOut1	Reorg3
MAOut4		MAOut3		MAOut3		MAOut2	Reorg4
MAOut5		MAOut4		MAOut4		MAOut3	Reorg5
MAOut6		MAOut5		MAOut5		MAOut4	Reorg6
MAOut7		MAOut6		MAOut6		MAOut5	Reorg7
MAOut8		MAOut7		MAOut7		MAOut6	Reorg8
MAOut9		MAOut8		MAOut8		MAOut7	Reorg9
MAOut10		MAOut9		MAOut9		MAOut8	Reorg10
MAOut11		MAOut10		MAOut10		MAOut9	Reorg11
MAOut12		MAOut11		MAOut11		MAOut10	Reorg12
MAOut13		MAOut12		MAOut12		MAOut11	Reorg13
MAOut14		MAOut13		MAOut13		MAOut12	Reorg14
MAOut15		MAOut14		MAOut14		MAOut13	Reorg15

CR17 bit 1=1 CR17 bit 0=1	CR17 bit 1=1 CR17 bit 0=0	CR17 bit 1=0 CR17 bit 0=1	CR17 bit 1=0 CR17 bit 0=0	Bits Sent to the Frame Buffer Address Decoder		
Reorg0	OR	Reorg0	OR	Reorg0	FBIn0	
Reorg1		Reorg1		Reorg1	Reorg1	FBIn1
Reorg2		Reorg2		Reorg2	Reorg2	FBIn2
Reorg3		Reorg3		Reorg3	Reorg3	FBIn3
Reorg4		Reorg4		Reorg4	Reorg4	FBIn4
Reorg5		Reorg5		Reorg5	Reorg5	FBIn5
Reorg6		Reorg6		Reorg6	Reorg6	FBIn6
Reorg7		Reorg7		Reorg7	Reorg7	FBIn7
Reorg8		Reorg8		Reorg8	Reorg8	FBIn8
Reorg9		Reorg9		Reorg9	Reorg9	FBIn9
Reorg10		Reorg10		Reorg10	Reorg10	FBIn10
Reorg11		Reorg11		Reorg11	Reorg11	FBIn11
Reorg12		Reorg12		Reorg12	Reorg12	FBIn12
Reorg13		RSOut0		Reorg13	RSOut0	FBIn13
Reorg14		Reorg14		RSOut1	RSOut1	FBIn14
Reorg15		Reorg15		Reorg15	Reorg15	FBIn15

## CR18 Line Compare Register

Read/Write at I/O address 3B5h/3D5h with 3B4h/3D4h set to index 18h

7	6	5	4	3	2	1	0
Line Compare Bits 7-0							

### 7-0 Line Compare Bits 7-0

This register provides the 8 least significant bits of a 10-bit value that specifies the scan line at which the memory address counter restarts at the value of 0. Bit 6 of the Maximum Scan Line Register (CR09) supplies the most significant bit, and bit 4 of the Overflow Register (CR07) supplies the second most significant bit.

Normally, this 10-bit value is set to specify a scan line after the last scan line of the active display area. When this 10-bit value is set to specify a scan line within the active display area, it causes that scan line and all subsequent scan lines in the active display area to display video data starting at the very first byte of the frame buffer. The result is what appears to be a screen split into a top and bottom part, with the image in the top part being repeated in the bottom part.

When used in cooperation with the Start Address High Register (CR0C) and the Start Address Low Register (CR0D), it is possible to create a split display, as described earlier, but with the top and bottom parts displaying different data. The top part will display whatever data exists in the frame buffer starting at the address specified in the two start address registers (CR0C and CR0D), while the bottom part will display whatever data exists in the frame buffer starting at the first byte of the frame buffer.

## **CR22 Memory Read Latch Data Register**

read-only at I/O address 3B5h/3D5h with 3B4h/3D4h set to index 22h

7	6	5	4	3	2	1	0
Memory Read Latch Data							

### **7-0 Memory Read Latch Data**

This register provides the value currently stored in 1 of the 4 memory read latches. Bits 1 and 0 of the Read Map Select Register (GR04) select which of the 4 memory read latches may be read using this register.

## **CR30 Extended Vertical Total Register**

Read/Write at I/O address 3B5h/3D5h with 3B4h/3D4h set to index 30h

7	6	5	4	3	2	1	0
Reserved				Vertical Total Bits 11-8			

### **7-4 Reserved**

Whenever this register is written to, these bits should be set to 0.

### **3-0 Vertical Total Bits 11-8**

The vertical total is a 10-bit or 12-bit value that specifies the total number of scan lines. This includes the scan lines both inside and outside of the active display area.

In standard VGA modes, where bit 0 of the I/O Control Register (XR09) is set to 0, the vertical total is specified with a 10-bit value. The 8 least significant bits of the vertical total are supplied by bits 7-0 of the Vertical Total Register (CR06), and the 2 most significant bits are supplied by bits 5 and 0 of the Overflow Register (CR07). In standard VGA modes, these bits 3-0 of this register (CR30) are not used.

In extended modes, where bit 0 of the I/O Control Register (XR09) is set to 1, the vertical total is specified with a 12-bit value. The 8 least significant bits of this value are supplied by bits 7-0 of the Vertical Total Register (CR06), and the 4 most significant bits are supplied by bits 3-0 of this register (CR30).

This 10-bit or 12-bit value should be programmed to be equal to the total number of scan lines, minus 2.

## CR31 Extended Vertical Display End Register

Read/Write at I/O address 3B5h/3D5h with 3B4h/3D4h set to index 31h

7	6	5	4	3	2	1	0
Reserved				Vertical Display End Bits 11-8			

### 7-4 Reserved

Whenever this register is written to, these bits should be set to 0.

### 3-0 Vertical Display End Bits 11-8

The vertical display enable end is a 10-bit or 12-bit value that specifies the number of the last scan line within the active display area.

In standard VGA modes, where bit 0 of the I/O Control Register (XR09) is set to 0, the vertical display enable end is specified with a 10-bit value. The 8 least significant bits of the vertical display enable end are supplied by bits 7-0 of the Vertical Display Enable End Register (CR12), and the 2 most significant bits are supplied by bits 6 and 1 of the Overflow Register (CR07). In standard VGA modes bits 3-0 of CR31 are not used.

In extended modes, where bit 0 of the I/O Control Register (XR09) is set to 1, the vertical display enable end is specified with a 12-bit value. The 8 least significant bits of the vertical display enable end are supplied by bits 7-0 of the Vertical Display Enable End Register (CR12), and the 4 most significant bits are supplied by these 4 bits of this register (CR31).

This 10-bit or 12-bit value should be programmed to be equal to the number of the last scan line within the active display area. Since the active display area always starts on the 0th scan line, this number should be equal to the total number of scan lines within the active display area, minus 1.

## CR32 Extended Vertical Sync Start Register

Read/Write at I/O address 3B5h/3D5h with 3B4h/3D4h set to index 32h

7	6	5	4	3	2	1	0
Reserved				Vertical Sync Start Bits 11-8			

### 7-4 Reserved

Whenever this register is written to, these bits should be set to 0.

### 3-0 Vertical Sync Start Bits 11-8

The vertical sync start is a 10-bit or 12-bit value that specifies the beginning of the vertical sync pulse relative to the beginning of the active display area.

In standard VGA modes, where bit 0 of the I/O Control Register (XR09) is set to 0, the vertical sync start is specified with a 10-bit value. The 8 least significant bits of the vertical sync start are supplied by bits 7-0 of the Vertical Sync Start Register (CR10), and the 2 most significant bits are supplied by bits 7 and 2 of the Overflow Register (CR07). In standard VGA modes, bits 3-0 of CR32 are not used.

In extended modes, where bit 0 of the I/O Control Register (XR09) is set to 1, the vertical display end is specified with a 12-bit value. The 8 least significant bits of the vertical sync start are supplied by bits 7-0 of the Vertical Sync Start Register (CR10), and the 4 most significant bits are supplied by bits 3-0 of this register (CR32).

This 10-bit or 12-bit value should be programmed to be equal to the number of scan lines from the beginning of the active display area to the start of the vertical sync pulse. Since the active display area always starts on the 0th scan line, this number should be equal to the number of the scan line on which the vertical sync pulse begins.

## CR33 Extended Vertical Blanking Start Register

Read/Write at I/O address 3B5h/3D5h with 3B4h/3D4h set to index 33h

7	6	5	4	3	2	1	0
Reserved				Vertical Blanking Start Bits 11-8			

### 7-4 Reserved

Whenever this register is written to, these bits should be set to 0.

### 3-0 Vertical Blanking Start Bits 11-8

The vertical blanking start is a 10-bit or 12-bit value that specifies the beginning of the vertical blanking period relative to the beginning of the active display area.

In standard VGA modes, where bit 0 of the I/O Control Register (XR09) is set to 0, the vertical blanking start is specified with a 10-bit value. The 8 least significant bits of the vertical blanking start are supplied by bits 7-0 of the Vertical Blanking Start Register (CR15), and the most and second-most significant bits are supplied by bit 5 of the Maximum Scan Line Register (CR09) and bit 3 of the Overflow Register (CR07), respectively. In standard VGA modes, bits 3-0 of this register (CR33) are not used.

In extended modes, where bit 0 of the I/O Control Register (XR09) is set to 1, the vertical blanking start is specified with a 12-bit value. The 8 least significant bits of this value are supplied by bits 7-0 of the Vertical Blanking Start Register (CR15), and the 4 most significant bits are supplied by bits 3-0 of this register (CR33).

This 10-bit or 12-bit value should be programmed to be equal to the number of scan line from the beginning of the active display area to the beginning of the blanking period. Since the active display area always starts on the 0th scan line, this number should be equal to the number of the scan line on which the vertical blanking period begins.



## **CR40 Extended Start Address Register**

Read/Write at I/O address 3B5h/3D5h with 3B4h/3D4h set to index 40h

7	6	5	4	3	2	1	0
Strt Addr En	Reserved			Start Address Bits 19-16			

### **7 Extended Mode Start Address Enable**

This bit is used only in extended modes, where bit 0 of the I/O Control Register (XR09) is set to 1, to signal the hardware to update the start address. In extended modes, the start address is specified with a 20 bit value. These 20 bits, which are provided by the Start Address Low Register (CR0D), the Start Address High Register (CR0C) and bits 3-0 of this register, are double-buffered and synchronized to VSYNC to ensure that changes occurring on the screen as a result of changes in the start address always have a smooth or instantaneous appearance. To change the start address in extended modes, all three registers must be set for the new value, and then this bit of this register must be set to 1. Only if this is done, will the hardware update the start address on the next VSYNC. When this update has been performed, the hardware will set bit 7 of this register back to 0.

### **6-4 Reserved**

Whenever this register is written to, these bits should be set to 0.

### **3-0 Start Address Bits 19-16**

The start address is a 16-bit or a 20-bit value that specifies the memory address offset from the beginning of the frame buffer at which the data to be shown in the active display area begins.

In standard VGA modes, where bit 0 of the I/O Control Register (XR09) is set to 0, the start address is specified with a 16-bit value. The eight bits of the Start Address High Register (CR0C) provide the eight most significant bits of this value, while the eight bits of the Start Address Low Register (CR0D) provide the eight least significant bits.

In extended modes, where bit 0 of the I/O Control Register (XR09) is set to 1, the start address is specified with a 20-bit value. The four most significant bits are provided by bits 3-0 of this register, bits 15 through 8 of this value are provided by the Start Address High Register (CR0C), and the eight least significant bits are provided by the Start Address Low Register (CR0D). Note that in extended modes, these 20 bits are double-buffered and synchronized to VSYNC to ensure that changes occurring on the screen as a result of changes in the start address always have a smooth or instantaneous appearance. To change the start address in extended modes, all three registers must be set for the new value, and then bit 7 of this register must be set to 1. Only if this is done, will the hardware update the start address on the next VSYNC. When this update has been performed, the hardware will set bit 7 of this register back to 0.

## CR41 Extended Offset Register

Read/Write at I/O address 3B5h/3D5h with 3B4h/3D4h set to index 41h

7	6	5	4	3	2	1	0
Reserved				Offset Bits 11-8			

### 7-4 Reserved

Whenever this register is written to, these bits should be set to 0.

### 3-0 Offset Bits 11-8

The offset is an 8-bit or 12-bit value describing the number of words or doublewords of frame buffer memory occupied by each horizontal row of characters. Whether this value is interpreted as the number of words or doublewords is determined by the settings of the bits in the Clocking Mode Register (SR01).

In standard VGA modes, where bit 0 of the I/O Control Register (XR09) is set to 0, the offset is described with an 8-bit value, all the bits of which are provided by the Offset Register (CR13).

In extended modes, where bit 0 of the I/O Control Register (XR09) is set to 1, the offset is described with a 12-bit value. The four most significant bits of this value are provided by bits 3-0 of this register, and the eight least significant bits are provided by the Offset Register (CR13).

This 8-bit or 12-bit value should be programmed to be equal to either the number of words or doublewords (depending on the setting of the bits in the Clocking Mode Register, SR01) of frame buffer memory that is occupied by each horizontal row of characters.

## CR70 Interlace Control Register

Read/Write at I/O address 3B5h/3D5h with 3B4h/3D4h set to index 70h

7	6	5	4	3	2	1	0
Interlace Enable	CRT Half-Line Value						

### 7 Interlace Enable

0 Selects non-interlaced CRT output. This is the default after reset.

1 Selects interlaced CRT output.

### 6-0 CRT Half-Line Value

When interlaced CRT output has been selected, the value in this register specifies the position along the length of a scan line at which the half-line vertical sync pulse occurs for the odd frame. This half-line vertical sync pulse begins at a position between two horizontal sync pulses on the last scan line, rather than coincident with the beginning of a Horizontal sync pulse at the end of a scan line.

## CR71 NTSC/PAL Video Output Control Register

Read/Write at I/O address 3B5h/3D5h with 3B4h/3D4h set to index 71h

7	6	5	4	3	2	1	0
NTSC/ PAL Sel	Pedestal Enable	Blanking Delay Ctrl	Composite Sync Character Clk Delay		Composite Sync Pixel Clk Delay		

### 7 NTSC/PAL Select

- 0 Selects NTSC-formatted video output.
- 1 Selects PAL-formatted video output.

### 6 Pedestal Enable

- 0 Disables the provision of an additional voltage pedestal on red, green and blue analog output lines during the active video portions of each horizontal line.
- 1 Enables the provision of an additional voltage pedestal on the red, green, and blue analog output lines during the active video portions of each horizontal line.

### 5 Blanking Delay Control

- 0 Blanking period is not delayed on odd frames.
- 1 Blanking period is delayed by half a scan line on odd frames.

### 4-3 Composite Sync Character Clock Delay

These 2 bits specify the number of character clocks (from 0 to 3) by which the composite sync may be delayed.

### 2-0 Composite Sync Pixel Clock Delay

These 3 bits specify the number of pixel clocks (from 0 to 7) by which the composite sync may be delayed.

## **CR72 NTSC/PAL Horizontal Serration 1 Start Register**

Read/Write at I/O address 3B5h/3D5h with 3B4h/3D4h set to index 72h

7	6	5	4	3	2	1	0
Horizontal Serration 1 Start							

### **7-0 Horizontal Serration 1 Start**

These 8 bits specify the start position along the length of a scan line of the first horizontal serration pulse for composite sync generation.

## **CR73 NTSC/PAL Horizontal Serration 2 Start Register**

Read/Write at I/O address 3B5h/3D5h with 3B4h/3D4h set to index 73h

7	6	5	4	3	2	1	0
Horizontal Serration 2 Start							

### **7-0 Horizontal Serration 2 Start**

These 8 bits specify the start position along the length of a scan line of the second horizontal serration pulse for composite sync generation.

## CR74 NTSC/PAL Horizontal Pulse Width Register

Read/Write at I/O address 3B5h/3D5h with 3B4h/3D4h set to index 74h

7	6	5	4	3	2	1	0
Reserved		Round Off	NTSC/PAL Horizontal Pulse Width				

### 7-6 Reserved

### 5 NTSC/PAL Horizontal Pulse Width Round Off Control

- 0 Enables the generation of horizontal equalization pulses with a width that is approximately equal to half the width of the horizontal sync pulse. The actual width is determined using bits 4-0 of this register.
- 1 Disables the generation of horizontal equalization pulses.

### 4-0 NTSC/PAL Horizontal Pulse Width

These 5 bits specify the pulse width of the horizontal equalization pulse used to generate the NTSC/PAL-compliant composite sync. Normally, the width of this horizontal equalization pulse is approximately half the width of the horizontal sync pulse.

These 5 bits should be programmed with a value equal to the actual pulse width, subtracted by 1. The width of the actual equalization pulse can be calculated as follows:

$$\text{pulse width} = ((\text{CR74}[4:0] - \text{CR74}[5]) \div 2) + 1$$

## CR75 NTSC/PAL Filtering Burst Read Length Register

Read/Write at I/O address 3B5h/3D5h with 3B4h/3D4h set to index 75h

7	6	5	4	3	2	1	0
Reserved (Writable) (xxxx)				Memory Burst Read Length (xxxx)			

### 7-4 Reserved

These bits should always be written with the value of 0.

### 3-0 Memory Burst Access Segment Length

The flicker reduction filtering processes are performed on pixel data as it is sequentially read from the frame buffer to be displayed. These filtering processes involve the averaging of current pixel data that is about to be displayed with data for adjacent pixels. Depending upon which filtering processes are selected, accesses to the frame buffer can become non-sequential. To optimize the use of the frame buffer, burst accesses of one or more quadwords are performed to read this data. These 4 bits provide a means of adjusting how many quadwords of pixel data are read from the frame buffer in each burst access.

## CR76 NTSC/PAL Filtering Burst Read Quantity Register

Read/Write at I/O address 3B5h/3D5h with 3B4h/3D4h set to index 76h

7	6	5	4	3	2	1	0
Memory Burst Access Segments Per Scan Line (xxxx:xxxx)							

### 7-0 Memory Burst Access Segments Per Scan Line

These 8 bits specify the number of burst reads required to supply both current pixel data and pixel data from adjacent pixels for each scan line's worth of displayable pixel data. Refer to the NTSC/PAL Filtering Burst Read Length Register (CR75) for an explanation of these burst reads.

## CR77 NTSC/PAL Filtering Control Register

Read/Write at I/O address 3B5h/3D5h with 3B4h/3D4h set to index 77h

7	6	5	4	3	2	1	0
Text Mode Line Halving (0)	Reserved (Writable) (000)			Hor. Filter Enable (0)	Ver. Filter Enable (0)	Clk Doubling Enable (0)	Filtering Enable (0)

### 7 VGA Text Mode Scan Line Halving

- 0 Disables VGA text mode scan line halving.
- 1 Enables VGA text mode scan line halving, where the setting carried in the Maximum Scan Line Register (CR09) and that carried by bits 4-0 of the Text Cursor End Register (CR0B) are halved. This is done to cut the number of scan lines actually sent to the display from VGA standard quantities (such as 400) down to quantities that are more manageable for televisions (such as 200) without actually programming CR09 and bits 4-0 of CR0B with values that are different from VGA standards. This function is meant to be used in conjunction with character fonts that are only half as high as those normally used in VGA text modes.

### 6-4 Reserved (Writable)

These bits should always be written with the value of 0.

### 3 Horizontal Flicker Reduction Filtering Enable

**Note:** Bits 1 and 0 of this register must both be set to 1 in order to enable the flicker reduction filtering hardware, before horizontal flicker reduction filtering can be enabled through this bit.

- 0 Disables horizontal flicker reduction filtering
- 1 Enables horizontal flicker reduction filtering where the current pixel is averaged with the pixels immediately to the left and right on the same scan line. This averaging process uses weighted averaging. The current pixel's value is divided by 2, the values of each of the two adjacent pixels is divided by 4, and the resulting three values are added to create the value that is displayed.

### 2 Vertical Flicker Reduction Filtering Enable

**Note:** Bits 1 and 0 of this register must both be set to 1 in order to enable the flicker reduction filtering hardware, before vertical flicker reduction filtering can be enabled through this bit.

- 0 Disables vertical flicker reduction filtering
- 1 Enables vertical flicker reduction filtering where the pixels of the current scan line are averaged with the pixels of the next scan line as the pixels of the current scan line are being displayed.

**CR77 NTSC/PAL Filtering Control Register (continued)****1 Internal Clock Doubling Enable**

- 0 One of the internal clocks used by the graphics controller remains at normal clock rates.
- 1 One of the internal clocks used by the graphics controller is doubled in frequency.

**0 Flicker Reduction Filtering Enable**

**Note:** Bit 1 of this register should be set to enable the doubling of an internal clock, before the use of the flicker reduction hardware is enabled by setting this bit to 1.

- 0 Disables all flicker reduction filter hardware.
- 1 Enables the use of the flicker reduction filter hardware.



## ***CHAPTER 10***

# ***SEQUENCER REGISTERS***

**This page intentionally left blank.**

## 10.0 SEQUENCER REGISTERS

Name	Function	Access (via 3C5)	Index Value In 3C4 (SRX)
SR00	Reset Register	Read/Write	00
SR01	Clocking Mode Register	Read/Write	01
SR02	Plane Mask Register	Read/Write	02
SR03	Character Map Select Register	Read/Write	03
SR04	Memory Mode Register	Read/Write	04
SR07	Horizontal Character Counter Reset	Read/Write	07

The sequencer registers are accessed by writing the index of the desired register into the VGA Sequencer Index Register (SRX) at I/O address 3C4, and then accessing the desired register through the data port for the sequencer registers at I/O address 3C5.

## **SRX Sequencer Index Register**

Read/Write at I/O address 3C4h

This register is cleared to 00h by reset.

7	6	5	4	3	2	1	0
Reserved					Sequencer Register Index		

### **7-3 Reserved**

### **2-0 Sequencer Register Index**

These three bits are used to select any one of the sequencer registers, SR00 through SR07, to be accessed via the data port at I/O location 3C5.

**Note:** SR02 is referred to in the VGA standard as the Map Mask Register. However, the word “map” is used with multiple meanings in the VGA standard and was, therefore, deemed too confusing, hence the reason for calling it the Plane Mask Register.

**Note:** SR07 is a standard VGA register that was not documented by IBM.  
It is not a CHIPS extension.

## SR00 Reset Register

Read/Write at I/O address 3C5h with 3C4h set to index 00h

7	6	5	4	3	2	1	0
Reserved						Sync Reset	Async Reset

### 7-2 Reserved

#### 1 Synchronous Reset

Setting this bit to 0 commands the sequencer to perform a synchronous clear and then halt. The sequencer should be reset via this bit before changing the Clocking Mode Register (SR01) if the memory contents are to be preserved. However, leaving this bit set to 0 for longer than a few tenths of a microsecond can still cause data loss in the frame buffer. No register settings are changed by performing this type of reset.

- 0 Forces synchronous reset and halt
- 1 Permits normal operation

#### 0 Asynchronous Reset

Setting this bit to 0 commands the sequencer to perform a clear and then halt. Resetting the sequencer via this bit can cause data loss in the frame buffer. No register settings are changed by performing this type of reset.

- 0 Forces asynchronous reset
- 1 Permits normal operation

## SR01 Clocking Mode Register

Read/Write at I/O address 3C5h with 3C4h set to index 01h

7	6	5	4	3	2	1	0
Reserved		Screen Off	Shift 4	Dot Clock Divide	Shift Load	Reserved	8/9 Dot Clocks

### 7-6 Reserved

### 5 Screen Off

- 0 Permits normal operation
- 1 Disables video output (blanks the screen) and turns off the picture-generating logic, thereby, allowing the full memory bandwidth to be available for CPU accesses. Synchronization pulses to the display, however, are maintained. Setting this bit to 1 can be used as a way to more rapidly update the frame buffer.

### 4 Shift 4

- 0 Causes the video data shift registers to be loaded every 1 or 2 character clock cycles, depending on bit 2 of this register.
- 1 Causes the video data shift registers to be loaded every 4 character clock cycles.

### 3 Dot Clock Divide

Setting this bit to 1 divides the dot clock by two and stretches all timing periods. This bit is used in standard VGA 40-column text modes to stretch timings to create horizontal resolutions of either 320 or 360 pixels (as opposed to 640 or 720 pixels, normally used in standard VGA 80-column text modes).

- 0 Pixel clock is left unaltered.
- 1 Pixel clock is divided by 2.

### 2 Shift Load

This bit is ignored if bit 4 of this register is set to 1.

- 0 Causes the video data shift registers to be loaded on every character clock, if bit 4 of this register is set to 0.
- 1 Causes the video data shift registers to be loaded every 2 character clocks, provided that bit 4 of this register is set to 0.

### 1 Reserved

### 0 8/9 Dot Clocks

- 0 Selects 9 dot clocks (9 horizontal pixels) per character in text modes with a horizontal resolution of 720 pixels
- 1 Selects 8 dot clocks (8 horizontal pixels) per character in text modes with a horizontal resolution of 640 pixels

## SR02 Plane Mask Register

Read/Write at I/O address 3C5h with 3C4h set to index 02h

7	6	5	4	3	2	1	0
Reserved				Memory Plane 3	Memory Plane 2	Memory Plane 1	Memory Plane 0

### 7-4 Reserved

### 3-0 Memory Plane 3 through Memory Plane 0

These four bits of this register control processor write access to the four memory maps:

- 0 Disables CPU write access to the given memory plane
- 1 Enables CPU write access to the given memory plane

In both the Odd/Even Mode and the Chain 4 Mode, these bits still control access to the corresponding color plane.

**Note:** This register is referred to in the VGA standard as the Map Mask Register. However, the word “map” is used with multiple meanings in the VGA standard and was, therefore, deemed too confusing, hence the reason for calling it the Plane Mask Register.

## SR03 Character Map Select Register

Read/Write at I/O address 3C5h with 3C4h set to index 03h

7	6	5	4	3	2	1	0
Reserved		Char Map A Select (bit 0)	Char Map B Select (bit 0)	Character Map A Select (bits 2 and 1)		Character Map B Select (bits 2 and 1)	

### 7-6 Reserved

### 5, 3-2 Character Map Select Bits for Character Map B

These three bits are used to select the character map (character generator tables) to be used as the secondary character set (font). Note that the numbering of the maps is not sequential.

Bit 3	Bit 2	Bit 5	Map Number	Table Location
0	0	0	0	1st 8KB of plane 2 at offset 0
0	0	1	4	2nd 8KB of plane 2 at offset 8K
0	1	0	1	3rd 8KB of plane 2 at offset 16K
0	1	1	5	4th 8KB of plane 2 at offset 24K
1	0	0	2	5th 8KB of plane 2 at offset 32K
1	0	1	6	6th 8KB of plane 2 at offset 40K
1	1	0	3	7th 8KB of plane 2 at offset 48K
1	1	1	7	8th 8KB of plane 2 at offset 56K

### 4, 1-0 Character Map Select Bits for Character Map A

These three bits are used to select the character map (character generator tables) to be used as the primary character set (font). Note that the numbering of the maps is not sequential.

Bit 1	Bit 0	Bit 4	Map Number	Table Location
0	0	0	0	1st 8KB of plane 2 at offset 0
0	0	1	4	2nd 8KB of plane 2 at offset 8K
0	1	0	1	3rd 8KB of plane 2 at offset 16K
0	1	1	5	4th 8KB of plane 2 at offset 24K
1	0	0	2	5th 8KB of plane 2 at offset 32K
1	0	1	6	6th 8KB of plane 2 at offset 40K
1	1	0	3	7th 8KB of plane 2 at offset 48K
1	1	1	7	8th 8KB of plane 2 at offset 56K

**Note:** In text modes, bit 3 of the video data's attribute byte normally controls the foreground intensity. This bit may be redefined to control switching between character sets. This latter function is enabled whenever there is a difference in the values of the Character Font Select A and the Character Font Select B bits. If the two values are the same, the character select function is disabled and attribute bit 3 controls the foreground intensity.

**Note:** Bit 1 of the Memory Mode Register (SR04) must be set to 1 for the character font select function of this register to be active. Otherwise, only character maps 0 and 4 are available.



## SR04 Memory Mode Register

Read/Write at I/O address 3C5h with 3C4h set to index 04h

7	6	5	4	3	2	1	0
Reserved				Chain 4	Odd/ Even	Extended Memory	Reserved

### 7-4 Reserved

### 3 Chain 4 Mode

- 0 The manner in which the frame buffer memory is mapped is determined by the setting of bit 2 of this register.
- 1 The frame buffer memory is mapped in such a way that the function of address bits 0 and 1 are altered so that they select planes 0 through 3.

The selections made by this bit affect both CPU Read and write accesses to the frame buffer.

### 2 Odd/Even Mode

- 0 The frame buffer memory is mapped in such a way that the function of address bit 0 such that even addresses select planes 0 and 2 and odd addresses select planes 1 and 3.
- 1 Addresses sequentially access data within a bit map, and the choice of which map is accessed is made according to the value of the Plane Mask Register (SR02).

Bit 3 of this register must be set to 0 for this bit to be effective. The selections made by this bit affect only CPU writes to the frame buffer.

### 1 Extended Memory Enable

- 0 Disable CPU accesses to more than the first 64KB of VGA standard memory.
- 1 Enable CPU accesses to the rest of the 256KB total VGA memory beyond the first 64KB.

This bit must be set to 1 to enable the selection and use of character maps in plane 2 via the Character Map Select Register (SR03).

### 0 Reserved

## SR07 Horizontal Character Counter Reset

Read/Write at I/O address 3C5h with 3C4h set to index 07h

7	6	5	4	3	2	1	0
Horizontal Character Counter							

Writing this register with any data will cause the horizontal character counter to be held in reset (the character counter output will remain 0) until a write occurs to any other sequencer register location with SRX set to an index of 0 through 6.

The vertical line counter is clocked by a signal derived from the horizontal display enable (which does not occur if the horizontal counter is held in reset). Therefore, if a write occurs to this register occurs during the vertical retrace interval, both the horizontal and vertical counters will be set to 0. A write to any other sequencer register location (with SRX set to an index of 0 through 6) may then be used to start both counters with reasonable synchronization to an external event via software control.

This is a standard VGA register which was not documented by IBM.

# **CHAPTER 11**

# **GRAPHICS CONTROLLER REGISTERS**

**This page intentionally left blank.**

## 11.0 GRAPHICS CONTROLLER REGISTERS

Name	Function	Access (via 3CF)	Index Value In 3CE (GRX)
GR00	Set/Reset Register	Read/Write	00h
GR01	Enable Set/Reset Register	Read/Write	01h
GR02	Color Compare Register	Read/Write	02h
GR03	Data Rotate Register	Read/Write	03h
GR04	Read Map Select Register	Read/Write	04h
GR05	Graphics Mode Register	Read/Write	05h
GR06	Miscellaneous Register	Read/Write	06h
GR07	Color Don't Care Register	Read/Write	07h
GR08	Bit Mask Register	Read/Write	08h

The graphics controller registers are accessed by writing the index of the desired register into the VGA Graphics Controller Index Register (GRX) at I/O address 3CE, and then accessing the desired register through the data port for the graphics controller registers located at I/O address 3CF.

## GRX Graphics Controller Index Register

Read/Write at I/O address 3CEh

This register is cleared to 00h by reset.

7	6	5	4	3	2	1	0
Reserved				Graphics Controller Register Index			

### 7-4 Reserved

### 3-0 Graphics Controller Register Index

These four bits are used to select any one of the graphics controller registers, GR00 through GR08, to be accessed via the data port at I/O location 3CF.

## GR00 Set/Reset Register

Read/Write at I/O address 3CFh with 3CEh set to index 00h

7	6	5	4	3	2	1	0
Reserved				Set/Reset Plane 3	Set/Reset Plane 2	Set/Reset Plane 1	Set/Reset Plane 0

### 7-4 Reserved

### 3-0 Set/Reset Plane 3 through Set/Reset Plane 0

When the Write Mode bits (bits 0 and 1) of the Graphics Mode Register (GR05) are set to select Write Mode 0, all 8 bits of each byte of each memory plane are set to either 1 or 0 as specified in the corresponding bit in this register if the corresponding bit in the Enable Set/Reset Register (GR01) is set to 1.

When the Write Mode bits (bits 0 and 1) of the Graphics Mode Register (GR05) are set to select Write Mode 3, all CPU data written to the frame buffer is rotated, then logically ANDed with the contents of the Bit Mask Register (GR08), and then treated as the addressed data's bit mask, while value of these four bits of this register are treated as the color value.

## GR01 Enable Set/Reset Register

Read/Write at I/O address 3CFh with 3CEh set to index 01h

7	6	5	4	3	2	1	0
Reserved				Enbl Set/ Reset Pln 3	Enbl Set/ Reset Pln 2	Enbl Set/ Reset Pln 1	Enbl Set/ Reset Pln 0

### 7-4 Reserved

### 3-0 Enable Set/Reset Plane 3 through Enable Set/Reset Plane 0

0 The corresponding memory plane can be read from or written to by the CPU without any special bitwise operations taking place.

1 The corresponding memory plane is set to 0 or 1 as specified in the Set/Reset Register (GR00).

This register works in conjunction with the Set/Reset Register (GR00). The Write Mode bits (bits 0 and 1) must be set for Write Mode 0 for this register to have any effect.

## GR02 Color Compare Register

Read/Write at I/O address 3CFh with 3CEh set to index 02h

7	6	5	4	3	2	1	0
Reserved				Color Comp Plane 3	Color Comp Plane 2	Color Comp Plane 1	Color Comp Plane 0

### 7-4 Reserved

### 3-0 Color Compare Plane 3 through Color Compare Plane 0

When the Read Mode bit (bit 3) of the Graphics Mode Register (GR05) is set to select Read Mode 1, all 8 bits of each byte of each of the 4 memory planes of the frame buffer corresponding to the address from which a CPU read access is being performed are compared to the corresponding bits in this register (if the corresponding bit in the Color Don't Care Register (GR07) is set to 1). The value that the CPU receives from the read access is an 8-bit value that shows the result of this comparison, wherein value of 1 in a given bit position indicates that all of the corresponding bits in the bytes across all of the memory planes that were included in the comparison had the same value as their memory plane's respective bits in this register.

## GR03 Data Rotate Register

Read/Write at I/O address 3CFh with 3CEh set to index 03h

7	6	5	4	3	2	1	0
Reserved			Function Select		Rotate Count		

### 7-5 Reserved

### 4-3 Function Select

These bits specify the logical function (if any) to be performed on data that is meant to be written to the frame buffer (using the contents of the memory read latch) just before it is actually stored in the frame buffer at the intended address location.

Bits 4 3	Result
0 0	Data being written to the frame buffer remains unchanged, and is simply stored in the frame buffer.
0 1	Data being written to the frame buffer is logically ANDed with the data in the memory read latch before it is actually stored in the frame buffer.
1 0	Data being written to the frame buffer is logically ORed with the data in the memory read latch before it is actually stored in the frame buffer.
1 1	Data being written to the frame buffer is logically XORed with the data in the memory read latch before it is actually stored in the frame buffer.

### 2-0 Rotate Count

These bits specify the number of bits to the right to rotate any data that is meant to be written to the frame buffer just before it is actually stored in the frame buffer at the intended address location.



## GR04 Read Plane Select Register

Read/Write at I/O address 3CFh with 3CEh set to index 04h

7	6	5	4	3	2	1	0
Reserved						Read Plane Select	

### 7-2 Reserved

### 1-0 Read Plane Select

These two bits select the memory plane from which the CPU reads data in Read Mode 0. In Odd/Even Mode, bit 0 of this register is ignored. In Chain 4 Mode, both bits 1 and 0 of this register are ignored. The four memory planes are selected as follows:

Bits 1 0	Plane Selected
0 0	Plane 0
0 1	Plane 1
1 0	Plane 2
1 1	Plane 3

These two bits also select which of the four memory read latches may be read via the Memory Read Latch Data Register (CR22). The choice of memory read latch corresponds to the choice of plane specified in the table above. The Memory Read Latch Data register and this additional function served by 2 bits are features of the VGA standard that were never documented by IBM.

## GR05 Graphics Mode Register

Read/Write at I/O address 3CFh with 3CEh set to index 05h

7	6	5	4	3	2	1	0
Reserved	Shift Register Control		Odd/ Even	Read Mode	Reserved	Write Mode	

### 7 Reserved

### 6-5 Shift Register Control

In standard VGA modes, pixel data is transferred from the 4 graphics memory planes to the palette via a set of 4 serial output bits. These 2 bits of this register control the format in which data in the 4 memory planes is serialized for these transfers to the palette.

- 0, 0 One bit of data at a time from parallel bytes in each of the 4 memory planes is transferred to the palette via the 4 serial output bits, with 1 of each of the serial output bits corresponding to a memory plane. This provides a 4-bit value on each transfer for 1 pixel, making possible a choice of 1 of 16 colors per pixel.

Serial Out	1st Xfer	2nd Xfer	3rd Xfer	4th Xfer	5th Xfer	6th Xfer	7th Xfer	8th Xfer
Bit 3	plane 3 bit 7	plane 3 bit 6	plane 3 bit 5	plane 3 bit 4	plane 3 bit 3	plane 3 bit 2	plane 3 bit 1	plane 3 bit 0
Bit 2	plane 2 bit 7	plane 2 bit 6	plane 2 bit 5	plane 2 bit 4	plane 2 bit 3	plane 2 bit 2	plane 2 bit 1	plane 2 bit 0
Bit 1	plane 1 bit 7	plane 1 bit 6	plane 1 bit 5	plane 1 bit 4	plane 1 bit 3	plane 1 bit 2	plane 1 bit 1	plane 1 bit 0
Bit 0	plane 0 bit 7	plane 0 bit 6	plane 0 bit 5	plane 0 bit 4	plane 0 bit 3	plane 0 bit 2	plane 0 bit 1	plane 0 bit 0

- 0, 1 Two bits of data at a time from parallel bytes in each of the 4 memory planes are transferred to the palette in a pattern that alternates per byte between memory planes 0 and 2, and memory planes 1 and 3. First the even-numbered and odd-numbered bits of a byte in memory plane 0 are transferred via serial output bits 0 and 1, respectively, while the even-numbered and odd-numbered bits of a byte in memory plane 2 are transferred via serial output bits 2 and 3. Next, the even-numbered and odd-numbered bits of a byte in memory plane 1 are transferred via serial output bits 0 and 1, respectively, while the even-numbered and odd-numbered bits of memory plane 3 are transferred via serial out bits 1 and 3. This provides a pair of 2-bit values (one 2-bit value for each of 2 pixels) on each transfer, making possible a choice of 1 of 4 colors per pixel.

Serial Out	1st Xfer	2nd Xfer	3rd Xfer	4th Xfer	5th Xfer	6th Xfer	7th Xfer	8th Xfer
Bit 3	plane 2 bit 7	plane 2 bit 5	plane 2 bit 3	plane 2 bit 1	plane 3 bit 7	plane 3 bit 5	plane 3 bit 3	plane 3 bit 1
Bit 2	plane 2 bit 6	plane 2 bit 4	plane 2 bit 2	plane 2 bit 0	plane 3 bit 6	plane 3 bit 4	plane 3 bit 2	plane 3 bit 0
Bit 1	plane 0 bit 7	plane 0 bit 5	plane 0 bit 3	plane 0 bit 1	plane 1 bit 7	plane 1 bit 5	plane 1 bit 3	plane 1 bit 1
Bit 0	plane 0 bit 6	plane 0 bit 4	plane 0 bit 2	plane 0 bit 0	plane 1 bit 6	plane 1 bit 4	plane 1 bit 2	plane 1 bit 0

This alternating pattern is meant to accommodate the use of the Odd/Even mode of organizing the 4 memory planes, which is used by standard VGA modes 2h and 3h.

## GR05 Graphics Mode Register (continued)

### 6-5 Shift Register Control (continued)

- 1, x Four bits of data at a time from parallel bytes in each of the 4 memory planes are transferred to the palette in a pattern that iterates per byte through memory planes 0 through 3. First the 4 most significant bits of a byte in memory plane 0 are transferred via the 4 serial output bits, followed by the 4 least significant bits of the same byte. Next, the same transfers occur from the parallel byte in memory planes 1, 2 and lastly, 3. Each transfer provides either the upper or lower half of an 8 bit value for the color for each pixel, making possible a choice of 1 of 256 colors per pixel.

Serial Out	1st Xfer	2nd Xfer	3rd Xfer	4th Xfer	5th Xfer	6th Xfer	7th Xfer	8th Xfer
Bit 3	plane 0 bit 7	plane 0 bit 3	plane 1 bit 7	plane 1 bit 3	plane 2 bit 7	plane 2 bit 3	plane 3 bit 7	plane 3 bit 3
Bit 2	plane 0 bit 6	plane 0 bit 2	plane 1 bit 6	plane 1 bit 2	plane 2 bit 6	plane 2 bit 2	plane 3 bit 6	plane 3 bit 2
Bit 1	plane 0 bit 5	plane 0 bit 1	plane 1 bit 5	plane 1 bit 1	plane 2 bit 5	plane 2 bit 1	plane 3 bit 5	plane 3 bit 1
Bit 0	plane 0 bit 4	plane 0 bit 0	plane 1 bit 4	plane 1 bit 0	plane 2 bit 4	plane 2 bit 0	plane 3 bit 4	plane 3 bit 0

This pattern is meant to accommodate mode 13h, a standard VGA 256-color graphics mode.

### 4 Odd/Even Mode

- 0 Addresses sequentially access data within a bit map, and the choice of which map is accessed is made according to the value of the Plane Mask Register (SR02).
- 1 The frame buffer is mapped in such a way that the function of address bit 0 such that even addresses select memory planes 0 and 2 and odd addresses select memory planes 1 and 3.

**Note:** This works in a way that is the inverse of (and is normally set to be the opposite of) bit 2 of the Memory Mode Register (SR04).

### 3 Read Mode

- 0 During a CPU read from the frame buffer, the value returned to the CPU is data from the memory plane selected by bits 1 and 0 of the Read Plane Select Register (GR04).
- 1 During a CPU read from the frame buffer, all 8 bits of the byte in each of the 4 memory planes corresponding to the address from which a CPU read access is being performed are compared to the corresponding bits in this register (if the corresponding bit in the Color Don't Care Register (GR07) is set to 1). The value that the CPU receives from the read access is an 8-bit value that shows the result of this comparison, wherein value of 1 in a given bit position indicates that all of the corresponding bits in the bytes across all 4 of the memory planes that were included in the comparison had the same value as their memory plane's respective bits in this register.

**GR05 Graphics Mode Register (continued)****2 Reserved****1-0 Write Mode**

- 0, 0 Write Mode 0 -- During a CPU write to the frame buffer, the addressed byte in each of the 4 memory planes is written with the CPU write data after it has been rotated by the number of counts specified in the Data Rotate Register (GR03). If, however, the bit(s) in the Enable Set/Reset Register (GR01) corresponding to one or more of the memory planes is set to 1, then those memory planes will be written to with the data stored in the corresponding bits in the Set/Reset Register (GR00).
- 0, 1 Write Mode 1 -- During a CPU write to the frame buffer, the addressed byte in each of the 4 memory planes is written to with the data stored in the memory read latches. (The memory read latches stores an unaltered copy of the data last read from any location in the frame buffer.)
- 1, 0 Write Mode 2 -- During a CPU write to the frame buffer, the least significant 4 data bits of the CPU write data is treated as the color value for the pixels in the addressed byte in all 4 memory planes. The 8 bits of the Bit Mask Register (GR08) are used to selectively enable or disable the ability to write to the corresponding bit in each of the 4 memory planes that correspond to a given pixel. A setting of 0 in a bit in the Bit Mask Register at a given bit position causes the bits in the corresponding bit positions in the addressed byte in all 4 memory planes to be written with value of their counterparts in the memory read latches. A setting of 1 in a Bit Mask Register at a given bit position causes the bits in the corresponding bit positions in the addressed byte in all 4 memory planes to be written with the 4 bits taken from the CPU write data to thereby cause the pixel corresponding to these bits to be set to the color value.
- 1, 1 Write Mode 3 -- During a CPU write to the frame buffer, the CPU write data is logically ANDed with the contents of the Bit Mask Register (GR08). The result of this ANDing is treated as the bit mask used in writing the contents of the Set/Reset Register (GR00) are written to addressed byte in all 4 memory planes.

## GR06 Miscellaneous Register

Read/Write at I/O address 3CFh with 3CEh set to index 06h

7	6	5	4	3	2	1	0
Reserved				Memory Map Mode		Chain Odd/Even	Graphics / Text Mode

### 7-4 Reserved

### 3-2 Memory Map Mode

These 2 bits control the mapping of the frame buffer into the CPU address space as follows:

Bits 3 2	Frame Buffer Address Range
0 0	A0000h - BFFFFh
0 1	A0000h - AFFFFh
1 0	B0000h - B7FFFh
1 1	B8000h - BFFFFh

**Note:** This function is both in standard VGA modes, and in extended modes that do not provide linear frame buffer access.

### 1 Chain Odd/Even

This bit provides the ability to alter the interpretation of address bit A0, so that it may be used in selecting between the odd-numbered memory planes (planes 1 and 3) and the even-numbered memory planes (planes 0 and 2).

0 A0 functions normally.

1 A0 is switched with a high order address bit, in terms of how it is used in address decoding. The result is that A0 is used to determine which memory plane is being accessed:

A0 = 0: planes 0 and 2

A0 = 1: planes 1 and 3

### 0 Graphics/Text Mode

0 Selects text mode.

1 Selects graphics mode.

## GR07 Color Don't Care Register

Read/Write at I/O address 3CFh with 3CEh set to index 07h

7	6	5	4	3	2	1	0
Reserved				Ignore Color Plane 3	Ignore Color Plane 2	Ignore Color Plane 1	Ignore Color Plane 0

### 7-4 Reserved

### 3-0 Ignore Color Plane 3 through Ignore Color Plane 0

- 0 The corresponding bit in the Color Compare Register (GR02) will not be included in color comparisons.
- 1 The corresponding bit in the Color Compare Register (GR02) is used in color comparisons.

**Note:** These bits have effect only when bit 3 of the Graphics Mode Register (GR05) is set to 1 to select read mode 1.

## GR08 Bit Mask Register

Read/Write at I/O address 3CFh with 3CEh set to index 08h

7	6	5	4	3	2	1	0
Bit Mask							

### 7-0 Bit Mask

- 0 The corresponding bit in each of the 4 memory planes is written to with the corresponding bit in the memory read latches.
- 1 Manipulation of the corresponding bit in each of the 4 memory planes via other mechanisms is enabled.

**Note:** This bit mask applies to any writes to the addressed byte of any or all of the 4 memory planes, simultaneously.

**Note:** This bit mask is applicable to any data written into the frame buffer by the CPU, including data that is also subject to rotation, logical functions (AND, OR, XOR), and Set/Reset. To perform a proper read-modify-write cycle into frame buffer, each byte must first be read from the frame buffer by the CPU (and this will cause it to be stored in the memory read latches), this Bit Mask Register must be set, and the new data then written into the frame buffer by the CPU.

## **CHAPTER 12**

# **ATTRIBUTE CONTROLLER REGISTERS**

**This page intentionally left blank.**



## 12.0 ATTRIBUTE CONTROLLER REGISTERS

Name	Function	Access	Index
AR00-AR0F	Color Data Registers	Read/Write	00-0F
AR10	Mode Control Register	Read/Write	10
AR11	Overscan Color Register	Read/Write	11
AR12	Memory Plane Enable Register	Read/Write	12
AR13	Horizontal Pixel Panning Register	Read/Write	13
AR14	Color Select Register	Read/Write	14

Unlike the other sets of sub-addressed registers, the attribute controller registers are not accessed through a scheme employing entirely separate index and data ports. I/O address 3C0h is used both as the read and write for the index register, and as the write address for the data port. I/O address 3C1h is the read address for the data port.

To write to one of the attribute controller registers, the index of the desired register must be written to I/O address 3C0h, and then the data is written to the very same I/O address. A flip-flop alternates with each write to I/O address 3C0h to change its function from writing the index to writing the actual data, and back again. This flip-flop may be deliberately set so that I/O address 3C0h is set to write to the index (which provides a way to set it to a known state) by performing a read operation from Input Status Register 1 (ST01) at I/O address 3BAh or 3DAh (depending on whether the graphics system has been set to emulate an MDA or a CGA).

To read from one of the attribute controller registers, the index of the desired register must be written to I/O address 3C0h, and then the data is read from I/O address 3C1h. A read operation from I/O address 3C1h does not reset the flip-flop to writing to the index. Only a write to 3C0h or a read from 3BAh or 3DAh, as described above, will toggle the flip-flop back to writing to the index.

## ARX Attribute Controller Index Register

Read/Write at I/O address 3C0h

7	6	5	4	3	2	1	0
Reserved		Video/Reg Enable	Attribute Controller Register Index				

### 7-6 Reserved

### 5 Video Output / Register Access Enable

- 0 Disables video, allowing the attribute controller color registers (AR0 0-AR0F) to be accessed by the CPU.
- 1 Enables video, causing the attribute controller color registers (AR00-AR0F) to be rendered inaccessible by the CPU.

**Note:** In the VGA standard, this is called the “Palette Address Source” bit.

### 4-0 Attribute Controller Register Index

These five bits are used to select any one of the attribute controller registers, AR00 through AR14, to be accessed.

**Note:** AR12 is referred to in the VGA standard as the Color Plane Enable Register. The words “plane,” “color plane,” “display memory plane,” and “memory map” have been all been used in IBM™ literature on the VGA standard to describe the four separate regions in the frame buffer where the pixel color or attribute information is split up and stored in standard VGA planar modes. This use of multiple terms for the same subject was deemed to be confusing, therefore, AR12 is called the Memory Plane Enable Register.

## AR00-AR0F Palette Registers 0-F

Read at 3C1h, Write at 3C0h with 3C0h set to indexes 00h to 0Fh

7	6	5	4	3	2	1	0
Reserved		Palette Bits P5-P0					

### 7-6 Reserved

### 5-0 Palette Bits P5-P0

In each of these 16 registers, these are the lower 6 of 8 bits that are used to map either text attributes or pixel color input values (for modes that use 16 colors) to the 256 possible colors available to be selected in the palette.

**Note:** Bits 3 and 2 of the Color Select Register (AR14) supply bits P7 and P6 for the values contained in all 16 of these registers. Bits 1 and 0 of the Color Select Register (AR14) can also replace bits P5 and P4 for the values contained in all 16 of these registers if bit 7 of the Mode Control Register (AR10) is set to 1.

## AR10 Mode Control Register

Read at 3C1h, Write at 3C0h with 3C0h set to index 10h

7	6	5	4	3	2	1	0
Palette Bits P5, P4 Select	Pixel Width/Clock Select	Pixel Panning Compat	Reserved	En Blink/ Select Bkgnd Int	En Line Gr Char Code	Select Display Type	Graphics/ Alpha Mode

### 7 Palette Bits P5, P4 Select

- 0 P5 and P4 for each of the 16 selected colors (for modes that use 16 colors) are individually provided by bits 5 and 4 of their corresponding Palette Registers (AR00-0F).
- 1 P5 and P4 for all 16 of the selected colors (for modes that use 16 colors) are provided by bits 1 and 0 of Color Select Register (AR14).

### 6 Pixel Width/Clock Select

- 0 Six bits of video data (translated from 4 bits via the palette) are output every dot clock.
- 1 Two sets of 4 bits of data are assembled to generate 8 bits of video data which is output every other dot clock, and the Palette Registers (AR00-0F) are bypassed.

**Note:** This bit is set to 0 for all of the standard VGA modes, except mode 13h.

### 5 Pixel Panning Compatibility

- 0 Scroll both the upper and lower screen regions horizontally as specified in the Pixel Panning Register (AR13).
- 1 Scroll only the upper screen region horizontally as specified in the Pixel Panning Register (AR13).

**Note:** This bit has application only when split-screen mode is being used, where the display area is divided into distinct upper and lower regions which function somewhat like separate displays.

### 4 Reserved

### 3 Enable Blinking/Select Background Intensity

- 0 Disables blinking in graphics modes, and for text modes, sets bit 7 of the character attribute bytes to control background intensity, instead of blinking.
- 1 Enables blinking in graphics modes and for text modes, sets bit 7 of the character attribute bytes to control blinking, instead of background intensity.

**Note:** The blinking rate is derived by dividing the VSYNC signal. The Blink Rate Control Register (FR19) defines the blinking rate.

**AR10 Mode Control Register (continued)****2 Enable Line Graphics Character Code**

- 0 Every 9th pixel of a horizontal line (i.e., the last pixel of each horizontal line of each 9-pixel wide character box) is assigned the same attributes as the background of the character of which the given pixel is a part.
- 1 Every 9th pixel of a horizontal line (i.e., the last pixel of each horizontal line of each 9-pixel wide character box) is assigned the same attributes as the 8th pixel of the character of which the given pixel is a part. This setting is intended to accommodate the line-drawing characters of the PC's extended ASCII character set -- characters with an extended ASCII code in the range of B0h to DFh.

**Note:** In IBM<sup>®</sup> literature describing the VGA standard, the range of extended ASCII codes that are said to include the line-drawing characters is mistakenly specified as C0h to DFh, rather than the correct range of B0h to DFh.

**1 Select Display Type**

- 0 Attribute bytes in text modes are interpreted as they would be for a color display.
- 1 Attribute bytes in text modes are interpreted as they would be for a monochrome display.

**0 Graphics/Alphanumeric Mode**

- 0 Selects alphanumeric (text) mode.
- 1 Selects graphics mode.

## **AR11 Overscan Color Register**

Read at 3C1h, Write at 3C0h with 3C0h set to index 11h

7	6	5	4	3	2	1	0
Overscan Color							

### **7-0 Overscan**

These 8 bits select the overscan (border) color. The border color is displayed during the blanking intervals. For monochrome displays, this value should be set to 00h.

## AR12 Memory Plane Enable Register

Read at 3C1h, Write at 3C0h with 3C0h set to index 12h

7	6	5	4	3	2	1	0
Reserved		Video Status Mux		Enable Plane 3	Enable Plane 2	Enable Plane 1	Enable Plane 0

### 7-6 Reserved

### 5-4 Video Status Mux

These 2 bits are used to select 2 of the 8 possible palette bits (P7-P0) to be made available to be read via bits 5 and 4 of the Input Status Register 1 (ST01). The table below shows the possible choices.

Palette Bits Made Readable Via Register ST01		
Bits 5 4	Bit 5 of ST01	Bit 4 of ST01
0 0	P2	P0
0 1	P5	P4
1 0	P3	P1
1 1	P7	P6

**Note:** These bits are largely unused by current software. They are provided for EGA compatibility.

### 3-0 Enable Plane 3-0

These 4 bits individually enable the use of each of the 4 memory planes in providing 1 of the 4 bits used in video output to select 1 of 16 possible colors from the palette to be displayed.

- 0 Disable the use of the corresponding memory plane in video output to select colors, forcing the bit that the corresponding memory plane would have provided to a value of 0.
- 1 Enable the use of the corresponding memory plane in video output to select colors.

**Note:** AR12 is referred to in the VGA standard as the Color Plane Enable Register. The words “plane,” “color plane,” “display memory plane,” and “memory map” have been all been used in IBM<sup>®</sup> literature on the VGA standard to describe the 4 separate regions in the frame buffer that are amongst which pixel color or attributes information is split up and stored in standard VGA planar modes. This use of multiple terms for the same subject was deemed to be confusing, therefore AR12 is called the Memory Plane Enable Register.

## AR13 Horizontal Pixel Panning Register

Read at 3C1h, Write at 3C0h with 3C0h set to index 13h

7	6	5	4	3	2	1	0
Reserved				Leftward Horizontal Pixel Shift			

### 7-4 Reserved

### 3-0 Leftward Horizontal Pixel Shift

Bits 3-0 of this register hold a 4-bit value that selects number of pixels by which the image is shifted horizontally to the left. This function is available in both text and graphics modes.

In text modes with a 9-pixel wide character box, the image can be shifted up to 9 pixels to the left. In text modes with an 8-pixel wide character box, and in graphics modes other than those with 256 colors, the image can be shifted up to 8 pixels to the left.

In standard VGA mode 13h (where bit 6 of the Mode Control Register, AR10, is set to 1 to support 256 colors), bit 0 of this register must remain set to 0, and the image may be shifted up to only 4 pixels to the left. In this mode, the number of pixels by which the image is shifted can be further controlled using bits 6 and 5 of the Preset Row Scan Register (CR08).

Value in Bits 3-0	Number of Pixels by Which Image is Shifted to the Left		
	9-Pixel Wide Text Mode	8-Pixel Wide Text & Graphics Modes (Except 256-Color)	256-Color Graphics Mode
0h	1	0	0
1h	2	1	Undefined
2h	3	2	1
3h	4	3	Undefined
4h	5	4	2
5h	6	5	Undefined
6h	7	6	3
7h	8	7	Undefined
8h	0	Undefined	Undefined



## AR14 Color Select Register

Read at 3C1h, Write at 3C0h with 3C0h set to index 14h

7	6	5	4	3	2	1	0
Reserved				P7	P6	Alt P5	Alt P4

### 7-4 Reserved

### 3-2 Palette Bits P7 and P6

These are the 2 upper-most of the 8 bits that are used to map either text attributes or pixel color input values (for modes that use 16 colors) to the 256 possible colors contained in the palette. These 2 bits are common to all 16 sets of bits P5 through P0 that are individually supplied by Palette Registers 0-F (AR00-AR0F).

### 1-0 Alternate Palette Bits P5 and P4

These 2 bits can be used as an alternate version of palette bits P5 and P4. Unlike the P5 and P4 bits that are individually supplied by Palette Registers 0-F (AR00-AR0F), these 2 alternate palette bits are common to all 16 of Palette Registers. Bit 7 of the Mode Control Register (AR10) is used to select between the use of either the P5 and P4 bits that are individually supplied by the 16 Palette Registers or these 2 alternate palette bits.

**This page intentionally left blank.**

## **CHAPTER 13**

# ***PALETTE DAC REGISTERS***

**This page intentionally left blank.**

## 13.0 PALETTE DAC REGISTERS

Name	Function	Access	I/O Address
DACMASK	Pixel Data Mask Register	Read/Write	3C6h
DACSTATE	DAC State Register	Read-Only	3C7h
DACRX	Palette Read Index Register	Write-Only	3C7h
DACWX	Palette Write Index Register	Read/Write	3C8h
DACDATA	Palette Data Register	Read/Write	3C9h

The palette DAC has two main components: a palette in which a selection of 256 colors may be stored, and a set of three digital to analog (D-to-A) converters, one each for the red, green and blue components used to produce a color on a CRT display. The palette DAC is also frequently called the RAMDAC, to emphasize the presence of memory alongside the three D-to-A converters, and the palette, itself, is often referred to as the CLUT or color look-up table.

During normal use, the palette DAC is operated either in direct-color mode or indexed-color mode. Direct color mode is used with pixel depths of 15, 16, or 24 bits per pixel. In direct color mode, the pixel data received from the frame buffer, through the sequencer and the attribute controller, directly specifies the color for a given pixel. This pixel data is pre-formatted such that certain bits of the pixel data for each pixel are used to provide the red, green and blue output values for each of the three corresponding 8-bit D-to-A converters. Indexed-color mode is used with pixel depths of 8 bits per pixel or less. In indexed-color mode, the incoming pixel data for each pixel is actually an 8-bit index that is used to choose one of the 256 color data positions within the palette. Each color data position holds a 24-bit color value that specifies the actual 8-bit red, green, and blue values for each of the three corresponding 8-bit D-to-A converters. In essence, the colors for each pixel are specified indirectly, with the actual choice of colors taking place in the color data positions of the palette, while the incoming pixel data chooses from among these color data positions. This method allows the full range of over 16 million possible colors to be accessible in modes with only 8 or fewer bits per pixel.

The color data stored in these 256 color data positions can be accessed only through a complex sub-addressing scheme, using a data register and two index registers. The Palette Data Register at I/O address 3C9h is the data port. The Palette Read Index Register at I/O address 3C7h and the Palette Write Index Register at I/O address 3C8h are the two index registers. The Palette Read Index Register is the index register that is used to choose the color data position that is to be read from via the data port, while the Palette Write Index Register is the index register that is used to choose the color data position that is to be written to through the same data port. This arrangement allows the same data port to be used for reading from and writing to two different color data positions. Reading and writing the color data at a color data position involves three successive reads or writes since the color data stored at each color data position consists of three bytes.

To read a color data position, the index of the desired color data position must first be written to the Palette Read Index Register. Then all three bytes of data in a given color data position may be read at the Palette Data Register. The first byte read from the Palette Data Register retrieves the 8-bit value specifying the intensity of the red color component, while the second and third bytes read are the corresponding 8-bit values for the green and blue color components, respectively. After completing the third read operation, the Palette Read Index Register is automatically incremented so that the data of the next color data position becomes accessible for being read. This allows the contents of all 256 color data positions of the palette to be read by specifying only the index of the 0th color data position in the Palette Read Index

Register, and then simply performing 768 successive reads from the Palette Data Register.

Writing a color data position entails a very similar procedure. The index of the desired color data position must first be written to the Palette Write Index Register. Then all three bytes of data to specify a given color may be written to the Palette Data Register. The first byte written to the Palette Data Register specifies the intensity of the red color component, the second byte specifies the intensity for the green color component, and the third byte specifies the same for the blue color component. One important detail is that all three of these bytes must be written before the hardware will actually update these three values in the given color data position. When all three bytes have been written, the Palette Write Index Register is automatically

incremented so that the data of the next color data position becomes accessible for being written. This allows the contents of all 256 color data positions of the palette to be written by specifying only the index of the 0th color data position in the Palette Write Index Register, and then simply performing 768 successive writes to the Palette Data Register.

In addition to the standard set of 256 color data positions of the palette, there is also an alternate set of 8 color data positions used to specify the colors used to draw cursors 1 and 2, and these are also accessed using the very same sub-addressing scheme. A bit in the Pixel Pipeline Configuration 0 Register (XR80) determines whether the standard 256 color data positions or the alternate 8 color data positions are to be accessed through this sub-addressing scheme.

## DACMASK PIXEL DATA MASK REGISTER

Read/Write at I/O Address 3C6h

7	6	5	4	3	2	1	0
Pixel Data Mask							

### 7-0 Pixel Data Mask

In indexed-color mode, the 8 bits of this register are logically ANDed with the 8 bits of pixel data received from the frame buffer for each pixel. The result of this ANDing process becomes the actual index used to select color data positions within the palette. This has the effect of limiting the choice of color data positions that may be specified by the incoming 8-bit data.

A value of 0 in a bit in this register results in the corresponding bit in the resulting 8-bit index being forced to 0, while a value of 1 in a bit in this register allows the corresponding bit in the resulting index to reflect the actual value of the corresponding bit in the incoming 8-bit pixel data.

In direct-color mode, the palette is not used, and the data in this register is entirely ignored.

## DACSTATE DAC STATE REGISTER

Read-only at I/O Address 3C7h

7	6	5	4	3	2	1	0
Reserved						DAC State	

### 7-2 Reserved

### 1-0 DAC State

These indicate which of the two index registers was most recently written to.

- 00 Palette Write Index Register at I/O Address 3C8h
- 01 Undefined
- 10 Palette Read Index Register at I/O Address 3C7h
- 11 Undefined

## DACRX PALETTE READ INDEX REGISTER

Write-only at I/O Address 3C7h

7	6	5	4	3	2	1	0
Palette Read Index							

### 7-0 Palette Read Index

The 8-bit index value programmed into this register chooses which of 256 standard color data positions within the palette (or which of 8 alternate color data positions, depending on the state of a bit in the Pixel Pipeline Control 0 Register) are to be made accessible for being read from via the Palette Data Register (DACDATA).

The index value held in this register is automatically incremented when all three bytes of the color data position selected by the current index have been read.

## DACWX PALETTE WRITE INDEX REGISTER

Read/Write at I/O Address 3C8

7	6	5	4	3	2	1	0
Palette Write Index							

### 7-0 Palette Write Index

The 8-bit index value programmed into this register chooses which of 256 standard color data positions within the palette (or which of 8 alternate color data positions, depending on the state of a bit in the Pixel Pipeline Control 0 Register) are to be made accessible for being written to via the Palette Data Register (DACDATA).

The index value held in this register is automatically incremented when all three bytes of the color data position selected by the current index have been written.



## DACDATA PALETTE DATA REGISTER

Read/Write at I/O Address 3C9h

7	6	5	4	3	2	1	0
Palette Data							

### 7-0 Palette Data

This byte-wide data port provides read or write access to the three bytes of data of each color data position selected using the Palette Read Index Register (DACRX) or the Palette Write Index Register (DACWX).

The three bytes in each color data position are read or written in three successive read or write operations. The first byte read or written always specifies the intensity of the red component of the color specified in the selected color data position. The second byte is always for the green component, and the third byte is always for the blue component.

When writing data to a color data position, all three bytes must be written before the hardware will actually update the three bytes of the selected color data position.

When reading or writing to a color data position, it is important to ensure that neither the Palette Read Index Register (DACRX) or the Palette Write Index Register (DACWX) are written to before all three bytes are read or written. A write to either of these two registers causes the circuitry that automatically cycles through providing access to the bytes for red, green and blue components to be reset such that the byte for the red component is the one that will be accessed by the next read or write operation via this register.

**This page intentionally left blank**

# ***CHAPTER 14***

# ***EXTENSION REGISTERS***

**This page intentionally left blank.**

## 14.0 EXTENSION REGISTERS

Name	Register Function	Access Port 3D7	Index Value Port 3D6 (XRX)
XR00	Vendor ID Low	Read-Only	00h
XR01	Vendor ID High	Read-Only	01h
XR02	Device ID Low	Read-Only	02h
XR03	Device ID High	Read-Only	03h
XR04	Revision ID	Read-Only	04h
XR05	Linear Base Address Low Register	Read-Only	05h
XR06	Linear Base Address High Register	Read-Only	06h
XR08	Host Bus Configuration Register	Read-Only	08h
XR09	Access Control Register	Read/Write	09h
XR0A	Frame Buffer Mapping Register	Read/Write	0Ah
XR0B	PCI Burst Write Support Register	Read/Write	0Bh
XR0E	Frame Buffer Page Select Register	Read/Write	0Eh
XR10	Subsystem Vendor ID Low	Read/Write	10h
XR11	Subsystem Vendor ID High	Read/Write	11h
XR12	Subsystem Vendor Device ID Low	Read/Write	12h
XR13	Subsystem Vendor Device ID High	Read/Write	13h
XR20	BitBLT Engine Configuration Register	Read/Write	20h
XR40	DRAM Access Control Register	Read/Write	40h
XR41	DRAM Type Register	Read/Write	41h
XR42	DRAM Configuration Register	Read/Write	42h
XR43	DRAM Interface Register	Read/Write	43h
XR44	DRAM Timing Register	Read/Write	44h
XR60	Video Pin Control Register	Read/Write	60h
XR61	DPMS Sync Control Register	Read/write	61h
XR62	GPIO Pin Control Register	Read/Write	62h
XR63	GPIO Pin Data Register	Read/Write	63h
XR67	Pin Tri-State Control Register	Read/Write	67h
XR70	Configuration Pins 0 Register	Read-Only	70h
XR71	Configuration Pins 1 Register	Read-Only	71h
XR80	Pixel Pipeline Configuration 0 Register	Read/Write	80h
XR81	Pixel Pipeline Configuration 1 Register	Read/Write	81h
XR82	Pixel Pipeline Configuration 2 Register	Read/Write	82h

Name	Register Function	Access Port 3D7	Index Value Port 3D6 (XRX)
XRA0	Cursor 1 Control Register	Read/Write	A0h
XRA1	Cursor 1 Vertical Extension Register	Read/Write	A1h
XRA2	Cursor 1 Base Address Low Register	Read/Write	A2h
XRA3	Cursor 1 Base Address High Register	Read/Write	A3h
XRA4	Cursor 1 X-Position Low Register	Read/Write	A4h
XRA5	Cursor 1 X-Position High Register	Read/Write	A5h
XRA6	Cursor 1 Y-Position Low Register	Read/Write	A6h
XRA7	Cursor 1 Y-Position High Register	Read/Write	A7h
XRA8	Cursor 2 Control Register	Read/Write	A8h
XRA9	Cursor 2 Vertical Extension Register	Read/Write	A9h
XRAA	Cursor 2 Base Address Low Register	Read/Write	AAh
XRAB	Cursor 2 Base Address High Register	Read/Write	ABh
XRAC	Cursor 2 X-Position Low Register	Read/Write	ACH
XRAD	Cursor 2 X-Position High Register	Read/Write	ADh
XRAE	Cursor 2 Y-Position Low Register	Read/Write	A Eh
XRAF	Cursor 2 Y-Position High Register	Read/Write	AFh
XRC0	Dot Clock 0 VCO M-Divisor Low Register	Read/Write	C0h
XRC1	Dot Clock 0 VCO N-Divisor Low Register	Read/Write	C1h
XRC2	Dot Clock 0 VCO M/N-Divisor High Register	Read/Write	C2h
XRC3	Dot Clock 0 Divisor Select Register	Read/Write	C3h
XRC4	Dot Clock 1 VCO M-Divisor Low Register	Read/Write	C4h
XRC5	Dot Clock 1 VCO N-Divisor Low Register	Read/Write	C5h
XRC6	Dot Clock 1 VCO M/N-Divisor High Register	Read/Write	C6h
XRC7	Dot Clock 1 Divisor Select Register	Read/Write	C7h
XRC8	Dot Clock 2 VCO M-Divisor Low Register	Read/Write	C8h
XRC9	Dot Clock 2 VCO N-Divisor Low Register	Read/Write	C9h
XRCA	Dot Clock 2 VCO M/N-Divisor High Register	Read/Write	CAh
XRCB	Dot Clock 2 Divisor Select Register	Read/Write	CBh
XRCC	Memory Clock VCO M-Divisor Register	Read/Write	CCh
XRCD	Memory Clock VCO N-Divisor Register	Read/Write	CDh
XRCE	Memory Clock VCO Divisor Select Register	Read/Write	CEh
XRCF	Clock Configuration Register	Read/Write	CFh
XRD0	Powerdown Control Register	Read/Write	D0h
XRD1	Power Conservation Control Register	Read/Write	D1h
XRD2	2KHz Down Counter Register	Read-Only	D2h
XRE0-XREB	Software Flag Registers	Read/Write	E0h-E9h
XRF8-XRFC	Test Registers	Read/Write	F8h-FCh

***XRX Extension Register Index Register***

Read/Write at I/O Address 3D6h

This register is cleared to 00h by reset

7	6	5	4	3	2	1	0
Extension Register Index (0000:0000)							

**7-0 Extension Register Index**

These 8 bits are used to select any one of the extension registers to be accessed via the data port at I/O location 3D7h.

## XR00 Vendor ID Low

Read-Only at I/O Address 3D7h with 3D6h set to Index 00h

7	6	5	4	3	2	1	0
Vendor ID Bits 7-0 (2Ch)							

### 7-0 Vendor ID Bits 7-0

These 8 bits always carry the value 2Ch. This is the lower byte of CHIPS' vendor ID for PCI devices. Both bytes of this ID are also readable from the Vendor ID register at offset 00h in the PCI configuration space.

## XR01 Vendor ID High

Read-Only at I/O Address 3D7h with 3D6h set to Index 01h

7	6	5	4	3	2	1	0
Vendor ID Bits 15-8 (10h)							

### 7-0 Vendor ID Bits 15-8

These 8 bits always carry the value 10h. This is the upper byte of CHIPS' vendor ID for PCI devices. Both bytes of this ID are also readable from the Vendor ID register at offset 00h in the PCI configuration space.



### **XR02 Device ID Low**

Read-Only at I/O Address 3D7h with 3D6h set to Index 02h

7	6	5	4	3	2	1	0
Device ID Bits 7-0 (E5h)							

#### **7-0 Device ID Bits 7-0**

These bits always carry the value E5h. This is the lower byte of the 65555's device ID as a PCI device. Both bytes of this ID are also readable from the Device ID register at offset 02h in the PCI configuration space.

### **XR03 Device ID High**

Read-Only at I/O Address 3D7h with 3D6h set to Index 03h

7	6	5	4	3	2	1	0
Device ID High (00h)							

#### **7-0 Device ID High**

These bits always carry the value 00h. This is the upper byte of the 65555's device ID as a PCI device. Both bytes of this ID are also readable from the Device ID register at offset 02h in the PCI configuration space.

## XR04 Revision ID

Read-Only at I/O Address 3D7h with 3D6h set to Index 04h

7	6	5	4	3	2	1	0
Chip Manufacturing Code (xxxx)				Chip Revision Code (xxxx)			

**Note:** This register is identical to the Revision register (REV) at offset 08h in the PCI configuration space.

### 7-4 Chip Manufacturing Code

These four bits carry the fabrication code.

### 3-0 Chip Revision Code

These four bits carry the revision code. Revision codes start at 0 and are incremented for each new silicon revision.

## XR05 Linear Base Address Low Register

Read-Only at I/O Address 3D7h with 3D6h set to Index 05h

7	6	5	4	3	2	1	0
Mem Space Base Bit 23 (0)	Reserved (000:0000)						

### 7 Memory Space Base Address Bit 23

This bit is provided only for backward compatibility, only. It is a hold-over from earlier CHIPS graphics controllers.

The 65555 requires a 16MB memory space on the host bus through which the linear frame buffer and memory-mapped registers are accessed. This 16MB memory space always begins on a 16MB address boundary, and so bit 23 of the linear base address of this 16MB memory space always has the value of 0. Therefore, this bit always returns the value of 0 when read. This base address is set through the MBASE register at offset 10h in the PCI configuration space.

### 6-0 Reserved

These bits always return the value of 0 when read.

## XR06 Linear Base Address High Register

Read-Only at I/O Address 3D7h with 3D6h set to Index 06h

7	6	5	4	3	2	1	0
Memory Space Base Address Bits 31-24 (xxxx:xxxx)							

### 7-0 Memory Space Base Address Bits 31-24

The 65555 requires a 16MB memory space on the host bus through which the linear frame buffer and memory-mapped registers are accessed. These 8 bits provide read-only access to bits 31-24, the 8 most significant bits of the linear base address at which the 16MB memory space begins. This base address is set through the MBASE register at offset 10h in the PCI configuration space.

## XR08 Host Bus Configuration Register

Read-Only at I/O Address 3D7h with 3D6h set to Index 08h

7	6	5	4	3	2	1	0
Reserved (0000:00)						PCI VGA Addr Dec (x)	Reserved (0)

### 7-2 Reserved

These bits always return the value of 0 when read.

### 1 PCI VGA Address Decode Enable

This bit reflects the state of memory interface address pin AA1 during reset.

- 0 Indicates that VGA I/O Address decoding is disabled on the PCI Bus, so access to the registers via I/O read and write operations is disabled.
- 1 Indicates that VGA I/O Address decoding is enabled on the PCI Bus, so access to the registers via I/O read and write operations is enabled.

**Note:** The reset state of this pin is also readable via bit 1 of the Configuration Pins 0 Register (XR70).

### 0 Reserved

This bit always returns the value of 0 when read.

## XR09 Access Control Register

Read/Write at I/O Address 3D7h with 3D6h set to Index 09h

7	6	5	4	3	2	1	0
Planar Off Shift En (0)	Reserved (000:00)					Attr Ctrl Ext Enable (0)	CRT Ctrl Ext Enable (0)

### 7 Planar Mode Frame Buffer Mapping Offset Shifting Enable

- 0 Disables planar mode frame buffer mapping offset shifting.
- 1 Enables planar mode frame buffer mapping offset shifting. When accessing the frame buffer through the A0000h-AFFFFh memory space in planar mode, setting this bit to 1 causes bits 1 and 0 of the Frame Buffer Page Select Register (XR0E) to be added to bits 15 and 14, respectively, of the address used in accessing the frame buffer through that memory space.

Note: This feature should only be enabled when the graphics controller has been configured for planar mode -- otherwise, unpredictable results may occur.

### 6-2 Reserved

These bits always return the value of 0 when read.

### 1 Attribute Controller Extensions Enable

- 0 Allow the sub-addressing scheme used to access the attribute controller registers to remain unchanged from the VGA standard. This is the default after reset.
- 1 Alter attribute controller sub-addressing scheme used to access the attribute controller registers so that I/O Address 3C0h is used solely as the index register and I/O Address 3C1h is used as the data port for both read and write operations.

### 0 CRT Controller Extensions Enable

- 0 Use only the CRT controller registers defined in the VGA standard to extend the number of bits used to specify the timing, resolution, and addressing parameters to beyond eight bits. This is the default after reset.
- 1 Use only the additional CHIPS CRT controller registers to extend the number of bits used to specify the timing, resolution, and addressing parameters to beyond eight bits.

## XR0A Frame Buffer Mapping Register

Read/Write at I/O Address 3D7h with 3D6h set to Index 0Ah

7	6	5	4	3	2	1	0
Reserved (00)		Endian Byte Swapping Control (00)		Reserved (0)	Planar to Non X-late (0)	Linear Mapping (0)	Paged Mapping (0)

### 7-6 Reserved

These bits always return the value of 0 when read.

### 5-4 Endian Byte Swapping Control

These 2 bits enable and select the type of byte-swapping performed on all word and doubleword data written to and read from the graphics controller by the CPU as follows:

Bits 5 4	Type of Endian Byte Swapping
0 0	No byte swapping. This is the default after reset.
0 1	Performs byte swapping wherein byte 0 is swapped with byte 1, and byte 2 is swapped with byte 3.
1 0	Performs byte swapping wherein byte 0 is swapped with byte 3, and byte 1 is swapped with byte 2.
1 1	Reserved

### 3 Reserved

This bit always returns the value of 0 when read.

### 2 Planar to Non-Planar Address Translation Enable

This bit provides a single-bit switch that can be used to alter the manner in which the frame buffer memory appears from the perspective of the host bus to be organized so that it looks as though the bits for each pixel are organized sequentially, rather than in planes, even though it may well still be organized in planes. This is done through a hardware-based address translation scheme. The result is intended to be very similar to setting the frame buffer memory to chain-4 mode using the graphics controller registers.

This switch is meant to be turned on occasionally as a convenience to programmers when the graphics controller is being used in standard VGA modes, in order to allow a given drawing operation or frame buffer save or restore operation to be carried out more easily. Altering this bit has no effect on the settings in the graphics controller registers (the GRxx series registers) that are normally used to specify the way in which the frame buffer memory is organized. It is recommended, however, that bits 3 and 2 of the Miscellaneous Register (GR06) be set so that the frame buffer memory is accessible using the A0000-AFFFF memory space during the time that this feature is used.

0 Disables address translation in support of packed mode. This is the default after reset.

1 Enables address translation in support of packed mode.

***XR0A Frame Buffer Mapping Register (continued)*****1 Frame Buffer Linear Mapping Enable**

- 0 Disables the linear mapping of the frame buffer.
- 1 Enables the linear mapping of the frame buffer.

**0 Frame Buffer Page Mapping Enable**

- 0 Disables the mapping of the frame buffer in 64KB pages into the A0000h-AFFFFh memory address space.
- 1 Enables the mapping of the frame buffer in 64KB pages into the A0000h-AFFFFh memory address space.

**Note:** The selection of which 64KB page is to mapped into memory addresses A0000h-AFFFFh is made using bits 6-0 of the Page Selector Register (XR0E).

## XR0B PCI Burst Write Support Register

Read/Write at I/O Address 3D7h with 3D6h set to Index 0Bh

7	6	5	4	3	2	1	0
Reserved (0000)				Burst Wrt Depth (0)	Burst Write Depth (0)	Reserved (0)	Burst Write Enable (0)

### 7-4 Reserved

These bits always return the value of 0 when read.

### 3 Font Expansion PCI Burst Write Buffer Depth

- 0 The buffer used to receive PCI burst writes is always 4 or 8 doublewords deep as selected by bit 2 of this register, regardless of whether or not font expansion is being used. This is the default after a reset.
- 1 The buffer used to receive PCI burst writes is limited to being 1 doubleword deep when the font expansion feature is being used.

### 2 PCI Burst Write Buffer Depth

- 0 The buffer used to receive PCI burst writes is set to be 8 doublewords deep.
- 1 The buffer used to receive PCI burst writes is set to be 4 doublewords deep.

**Note:** The use of this bit to choose the depth of the PCI burst write buffer can be overridden by bit 3 of this register.

### 1 Reserved

This bit always returns the value of 0 when read.

### 0 PCI Burst Write Support Enable

- 0 Disables support for receiving PCI burst write cycles.
- 1 Enables support for receiving PCI burst write cycles.



## ***XR0E Frame Buffer Page Select Register***

Read/Write at I/O Address 3D7h with 3D6h set to Index 0Eh

7	6	5	4	3	2	1	0
Reserved (0)	Page Select (000:0000)					Added CPU Address (0)	

### **7 Reserved**

This bit always returns the value of 0 when read.

### **6-2 Page Select**

These seven bits select which 64KB page of the frame buffer is to be mapped into the A0000h-AFFFFh memory address space.

**Note:** Bit 0 of the Address Mapping Register (XR0A) must be set to 1 to enable this mapping feature.

### **1-0 Planar Mode Frame Buffer Mapping Offset Shift Factor**

When bit 7 of the Access Control Register (XR09) is set to 1, these two bits are added to bits 15 and 14 of the address used in accessing the frame buffer through the A0000h-AFFFFh memory space in planar mode.

## XR10 Subsystem Vendor ID Low

Read-Only at I/O Address 3D7h with 3D6h set to Index 10h

7	6	5	4	3	2	1	0
Subsystem Vendor ID Bits 7-0 (2Ch)							

### 7-0 Subsystem Vendor ID Bits 7-0

These bits are used to program the lower 8 bits of the 16-bit SUBVENDID register (at offset 2C of the PCI configuration space) which carries the vendor ID of the vendor of the subsystem in which this graphics controller is used, such as an add-in graphics card. After reset, this register defaults to 2Ch, the lower 8 bits of the vendor ID assigned to CHIPS by the PCI special interest group. The lower 8 bits of the vendor ID of the actual subsystem vendor must be programmed into the graphics controller by writing it to this register.

## XR11 Subsystem Vendor ID High

Read-Only at I/O Address 3D7h with 3D6h set to Index 11h

7	6	5	4	3	2	1	0
Subsystem Vendor ID Bits 15-8 (10h)							

### 7-0 Subsystem Vendor ID Bits 15-8

These bits are used to program the upper 8 bits of the 16-bit SUBVENDID register (at offset 2C of the PCI configuration space) which carries the vendor ID of the vendor of the subsystem in which this graphics controller is used, such as an add-in graphics card. After reset, this register defaults to 10h, the upper 8 bits of the vendor ID assigned to CHIPS by the PCI special interest group. The upper 8 bits of the vendor ID of the actual subsystem vendor must be programmed into the graphics controller by writing it to this register.

## XR12 Subsystem Vendor Device ID Low

Read-Only at I/O Address 3D7h with 3D6h set to Index 12h

7	6	5	4	3	2	1	0
Subsystem Vendor Device ID Bits 7-0 (E5h)							

### 7-0 Subsystem Vendor Device ID Bits 7-0

These bits are intended to program the lower 8 bits of the 16-bit SUBDEVID register (at offset 2E of the PCI configuration space) which carries the device ID specified by the vendor of the subsystem in which this graphics controller is used, such as an add-in graphics card. After reset, this register defaults to E5h, the lower 8 bits of the device ID assigned to the 65555 by CHIPS. The lower 8 bits of the device ID desired by the actual subsystem vendor must be programmed into this graphics controller by writing it to this register.

## XR13 Subsystem Vendor Device ID High

Read-Only at I/O Address 3D7h with 3D6h set to Index 13h

7	6	5	4	3	2	1	0
Subsystem Vendor Device ID Bits 15-8 (00h)							

### 7-0 Subsystem Vendor Device ID Bits 15-8

These bits are intended to program the upper 8 bits of the 16-bit SUBDEVID register (at offset 2E of the PCI configuration space) which carries the device ID specified by the vendor of the subsystem in which this graphics controller is used, such as an add-in graphics card. After reset, this register defaults to 00h, the upper 8 bits of the device ID assigned to the 65555 by CHIPS. The upper 8 bits of the device ID desired by the actual subsystem vendor must be programmed into this graphics controller by writing it to this register.

## XR20 BitBLT Configuration Register

Read/Write at I/O Address 3D7h with 3D6h set to Index 20h

7	6	5	4	3	2	1	0
Reserved (00)		BitBLT Engine Color Depth (00)		Reserved (Writable) (00)		BitBLT Reset (0)	BitBLT Status (0)

### 7-6 Reserved

These bits always have the value of 0 when read.

### 5-4 BitBLT Engine Color Depth

When bit 23 of the BitBLT Control Register (BR04) is set to 0, these two bits configure the BitBLT engine for one of three possible color depths. If bit 23 of the BitBLT Control Register (BR04) is set to 1, then this function is performed by bits 25 and 24 of that same register. It is strongly recommended that, when possible, the color depth of the BitBLT engine be set to match the color depth to which the graphics system has been set.

Bits 5 4	BitBLT Engine Color Depth
0 0	8 bits per pixel (1 byte per pixel). This is the default after reset.
0 1	16 bits per pixel (2 bytes per pixel)
1 0	24 bits per pixel (3 bytes per pixel)
1 1	Reserved

The choice of color depth configures the BitBLT engine to work with one, two or three bytes per pixel. This directly affects the number of bytes of graphics data that the BitBLT engine will read and write for a given number of pixels. In the case of monochrome source or pattern data, this setting directly affects the color depth into which such monochrome data will be converted during the color expansion process.

If the graphics system has been set to a color depth that is not supported by the BitBLT engine, then it is strongly recommended that the BitBLT engine not be used. See appendix E for more information.

### 3-2 Reserved (Writable)

These bits always return the value of 0 when read.

### 1 BitBLT Reset

0 Writing a value of 0 to this bit permits normal operation of the BitBLT engine. This is the default value after reset.

1 Writing a value of 1 to this bit resets the BitBLT engine.

### 0 BitBLT Engine Status

0 Indicates that the BitBLT engine is idle. This is the default after reset.

1 Indicates that the BitBLT engine is busy.

## XR40 DRAM Access Control Register

Read/Write at I/O Address 3D7h with 3D6h set to Index 40h

7	6	5	4	3	2	1	0
Reserved (Writable) (0000:00)						Address Wrap (0)	Memory Access (0)

### 7-2 Reserved (Writable)

These bits should always be set to the value of 0.

### 1 Address Wrap

- 0 Only bits 0 through 17 of the memory address decode are used, causing the memory address to wrap at 256K for all memory accesses either through the VGA porthole or linearly.
- 1 All memory address bits are used, allowing access to all of the memory up to 4 megabytes.

### 0 Memory Access Width

- 0 Selects the use of 16-bit accesses to memory to accommodate the standard VGA modes and extended resolution modes with 4-bit color. This is the default after reset.
- 1 Selects the use of 64-bit accesses to memory to accommodate high resolution modes.

## XR41 DRAM Type Register

Read/Write at I/O Address 3D7h with 3D6h set to Index 41h

7	6	5	4	3	2	1	0
Reserved (Writable) (0000:00)						DRAM Type (xx)	

### 7-2 Reserved (Writable)

These bits should always be set to the value of 0.

### 1-0 DRAM Type

- 00 Reserved
- 01 Extended Data-Out (EDO) DRAM
- 10 Reserved
- 11 Reserved

## XR42 DRAM Configuration Register

Read/Write at I/O Address 3D7h with 3D6h set to Index 42h

7	6	5	4	3	2	1	0
Reserved (Writable) (0000:0)					Sec Bank Col Width (0)	Column Address Width (01)	

### 7-3 Reserved (Writable)

These bits should always be set to the value of 0.

### 2 Second Bank Column Width

- 0 The width of the column address used for the second bank follows that which is set for the first bank using bits 1 and 0 of this register. This is the default after reset.
- 1 The width of the column address used for the second bank is fixed to only 8 valid address bits. This is used in some 3MB configurations in which 128K-deep DRAMs requiring an 8-bit column address are used in the second bank.

**Note:** Bits 5 to 3 of the DRAM Interface Register must be set to select a 64-bit memory interface width and split RAS operation when this bit is set to 1.

### 1-0 Column Address Width

- 00 8-bit wide column address. AA0-AA7 are used during CAS cycle
- 01 9-bit wide column address. AA0-AA8 are used during CAS cycle  
This is the default after reset.
- 10 Reserved
- 11 Reserved

## XR43 DRAM Interface Register

Read/Write at I/O Address 3D7h with 3D6h set to Index 43h

7	6	5	4	3	2	1	0
Reserved (Writable) (00)		Memory Interface Width (xx)		Split RAS Enable (0)	Reserved (Writable) (000)		

### 7-6 Reserved (Writable)

These bits should always be set to the value of 0.

### 5-4 Memory Interface Width

- 00 Reserved
- 01 64-bit
- 10 Reserved
- 11 Reserved

**Note:** These should always be set to select a 64-bit memory interface.

### 3 Split RAS Enable

- 0 This causes both RAS0# and RAS1# function identically. This is the default after reset.
- 1 The selection and use of either RAS0# or RAS1# becomes dependent upon either the 18th or 19th bit of the frame buffer address, depending upon whether a 32-bit or a 64-bit memory interface data bus width is selected using bits 5 and 4 of this register.

**Note:** These should always be set to enable split RAS operation when bit 2 of the DRAM Configuration Register (XR42) is set to 1.

### 2-0 Reserved (Writable)

These bits should always be set to the value of 0.



## XR44 DRAM Timing Register

Read/Write at I/O Address 3D7h with 3D6h set to Index 44h

7	6	5	4	3	2	1	0
Shortened RAS Cycle (0)	Reserved (Writable)						
	(000:0000)						

### 7 Shortened RAS Cycle

0 Sets tRCD = 2.5 MCLKs. This is the default after reset.

1 Sets tRCD = 1.5 MCLKs.

### 6-0 Reserved

These bits should always be written with the value of 0.

## XR60 Video Pin Control Register

Read/Write at I/O Address 3D7h with 3D6h set to Index 60h

7	6	5	4	3	2	1	0
Reserved (0)	PCLK Pin Out Select (0)	Reserved (00:00)				Video Data Port Configuration (00)	

### 7 Reserved

This bit always returns the value of 0 when read.

### 6 PCLK Pin Output Select

0 Selects the current DCLK to be output on the PCLK pin. This is the default after reset.

1 Selects the current DCLK divided by 2 to be output on the PCLK pin.

### 5-2 Reserved

These bits always return the value of 0 when read.

### 1-0 Video Data Port Configuration

00 Disables the video data port feature.

01 Enables the video data port and configures it to be used to support a standard VGA feature connector.

10 Reserved

11 Enables the video data port and configures it to be used to support a ZV-type input.

## XR61 DPMS Sync Control Register

Read/Write at I/O Address 3D7h with 3D6h set to Index 61h

7	6	5	4	3	2	1	0
Reserved (0)	DPMS VSYNC (0)	DPMS HSYNC (0)	DPMS State Control (0)	DPMS VSYNC Sel (0)	DPMS VSYNC Data (0)	DPMS HSYNC Sel (0)	DPMS HSYNC Data (0)

### 7 Reserved

This bit always has the value of 0 when read.

### 6 DPMS VSYNC Output Select 2

- 0 The value carried by bit 2 of this register is output on the VSYNC pin. This is the default after reset.
- 1 The internal power sequencing clock is output on the VSYNC pin.

### 5 DPMS HSYNC Output Select 2

- 0 The value carried by bit 0 of this register is output on the HSYNC pin. This is the default after reset.
- 1 The internal power sequencing clock is output on the HSYNC pin.

### 4 DPMS HSYNC/VSYNC State Control

- 0 HSYNC and VSYNC pins are tri-stated during standby or panel-off modes. This is the default after reset.
- 1 HSYNC and VSYNC pins are driven during standby or panel-off modes with whatever data or signals that are selected by the other bits in this register.

### 3 DPMS VSYNC Output Select 1

- 0 The VSYNC signal is output on the VSYNC pin. This is the default after reset.
- 1 Bit 6 of this register is used to select what is output on the VSYNC pin.

### 2 DPMS VSYNC Output Data

The value to which this bit is set is set is output on the VSYNC pin if bits 6 and 3 of this register are set to 0 and 1, respectively.

### 1 DPMS HSYNC Output Select 1

- 0 The HSYNC signal is output on the HSYNC pin. This is the default after reset.
- 1 Bit 5 of this register is used to select what is output on the HSYNC pin.

### 0 DPMS HSYNC Output Data

The value to which this bit is set is set is output on the HSYNC pin if bits 5 and 1 of this register are set to 0 and 1, respectively.

## XR62 GPIO Pin Control Register

Read/Write at I/O Address 3D7h with 3D6h set to Index 62h

7	6	5	4	3	2	1	0
Reserved (0000)				GPIO3 Direction (0)	GPIO2 Direction (0)	Reserved (0)	

**Note:** See the FP Pin Control 2 Register (FR0C) for direction control of GPIO0 and GPIO1.

### 7-4 Reserved

These bits always return the value of 0 when read.

### 3 GPIO3 Direction Control

0 GPIO3 acts as an input. This is the default after reset.

1 GPIO3 acts as an output.

### 2 GPIO2 Direction Control

0 GPIO2 acts as an input. This is the default after reset.

1 GPIO2 acts as an output.

### 1-0 Reserved

These bits always return the value of 0 when read.

## XR63 GPIO Pin Data Register

Read/Write at I/O Address 3D7h with 3D6h set to Index 63h

7	6	5	4	3	2	1	0
Reserved (0000)				GPIO3 Data (x)	GPIO2 Data (x)	GPIO1 Data (x)	GPIO0 Data (x)

### 7-4 Reserved

These bits always return the value of 0 when read.

### 3 GPIO3 Data

This bit is used in either reading or setting the state of GPIO3.

### 2 GPIO2 Data

This bit is used in either reading or setting the state of GPIO2.

### 1 GPIO1 Data

This bit is used in either reading or setting the state of GPIO1.

### 0 GPIO0 Data

This bit is used in either reading or setting the state of GPIO0.

## XR67 Pin Tri-State Control Register

Read/Write at I/O Address 3D7h with 3D6h set to Index 67h

7	6	5	4	3	2	1	0
Reserved (0000:00)						Data Port Tri-State (0)	Mem Int. Tri-State (0)

### 7-2 Reserved

These bits always return the value of 0 when read.

### 1 Video Data Port Tri-State

0 Video data port pins are not tri-stated. This the default after reset.

1 Video data port pins are tri-stated.

### 0 Memory Interface Tri-State

0 Memory interface pins are not tri-stated. This is the default after reset.

1 Memory interface pins are tri-stated.

## XR70 Configuration Pin 0 Register

Read-Only at I/O Address 3D7h with 3D6h set to Index 70h

7	6	5	4	3	2	1	0
AA7 (x)	AA6 (x)	AA5 (x)	AA4 (x)	AA3 (x)	AA2 (x)	AA1 (x)	AA0 (1)

The bits of this register indicate the state of each of these pins at the time the graphics controller is reset.

During a reset, the graphics controller does not drive these pins, thereby allowing them to either be pulled high by relatively weak internal resistors, or to be pulled low by external resistors (4.7K recommended). Instead, during reset, the graphics controller latches the state of these pins, and the latched values are used by the graphics controller to provide a limited degree of hardware-based configuration of some features. Some of these latched values directly affect the hardware, while others are simply reflected in this register so as to be read by configuration software, usually the BIOS.

### 7 Memory Interface Address Pin AA7

- 0 Enables clock test mode.
- 1 Disables clock test mode.

**Note:** Clock test mode allows the internal clock synthesizers to be tested, by placing the output of the MCLK synthesizer on the ROMOE# pin (the pin used to drive the chip select pin of the BIOS ROM) and the output of the VCLK synthesizer on the PCLK pin (the clock pin used for the video data port).

### 6 Memory Interface Address Pin AA6

- 0 The ACTI and ENABKL outputs are forced to be tri-stated.
- 1 The ACTI and ENABKL outputs are permitted to function normally.

### 5 Memory Interface Address Pin AA5

Reserved.

No interpretation has yet been assigned to the state of this bit, and the hardware does not interpret the state of the corresponding pin during reset.

### 4 Memory Interface Address Pin AA4

- 0 The REFCLK and TCLK pins are used as inputs to receive MCLK and DCLK, respectively, from an external source.
- 1 MCLK and DCLK are provided by the internal clock generators.

**Note:** The default selection of sources for MCLK and DCLK may be individually changed by changing the settings of bits 2 and 1 of the Memory Clock Divisor Select Register (XR70). Both of those two bits also use the state of pin AA4 at reset to determine their default values.

**XR70 Configuration Pins 0 Register (continued)****3 Memory Interface Address Pin AA3**

Reserved.

No individual interpretation has yet been assigned to the state of this bit, and the hardware does not interpret the state of the corresponding pin during reset.

**2 Memory Interface Address Pin AA2**

Reserved.

No interpretation has yet been assigned to the state of this bit, and the hardware does not interpret the state of the corresponding pin during reset.

**1 Memory Interface Address Pin AA1**

0 Indicates that VGA I/O Address decoding is disabled on the PCI Bus, so access to the registers via I/O read and write operations is disabled.

1 Indicates that VGA I/O Address decoding is enabled on the PCI Bus, so access to the registers via I/O read and write operations is enabled.

**Note:** The reset state of this pin is also readable via bit 1 of the Host Bus Configuration Register (XR08).

**0 Reserved**

This bit always returns the value of 1 when read.



## XR71 Configuration Pin 1 Register

Read-Only at I/O Address 3D7h with 3D6h set to Index 71h

7	6	5	4	3	2	1	0
MAD7 (x)	MAD6 (x)	MAD5 (x)	MAD4 (x)	MAD3 (x)	MAD2 (x)	AA9 (x)	AA8 (x)

The bits of this register indicate the state of each of these pins at the time the graphics controller is reset.

During a reset, the graphics controller does not drive these pins, thereby allowing them to either be pulled high by relatively weak internal resistors, or to be pulled low by external resistors (4.7K recommended). Instead, during reset, the graphics controller latches the state of these pins, and the latched values are used by the graphics controller to provide a limited degree of hardware-based configuration of some features. Some of these latched values directly affect the hardware, while others are simply reflected in this register so as to be read by configuration software, usually the BIOS.

### 7 Memory Interface Data Pin MAD7

Reserved. No individual interpretation has yet been assigned to the state of this bit, and the hardware does not interpret the state of the corresponding pin during reset.

### 6 Memory Interface Data Pin MAD6

Reserved for BIOS for use as bit 3 of a 4-bit code specifying the panel type.

### 5 Memory Interface Data Pin MAD5

Reserved for BIOS for use as bit 2 of a 4-bit code specifying the panel type.

### 4 Memory Interface Data Pin MAD4

Reserved for BIOS for use as bit 1 of a 4-bit code specifying the panel type.

### 3 Memory Interface Data Pin MAD3

Reserved for BIOS for use as bit 0 of a 4-bit code specifying the panel type.

### 2 Memory Interface Data Pin MAD2

Reserved. No individual interpretation has yet been assigned to the state of this bit, and the hardware does not interpret the state of the corresponding pin during reset.

### 1 Memory Interface Address Pin AA9

Reserved. No individual interpretation has yet been assigned to the state of this bit, and the hardware does not interpret the state of the corresponding pin during reset.

### 0 Memory Interface Address Pin AA8

Reserved. No individual interpretation has yet been assigned to the state of this bit, and the hardware does not interpret the state of the corresponding pin during reset.

## XR80 Pixel Pipeline Configuration 0 Register

Read/Write at I/O Address 3D7h with 3D6h set to Index 80h

7	6	5	4	3	2	1	0
6-Bit/8-Bit DAC Select (0)	Reserved (0)	Pixel Averaging (0)	Reserved (0)	Extended Status Read (0)	Flat Panel Overscan (0)	CRT Overscan (0)	Palette Addr Select (0)

### 7 6-Bit/8-Bit DAC Select

- 0 All three D-to-A converters are set for 6-bit operation. This is the default after reset.
- 1 All three D-to-A converters are set for 8-bit operation.

### 6 Reserved

This bit always returns the value of 0 when read.

### 5 Pixel Averaging Enable

Pixel averaging causes the red, green and blue color component values of a replicated pixel created by the horizontal stretching process to be averaged with those of the next pixel.

- 0 Disables pixel averaging. This is the default after reset.
- 1 Enables pixel averaging.

**Note:** The pixel averaging feature applies only to flat panel displays, not CRT's, and it applies only when horizontal stretching is active (see the description of the Horizontal Stretching Register, FR41, for more details).

### 4 Reserved

This bit always returns the value of 0 when read.

**XR80 Pixel Pipeline Configuration 0 Register (continued)**

**3 Extended Status Read Enable**

- 0 Disable extended status read feature. This is the default after reset.
- 1 Enable extended status read feature.

**Note:** When enabled, the extended status read feature changes the functionality of three of the palette DAC registers in order to allow the status of the internal state machines and values of the read and green data in the input holding register to be read. The affected palette DAC registers and their alternate functions are as follows:

Affected Register	Alternate Function
Pixel Data Mask Register (DACMASK)	Returns the value of the red pixel data currently in the data holding register.
Palette Write Mode Index Register (DACWX)	Returns the value of the green pixel data currently in the data holding register.
Palette DAC State Register (DACSTATE)	Returns the status of the internal state machines in bits 7-2.

This feature must be disabled to permit normal accesses to the registers and color data locations within the palette DAC.

**2 Flat Panel Overscan Color Enable**

- 0 Disable the use of the flat panel overscan color (Overscan[1]). This is the default after reset.
- 1 Enable the use of the flat panel overscan color (Overscan[1]).

**1 CRT Overscan Enable**

- 0 Disable the use of the CRT overscan color (Overscan[0]). This is the default after reset.
- 1 Enable the use of the CRT overscan color (Overscan[0]).

**0 Palette Addressing Select**

- 0 Select the standard 256-position palette for the main display image to be accessed via the palette DAC's sub-addressing scheme. This is the default after reset.
- 1 Select the separate 8-position palette for cursor 1 and cursor 2 to be accessed via the palette DAC's sub-addressing scheme.

## XR81 Pixel Pipeline Configuration 1 Register

Read/Write at I/O Address 3D7h with 3D6h set to Index 81h

7	6	5	4	3	2	1	0
Reserved (000)			VGA Std Delay (0)	Graphics System Color Depth (0000)			

### 7-5 Reserved

These bits always return the value of 0 when read.

### 4 VGA Standard Signal Delay Enable

- 0 The CRT Display Enable and CRT Blank are delayed slightly for compatibility with the VGA standard. This is the default after reset.
- 1 The CRT Display Enable and CRT Blank are not delayed.

**Note:** This is a signal delay relative to the CRT horizontal and vertical sync signals. This bit applies only when the standard VGA pixel pipeline is used by the CRT display engine (i.e., when bit 0 of FR01 is set to 1). When the flat panel display engine is enabled (i.e., when bit 1 of FR01 is set to 1), then this bit is ignored and no such delay takes place. This bit should be set to 0 for all standard VGA modes, and should be set to 1 for all extended modes.

### 3-0 Graphics System Color Depth

Bits 3 2 1 0	Color Depth Selected for Graphics System
0 0 0 0	Configures the CRT pipeline for standard VGA text and graphics modes, and for 1bpp, 2bpp and 4bpp extended graphics modes. This is the default after reset.
0 0 0 1	Reserved
0 0 1 0	Configures the CRT pipeline for 8bpp extended graphics modes.
0 0 1 1	Reserved
0 1 0 0	Configures the CRT pipeline for 16bpp extended graphics modes wherein the graphics data follows a fixed Targa-compatible 5-5-5 RGB format.
0 1 0 1	Configures the CRT pipeline for 16bpp extended graphics modes wherein the graphics data follows a fixed XGA-compatible 5-6-5 RGB format.
0 1 1 0	Configures the CRT pipeline for packed 24bpp extended graphics modes wherein only 3 bytes are allocated for each pixel.
0 1 1 1	Configures the CRT pipeline for non-packed 24bpp (32bpp) extended graphics modes wherein 4 bytes are allocated for each pixel, so that the graphics data for each pixel is doubleword-aligned. The 4th byte allocated for each pixel is unused.
1 x x x	Reserved

## XR82 Pixel Pipeline Configuration 2 Register

Read/Write at I/O Address 3D7h with 3D6h set to Index 82h

7	6	5	4	3	2	1	0
Reserved (0000)				Graphics Gamma (0)	Video Gamma (0)	Comp. Sync on Green (0)	Blank Pedestal (0)

### 7-4 Reserved

These bits always return the value of 0 when read.

### 3 Graphics Data Gamma Correction Enable

- 0 Graphics data bypasses the palette when the graphics system is set to a color depth of 16, 24 or 32 bits per pixel. This is the default after reset.
- 1 Graphics data goes through the palette when the graphics system is set to a color depth of 16, 24 or 32 bits per pixel, allowing the palette to be used to perform gamma correction.

### 2 Video Data Gamma Correction Enable

- 0 Video data bypasses the palette. This is the default after reset.
- 1 Video data goes through the palette, allowing the palette to be used to perform gamma correction.

### 1 Composite Sync on Green Enable

- 0 Disables the provision of composite sync on the green analog output. This is the default after reset.
- 1 Enables the provision of composite sync on the green analog output.

### 0 Blank Pedestal Enable

- 0 Disables the provision of a pedestal output level during blanking periods. This is the default after reset.
- 1 Enables the provision of a pedestal output level during blanking periods.

## XRA0 Cursor 1 Control Register

Read/Write at I/O Address 3D7h with 3D6h set to Index A0h

7	6	5	4	3	2	1	0
Cursor 1 Blink En (0)	Cursor 1 V Stretch (0)	Cursor 1 H Stretch (0)	Coordinate Origin Sel (0)	Vertical Extension (0)	Cursor 1 Mode Select (000)		

### 7 Cursor 1 Blink Enable

- 0 Disables blinking. This is the default after reset.
- 1 Enables blinking. Blinking rate set in register FR19.

### 6 Cursor 1 Vertical Stretching Enable

- 0 Disables vertical stretching for cursor 1. This is the default after reset.
- 1 Enables vertical stretching for cursor 1.

**Note:** Just as is the case with the vertical stretching for the main display image, vertical stretching for cursor 1 applies only to flat panel displays.

### 5 Cursor 1 Horizontal Stretching Enable

- 0 Disables horizontal stretching for cursor 1. This is the default after reset.
- 1 Enables horizontal stretching for cursor 1.

**Note:** Just as is the case with the horizontal stretching for the main display image, horizontal stretching for cursor 1 applies only to flat panel displays.

### 4 Cursor 1 Coordinate System Origin Select

- 0 Selects the outermost upper left-hand corner of the screen border as the origin for the coordinate system used to position cursor 1. This is the default after reset.
- 1 Selects the upper left-hand corner of the active display area as the origin for the coordinate system used to position cursor 1.

### 3 Cursor 1 Vertical Extension Enable

- 0 Disables the vertical extension feature for cursor 1. This is the default after reset.
- 1 Enables the vertical extension feature for cursor 1, thereby permitting the height of cursor 1 may be specified independently of its mode selection through the use of the Cursor 1 Vertical Extension Register (XRA1).

### ***XRA0 Cursor 1 Control Register (continued)***

#### **2-0 Cursor 1 Mode Select**

These three bits select the mode for cursor 1. See appendix F for more details concerning the cursor modes.

<b>Bits 2 1 0</b>	<b>Cursor Mode</b>
0 0 0	Cursor 1 is disabled. This is the default after reset.
0 0 1	32x32 2bpp AND/XOR 2-plane mode
0 1 0	128x128 1bpp 2-color mode
0 1 1	128x128 1bpp 1-color and transparency mode
1 0 0	64x64 2bpp 3-color and transparency mode
1 0 1	64x64 2bpp AND/XOR 2-plane mode
1 1 0	64x64 2bpp 4-color mode
1 1 1	Reserved

## XRA1 Cursor 1 Vertical Extension Register

Read/Write at I/O Address 3D7h with 3D6h set to Index A1h

7	6	5	4	3	2	1	0
Cursor 1 Vertical Extension (00h)							

### 7-0 Cursor 1 Vertical Extension

When the vertical extension feature for cursor 1 is enabled by setting bit 3 of the Cursor 1 Control Register (XRA0) to 1, these 8 bits of this register are used to specify the height of cursor 1 in scan lines. The number of scan lines must be a multiple of four.

This register should be programmed with a value derived from the following equation:

$$\text{value} = ((\text{number of scan lines}) \div 4) - 1$$



## XRA2 Cursor 1 Base Address Low Register

Read/Write at I/O Address 3D7h with 3D6h set to Index A2h

7	6	5	4	3	2	1	0
Cursor 1 Base Address Bits 15-12 (0000)				Cursor 1 Pattern Select (0000)			

### 7-4 Cursor 1 Base Address Bits 15-12

These four bits provide part of a 22-bit value that specifies the offset from the beginning of the frame buffer memory space where the 4KB cursor data space for cursor 1 is to be located. The six most-significant bits of this 22-bit value are supplied by the Cursor 1 Base Address High Register (XRA3).

### 3-0 Cursor 1 Pattern Select

These four bits allow 1 of up to as many as 16 possible patterns contained in the cursor data space for cursor 1 to be selected to be displayed.

The actual number of patterns depends on the size of each pattern, since the cursor data space is limited to a total of 4KB in size. The size of each pattern depends, at least in part, on the choice of cursor mode. See appendix F for more details concerning the cursor modes.

## XRA3 Cursor 1 Base Address High Register

Read/Write at I/O Address 3D7h with 3D6h set to Index A3h

7	6	5	4	3	2	1	0
Reserved (00)		Cursor 1 Base Address Bits 21-16 (00:0000)					

### 7-6 Reserved

These bits always return the value of 0 when read.

### 3-0 Cursor 1 Base Address Bits 21-16

These six bits provide the six most significant bits of a 22-bit value that specifies the offset from the beginning of the frame buffer memory space where the 4KB cursor data space for cursor 1 is to be located. The four next most-significant bits of this 22-bit value are supplied by the Cursor 1 Base Address Low Register (XRA2).

## XRA4 Cursor 1 X-Position Low Register

Read/Write at I/O Address 3D7h with 3D6h set to Index A4h

7	6	5	4	3	2	1	0
Cursor 1 X-Position Magnitude Bits 7-0 (00h)							

### 7-0 Cursor 1 X-Position Magnitude Bits 7-0

This register provides the eight least significant magnitude bits of a signed 12-bit value that specifies the horizontal position of cursor 1. The three most significant magnitude bits and the sign bit of this value are provided by bits 2-0 and bit 7, respectively, of the Cursor 1 X-Position High Register (XRA5).

## XRA5 Cursor 1 X-Position High Register

Read/Write at I/O Address 3D7h with 3D6h set to Index A5h

7	6	5	4	3	2	1	0
X-Pos Sign Bit (0)	Reserved (000:0)				Cursor 1 X-Position Magnitude Bits 10-8 (000)		

### 7 Cursor 1 X-Position Sign Bit

This bit provides the sign bit of a signed 12-bit value that specifies the horizontal position of cursor 1. The magnitude bits are provided by the Cursor 1 X-Position Low Register (XRA4) and bits 2-0 of this register.

### 6-3 Reserved

These bits always return the value of 0 when read.

### 2-0 Cursor 1 X-Position Magnitude Bits 10-8

These three bits provide the three most significant magnitude bits of a signed 12-bit value that specifies the horizontal position of cursor 1. The eight least significant magnitude bits of this value are provided by bits 7-0 of the Cursor 1 X-Position Low Register (XRA4). The sign bit is provided by bit 7 of this register.

## XRA6 Cursor 1 Y-Position Low Register

Read/Write at I/O Address 3D7h with 3D6h set to Index A6h

7	6	5	4	3	2	1	0
Cursor 1 Y-Position Magnitude Bits 7-0 (00h)							

### 7-0 Cursor 1 Y-Position Magnitude Bits 7-0

This register provides the eight least significant magnitude bits of a signed 12-bit value that specifies the vertical position of cursor 1. The three most significant magnitude bits and the sign bit of this value are provided by bits 2-0 and bit 7, respectively, of the Cursor 1 Y-Position High Register (XRA7).

## XRA7 Cursor 1 Y-Position High Register

Read/Write at I/O Address 3D7h with 3D6h set to Index A7h

7	6	5	4	3	2	1	0
Y-Pos Sign Bit (0)	Reserved (000:0)				Cursor 1 Y-Position Magnitude Bits 10-8 (000)		

### 7 Cursor 1 Y-Position Sign Bit

This bit provides the sign bit of a signed 12-bit value that specifies the horizontal position of cursor 1. The magnitude bits are provided by the Cursor 1 Y-Position Low Register (XRA6) and bits 2-0 of this register.

### 6-3 Reserved

These bits always return the value 0 when read.

### 2-0 Cursor 1 Y-Position Magnitude Bits 10-8

These three bits provide the three most significant magnitude bits of a signed 12-bit value that specifies the horizontal position of cursor 1. The eight least significant magnitude bits of this value are provided by bits 7-0 of the Cursor 1 Y-Position Low Register (XRA6). The sign bit is provided by bit 7 of this register.

## XRA8 Cursor 2 Control Register

Read/Write at I/O Address 3D7h with 3D6h set to Index A8h

7	6	5	4	3	2	1	0
Cursor 2 Blink En (0)	Cursor 2 V Stretch (0)	Cursor 2 H Stretch (0)	Coordinate Origin Sel (0)	Vertical Extension (0)	Cursor 2 Mode Select (000)		

### 7 Cursor 2 Blink Enable

- 0 Disables blinking. This is the default after reset.
- 1 Enables blinking. Blinking rate set in register FR19.

### 6 Cursor 2 Vertical Stretching Enable

- 0 Disables vertical stretching for cursor 2. This is the default after reset.
- 1 Enables vertical stretching for cursor 2.

**Note:** Just as is the case with the vertical stretching for the main display image, vertical stretching for cursor 2 applies only to flat panel displays.

### 5 Cursor 2 Horizontal Stretching Enable

- 0 Disables horizontal stretching for cursor 2. This is the default after reset.
- 1 Enables horizontal stretching for cursor 2.

**Note:** Just as is the case with the horizontal stretching for the main display image, horizontal stretching for cursor 2 applies only to flat panel displays.

### 4 Cursor 2 Coordinate System Origin Select

- 0 Selects the outermost upper left-hand corner of the screen border as the origin for the coordinate system used to position cursor 2. This is the default after reset.
- 1 Selects the upper left-hand corner of the active display area as the origin for the coordinate system used to position cursor 2.

### 3 Cursor 2 Vertical Extension Enable

- 0 Disables the vertical extension feature for cursor 2. This is the default after reset.
- 1 Enables the vertical extension feature for cursor 2, thereby permitting the height of cursor 2 may be specified independently of its mode selection through the use of the Cursor 2 Vertical Extension Register (XRA9).

**XRA8 Cursor 2 Control Register (continued)**

**2-0 Cursor 2 Mode Select**

These three bits select the mode for cursor 2. See appendix F for more details concerning the cursor modes.

Bits 2 1 0	Cursor Mode
0 0 0	Cursor 2 is disabled. This is the default after reset.
0 0 1	32x32 2bpp AND/XOR 2-plane mode
0 1 0	128x128 1bpp 2-color mode
0 1 1	128x128 1bpp 1-color and transparency mode
1 0 0	64x64 2bpp 3-color and transparency mode
1 0 1	64x64 2bpp AND/XOR 2-plane mode
1 1 0	64x64 2bpp 4-color mode
1 1 1	Reserved

## XRA9 Cursor 2 Vertical Extension Register

Read/Write at I/O Address 3D7h with 3D6h set to Index A9h

7	6	5	4	3	2	1	0
Cursor 2 Vertical Extension (00h)							

### 7-0 Cursor 2 Vertical Extension

When the vertical extension feature for cursor 2 is enabled by setting bit 3 of the Cursor 2 Control Register (XRA8) to 1, these 8 bits of this register are used to specify the height of cursor 2 in scan lines. The number of scan lines must be a multiple of four.

This register should be programmed with a value derived from the following equation:

$$\text{value} = ((\text{number of scan lines}) \div 4) - 1$$

## XRAA Cursor 2 Base Address Low Register

Read/Write at I/O Address 3D7h with 3D6h set to Index AAh

7	6	5	4	3	2	1	0
Cursor 2 Base Address Bits 15-12 (0000)				Cursor 2 Pattern Select (0000)			

### 7-4 Cursor 2 Base Address Bits 15-12

These four bits provide part of a 22-bit value that specifies the offset from the beginning of the frame buffer memory space where the 4KB cursor data space for cursor 2 is to be located. The six most-significant bits of this 22-bit value are supplied by the Cursor 2 Base Address High Register (XRAB).

### 3-0 Cursor 2 Pattern Select

These four bits allow 1 of up to as many as 16 possible patterns contained in the cursor data space for cursor 2 to be selected to be displayed.

The actual number of patterns depends on the size of each pattern, since the cursor data space is limited to a total of 4KB in size. The size of each pattern depends, at least in part, on the choice of cursor mode. See appendix F for more details concerning the cursor modes.

## ***XRAB Cursor 2 Base Address High Register***

Read/Write at I/O Address 3D7h with 3D6h set to Index ABh

7	6	5	4	3	2	1	0
Reserved (00)		Cursor 2 Base Address Bits 21-16 (00:0000)					

### **7-6 Reserved**

These bits always return the value of 0 when read.

### **3-0 Cursor 2 Base Address Bits 21-16**

These six bits provide the six most significant bits of a 22-bit value that specifies the offset from the beginning of the frame buffer memory space where the 4KB cursor data space for cursor 2 is to be located. The four next most-significant bits of this 22-bit value are supplied by the Cursor 2 Base Address Low Register (XRAA).

## ***XRAC Cursor 2 X-Position Low Register***

Read/Write at I/O Address 3D7h with 3D6h set to Index ACh

7	6	5	4	3	2	1	0
Cursor 2 X-Position Magnitude Bits 7-0 (00h)							

### **7-0 Cursor 2 X-Position Magnitude Bits 7-0**

This register provides the eight least significant magnitude bits of a signed 12-bit value that specifies the horizontal position of cursor 2. The three most significant magnitude bits and the sign bit of this value are provided by bits 2-0 and bit 7, respectively, of the Cursor 2 X-Position High Register (XRAD).



## XRAD Cursor 2 X-Position High Register

Read/Write at I/O Address 3D7h with 3D6h set to Index ADh

7	6	5	4	3	2	1	0
X-Pos Sign Bit (0)	Reserved (000:0)				Cursor 2 X-Position Magnitude Bits 10-8 (000)		

### 7 Cursor 2 X-Position Sign Bit

This bit provides the sign bit of a signed 12-bit value that specifies the horizontal position of cursor 2. The magnitude bits are provided by the Cursor 2 X-Position Low Register (XRAC) and bits 2-0 of this register.

### 6-3 Reserved

These bits always return the value of 0 when read.

### 2-0 Cursor 2 X-Position Magnitude Bits 10-8

These three bits provide the three most significant magnitude bits of a signed 12-bit value that specifies the horizontal position of cursor 2. The eight least significant magnitude bits of this value are provided by bits 7-0 of the Cursor 2 X-Position Low Register (XRAC). The sign bit is provided by bit 7 of this register.

## ***XRAE Cursor 2 Y-Position Low Register***

Read/Write at I/O Address 3D7h with 3D6h set to Index AEh

7	6	5	4	3	2	1	0
Cursor 2 Y-Position Magnitude Bits 7-0 (00h)							

### **7-0 Cursor 2 Y-Position Magnitude Bits 7-0**

This register provides the eight least significant magnitude bits of a signed 12-bit value that specifies the vertical position of cursor 2. The three most significant magnitude bits and the sign bit of this value are provided by bits 2-0 and bit 7, respectively, of the Cursor 2 Y-Position High Register (XRAF).

## ***XRAF Cursor 2 Y-Position High Register***

Read/Write at I/O Address 3D7h with 3D6h set to Index AFh

7	6	5	4	3	2	1	0
Y-Pos Sign Bit (0)	Reserved (000:0)				Cursor 2 Y-Position Magnitude Bits 10-8 (000)		

### **7 Cursor 2 Y-Position Sign Bit**

This bit provides the sign bit of a signed 12-bit value that specifies the horizontal position of cursor 2. The magnitude bits are provided by the Cursor 2 Y-Position Low Register (XRAE) and bits 2-0 of this register.

### **6-3 Reserved**

These bits always return the value of 0 when read.

### **2-0 Cursor 2 Y-Position Magnitude Bits 10-8**

These three bits provide the three most significant magnitude bits of a signed 12-bit value that specifies the horizontal position of cursor 2. The eight least significant magnitude bits of this value are provided by bits 7-0 of the Cursor 2 Y-Position Low Register (XRAE). The sign bit is provided by bit 7 of this register.

## XRC0 Dot Clock 0 VCO M-Divisor Low Register

Read/Write at I/O Address 3D7h with 3D6h set to Index C0h

7	6	5	4	3	2	1	0
Dot Clock 0 VCO M-Divisor Bits 7-0							

**Note:** All four of the registers used in specifying the loop parameters for the synthesizer used to generate dot clock 0 (XRC0-XRC3) must be written, and in sequence, before the hardware will update the synthesizer settings. This is meant to prevent wild fluctuations in the synthesizer output as new values are being written to these registers.

### 7-0 Dot Clock 0 VCO M-Divisor Bits 7-0

This register provides the eight least significant bits of a 10-bit value that specifies the M-divisor, one of the loop parameters used in controlling the frequency of the output of the synthesizer used to generate dot clock 0. Bits 1 and 0 of the Dot Clock 0 VCO M/N Divisor High Register (XRC2) provide the two most significant bits.

A series of calculations are used to derive this value and the values for the other loop parameters given a desired output frequency and a series of constraints placed on different components within the synthesizer used to generate dot clock 0. See appendix A for a detailed description of the process used to derive the loop parameter values.

## **XRC1 Dot Clock 0 VCO N-Divisor Low Register**

Read/Write at I/O Address 3D7h with 3D6h set to Index C1h

7	6	5	4	3	2	1	0
Dot Clock 0 VCO N-Divisor Bits 7-0							

**Note:** All four of the registers used in specifying the loop parameters for the synthesizer used to generate dot clock 0 (XRC0-XRC3) must be written, and in sequence, before the hardware will update the synthesizer settings. This is meant to prevent wild fluctuations in the synthesizer output as new values are being written to these registers.

### **7-0 Dot Clock 0 VCO N-Divisor Bits 7-0**

This register provides the eight least significant bits of a 10-bit value that specifies the N-divisor, one of the loop parameters used in controlling the frequency of the output of the synthesizer used to generate dot clock 0. Bits 5 and 4 of the Dot Clock 0 VCO M/N Divisor High Register (XRC2) provide the two most significant bits.

A series of calculations are used to derive this value and the values for the other loop parameters given a desired output frequency and a series of constraints placed on different components within the synthesizer used to generate dot clock 0. See appendix A for a detailed description of the process used to derive the loop parameter values.

## XRC2 Dot Clock 0 VCO M/N-Divisor High Register

Read/Write at I/O Address 3D7h with 3D6h set to Index C2h

7	6	5	4	3	2	1	0
Reserved		Dot Clock 0 VCO N-Divisor Bits 9-8		Reserved		Dot Clock 0 VCO M-Divisor Bits 9-8	

**Note:** All four of the registers used in specifying the loop parameters for the synthesizer used to generate dot clock 0 (XRC0-XRC3) must be written, and in sequence, before the hardware will update the synthesizer settings. This is meant to prevent wild fluctuations in the synthesizer output as new values are being written to these registers.

### 7-6 Reserved

These bits always return the value of 0 when read.

### 5-4 Dot Clock 0 VCO N-Divisor Bits 9-8

These two bits provide the two most significant bits of a 10-bit value that specifies the N-divisor, one of the loop parameters used in controlling the frequency of the output of the synthesizer used to generate dot clock 0. The Dot Clock 0 VCO N Divisor Low Register (XRC1) provides the eight least significant bits.

A series of calculations are used to derive this value and the values for the other loop parameters given a desired output frequency and a series of constraints placed on different components within the synthesizer used to generate dot clock 0. See appendix A for a detailed description of the process used to derive the loop parameter values.

### 3-2 Reserved

These bits always return the value of 0 when read.

### 1-0 Dot Clock 0 VCO M-Divisor Bits 9-8

These two bits provide the two most significant bits of a 10-bit value that specifies the M-divisor, one of the loop parameters used in controlling the frequency of the output of the synthesizer used to generate dot clock 0. The Dot Clock 0 VCO M Divisor Low Register (XRC0) provides the eight least significant bits.

A series of calculations are used to derive this value and the values for the other loop parameters given a desired output frequency and a series of constraints placed on different components within the synthesizer used to generate dot clock 0. See appendix A for a detailed description of the process used to derive the loop parameter values.

## XRC3 Dot Clock 0 Divisor Select Register

Read/Write at I/O Address 3D7h with 3D6h set to Index C3h

7	6	5	4	3	2	1	0
Reserved	Post Divisor Select			Reserved	VCO Loop Divisor	Reference Clock Input Divisor	

**Note:** All four of the registers used in specifying the loop parameters for the synthesizer used to generate dot clock 0 (XRC0-XRC3) must be written, and in sequence, before the hardware will update the synthesizer settings. This is meant to prevent wild fluctuations in the synthesizer output as new values are being written to these registers.

### 7 Reserved

This bit always returns the value of 0 when read.

### 6-4 Post Divisor Select

These three bits select a value that specifies the post divisor, one of the loop parameters used in controlling the frequency of the output of the synthesizer used to generate dot clock 0. The manner in which these bits are used to choose this value is shown in the table below:

Bits 6 5 4	Post Divisor
0 0 0	1
0 0 1	2
0 1 0	4
0 1 1	8
1 0 0	16
1 0 1	32
1 1 0	Reserved
1 1 1	Reserved

A series of calculations are used to derive this value and the values for the other loop parameters given a desired output frequency and a series of constraints placed on different components within the synthesizer used to generate dot clock 0. See the appendix A for a detailed description of the process used to derive the loop parameter values.

### 3 Reserved

This bit always returns the value of 0 when read.

### ***XRC3 Dot Clock 0 Divisor Select Register (continued)***

#### **2 VCO Loop Divisor Select**

This bit selects a value that specifies the VCO loop divide, one of the loop parameters used in controlling the frequency of the output of the synthesizer used to generate dot clock 0.

- 0 Selects a VCO loop divide value of 4.
- 1 Selects a VCO loop divide value of 16.

A series of calculations are used to derive this value and the values for the other loop parameters given a desired output frequency and a series of constraints placed on different components within the synthesizer used to generate dot clock 0. See appendix A for a detailed description of the process used to derive the loop parameter values.

#### **1-0 Reference Input Clock Divisor Select**

These two bits select a value that specifies the reference input clock divisor, one of the loop parameters used in controlling the frequency of the output of the synthesizer used to generate dot clock 0. The manner in which these bits are used to choose this value is shown in the table below:

Bits 1 0	Reference Input Clock Divisor	Comment
0 0	4	
0 1	1	The reference input clock is not divided.
1 0	20	
1 1	5	A value intended to make it somewhat easier to generate a clock for use in creating output with NTSC-compatible timings.

A series of calculations are used to derive this value and the values for the other loop parameters given a desired output frequency and a series of constraints placed on different components within the synthesizer used to generate dot clock 0. See appendix A for a detailed description of the process used to derive the loop parameter values.

## **XRC4 Dot Clock 1 VCO M-Divisor Low Register**

Read/Write at I/O Address 3D7h with 3D6h set to Index C4h

7	6	5	4	3	2	1	0
Dot Clock 1 VCO M-Divisor Bits 7-0							

**Note:** All four of the registers used in specifying the loop parameters for the synthesizer used to generate dot clock 1 (XRC4-XRC7) must be written, and in sequence, before the hardware will update the synthesizer settings. This is meant to prevent wild fluctuations in the synthesizer output as new values are being written to these registers.

### **7-0 Dot Clock 1 VCO M-Divisor Bits 7-0**

This register provides the eight least significant bits of a 10-bit value that specifies the M-divisor, one of the loop parameters used in controlling the frequency of the output of the synthesizer used to generate dot clock 1. Bits 1 and 0 of the Dot Clock 1 VCO M/N Divisor High Register (XRC6) provide the two most significant bits.

A series of calculations are used to derive this value and the values for the other loop parameters given a desired output frequency and a series of constraints placed on different components within the synthesizer used to generate dot clock 1. See appendix A for a detailed description of the process used to derive the loop parameter values.



## XRC5 Dot Clock 1 VCO N-Divisor Low Register

Read/Write at I/O Address 3D7h with 3D6h set to Index C5h

7	6	5	4	3	2	1	0
Dot Clock 1 VCO N-Divisor Bits 7-0							

**Note:** All four of the registers used in specifying the loop parameters for the synthesizer used to generate dot clock 1 (XRC4-XRC7) must be written, and in sequence, before the hardware will update the synthesizer settings. This is meant to prevent wild fluctuations in the synthesizer output as new values are being written to these registers.

### 7-0 Dot Clock 1 VCO N-Divisor Bits 7-0

This register provides the eight least significant bits of a 10-bit value that specifies the N-divisor, one of the loop parameters used in controlling the frequency of the output of the synthesizer used to generate dot clock 1. Bits 5 and 4 of the Dot Clock 1 VCO M/N Divisor High Register (XRC6) provide the two most significant bits.

A series of calculations are used to derive this value and the values for the other loop parameters given a desired output frequency and a series of constraints placed on different components within the synthesizer used to generate dot clock 1. See appendix A for a detailed description of the process used to derive the loop parameter values.

## XRC6 Dot Clock 1 VCO M/N-Divisor High Register

Read/Write at I/O Address 3D7h with 3D6h set to Index C6h

7	6	5	4	3	2	1	0
Reserved		Dot Clock 1 VCO N-Divisor Bits 9-8		Reserved		Dot Clock 1 VCO M-Divisor Bits 9-8	

**Note:** All four of the registers used in specifying the loop parameters for the synthesizer used to generate dot clock 1 (XRC4-XRC7) must be written, and in sequence, before the hardware will update the synthesizer settings. This is meant to prevent wild fluctuations in the synthesizer output as new values are being written to these registers.

### 7-6 Reserved

These bits always return the value of 0 when read.

### 5-4 Dot Clock 1 VCO N-Divisor Bits 9-8

These two bits provide the two most significant bits of a 10-bit value that specifies the N-divisor, one of the loop parameters used in controlling the frequency of the output of the synthesizer used to generate dot clock 1. The Dot Clock 1 VCO N Divisor Low Register (XRC5) provides the eight least significant bits.

A series of calculations are used to derive this value and the values for the other loop parameters given a desired output frequency and a series of constraints placed on different components within the synthesizer used to generate dot clock 1. See appendix A for a detailed description of the process used to derive the loop parameter values.

### 3-2 Reserved

These bits always return the value of 0 when read.

### 1-0 Dot Clock 1 VCO M-Divisor Bits 9-8

These two bits provide the two most significant bits of a 10-bit value that specifies the M-divisor, one of the loop parameters used in controlling the frequency of the output of the synthesizer used to generate dot clock 1. The Dot Clock 1 VCO M Divisor Low Register (XRC0) provides the eight least significant bits.

A series of calculations are used to derive this value and the values for the other loop parameters given a desired output frequency and a series of constraints placed on different components within the synthesizer used to generate dot clock 1. See appendix A for a detailed description of the process used to derive the loop parameter values.

## XRC7 Dot Clock 1 Divisor Select Register

Read/Write at I/O Address 3D7h with 3D6h set to Index C7h

7	6	5	4	3	2	1	0
Reserved	Post Divisor Select			Reserved	VCO Loop Divisor	Reference Clock Input Divisor	

**Note:** All four of the registers used in specifying the loop parameters for the synthesizer used to generate dot clock 1 (XRC4-XRC7) must be written, and in sequence, before the hardware will update the synthesizer settings. This is meant to prevent wild fluctuations in the synthesizer output as new values are being written to these registers.

### 7 Reserved

This bit always returns the value of 0 when read.

### 6-4 Post Divisor Select

These three bits select a value that specifies the post divisor, one of the loop parameters used in controlling the frequency of the output of the synthesizer used to generate dot clock 1. The manner in which these bits are used to choose this value is shown in the table below:

Bits 6 5 4	Post Divisor
0 0 0	1
0 0 1	2
0 1 0	4
0 1 1	8
1 0 0	16
1 0 1	32
1 1 0	Reserved
1 1 1	Reserved

A series of calculations are used to derive this value and the values for the other loop parameters given a desired output frequency and a series of constraints placed on different components within the synthesizer used to generate dot clock 0. See appendix A for a detailed description of the process used to derive the loop parameter values.

### 3 Reserved

This bit always returns the value of 0 when read.

## ***XRC7 Dot Clock 1 Divisor Select Register (continued)***

### **2 VCO Loop Divisor Select**

This bit selects a value that specifies the VCO loop divide, one of the loop parameters used in controlling the frequency of the output of the synthesizer used to generate dot clock 1.

- 0 Selects a VCO loop divide value of 4.
- 1 Selects a VCO loop divide value of 16.

A series of calculations are used to derive this value and the values for the other loop parameters given a desired output frequency and a series of constraints placed on different components within the synthesizer used to generate dot clock 1. See appendix A for a detailed description of the process used to derive the loop parameter values.

### **1-0 Reference Input Clock Divisor Select**

These two bits select a value that specifies the reference input clock divisor, one of the loop parameters used in controlling the frequency of the output of the synthesizer used to generate dot clock 1. The manner in which these bits are used to choose this value is shown in the table below:

<b>Bits 1 0</b>	<b>Reference Input Clock Divisor</b>	<b>Comment</b>
0 0	4	
0 1	1	The reference input clock is not divided.
1 0	20	
1 1	5	A value intended to make it somewhat easier to generate a clock for use in creating output with NTSC-compatible timings.

A series of calculations are used to derive this value and the values for the other loop parameters given a desired output frequency and a series of constraints placed on different components within the synthesizer used to generate dot clock 1. See appendix A for a detailed description of the process used to derive the loop parameter values.

## XRC8 Dot Clock 2 VCO M-Divisor Low Register

Read/Write at I/O Address 3D7h with 3D6h set to Index C8h

7	6	5	4	3	2	1	0
Dot Clock 2 VCO M-Divisor Bits 7-0							

**Note:** All four of the registers used in specifying the loop parameters for the synthesizer used to generate dot clock 2 (XRC8-XRCB) must be written, and in sequence, before the hardware will update the synthesizer settings. This is meant to prevent wild fluctuations in the synthesizer output as new values are being written to these registers.

### 7-0 Dot Clock 2 VCO M-Divisor Bits 7-0

This register provides the eight least significant bits of a 10-bit value that specifies the M-divisor, one of the loop parameters used in controlling the frequency of the output of the synthesizer used to generate dot clock 2. Bits 1 and 0 of the Dot Clock 2 VCO M/N Divisor High Register (XRCA) provide the two most significant bits.

A series of calculations are used to derive this value and the values for the other loop parameters for a given desired output frequency and a series of constraints placed on different components within the synthesizer used to generate dot clock 2. See appendix A for a detailed description of the process used to derive the loop parameter values.

## **XRC9 Dot Clock 2 VCO N-Divisor Low Register**

Read/Write at I/O Address 3D7h with 3D6h set to Index C9h

7	6	5	4	3	2	1	0
Dot Clock 2 VCO N-Divisor Bits 7-0							

**Note:** All four of the registers used in specifying the loop parameters for the synthesizer used to generate dot clock 2 (XRC8-XRCB) must be written, and in sequence, before the hardware will update the synthesizer settings. This is meant to prevent wild fluctuations in the synthesizer output as new values are being written to these registers.

### **7-0 Dot Clock 2 VCO N-Divisor Bits 7-0**

This register provides the eight least significant bits of a 10-bit value that specifies the N-divisor, one of the loop parameters used in controlling the frequency of the output of the synthesizer used to generate dot clock 2. Bits 5 and 4 of the Dot Clock 2 VCO M/N Divisor High Register (XRCA) provide the two most significant bits.

A series of calculations are used to derive this value and the values for the other loop parameters for a given desired output frequency and a series of constraints placed on different components within the synthesizer used to generate dot clock 2. See appendix A for a detailed description of the process used to derive the loop parameter values.

## ***XRCA Dot Clock 2 VCO M/N-Divisor High Register***

Read/Write at I/O Address 3D7h with 3D6h set to Index CAh

7	6	5	4	3	2	1	0
Reserved		Dot Clock 2 VCO N-Divisor Bits 9-8		Reserved		Dot Clock 2 VCO M-Divisor Bits 9-8	

**Note:** All four of the registers used in specifying the loop parameters for the synthesizer used to generate dot clock 2 (XRC8-XRCB) must be written, and in sequence, before the hardware will update the synthesizer settings. This is meant to prevent wild fluctuations in the synthesizer output as new values are being written to these registers.

### **7-6 Reserved**

These bits always return the value of 0 when read.

### **5-4 Dot Clock 2 VCO N-Divisor Bits 9-8**

These two bits provide the two most significant bits of a 10-bit value that specifies the N-divisor, one of the loop parameters used in controlling the frequency of the output of the synthesizer used to generate dot clock 2. The Dot Clock 2 VCO N Divisor Low Register (XRC9) provides the eight least significant bits.

A series of calculations are used to derive this value and the values for the other loop parameters for a given desired output frequency and a series of constraints placed on different components within the synthesizer used to generate dot clock 2. See appendix A for a detailed description of the process used to derive the loop parameter values.

### **3-2 Reserved**

These bits always the value of 0 when read.

### **1-0 Dot Clock 0 VCO M-Divisor Bits 9-8**

These two bits provide the two most significant bits of a 10-bit value that specifies the M-divisor, one of the loop parameters used in controlling the frequency of the output of the synthesizer used to generate dot clock 2. The Dot Clock 2 VCO M Divisor Low Register (XRC8) provides the eight least significant bits.

A series of calculations are used to derive this value and the values for the other loop parameters for given desired output frequency and a series of constraints placed on different components within the synthesizer used to generate dot clock 2. See appendix A for a detailed description of the process used to derive the loop parameter values.

## XRCB Dot Clock 2 Divisor Select Register

Read/Write at I/O Address 3D7h with 3D6h set to Index CBh

7	6	5	4	3	2	1	0
Reserved	Post Divisor Select			Reserved	VCO Loop Divisor	Reference Clock Input Divisor	

**Note:** All four of the registers used in specifying the loop parameters for the synthesizer used to generate dot clock 2 (XRC8-XRCB) must be written, and in sequence, before the hardware will update the synthesizer settings. This is meant to prevent wild fluctuations in the synthesizer output as new values are being written to these registers.

### 7 Reserved

This bit always returns the value of 0 when read.

### 6-4 Post Divisor Select

These three bits select a value that specifies the post divisor, one of the loop parameters used in controlling the frequency of the output of the synthesizer used to generate dot clock 2. The manner in which these bits are used to choose this value is shown in the table below:

Bits 6 5 4	Post Divisor
0 0 0	1
0 0 1	2
0 1 0	4
0 1 1	8
1 0 0	16
1 0 1	32
1 1 0	Reserved
1 1 1	Reserved

A series of calculations are used to derive this value and the values for the other loop parameters for a given desired output frequency and a series of constraints placed on different components within the synthesizer used to generate dot clock 2. See appendix A for a detailed description of the process used to derive the loop parameter values.

### 3 Reserved

This bit always returns the value of 0 when read.



**XRCB Dot Clock 2 Divisor Select Register (continued)**

**2 VCO Loop Divisor Select**

This bit selects a value that specifies the VCO loop divide, one of the loop parameters used in controlling the frequency of the output of the synthesizer used to generate dot clock 2.

- 0 Selects a VCO loop divide value of 4.
- 1 Selects a VCO loop divide value of 16.

A series of calculations are used to derive this value and the values for the other loop parameters for a given desired output frequency and a series of constraints placed on different components within the synthesizer used to generate dot clock 2. See appendix A for a detailed description of the process used to derive the loop parameter values.

**1-0 Reference Input Clock Divisor Select**

These two bits select a value that specifies the reference input clock divisor, one of the loop parameters used in controlling the frequency of the output of the synthesizer used to generate dot clock 2. The manner in which these bits are used to choose this value is shown in the table below:

Bits 1 0	Reference Input Clock Divisor	Comment
0 0	4	
0 1	1	The reference input clock is not divided.
1 0	20	
1 1	5	A value intended to make it somewhat easier to generate a clock for use in creating output with NTSC-compatible timings.

A series of calculations are used to derive this value and the values for the other loop parameters for a given desired output frequency and a series of constraints placed on different components within the synthesizer used to generate dot clock 2. See appendix A for a detailed description of the process used to derive the loop parameter values.

## XRCC Memory Clock VCO M-Divisor Register

Read/Write at I/O Address 3D7h with 3D6h set to Index CCh

7	6	5	4	3	2	1	0
Reserved	Memory Clock VCO M-Divisor						

**Note:** Before any value is written to this register, bit 7 of the Memory Clock Divisor Select Register (XRCE) should be set to 0 to select the default memory clock.

### 7 Reserved

This bit always returns the value of 0 when read.

### 6-0 Memory Clock VCO M-Divisor

These seven bits provide a 7-bit value that specifies the M-divisor, one of the loop parameters used in controlling the frequency of the output of the synthesizer used to generate the memory clock.

A series of calculations are used to derive this value and the values for the other loop parameters for a given desired output frequency and a series of constraints placed on different components within the synthesizer used to generate the memory clock. See appendix A for a detailed description of the process used to derive the loop parameter values.

## ***XRCD Memory Clock VCO N-Divisor Register***

Read/Write at I/O Address 3D7h with 3D6h set to Index CDh

7	6	5	4	3	2	1	0
Reserved	Memory Clock VCO N-Divisor						

**Note:** Before any value is written to this register, bit 7 of the Memory Clock Divisor Select Register (XRCE) should be set to 0 to select the default memory clock.

### **7 Reserved**

This bit always returns the value of 0 when read.

### **6-0 Memory Clock VCO N-Divisor**

These seven bits provide a 7-bit value that specifies the N-divisor, one of the loop parameters used in controlling the frequency of the output of the synthesizer used to generate the memory clock.

A series of calculations are used to derive this value and the values for the other loop parameters for a given desired output frequency and a series of constraints placed on different components within the synthesizer used to generate the memory clock. See appendix A for a detailed description of the process used to derive the loop parameter values.

## XRCE Memory Clock Divisor Select Register

Read/Write at I/O Address 3D7h with 3D6h set to Index CEh

7	6	5	4	3	2	1	0
Memory Clock Select	Post Divisor Select			Reserved			Reference Clk Input Divisor

**Note:** Before any value is written to bits other than bit 7 of register, bit 7 of this register should be set to 0 to select the default memory clock.

### 7 Memory Clock Select

- 0 The memory clock output is set to a preset frequency of 25.175MHz. This is the default after reset.
- 1 The memory clock output is controlled by the loop parameters given to the memory clock synthesizer using a group of three registers (XRCC-XRCE) which includes this one.

### 6-4 Post Divisor Select

These three bits select a value that specifies the post divisor, one of the loop parameters used in controlling the frequency of the output of the synthesizer used to generate dot clock 0. The manner in which these bits are used to choose this value is shown in the table below:

Bits 6 5 4	Post Divisor
0 0 0	1
0 0 1	2
0 1 0	4
0 1 1	8
1 0 0	16
1 0 1	32
1 1 0	Reserved
1 1 1	Reserved

A series of calculations are used to derive this value and the values for the other loop parameters for a given desired output frequency and a series of constraints placed on different components within the synthesizer used to generate dot clock 0. See appendix A for a detailed description of the process used to derive the loop parameter values.

***XRCE Memory Clock Divisor Select Register (continued)*****3-1 Reserved**

These bits always return the value of 0 when read.

**0 Reference Input Clock Divisor Select**

This bit select a value that specifies the reference input clock divisor, one of the loop parameters used in controlling the frequency of the output of the synthesizer used to generate dot clock 0.

0 The reference input clock is divided by 4.

1 The reference input clock is divided by 1, i.e., the reference input clock remains undivided.

A series of calculations are used to derive this value and the values for the other loop parameters for a given desired output frequency and a series of constraints placed on different components within the synthesizer used to generate dot clock 0. See appendix A for a detailed description of the process used to derive the loop parameter values.

## XRCF Memory Clock Divisor Select Register

Read/Write at I/O Address 3D7h with 3D6h set to Index CFh

7	6	5	4	3	2	1	0
Reserved (0000)				Power Seq Ref Clock (0)	Dot Clock Source (x)	Mem Clk Source (x)	Reserved (0)

The bits in this register provide an indication of selections that have been made using various strapping pins.

### 7-4 Reserved

These bits always return the value of 0 when read.

### 3 Power Sequencing Reference Clock Select

- 0 The clock used to time the steps of panel powerdown or powerup is the reference input clock divided by 384. Presuming that the reference clock is the usual 14.31818MHz, the frequency resulting from this division should be 37.5KHz. This is the default after reset.
- 1 The clock used to time the steps of panel powerdown or powerup is the 32KHz clock provided as an input on the GPIO1 pin. This same clock is usually also used to provide a time base for memory refreshes during standby mode.

### 2 Dot Clock Source

- 0 An external clock source received through the TCLK pin is used to provide the dot clock. All three of the synthesizers otherwise used to generate the three selectable dot clocks are disabled.
- 1 The three synthesizers used to generate the three selectable dot clocks are enabled.

**Note:** The default state of this bit reflects the state of pin AA4 during reset. The state of pin AA4 during reset is also readable via bit 4 of the Configuration Pins 0 Register (XR70). Bit 4 of XR70 is read-only, while this bit is writable, allowing the source of the dot clock to be changed after reset.

### 1 Memory Clock Source

- 0 An external clock source is used to provide the memory clock. The synthesizer otherwise used to generate the memory clock is disabled. The graphics controller is configured to receive this external clock source on either one of two pins depending on the state of pin AA4 during reset. If AA4 was pulled low by an external pull-down resistor during reset, then the graphics controller will be configured to receive the external clock on the REFCLK pin. If AA4 was allowed to be pulled high by the internal pull-up resistor during reset, then the graphics controller is configured to receive the external clock on the TDI pin.
- 1 The synthesizer used to generate memory clock is enabled.

**Note:** The default state of this bit reflects the state of pin AA4 during reset. The state of pin AA4 during reset is also readable via bit 4 of the Configuration Pins 0 Register (XR70). Bit 4 of XR70 is read-only, while this bit is writable, allowing the source of the memory clock to be changed after reset.

### 0 Reserved

This bit always returns the value of 0 when read.

## ***XRD0 Powerdown Control Register***

Read/Write at I/O Address 3D7h with 3D6h set to Index D0h

7	6	5	4	3	2	1	0
Reserved (0)	Video Port Enable (0)	Capture Enable (1)	Playback Enable (1)	MCLK VCO En (1)	DCLK VCO En (1)	Palette Enable (1)	DAC Enable (1)

### **7 Reserved**

This bit always returns the value of 0 when read.

### **6 Video Port Enable**

0 Disables the video port. This is the default after reset.

1 Enables the video port.

### **5 Video Capture Enable**

0 Disables video capture.

1 Enables video capture. This is the default after reset.

### **4 Video Playback Enable**

0 Disables video playback.

1 Enables video playback. This is the default after reset.

### **3 Memory Clock VCO Enable**

0 Disables the memory clock VCO.

1 Enables the memory clock VCO. This is the default after reset.

### **2 Dot Clock VCO Enable**

0 Disables the dot clock VCO.

1 Enables the dot clock VCO. This is the default after reset.

### **1 Palette Enable**

0 Disables the palette.

1 Enables the palette. This is the default after reset.

### **0 D-to-A Converters Enable**

0 Disables the D-to-A converters.

1 Enables the D-to-A converters. This is the default after reset.

## XRD1 Power Conservation Control Register

read/write at I/O address 3D7h with 3D6h set to index D1h

7	6	5	4	3	2	1	0
Reserved							BitBLT Idle Pwr Down (0)
(0000:000)							

### 7-6 Reserved

This bit always returns the value of 0 when read.

### 0 BitBLT Engine Idle-State Power Down

- 0 Does not cause the BitBLT engine to automatically power down when idle.
- 1 Causes the BitBLT engine to automatically power down when idle.

Note: Use of this feature does not affect the usability of the BitBLT engine, and does not impede access to the BitBLT registers. The manner in which the BitBLT engine is programmed is not affected by the use of this feature.

## XRD2 2KHz Down Counter Register

Read/Write at I/O Address 3D7h with 3D6h set to Index D2h

7	6	5	4	3	2	1	0
32KHz Down Counter Bits 7-0 (00h)							

### 7-0 2KHz Down Counter Bits 7-0

This register provides the output of a looping 8-bit counter that is continuously decremented at a rate of 2KHz. The 2KHz frequency is derived from the same 14.318MHz reference frequency received from an external oscillator that is used as the base frequency for the generation of both the dot clock and memory clock.

This register is meant to be used to provide a fixed time base that can be used by CHIPS' BIOS to properly time the various steps to perform a powerdown or powerup of the graphics controller.



***XRE0-XREB Software Flag Registers 0 to B***

Read/Write at I/O Address 3D7h with 3D6h set to Indexes E0h to EBh

7	6	5	4	3	2	1	0
Software Flag Bits (xxxx:xxxx)							

**7-0 Software Flag Bits**

When the graphics controller is used in a PC environment, the bits in each of these eight registers are used almost exclusively by CHIPS' BIOS as a "scratch pad" for internal functions. To a very limited extent, these registers are also used as a medium of communication between CHIPS' BIOS and CHIPS' device drivers. No value written to any bit of any of these registers has any effect on the hardware of the graphics controllers -- in essence, these registers can be regarded as providing a few bytes of memory so that CHIPS' software needn't use frame buffer or system memory.

***XRF8-XRFC Test Registers***

Read/Write at I/O Address 3D7h with 3D6h set to Index F8h to FCh

7	6	5	4	3	2	1	0
Test Register Bits (xxxx:xxxx)							

**7-0 Test Register Bits**

The bits in each of these registers are used to perform chip testing, and should never be written to.

**This page intentionally left blank.**

## ***CHAPTER 15***

# ***FLAT PANEL REGISTERS***

**This page intentionally left blank.**

## 15.0 FLAT PANEL REGISTERS

### FR00 Feature Register

Read Only at I/O Address 3D1h

7	6	5	4	3	2	1	0
Reserved					H/W Pop-up	Reserved	Flat Panel

#### 7-3 Reserved (0000-0)

#### 2 Hardware Pop-up

- 0 Hardware support for pop-up menu does not exist
- 1 Hardware support for pop-up menu exists

#### 1 Reserved (0)

#### 0 Flat Panel

- 0 Flat Panel module does not exist
- 1 Flat Panel module exists

### FR01 CRT/FP Control

Read/ Write at I/O Address 3D1h

7	6	5	4	3	2	1	0
Reserved (R/W)						CRT/FP Control	

#### 7-2 Reserved (R/W) (0000-00)

#### 1-0 CRT/FP Control

- 00 CRT & FP display engines disabled.
- 01 CRT mode enabled. (Default)
- 10 FP mode enabled.
- 11 Reserved.

## FR02 FP Mode Control

Read/ Write at I/O Address 3D1h

7	6	5	4	3	2	1	0
Reserved (R/W)				Reserved			

**7-4 Reserved (R/W) (reset state: 0000)**

**3-0 Reserved (0)**

## FR03 FP Dot Clock Source

Read / Write at I/O Address 3D1h

7	6	5	4	3	2	1	0
Reserved			Synthesizer	FP Select Bits		Reserved (R/W)	

**7-5 Reserved (000)**

### 4 FP Clock Synthesizer Select

0 Use Video Clock Synthesizer to generate FP dot clock (default)

1 Use Memory Clock Synthesizer to generate FP dot clock.

This bit selects the graphics/video clock synthesizer to generate the FP dot clock in FP mode (FR01[1]=1). Note that CRT display engine always uses dot clock from the graphics/video clock synthesizer.

### 3-2 FP Clock Select Bits (reset state: 00)

Select graphics/video clock synthesizer frequency when not in CRT mode (FR01[0]=0). In CRT mode, the graphics/video clock synthesizer frequency is selected by MSR[3:2]. See description of MSR[3:2].

00 Select clock 0

01 Select clock 1

1x Select clock 2

**1-0 Reserved (R/W) (reset state: 00)**

## FR04 Panel Power Sequencing Delay

Read / Write at I/O Address 3D1h

7	6	5	4	3	2	1	0
Power Up Delay				Power Down Delay			

### 7-4 Power Up Delay (reset state: 1000)

Programmable value of panel power sequencing during power up. This value can be programmed up to 54 milliseconds in increments of 3.4 milliseconds. A value of 0 is undefined.

### 3-0 Power Down Delay (reset state: 0001)

Programmable value of panel power-sequencing during power down. This value can be programmed up to 459 milliseconds in increments of 27.5 milliseconds. A value 0 is undefined.

This register controls panel power on/off sequencing delays. The generation of the clock for the panel power sequencing logic is controlled by XRCF[3]. The delay intervals above assume a 37.5 KHz clock generated by the 14.31818 MHz reference clock. If using a 32KHz input, scale the delay intervals accordingly.

## FR05 Power Down Control 1

Read / Write at I/O Address 3D1h

7	6	5	4	3	2	1	0
CRT Mode	Refresh	Panel Off	Host Stby	Off Mode	FP Norm Refresh		

### 7 CRT Mode Control

0 Flat panel data and control signals are tri-stated with weak internal pull-down (default)

1 Flat panel data and control signals are driven inactive.

This bit is effective only CRT mode (flat panel is not active).

### 6 Standby Refresh Control

0 Self-Refresh DRAM support (default)

1 Non self-refresh DRAM. Display memory refresh frequency is derived from power sequencing clock

This bit is effective only in Standby mode and controls display memory refresh.

### 5 Standby and Panel Off Control

0 Flat panel data and control signals are driven inactive (default)

1 Flat panel data and control signals are tri-stated with a weak internal pull-down.

This bit is effective in Flat Panel Mode during Standby and Panel Off modes. This bit does not affect CRT control signals which will be driven low.

### 4 Host Standby Mode

0 Normal Mode (default)

1 Standby Mode

This bit disables the CPU interface, but allows the display to remain active. All CPU interface activity is ignored except RESET#. This bit can be cleared (re-enabling the CPU interface) by RESET# or a low-to-high transition on STNDBY#

### 3 Panel Off Mode

0 Normal mode (default)

1 Panel Off mode

When this bit is set, the chip enters Panel Off mode. In this mode, CRT/FP screen refresh is inactive but CPU interface and display memory refresh are still active. Display memory and I/O registers can still be accessed.

### 2-0 FP Normal Refresh Count (default = 001)

These bits specify the number of memory refresh cycles per scanline. These bits should have a minimum value of 001.



## FR06 FP Power Down Control

Read / Write at I/O Address 3D1h

7	6	5	4	3	2	1	0
Reserved					SYNC Enable	Reserved (Writable)	Palette Power down

### 7-3 Reserved (0000-0)

#### 2 HSync and VSync Enable When DAC is Off

0 Deactivate HSync and VSync when internal DAC is disabled (default)

1 Allow HSync and VSync to remain active when internal DAC is disabled

This bit is effective when internal DAC is disabled (bit 0 of XRD0 is set to 0).

#### 1 Reserved (Writable)

This bit should always be written to with the value of 0.

#### 0 Panel-Off VGA Palette Power down Enable

0 Disable VGA Palette power down in Panel Off mode (default)

1 Enable VGA Palette power down in Panel Off mode

## FR08 FP Pin Polarity

Read / Write at I/O Address 3D1h

7	6	5	4	3	2	1	0
Alt VSYNC Polarity	Alt HSYNC Polarity	Graphic Polarity	Text Polarity	FLM Polarity	LP Polarity	Dis Enbl Polarity	Reserved (R/W)

### 7 Alternate CRT VSync Polarity

This bit is used instead of MSR bit 7 when not in CRT mode FR01[0]=0.

- 0 Positive polarity (default)
- 1 Negative polarity

### 6 Alternate CRT HSync Polarity

This bit is used instead of MSR bit 6 when not in CRT mode FR01[0]=0.

- 0 Positive polarity (default)
- 1 Negative polarity

### 5 FP Graphics Video Output Polarity

This bit affects FP video data output in graphics mode only.

- 0 Normal polarity (default)
- 1 Inverted polarity

### 4 FP Text Video Output Polarity

This bit affects FP video data output in text mode only.

- 0 Normal polarity (default)
- 1 Inverted polarity

### 3 FP VSync (FLM) Polarity

- 0 Positive polarity (default)
- 1 Negative polarity

### 2 FP HSync (LP) Polarity

- 0 Positive polarity (default)
- 1 Negative polarity

### 1 FP Display Enable Polarity

- 0 Positive polarity (default)
- 1 Negative polarity

### 0 Reserved (R/W)

## FR0A Programmable Output Drive

Read / Write at I/O Address 3D1h

7	6	5	4	3	2	1	0
Mem Addr Drive	HS, VS & ACTI	Reserved (Writable)	Mem Ctrl & Data	Bus Output Drive	FP Output Drive	Reserved (Writable)	

**Note:** This register controls the input threshold and output drive of the bus, video, and memory interface pins.

### 7 Memory Interface Address Output Drive Select

- 0 Lower drive (Default)
- 1 Higher drive

### 6 HSYNC, VSYNC, ACTI, PCLK output drive select 0

- 0 Lower drive (Default)
- 1 Higher drive

### 5 Reserved (Writable)

This bit should always be set to the value of 0.

### 4 Memory Interface Control and Data Output Drive Select

- 0 Lower drive (Default)
- 1 Higher drive

### 3 Bus Interface Output Drive Select

- 0 Higher drive (Default)
- 1 Lower drive

### 2 Flat Panel Interface Output Drive Select

- 0 Lower drive (Default)
- 1 Higher drive

### 1-0 Reserved (Writable)

These bits should always be set to the value of 0.

## FR0B FP Pin Control 1

Read / Write at I/O Address 3D1h

7	6	5	4	3	2	1	0
Tri State Mem C	Reserved (R/W)	Comp Sync	Reserved	Pins W4 & U6	Pins U3 & V2	Pin Y4	Pin V6

### 7 Tri-State Memory C Interface

- 0 Normal Operation (default)
- 1 3-State memory C output pins

### 6 Reserved (R/W) (reset state: 0)

### 5 Simple Composite Sync

- 0 Output CRT horizontal sync on HSYNC output pin.
- 1 Output CRT horizontal sync OR'd with CRT vertical sync on HSYNC output pin.  
Effective only when XR0B[2] = 0 and only affects HSYNC pin.

### 4 Reserved (0)

### 3 Pin W4 and Pin U6 Select

- 0 Enable VEE (ENAVEE) goes to pin W4. Enable Backlight (ENABKL) goes to pin U6. (default).
- 1 Enable VEE (ENAVEE) goes to pin U6. Enable Backlight (ENABKL) goes to pin U6.

### 2 Pin U3 and Pin V2 Select

- 0 CRT HSync signal goes to pin U3. CRT VSync signal goes to pin V2. (default)
- 1 Composite Sync (CSYNC) goes to pin U3. Modified VSync signal goes to pin V2.

### 1 Pin Y4 Select

- 0 FP HSync (LP) signal goes to pin Y4 (default)
- 1 FP Display Enable (FP Blank#) goes to pin Y4.

### 0 Pin V6 Select

- 0 FP "M" signal goes to pin V6 (default)
- 1 FP Display Enable (FP Blank#) goes to pin V6.

## FR0C Pin Control 2

Read / Write at I/O Address 3D1h

7	6	5	4	3	2	1	0
GPIO ENABKL		Reserved (R/W)	GPIO (ACTI)		Reserved (R/W)		

### 7-6 GPIO1 (C32KHz) Pin Control

00 Pin T4 is C32KHz input (default)

Also see XR63[3]

01 Reserved

10 Pin T4 is general purpose input 1 (GPIO1). Data is read into XR63[1]

11 Pin T4 is general purpose output 1 (GPIO1). Data comes from XR63[1]

### 5 Reserved (R/W) (reset state: 0)

### 4-3 GPIO0 (ACTI) Pin Control

00 Pin V1 is ACTI output (default)

01 Pin V1 is Composite Sync output

10 Pin V1 is general purpose input 0 (GPIO0). Data is read into XR63[0]

11 Pin V1 is general purpose output 0 (GPIO0). Data comes from XR63[0]

### 2-0 Reserved (R/W) (reset state: 000)

## FR0F Activity Timer Control

Read / Write at I/O Address 3D1h

7	6	5	4	3	2	1	0
Activity Timer	Timer Action	Reserved (R/W)	Activity Timer Count				

### 7 Enable Activity Timer

- 0 Disable activity timer (default on reset)
- 1 Enable activity timer

### 6 Activity Timer Action

- 0 When the activity timer count is reached, the ENABKL pin is activated (driven low to turn the backlight off)
- 1 When the activity timer count is reached, Panel Off mode is entered.

### 5 Reserved (R/W)

### 4-0 Activity Timer Count

For a 37.5KHz power sequencing clock, the counter resolution is 28.1 seconds. The minimum programmed value of 0 results in 28.1 seconds delay, and the maximum value of 1Eh results in a delay of about 15 minutes. A value of 1Fh results in no delay.

This register controls the activity timer functions. The activity timer is an internal counter that starts from a value programmed into this register (see bits 0-4 below) and is reset back to that count by read or write accesses to graphics memory or standard VGA I/O. Reading or writing extended VGA registers does not reset the counter. If no accesses occur, the counter increments until the end of its programmed interval, then activates either the ENABKL pin or Panel Off mode (as selected by bit-6 below). The timer count does not need to be reloaded once programmed and the timer enabled. Any access to the chip with the timer timed out (ENABKL active or Panel Off mode active) resets the timer and deactivates the ENABKL (or Panel Off mode) pin. The activity timer uses the same clock as the power sequencing logic. The delay intervals assume a 37.5 KHz clock. If using a 32KHz input, scale the delay intervals accordingly.

## FR10 FP Format 0

Read / Write at I/O Address 3D1h

7	6	5	4	3	2	1	0
Reserved (R/W)	Shift Clock Divide			Mono / Color		Panel Type	

**7 Reserved (R/W) (reset state: 0)**

**6-4 Shift Clock Divide (reset state: 000)**

These bits specify the frequency ratio between the internal dot clock (DCLK) and flat panel shift clock (SHFCLK) signal. See tables at the end of this register description.

**3-2 Panel Monochrome/Color Select**

- 00 Monochrome panel: NTSC weighting color reduction algorithm (default)
- 01 Monochrome panel: Equivalent weighting color reduction algorithm
- 10 Monochrome panel: Greenonly color reduction algorithm
- 11 Color panel

For monochrome panels, these bits select the algorithm used to reduce 18 and 24-bit color data to 6 and 8-bit color data.

**1-0 Panel Type**

- 00 Single Panel Single Drive (SS) (default)
- 01 Reserved
- 10 Reserved
- 11 Dual Panel Dual Drive (DD)

**Clock Divide Information for bits 6-4 follows:**

### Monochrome TFT

Bits [6-4]	SHFCLK	Pixel /SHFCLK	Max bpp
000	DCLK	1	8
001	DCLK/2	2	8
010	DCLK/4	4	4
011	DCLK/8	8	2
100	DCLK/16	16	1
101	—	—	—
110	—	—	—
111	—	—	—

**FR10 FP Format 0 (continued)****6-4 Shift Clock Divide (continued)**

These bits specify the frequency ratio between the internal dot clock (DCLK) and flat panel shift clock (SHFCLK) signal.

**Color TFT**

Bits [6-4]	SHFCLK	Pixel / SHFCLK	Max bpp
000	DCLK	1	24
001	DCLK/2	2	12
010	—	—	—
011	—	—	—
100	—	—	—
101	—	—	—
110	—	—	—
111	—	—	—

**4-bit pack Color STN-SS**

Bits [6-4]	SHFCLK	Pixel / SHFCLK	Max bpp
000	DCLK	1 1/3	4
001	DCLK/2	2 2/3	8
010	DCLK/4	5 1/3	16
011	—	—	—
100	—	—	—
101	—	—	—
110	—	—	—
111	—	—	—

**Monochrome STN-DD w/o frame accel.**

Bits [6-4]	SHFCLK	Pixel / SHFCLK	Max bpp
000	DCLK	1	2
001	DCLK/2	2	4
010	DCLK/4	4	8
011	DCLK/8	8	16
100	—	—	—
101	—	—	—
110	—	—	—
111	—	—	—

**Monochrome STN-DD w/o frame accel.**

Bits [6-4]	SHFCLK	Pixel / SHFCLK	Max bpp
000	—	—	—
001	DCLK/2	2	2
010	DCLK/4	4	4
011	DCLK/8	8	8
100	DCLK/16	16	16
101	—	—	—
110	—	—	—
111	—	—	—

**4-bit pack color STN-DD w/frame accel.**

Bits [6-4]	SHFCLK	Pixel / SHFCLK	Max bpp
000	DCLK	2 2/3	8
001	DCLK/2	5 1/3	16
010	—	—	—
011	—	—	—
100	—	—	—
101	—	—	—
110	—	—	—
111	—	—	—

**4-bit pack color STN-DD w/o frame accel.**

Bits [6-4]	SHFCLK	Pixel / SHFCLK	Max bpp
000	—	—	—
001	DCLK/2	2 2/3	8
010	DCLK/4	5 1/3	16
011	—	—	—
100	—	—	—
101	—	—	—
110	—	—	—
111	—	—	—

**3-bit pack color STN-DD w/ frame accel.**

Bits [6-4]	SHFCLK	Pixel / SHFCLK	Max bpp
000	DCLK	2	6
001	DCLK/2	4	12
010	DCLK/4	8	24
011	—	—	—
100	—	—	—
101	—	—	—
110	—	—	—
111	—	—	—

**3-bit pack color STN-DD w/o frame accel.**

Bits [6-4]	SHFCLK	Pixel / SHFCLK	Max bpp
000	—	—	—
001	DCLK/2	2	6
010	DCLK/4	4	12
011	DCLK/8	8	24
100	—	—	—
101	—	—	—
110	—	—	—
111	—	—	—



## FR11 FP Format 1

Read / Write at I/O Address 3D1h

7	6	5	4	3	2	1	0
Res Dither	Bits Per Pixel			Dither Enable		FRC	

### 7 FP Restrict Dither (reset state: 0)

- 0 Dithering can be enabled on all modes.
- 1 Dithering can be enabled only on modes with more than 256 colors.

### 6-4 Bits Per Pixel Select (reset state: 000)

**Gray/Color without dither**

Bits [6:4]	#MSBs Used	No FRC	2 Frame FRC	16Frame FRC
000	0	—	—	—
001	1	2	—	—
010	2	4	3	—
011	3	8	5	—
100	4	16	15	16
101	5	32	31	—
110	6	64	—	—
111	8	256	—	—

**Gray/Color with dither**

Bits [6:4]	#MSBs Used	No FRC	2 Frame FRC	16Frame FRC
000	0	—	—	—
001	1	5	—	—
010	2	13	9	—
011	3	29	25	—
100	4	61	57	61
101	5	125	121	—
110	6	253	—	—
111	8	—	—	—

**Notes:**

- 1) No FRC is the recommended setting when interfacing with color TFT panel with more than 12 bits per pixel (4K color) or interfacing with monochrome panel with internal gray scaling. When No FRC is chosen FR11[6:4] should be programmed equal to the number of bits/color of the panel. For example, a TFT panel with 18 bits/pixel color uses 6 bits/color. FR11[6:4] should be programmed to 110b.
- 2) 2 FRC should be used with color TFT panel with less than or equal to 12 bits per pixel (<4k color) or used with monochrome panel with internal gray scaling. When 2 FRC is chosen FR11[6:4] should be programmed equal to the number of bits/color of the panel plus 1. The extra bit is for the two frame FRC. For example, a TFT panel with 9 bits/pixel color uses 3 bits/color. FR11[6:4] should be programmed equal to 100b.
- 3) 16 FRC should be used with STN panel. To achieve 16 frame FRC, 4 bits are needed for each color (R, G, B)
- 4) When 2-bit dither is disabled, the theoretical Color/Gray level per R, G, and B is calculated by using the formula below:

$$\text{Theoretical Color/Gray level} = 2^X \quad \text{where X is number of bits/color selected}$$

When 2 FRC or 16 FRC is enabled the actual Color/Gray level per R, G, and B that can be achieved is less than the theoretical Color/Gray level.

- 5) When 2-bit dither is enabled, the theoretical Color/Gray level per R, G, and B is calculated by using the formula below:

$$\text{Theoretical Color/Gray Level} = 4 * 2^X \quad \text{where X is number of bits/color selected}$$

When 2-bit dither, 2 FRC, or 16 FRC is enabled the actual achievable Color/Gray level per R, G, and B is less than the theoretical Color/Gray level.

**FR11 FP Format 0(continued)****3-2 Dither Enable**

- 00 Disable dithering (default)
- 01 Enable 2-bit dithering
- 10 Reserved for 4-bit dithering
- 11 Reserved

**1-0 Frame Rate Control (FRC)**

- 00 No FRC. This setting may be used with all panels, especially for panels which can generate shades of gray/color internally (default)
- 01 16-Frame FRC. This setting may be used for panels which do not support internal grayscale such as color STN or monochrome STN panels. This setting simulates up to 16 gray/color levels per pixel as specified in FR11[6:4].
- 10 2-frame FRC. This setting may be used with color/monochrome panels, especially for panels which can generate shades of gray/color internally. Valid number of bits/pixel is specified in FR11[6:4].
- 11 Reserved.

FRC is grayscale simulation on frame-by-frame basis to generate shades of gray or color on panels that do not generate gray/color levels internally.

## FR12 FP Format 2

Read / Write at I/O Address 3D1h

7	6	5	4	3	2	1	0
Data Width		Force FP Data High	Force HSYNC	FP Blank# Select	Clk Mask STN-DD	Clock Mask	Clock Divide

### 7-6 FP Data Width

- 00 16-bit panel data width. For color TFT panel this is the 565 RGB interface (default)
- 01 24-bit panel data width. For color the TFT panel this is 888 RGB interface. This setting can also be used for the 24-bit color STN-DD panel.
- 10 Reserved.
- 11 36-bit panel data width (TFT panels only). Program 000 in shift clock divide bits of FR10.

### 5 Force FP Data Signals High during Vertical Blank

- 0 Flat panel data output signals are not forced high during vertical blanking.
- 1 Flat panel data output signals are forced high during vertical blanking.

### 4 Force FP HSync (LP) during Vertical Blank

- 0 FP Display Enable output is generated by inverting both FP Vertical and Horizontal Blank therefore FP Display Enable will not toggle active during Vertical Blank time. FP HSync (LP) is not generated during Vertical Blank except when bit 3 is set to 1.
- 1 FP Display Enable output is generated by inverting FP Horizontal Blank only therefore FP Display Enable will be active during Vertical Blank time. FP HSync (LP) will also be active during Vertical Blank.

This bit should be set only for SS panels which require FP HSync (LP) to be active during vertical blank time when bit 3 is 0. This bit should be reset when using DD panels or when bit 3 is 1. Default state is 0

### 3 FP Display Enable (FP Blank#) Select

- 0 The FP Display Enable is inactive during vertical blank time because the output comes from inverting both the FP Vertical and Horizontal blank. FP HSync is not generated during vertical blank except when bit 4 is set to 1. In 480-line DD panels, this option will generate exactly 240 FP HSync (LP) pulses. (default)
- 1 The FP Display Enable is active during Vertical blank time since the output comes from inverting the FP Horizontal Blank enable. FP HSync will also be active during vertical blank.

This bit controls FP Display Enable (FP Blank#) generation. This bit also affects FP HSync (LP) generation.

**FR12 FP Format 2 (continued)****2 Shift Clock Mask for STN-DD**

0 Allow Shift Clock output to toggle in first line of Vertical Blank (default)

1 Force Shift Clock output low in first line of Vertical Blank.

This is an option to eliminate dark lines in the middle of STN-DD panels.

**1 Shift Clock Mask**

0 Allow Shift Clock output to toggle outside the display enable interval (default)

1 Force Shift Clock output low outside the display enable interval.

**0 Shift Clock Divide**

0 Shift Clock to Dot Clock relationship is specified by FR10[6:4] (default)

1 Shift Clock is further divided by 2 and different video data is valid on the rising and falling edges of Shift Clock.

## FR13 FP Format 3

Read / Write at I/O Address 3D1h

7	6	5	4	3	2	1	0
Reserved (R/W)					Set Up Time	Color STN	

### 7-3 Reserved (R/W) (reset state: 0000-0)

#### 2 Increase Setup Time 16-bit Color STN-DD

- 0 Normal data setup time with respect to SHFCLK falling edge (default). Maximum SHFCLK frequency is DCLK/2 (1:1 duty cycle).
- 1 Extended data setup time with respect to SHFCLK falling edge. The setup time is increased by approximately half dot clock cycle. This is done by extending SHFCLK high time by half dot clock cycle. Maximum SHFCLK frequency is DCLK/2.5, 1.5:1 duty cycle).

This bit is effective only for 16-bit Color STN-DD when frame acceleration is enabled or for 8-bit Color STN-DD when frame acceleration is disabled.

#### 1-0 Color STN Pixel Packing

- 00 3-bit pack (default).
- 01 4-bit pack.
- 10 Reserved.
- 11 Extended 4-bit pack. Bits FR10[6:4] must be programmed to 001. This setting may only be used for 8-bit interface color STN SS panels.

This determines the type of pixel packing (the RGB pixel output sequence) for color STN panels. These bits must be programmed to 00 for monochrome STN panels and for all TFT panels.

## FR16 FRC Option Select

Read / Write at I/O Address 3D1h

7	6	5	4	3	2	1	0
Reserved (R/W)					FRC Opt 3	FRC Opt 2	FRC Opt 1

### 7-3 Reserved (R/W)

These bits should always be written with 0's for future compatibility.

### 2 FRC Option 3

This affects 2-frame FRC and normally should be set to 1.

0 FRC data changes every frame

1 FRC data changes every other frame

### 1 FRC Option 2

This affects 16-frame FRC and normally should be set to 1.

0 2x2 FRC sub-matrix

1 2x4 FRC sub-matrix

### 0 FRC Option 1

This affects 16-frame FRC and normally should be set to 1.

0 15x31 FRC matrix

1 16x32 FRC matrix

## FR17 Polynomial FRC Control

Read / Write at I/O Address 3D1h

7	6	5	4	3	2	1	0
Polynomial M Value				Polynomial N Value			

### 7-4 Polynomial 'M' Value

### 3-0 Polynomial 'N' Value

This register affects the quality of both 2-frame and the 16-frame FRC algorithm. It controls the FRC polynomial counters. These values determine in row and column offsets of the FRC counters. These panel dependent values are usually determined by trial and error. These values require readjustment when the horizontal or vertical parameters change, especially when Vertical Total parameter is changed.

## FR18 FP Text Mode Control

Read / Write at I/O Address 3D1h

7	6	5	4	3	2	1	0
Reserved						Text Enhancement	

### 7-2 Reserved (0)

### 1-0 Text Enhancement

- 00 Normal text (default)
- 01 Text attribute 07h and 0Fh are reversed to maximize the brightness of the normal DOS prompt. This affects both CRT and Flat Panel displays.
- 10 Text attribute 07h and 0Fh are reversed to maximize the brightness of the normal DOS prompt. This affects Flat Panel displays.
- 11 Reserved

## FR19 Blink Rate Control

Read / Write at I/O Address 3D1h

7	6	5	4	3	2	1	0
Char Blink Duty Cycle		Cursor Blink Rate					

### 7-6 Character Blink Duty Cycle

These bits specify the character blink (also called 'attribute blink') duty cycle in text mode.

Bit 7	Bit 6	Character Blink Duty Cycle
0	0	50%
0	1	25%
1	0	50% (default on Reset)
1	1	75%

For setting 00, the character blink period is equal to the cursor blink period. For all other settings, the character blink period is twice the cursor blink period (character blink is half as fast as cursor blink).

### 5-0 Cursor Blink Rate (default = 03h)

These bits specify the cursor blink period in terms of number of VSyncs (50% duty cycle). In text mode, the character blink period and duty cycle is controlled by bits 7-6 of this register. These bits should be programmed to:

$$\text{Programmed value} = (\text{Actual Value}) / 2 - 1$$

**Note:** In graphics mode, the pixel blink period is fixed at 32 VSyncs per cursor blink period with 50% duty cycle (16 on and 16 off).



## FR1A STN-DD Buffering Control

Read / Write at I/O Address 3D1h

7	6	5	4	3	2	1	0
Reserved (Writable) (0000:00)						Frame Accel Enable (0)	Buffering Enable (0)

### 7-2 Reserved (Writable)

These bits should always be set to the value of 0.

### 1 STN-DD Frame Acceleration Enable

Enabling STN-DD frame acceleration doubles the screen refresh rate on an attached STN-DD panel relative to an attached CRT (each CRT frame corresponds to two STN-DD panel frames). The required memory bandwidth does not increase. In the simultaneous display mode, if the CRT refresh rate is 60Hz, the STN-DD panel refresh rate is 120Hz when STN-DD frame acceleration is enabled. Under the same conditions, the STN-DD panel refresh rate is 60Hz when STN-DD frame acceleration is disabled. Usually, STN-DD panels display higher quality images when STN-DD frame acceleration is enabled. If STN-DD frame acceleration is disabled, then the STN-DD buffer must be large enough to hold an entire frame consisting of 3-bits per pixel organized as 10 pixels per 32-bit dword. With STN-DD frame acceleration enabled, the required STN-DD buffer size is half this amount (only half a frame need be stored).

### 0 STN-DD Buffering Enable

- 0 Disables STN-DD buffering. This is the default after reset.
- 1 Enables STN-DD buffering.

STN-DD buffering is required for STN-DD panel operation. For STN-SS panel operation, STN-DD buffering is not required so this bit must be set to 0.

## FR1E M (ACDCLK) Control

Read / Write at I/O Address 3D1h

7	6	5	4	3	2	1	0
ACDCLK Control	M(ACDCLK) Count (ACDCNT)						

### 7 M (ACDCLK) Control

0 The M (ACDCLK) phase changes depending on bits 0-6 of this register

1 The M (ACDCLK) phase changes every frame if the frame accelerator is not used. If the frame accelerator is used, the M (ACDCLK) phase changes every other frame.

This register is used only in flat panel mode.

### 6-0 M (ACDCLK) Count (ACDCNT)

These bits define the number of HSyncs between adjacent phase changes on the M (ACDCLK) output. These bits are effective only when bit 7 = 0 and the contents of this register are greater than 2.

$$\text{Programmed Value} = \text{Actual Value} - 2$$

## FR1F Diagnostic

Read / Write at I/O Address 3D1h

7	6	5	4	3	2	1	0
Reserved (R/W)		Pixel Data Output Mode		Misc Mod Control 2	Misc Mod Control 2	Byp VGA Palette	Diag Mode

### 7-6 Reserved (R/W) (reset state: 00)

### 5-4 Pixel Data Pin Diagnostic Output Mode

These bits control the output of pins: SHFCLK, LP, M, P[15:0] and CA[7:0].

00 Normal Operation (default)

01 Output the following internal signals:

Signal	Pins
PDCLK	FLM
RDDE	LP
RDBLANK	M
RDVIDEO[23:16]	CA[7:0]
RDVIDEO[15:0]	P[15:0]

10 Output the following internal signals on P[15:0]

PDDELETE, PDGDCK, PHHSTR[2:0], PHREMAIN[10:0]

11 Output the following internal signals on P[13:0]

SS1ROMBOE, FHC32KHZI, FHXMEMRQ, T2DDSPBP, T2DDSPEN, T2DHBLANK, MXSQRDBG[7:0]

### 3 FP Miscellaneous module control 2

0 Normal Operation (default)

1 Enable the ring oscillator. The wave- form is output on ACTI pin. In addition, it is also output on pin A25 if the configuration option of pin AA4 is chosen to output clocks on A24 and A25.

### 2 FP Miscellaneous module control 2

0 Normal Operation (default)

1 Bypass clock divider for testing purposes

### 1 Bypass VGA Palette

0 Normal Operation (default)

1 Bypass internal VGA palette

### 0 FP Interface Diagnostic Mode

0 Normal Operation (default)

1 FP Interface Diagnostic Mode

## FR20 FP Horizontal Panel Display Size LSB

Read / Write at I/O Address 3D1h

7	6	5	4	3	2	1	0
FP Horizontal Panel Size LSB							

### 7-0 FP Horizontal Panel Size LSB

Programmed Value = Actual Value – 1

This parameter signifies the end of FP Horizontal Display Enable and the start of FP Horizontal Blank time relative to the start of FP Horizontal Display Enable. The most significant bits are programmed in FR25[3:0]. In FP mode (FR01[1]=1), this parameter is counted using a counter which is clocked with FP dot clock divided by 8 in all modes and is independent of horizontal compensation.

## FR21 FP Horizontal Sync Start LSB

Read / Write at I/O Address 3D1h

7	6	5	4	3	2	1	0
FP Horizontal Sync Start LSB							

### 7-0 FP Horizontal Sync Start LSB

Programmed Value = Actual Value – 1

In FP mode, this parameter is counted using a counter which is clocked with the FP dot clock divided by 8 in all modes and is independent of horizontal compensation. This parameter signifies the start of CRT HSync when not in CRT mode (FR01[0]=0). The most significant bits are programmed in FR25[7:4].

## FR22 FP Horizontal Sync End

Read / Write at I/O Address 3D1h

7	6	5	4	3	2	1	0
Reserved (R/W)			FP Horizontal Sync End				

### 7-5 Reserved (R/W) (Reset state: xxx)

In FP mode, this parameter is counted using a counter which is clocked with the FP dot clock divided by 8 in all modes and is independent of horizontal compensation. This parameter signifies the end of CRT HSync when not in CRT mode (FR01[0]=0). Only the 5 least significant bits are programmed.

### 4-0 FP Horizontal Sync End

## FR23 FP Horizontal Total LSB

Read / Write at I/O Address 3D1h

7	6	5	4	3	2	1	0
FP Horizontal Total LSB							

### 7-0 FP Horizontal Total LSB

Programmed Value = Actual Value – 5

In FP mode, this parameter is counted using a counter which is clocked with the FP dot clock divided by 8 in all modes and is independent of horizontal compensation. This parameter signifies the end of FP Horizontal Blank time and the start of FP Horizontal Display Enable relative to the start of the previous FP Horizontal Display Enable, i.e., the total size from one Horizontal Enable to the next. The most significant bits are programmed in FR26[3:0].

## FR24 FP HSync (LP) Delay LSB

Read / Write at I/O Address 3D1h

7	6	5	4	3	2	1	0
FP HSYNC (LP) Delay LSB							

### 7-0 FP HSync (LP) Delay LSB

In FP mode, this parameter is counted using a counter which is clocked with the FP dot clock divided by 8 in all modes and is independent of horizontal compensation. This register is effective when FR27[7]=0 and signifies the start of FP HSync (LP) measured from start of FP Horizontal Display Enable. This allows FP HSync (LP) to be positioned independently from CRT HSync. The most significant bits are programmed in FR26[7:4].

Programmed Value = Actual Value – 1

## FR25 FP Horizontal Overflow 1

Read / Write at I/O Address 3D1h

7	6	5	4	3	2	1	0
Reserved for Sync Start MSB				Reserved for Panel Size MSB			

### 7-4 Reserved (0) for FP Horizontal Sync Start MSB

### 3-0 Reserved (0) for FP Horizontal Panel SizeMSB

See description of FR20 and FR21.

## FR26 FP Horizontal Overflow 2

Read / Write at I/O Address 3D1h

7	6	5	4	3	2	1	0
Reserved for FP HSYNC			Reserved for Sync Start MSB	Reserved for FP Horizontal Total (MSB)			

**7-5 Reserved (0) for FP HSync (LP) Delay (bits-11-9)**

**4 FP HSync (LP) Delay (bit-8)**

**3-0 Reserved (0) for FP Horizontal Total (MSB)**

See description of FR23 and FR24.

## FR27 FP HSync (LP) Width and Disable

Read / Write at I/O Address 3D1h

7	6	5	4	3	2	1	0
Delay Disable	FP HSync LP Width						

**7 FP HSync (LP) Delay Disable**

0 FP HSync (LP) delay enable

1 FP HSync (LP) delay disable

In FP mode, this parameter is counted using a counter which is clocked with the FP dot clock divided by 8 in all modes and is independent of horizontal compensation.

**6-0 FP HSync (LP) Width**

Programmed Value = Actual Value – 1

## FR30 FP Vertical Panel Size LSB

Read / Write at I/O Address 3D1h

7	6	5	4	3	2	1	0
FP Vertical Panel Size LSB							

In FP mode (FR01[1]=1), this register is used to establish the end of FP Vertical Display Enable and the start of FP Vertical Blank time. The most significant bits are programmed in FR35[3:0].

### 7-0 FP Vertical Panel Size LSB

Programmed Value = Actual Value – 1

## FR31 FP Vertical Sync Start LSB

Read / Write at I/O Address 3D1h

7	6	5	4	3	2	1	0
FP Vertical Sync Start LSB							

### 7-0 FP Vertical Sync Start LSB

Programmed Value = Actual Value – 1

In FP mode (FR01[1]=1), this register signifies the start of CRT VSync (FR01[0]=0). The most significant bits are programmed in FR35[7:4].



## FR32 FP Vertical Sync End

Read / Write at I/O Address 3D1h

7	6	5	4	3	2	1	0
Reserved				FP Vertical Sync End			

### 7-4 Reserved (R/W) (Reset state: xxxx)

In FP mode (FR01[1]=1), this register signifies the end of CRT VSync. Only the 4 least significant bits are programmed.

### 3-0 FP Vertical Sync End

Programmed Value = Actual Value – 1

## FR33 FP Vertical Total LSB

Read / Write at I/O Address 3D1h

7	6	5	4	3	2	1	0
Vertical Total LSB							

### 7-0 Vertical Total LSB

FP Programmed Value = Actual Value – 2

In FP mode (FR01[1]=1), this register is used to establish the end of FP Vertical Blank time and the start of FP Vertical Display Enable. The most significant bits are programmed in FR36[3:0].

## FR34 FP VSync (FLM) Delay LSB

Read / Write at I/O Address 3D1h

7	6	5	4	3	2	1	0
FP VSync (FLM) Delay LSB							

### 7-0 FP VSync (FLM) Delay LSB

Programmed Value = Actual Value – 1

In FP mode (FR01[1]=1), this register is effective when FR37[7]=0 and FR37[6]=0. This register signifies the start of FP VSync (FLM) measured from start of CRT VSync which is programmed in FR31. This allows FP VSync (FLM) to be located in a different position from CRT VSync. The most significant bits are programmed in FR36[6:4].

## FR35 FP Vertical Overflow 1

Read / Write at I/O Address 3D1h

7	6	5	4	3	2	1	0
Vertical Sync Start [11-8]				Vertical Panel Size [11-8]			

### 7-4 FP Vertical Sync Start (bits 11-8)

### 3-0 FP Vertical Panel Size (bits 11-8)

See description of FR30 and FR31.

**FR36 FP Vertical Overflow 2**

Read / Write at I/O Address 3D1h

7	6	5	4	3	2	1	0
FLM bit 11	FLM Delay			FP Vertical Total [11-8]			

**7    Reserved (0) for FP FLM Delay (bit-11)**

**6-4    FP FLM Delay (bits 10-8)**

**3-0    FP Vertical Total (bits 11-8)**

See description of FR33 and FR34.

## FR37 FP VSync (FLM) Disable

Read / Write at I/O Address 3D1h

7	6	5	4	3	2	1	0
FLM Delay	FLM Select	FP VSync (FLM) width			Reserved (0)	SHFCLK Delay	

When the FP Display engine is enabled (FR01[1]=1), it uses this register.

### 7 FP VSync (FLM) Delay Disable

This bit is effective when FR37[6]=0

- 0 FP VSync (FLM) delay enable
- 1 FP VSync (FLM) delay disable

### 6 FP VSync (FLM) select

- 0 FP VSync (FLM) is generated using FR37[7] and FP VSync (FLM) Delay (FR36[6:4] and FR34) .
- 1 FP VSync (FLM) is the same as CRT VSync. FR37[7] is ignored in this case.

### 5-3 FP Vsync (FLM) width.

These bits are effective only if bit 6 is 0.

Programmed value = actual value -1

### 2 Reserved (0)

### 1-0 SHFCLK Delay (Applies to ES1 and later)

- 00 No delay (same as previous revisions)
- 01 Small delay
- 10 Large delay
- 11 Reserved

## FR40 Horizontal Compensation Register

Read / Write at I/O Address 3D1h

7	6	5	4	3	2	1	0
Reserved		EGHC	THCP		ETHC	EHC	EHCP

This register is used in FP mode (FR01[1]=1)

### 7-6 Reserved (R/W) (reset state: 00)

### 5 FP Enable Graphics Horizontal Compensation (EGHC) (reset state: 0)

- 0 Disable graphics mode horizontal compensation.
- 1 Enable graphics mode horizontal compensation which consists of horizontal stretching and FR41 is used to specify stretching for different horizontal resolutions.

This bit is effective only when bit 0 is 1.

### 4-3 Text Horizontal Compensation Priority (THCP)

- 00 Allow 9-dot compression to 8-dot if needed. If horizontal panel size is wide enough, 8-dot text remains 8-dot text and 9-dot text remains 9-dot text. If horizontal panel size is not wide enough, then 8-dot text remains 8-dot text and 9-dot text is forced to 8-dot text (default)
- 01 No compression or expansion. 8-dot text remains 8-dot text and 9-dot text remains as 9-dot text regardless of horizontal panel size.
- 10 Allow 8-dot expansion to 9-dot, or 9-dot compression to 8-dot, depending on horizontal panel size. If horizontal panel size is wide enough, 8-dot text is forced to 9-dot text and 9-dot text remains 9-dot text. If horizontal panel size is not wide enough then 8-dot text remains 8-dot text and 9-dot text is forced to 8-dot text.
- 11 Allow 8-dot and 9-dot expansion to 10-dot, or 8-dot expansion to 9-dot, or 9-dot compression to 8-dot, depending on horizontal panel size. If horizontal panel size is wide enough, 8-dot text is forced to 10-dot text and 9-dot text is forced to 10-dot text. Otherwise, if horizontal panel size is wide enough, 8-dot text is forced to 9-dot text and 9-dot text remains 9-dot text. If horizontal panel size is not wide enough, then 8-dot text remains 8-dot text and 9-dot text is forced to 8-dot text.

These bits are effective only when bit 0 is 1 and bit 2 is 1. These bits determine the text mode compression/stretching method to be applied if horizontal panel size is wide enough. If after applying the specified text compression/stretching, the horizontal panel size is still wider than the stretched image then further stretching will be applied using the same algorithm used for horizontal graphics compensation.

**FR40 Horizontal Compensation Register (continued)****2 Enable Text Horizontal Compensation (ETHC)**

0 Disable text mode horizontal compensation (default)

1 Enable text mode horizontal compensation.

This bit is effective only when bit 0 is 1. Text mode horizontal compensation priority/method is specified in bits [4:3]

**1 Enable Horizontal Centering (EHC)**

0 Disable horizontal centering (default)

1 Enable horizontal centering. Horizontal left and right borders will be computed automatically.

**0 Enable Horizontal Compensation (EHCP)**

0 Disable horizontal compensation (default)

1 Enable horizontal compensation

## FR41 Horizontal Stretching Register

Read / Write at I/O Address 3D1h

7	6	5	4	3	2	1	0
Reserved (0000)				Reserved (R/W)	Hor Stretch 1024 Col	Hor Stretch 800 Col	Hor Stretch 640 Col

This register is used when FR01[1] and FR40[0]=1 and graphics mode is enabled. This register must be set before FR40.

### 7-4 Reserved (0)

### 3 Reserved (R/W) (reset state: 0)

### 2 FP Enable Horizontal Stretching for 1024-column Graphics Mode

0 Disable horizontal stretching for 1024-column graphics mode.

1 Enable horizontal stretching for 1024-column graphics mode.

**Note:** That 1024-column graphics mode includes 512-column graphics mode with horizontal pixel doubling enabled.

### 1 FP Enable Horizontal Stretching for 800-column Graphics Mode

0 Disable horizontal stretching for 800-column graphics mode.

1 Enable horizontal stretching for 800-column graphics mode.

**Note:** That 800-column graphics mode includes 400-column graphics mode with horizontal pixel doubling enabled.

### 0 FP Enable Horizontal Stretching for 640-column Graphics Mode

0 Disable horizontal stretching for 640-column graphics mode.

1 Enable horizontal stretching for 640-column graphics mode.

**Note:** The 640-column graphics mode includes 320-column graphics mode with horizontal pixel doubling enabled.

## FR48 Vertical Compensation Register

Read / Write at I/O Address 3D1h

7	6	5	4	3	2	1	0
Reserved (R/W) (000)			ETVS	Text Mode Stretch	EVLN	Vertical Centering	EVCP

### 7-5 Reserved (R/W) (reset state: 0)

When the FP Display engine is enabled (FR01[1]=1), it uses this register.

### 4 Enable Text Mode Vertical Stretching (ETVS)

- 0 Disable vertical stretching (default)
- 1 Enable vertical stretching

### 3 Text Mode Vertical Stretching Priority

- 0 Priority: ETVS, EVLN (default)
- 1 Priority: EVLN, ETVS

This bit is effective in text modes if bits 2 and 4 are set.

### 2 Enable Vertical Line Replication (EVLN)

- 0 Disable vertical line replication (default)
- 1 Enable vertical line replication

This bit is effective in text and graphics modes.

### 1 Enable Vertical Centering

- 0 Disable vertical centering (default)
- 1 Enable vertical centering

This bit is effective only when bit 0 is "1".

### 0 Enable Vertical Compensation (EVCP)

- 0 Disable vertical compensation feature (default)
- 1 Enable vertical compensation feature



## FR49-4C Text Mode Vertical Stretching 0 MSB

Read / Write at I/O Address 3D1h

7	6	5	4	3	2	1	0
Replication Specification							

### 7-0 Replication Specifications

- 00 No replication
- 01 Replicate once
- 10 Replicate twice
- 11 Replicate three times

Font lines beyond 16 are not replicated.

This register specifies the new text mode vertical stretching (along with FR4A, FR4B, FR4C). FR49(MSB), FR4A(LSB) and FR4B (MSB), FR4C(LSB) constitute two 16 bit registers. Each of the 16 pairs of bits specify scan line replication as shown above.

- FR49 Text Mode Vertical Stretching 1 MSB
- FR4A Text Mode Vertical Stretching 1 LSB
- FR4B Text Mode Vertical Stretching 0 MSB
- FR4C Text Mode Vertical Stretching 0 LSB

## FR4D Vertical Line Replication Register

Read / Write at I/O Address 3D1h

7	6	5	4	3	2	1	0
VLRHH				VLRHL			

### 7-4 FP Vertical Line Replication Height High (VLRHH)

### 3-0 FP Vertical Line Replication Height Low (VLRHL)

This register is used in FP mode (FR01[1]=1) and when vertical line replication is enabled. The 4 bit number specifies the number of lines between replicated lines. Double scanned lines are counted. The state machine starts stretching by using the lower nibble value. If the stretched display does not fit it uses the next higher value. The process continues until the count is equal to upper nibble value or the display fits. The lower nibble value must be less than or equal to upper nibble value. Set this register before FR40.

## FR4E Selective Vertical Stretching Disable Register

Read / Write at I/O Address 3D1h

7	6	5	4	3	2	1	0
Reserved (00)		Disable 600 Graph	Disable 480 Graph	Disable 400Graph	Disable 350 Graph	Disable 400 Text	Disable 350 Text

**7-6 Reserved (R/W) (reset state: xx)**

**5 Disable 600-line Graphics Stretching**

0 Disable stretching

1 Enable stretching

**4 Disable 480-line Graphics Stretching**

0 Disable stretching

1 Enable stretching

**3 Disable 400-line Graphics Stretching**

0 Disable stretching

1 Enable stretching

**2 Disable 350-line Graphics Stretching**

0 Disable stretching

1 Enable stretching

**1 Disable 400-line Text Stretching**

0 Disable stretching

1 Enable stretching

**0 Disable 350-line Text Stretching**

0 Disable stretching

1 Enable stretching

This register is used to selectively disable vertical stretching based on the vertical display end parameter. The register is qualified by master enable bits in FR48. Set this register before FR40.

## FR70 TMED Red Seed Register

Read / Write at I/O Address 3D1h

7	6	5	4	3	2	1	0
TMED Red Seed							

### 7-0 TMED Red Seed

The 8-bit value written to this register specifies the seed value used in the TMED algorithm for red pixel data to improve images on dual-scan passive matrix LCD panels.

## FR71 TMED Green Seed Register

Read / Write at I/O Address 3D1h

7	6	5	4	3	2	1	0
TMED Green Seed							

### 7-0 TMED Green Seed

The 8-bit value written to this register specifies the seed value used in the TMED algorithm for green pixel data to improve images on dual-scan passive matrix LCD panels.

## FR72 TMED Blue Seed Register

Read / Write at I/O Address 3D1h

7	6	5	4	3	2	1	0
TMED Blue Seed							

### 7-0 TMED Blue Seed

The 8-bit value written to this register specifies the seed value used in the TMED algorithm for blue pixel data to improve images on dual-scan passive matrix LCD panels.

## FR73 TMED Control Register

Read / Write at I/O Address 3D1h

7	6	5	4	3	2	1	0
TMED Enable	Scheme Select	Shades per Color Select		Horizontal Beat Suppression			

### 7 TMED Enable

- 0 Disables TMED.
- 1 Enables TMED.

### 6 TMED Scheme Select

- 0 Selects TMED energy distribution scheme 2.
- 1 Selects TMED energy distributionscheme 1.

### 5-4 TMED Shades per Color Select

- 00 Selects 33 shades for red, 65 shades for green, and 33 shades for blue.
- 01 Selects 65 shades for red, green, and blue.
- 10 Selects 129 shades for red, green, and blue.
- 11 Selects 256 shades for red, green, and blue.

### 3-0 TMED Horizontal Beat Suppression

The value written to these 4 bits specifies the horizontal beat suppression factor.

**This page intentionally left blank.**

## ***CHAPTER 16***

# ***MULTIMEDIA REGISTERS***

**This page intentionally left blank.**



## 16.0 MULTIMEDIA REGISTERS

### MR00 Module Capability

Read Only at I/O Address 3D3h

7	6	5	4	3	2	1	0
Reserved						Capture	Playback

**7-2 Reserved**

**1 Capture Available**

- 0 Absent
- 1 Included

**0 Playback Available**

- 0 Absent
- 1 Included

### MR01 Secondary Capability

Read Only at I/O Address 3D3h

7	6	5	4	3	2	1	0
Reserved							

**7-0 Reserved**

## 16.1 CAPTURE CONTROL

### MR02 Capture Control 1

Read / Write at I/O Address 3D3h

Default = 00h

7	6	5	4	3	2	1	0
Field Det Method	Field Det Polarity	VSYNC Polarity	HSYNC Polarity	RGB Mode	Color	Format	Interlace

#### 7 Field Detect Method

- 0 Trailing Edge of V
- 1 Leading Edge of V

#### 6 Field Detect Polarity

- 0 Normal
- 1 Inverted

#### 5 VSYNC Polarity

- 0 Low asserted
- 1 High asserted

#### 4 HSYNC Polarity

- 0 Low asserted
- 1 High asserted

#### 3 RGB Mode

- 0 RGB16
- 1 RGB15

#### 2 Color

- 0 YUV
- 1 RGB

#### 1 Format

- 0 Video
- 1 Game

#### 0 Interlace

- 0 Interlace Enabled
- 1 Non-Interlace

## MR03 Capture Control 2

Read / Write at I/O Address 3D3h

Default = 00h

7	6	5	4	3	2	1	0
V Scaling Method		Y-Scale	X-Scale	Field Select	Frame/Field Capture	Continuous/Single	Capture Enable

### 7-6 V Scaling Method

- 00 Normal
- 01 Reserved
- 10 Overwrite
- 11 Reserved

### 5 Y-Scale Enable

- 0 Disabled
- 1 Scaled on V

### 4 X-Scale Enable

- 0 Disabled
- 1 Scaled on H

### 3 Field Select

- 0 Field 0
- 1 Field 1

Bit-3 is only effective when Bit-2 = 1

### 2 Frame/Field Capture

- 0 Frame
- 1 Field

### 1 Continuous/Single Frame/Field Video Data Capture

- 0 Causes the continuous capturing of video data from the video data port.
- 1 Causes the capture of a single frame or field (depending on the setting of bit 2 of this register) from the video data port.

### 0 Capture Enable

- 0 Stop
- 1 Start

## MR04 Capture Control 3

Read / Write at I/O Address 3D3h

Default = 00h

7	6	5	4	3	2	1	0
Capture Drop En	Reserved	Buffer Pointer	Buffer Enable	Reserved	Horizontal Filter	Y-Capture	X- Capture

### 7 Capture Frame/Field Drop Enable

- 0 Causes the capture of video data from the video data port without dropping frames or fields.
- 1 Causes the dropping of the number of frames/fields specified in the Capture Frame/Field Drop Count Register (MR18) between every frame/field that is saved.

### 6 Reserved

### 5 Double Buffer Pointer

- 0 PTR1 in use
- 1 PTR 2 in use

### 4 Double Buffer Enable

- 0 Double buffering disabled
- 1 Double buffering enabled

### 3 Reserved

### 2 Horizontal Filter Enable

- 0 No Filter
- 1 Filter pixels with horizontal filter

### 1 Y-Capture Direction

- 0 Normal: top to bottom
- 1 Flipped: bottom to top

### 0 X-Capture Direction

- 0 Normal: left to right
- 1 Mirrored: right to left

**Note:** Changing the X- or Y- capture direction (Bits 1-0) will also require a change in the capture memory address pointer.

## MR05 Capture Control 4

Read / Write at I/O Address 3D3h

Default = 00h

7	6	5	4	3	2	1	0
Input Byte	UV SWAP	Pixel Qual Polarity	Pixel Qual Enable	VSYNC	Last Frame	Current Address	Actual Grab

### 7 Input Byte Swap

- 0 Y on low 8 input pins, UV on high 8 input pins
- 1 Y on high 8 input pins, UV on low 8 input pins (VESA style)

### 6 UV SWAP

- 0 Normal UV sequence
- 1 Exchange U and V

### 5 Pixel Qualifier Polarity

- 0 Non-inverted
- 1 Inverted

### 4 Pixel Qualifier Enable

- 0 Continuous pixels gated by blank
- 1 PIXEN qualifies valid pixels

### 3 Input VSYNC (read only)

(After polarity correction)

### 2 Last Frame Grabbed (read only)

- 0 PTR1
- 1 PTR2

(Effective only with double buffering)

### 1 Current Address Pointer (read only)

- 0 PTR1 (Capture memory pointer 1)
- 1 PTR2 (Capture memory pointer 2)

Indicates which buffer is being grabbed is double buffering is enabled.

### 0 Actual Grab (read only)

- 0 Hardware frame capture stopped
- 1 Hardware frame capture active (synchronized to V)

## MR06 Capture Memory Address PTR1

Read / Write at I/O Address 3D3h

7	6	5	4	3	2	1	0
Capture Memory Address PTR 1 [07:00]							

### 7-0 Capture Memory Address PTR1 [07:00]

(Bit 2-0: forced to 0)

## MR07 Capture Memory Address PTR1

Read / Write at I/O Address 3D3h

7	6	5	4	3	2	1	0
Capture Memory Address PTR 1 [15:08]							

### 7-0 Capture Memory Address PTR1 [15:08]

## MR08 Capture Memory Address PTR1

Read / Write at I/O Address 3D3h

7	6	5	4	3	2	1	0
Capture Memory Address PTR 1 [23:16]							

### 7-0 Capture Memory Address PTR1 [23:16]

### **MR09 Capture Memory Address PTR2**

Read / Write at I/O Address 3D3h

7	6	5	4	3	2	1	0
Capture Memory Address PTR 2 [07:00]							

**7-0 Capture Memory Address PTR2 [07:00]**

(Bit 2-0: forced to 0)

### **MR0A Capture Memory Address PTR2**

Read / Write at I/O Address 3D3h

7	6	5	4	3	2	1	0
Capture Memory Address PTR 2 [15:08]							

**7-0 Capture Memory Address PTR2 [15:08]**

### **MR0B Capture Memory Address PTR2**

Read / Write at I/O Address 3D3h

7	6	5	4	3	2	1	0
Capture Memory Address PTR 2 [23:16]							

**7-0 Capture Memory PTR2 [23:16]**

## MR0C Capture Line Memory Storage Width

Read / Write at I/O Address 3D3h

7	6	5	4	3	2	1	0
Memory Width (span) [07:00]							

### 7-0 Memory Width (Span) [07:00]

(Pixel Width) / 4 mem-quad words (scaled if horizontal scaling is enabled by setting bit 4 of MR03 to 1).

## MR0E Capture Window X-LEFT

Read / Write at I/O Address 3D3h

7	6	5	4	3	2	1	0
Capture Window X-LEFT [07:00]							

### 7-0 Capture Window X-LEFT [07:00]

## MR0F Capture Window X-LEFT

Read / Write at I/O Address 3D3h

7	6	5	4	3	2	1	0
Reserved					Capture Window X-LEFT [10:08]		

### 7-3 Reserved

### 2-0 Capture Window X-LEFT [10:08]

Maximum capture window size is 1024x1024.



**MR10 Capture Window X-RIGHT**

Read / Write at I/O Address 3D3h

7	6	5	4	3	2	1	0
Capture Window X-RIGHT [07:00]							

**7-0 Capture Window X-RIGHT [07:00]**

**MR11 Capture Window X-RIGHT**

Read / Write at I/O Address 3D3h

7	6	5	4	3	2	1	0
Reserved					Capture Window X-RIGHT [07:00]		

**7-3 Reserved**

**2-0 Capture Window X-RIGHT [10:08]**

**MR12 Capture Window Y-TOP**

Read / Write at I/O Address 3D3h

7	6	5	4	3	2	1	0
Capture Window Y-TOP [07:00]							

**7-0 Capture Window Y-TOP [07:00]**

## MR13 Capture Window Y-TOP

Read / Write at I/O Address 3D3h

7	6	5	4	3	2	1	0
Reserved					Capture Window Y-TOP [10:08]		

7-3 Reserved

2-0 Capture Window Y-TOP [10:08]

## MR14 Capture Window Y-BOTTOM

Read / Write at I/O Address 3D3h

7	6	5	4	3	2	1	0
Capture Window Y-BOTTOM [07:00]							

7-0 Capture Window Y-BOTTOM [07:00]

## MR15 Capture Window Y-BOTTOM

Read / Write at I/O Address 3D3h

7	6	5	4	3	2	1	0
Reserved					Acq. Window Y-BOTTOM [10:08]		

7-3 Reserved

2-0 Capture Window Y-BOTTOM [10:08]

**MR16 H-SCALE**

Read / Write at I/O Address 3D3h

7	6	5	4	3	2	1	0
H-SCALE [07:00]							

**7-0 H-SCALE [07:00]**

**MR17 V-SCALE**

Read / Write at I/O Address 3D3h

7	6	5	4	3	2	1	0
V-SCALE [07:00]							

**7-0 V-SCALE [07:00]**

**MR18 Capture Frame/Field Drop Count**

Read / Write at I/O Address 3D3h

7	6	5	4	3	2	1	0
AQ Capture Frame Count [07:00]							

**7-0 Capture Frame/Field Drop Count**

When the dropping of frames/fields is enabled by setting bit 2 of the Capture Control 2 Register (MR03) to 1, these 8 bits set the number of captured frames/fields to be dropped between every frame/field that is captured and saved.

## 16.2 PLAYBACK CONTROL

### MR1E Playback Control 1

Read / Write at I/O Address 3D3h

Default = 00h

7	6	5	4	3	2	1	0
Reserved			Playback Interface	V-ZOOM	H-ZOOM	Y-Playback	X-Playback

#### 7-5 Reserved

#### 4 Playback Interlace Enable

- 0 Non-Interlaced
- 1 Interlaced

#### 3 V-ZOOM Enable

- 0 Normal
- 1 Video playback is zoomed to the degree specified in the V-ZOOM Register (MR33).

#### 2 H-ZOOM Enable

- 0 Normal
- 1 Video playback is zoomed to the degree specified in the H-ZOOM Register (MR32).

#### 1 Y-Playback Direction

- 0 Normal: top to bottom
- 1 Flipped: bottom to top

Be sure to change memory pointer value of PTR1 (MR22 - MR24) and/or PTR2 (MR25 - MR27) if flipped.

#### 0 X-Playback Direction

- 0 Normal: left to right
- 1 Mirrored: right to left

Be sure to change memory pointer value of PTR1 (MR22 - MR24) and/or PTR2 (MR25 - MR27) if mirrored.

## MR1F Playback Control 2

Read / Write at I/O Address 3D3h

Default = 00h

7	6	5	4	3	2	1	0
V Inter Enable	V Inter Mode	H Inter Enable	Reserved	Color Mode	Reserved	UV Sign	Color Type

### 7 V Interpolate Enable

- 0 Disable
- 1 Enable

### 6 V Interpolate Mode

- 0 De-block
- 1 Running Average (when bit 7 is set)

### 5 H Interpolate Enable

### 4 Reserved

### 3 Color Mode Select

- 0 YUV
- 1 RGB

See color mode table below.

### 2 Reserved

### 1 UV Sign

- 0 UV Unsigned (signed offset)
- 1 UV Signed (2's complement)

### 0 Color Type Select (See bit 3)

- 0 Normal (U and V, or RGB16)
- 1 Exchange U and V positions, or RGB15

Color Mode table for bit-3:

Bit				Color Format
3	2	1	0	
0	x	0	0	YUV 4:2:2
0	x	0	1	YVU 4:2:2; UV Swap
0	x	1	0	YUV 4:2:2; UV=2's comp
0	x	1	1	YVU 4:2:2; UV=2'comp, UV swap
1	x	x	0	RGB16; R5G6B5 (B=LSB)
1	x	x	1	RGB15, xR5G5B5 (B=LSB)

## MR20 Playback Control 3

Read / Write at I/O Address 3D3h

Default = 00h

7	6	5	4	3	2	1	0
Playback Vert Ctr	Playback Width	Capture Ptr Ctrl	Buffer Flag	Capture Ptr Ctrl	Buffer Trigger	Reserved	

### 7 Playback Vertical Auto-Centering Enable

- 0 Allow software to employ a delay to properly center the playback window vertically. This is done usually via bit 4 of the Pixel Pipeline Configuration Register 1 (XR81).
- 1 Activate a hardware-based auto-centering mechanism.

### 6 Playback Width Source

- 0 Use MR28 for Playback width
- 1 Use MR34 for Playback width

### 5 Playback Pointer Control

- 0 The pointer to the location in the frame buffer from which frames/fields of video data are played back is selected by bit 4 of this register.
- 1 The pointer to the location in the frame buffer from which frames/fields of video data are played back is controlled by bit 3 of this register.

### 4 CPU Double Buffer Flag

- 0 Playback memory address PTR1
- 1 Playback memory address PTR2

### 3 Playback Pointer Control

- 0 The pointer to the location in the frame buffer from which frames/fields of video data are played back is selected by bit 4 of this register.
- 1 The pointer to the location in the frame buffer from which frames/fields of video data are played back toggles between the addresses indicated by PTR1 and PTR2 after each frame/field captured.

### 2 Double Buffer Trigger

- 0 Retain old PTR
- 1 Take new PTR on next VSYNC if bit 5=1

### 1-0 Reserved

## **MR21 Double Buffer Status**

Read Only at I/O Address 3D3h

7	6	5	4	3	2	1	0
Reserved						Buffer Pointer	Buffer Trigger

### **7-2 Reserved**

#### **1 Double Buffer Pointer in Use**

- 0 PTR1
- 1 PTR2

#### **0 Double Buffer Trigger Status**

- 0 Taken
- 1 Pending

## MR22 Playback Window Address PTR1

Read / Write at I/O Address 3D3h

7	6	5	4	3	2	1	0
Playback Memory Address PTR1 [07:00]							

### 7-0 Playback Memory Address PTR1 [07:00]

(Bit 2-0: forced to 0)

## MR23 Playback Window Address PTR1

Read / Write at I/O Address 3D3h

7	6	5	4	3	2	1	0
Playback Memory Address PTR1 [15:08]							

### 7-0 Playback Memory Address PTR1 [15:08]

## MR24 Playback Window Address PTR1

Read / Write at I/O Address 3D3h

7	6	5	4	3	2	1	0
Playback Memory Address PTR1 [23:16]							

### 7-0 Playback Memory Address PTR1 [23:16]



**MR25 Playback Window Address PTR2**

Read / Write at I/O Address 3D3h

7	6	5	4	3	2	1	0
Playback Memory Address PTR2 [07:00]					0	0	0

**7-0 Playback Memory Address PTR2 [07:00]**

(Bit 2-0: forced to 0)

**MR26 Playback Window Address PTR2**

Read / Write at I/O Address 3D3h

7	6	5	4	3	2	1	0
Playback Memory Address PTR2 [15:08]							

**7-0 Playback Memory Address PTR2 [15:08]**

**MR27 Playback Window Address PTR2**

Read / Write at I/O Address 3D3h

7	6	5	4	3	2	1	0
Playback Memory Address PTR2 [23:16]							

**7-0 Playback Memory Address PTR1 [23:16]**

## MR28 Playback Line Memory Fetch Width

Read / Write at I/O Address 3D3h

7	6	5	4	3	2	1	0
Playback Line Memory Fetch Width [07:00]							

### 7-0 Playback Line Memory Fetch Width [07:00]

These 8 bits specify the number of quadwords read by the playback engine from the frame buffer to display a horizontal line's worth of video data. Normally, this value is set equal to the actual number of quadwords required to store a horizontal line's worth of video data captured from the video data port -- i.e., normally this value is the same as that of register MR0C.

If bit 6 of the Playback Control 3 Register (MR0C) is set to 0, then this register also specifies the number of quadwords out of a horizontal line's worth of video data that is actually played back, starting at the left-most edge of the video playback window.

This value is calculated as follows:  $(\text{width of line in pixels}) / (\text{number of pixels per quadwords}) - 1$ .

## MR2A Playback Window X-LEFT

Read / Write at I/O Address 3D3h

Default = 00h

7	6	5	4	3	2	1	0
Playback Window X-LEFT [07:00]							

### 7-0 Playback Window X-LEFT [07:00]

## MR2B Playback Window X-LEFT

Read / Write at I/O Address 3D3h

Default = 00h

7	6	5	4	3	2	1	0
Reserved					Playback Window X-LEFT [10:08]		

### 7-3 Reserved

### 2-0 Playback Window X-LEFT [10:08]

**MR2C Playback Window X-RIGHT**

Read / Write at I/O Address 3D3h

Default = 00h

7	6	5	4	3	2	1	0
Playback Window X-RIGHT [07:00]							

**7-0 Playback Window X-RIGHT [07:00]**

**MR2D Playback Window X-RIGHT**

Read / Write at I/O Address 3D3h

Default = 00h

7	6	5	4	3	2	1	0
Reserved					Playback Window X-RIGHT [10:08]		

**7-3 Reserved**

**2-0 Playback Window X-RIGHT [10:08]**

**MR2E Playback Window Y-TOP**

Read / Write at I/O Address 3D3h

Default = 00h

7	6	5	4	3	2	1	0
Playback Window Y-TOP [07:00]							

**7-0 Playback Window Y-TOP [07:00]**

## MR2F Playback Window Y-TOP

Read / Write at I/O Address 3D3h

Default = 00h

7	6	5	4	3	2	1	0
Reserved					Playback Window Y-TOP [10:08]		

7-3 Reserved

2-0 Playback Window Y-TOP [10:08]

## MR30 Playback Window Y-BOTTOM

Read / Write at I/O Address 3D3h

Default = 00h

7	6	5	4	3	2	1	0
Playback Window Y-BOTTOM [07:00]							

7-0 Playback Window Y-BOTTOM [07:00]

## MR31 Playback Window Y-BOTTOM

Read / Write at I/O Address 3D3h

Default = 00h

7	6	5	4	3	2	1	0
Reserved					Playback Window Y-BOTTOM [10:08]		

7-3 Reserved

2-0 Playback Window Y-BOTTOM [10:08]

**MR32 H-ZOOM**

Read / Write at I/O Address 3D3h

Default = 00h

7	6	5	4	3	2	1	0
H-ZOOM						Reserved (00)	

**7-2 H-ZOOM**

When enabled by setting bit 2 of the Playback Control 1 Register (MR1E) to 1, these six bits are used to specify the zoom factor by which the playback image is magnified.

Zoom factor = 100h / (value of bits 7 to 2 of this register)

Examples of programmed values:

Bits 7 6 5 4 3 2	Resulting Zoom Factor
1 0 0 0 0 0	Magnify by 2
0 1 0 0 0 0	Magnify by 4
0 0 1 0 0 0	Magnify by 8

**1-0 Reserved**

These bits always return the value of 0 when read.

## MR33 V-ZOOM

Read / Write at I/O Address 3D3h

Default = 00h

7	6	5	4	3	2	1	0
V-ZOOM						Reserved (00)	

### 7-2 V-ZOOM

When enabled by setting bit 3 of the Playback Control 1 Register (MR1E) to 1, these six bits are used to specify the zoom factor by which the playback image is magnified.

Zoom factor = 100h / (value of bits 7 to 2 of this register)

Examples of programmed values:

Bits 7 6 5 4 3 2	Resulting Zoom Factor
1 0 0 0 0 0	Magnify by 2
0 1 0 0 0 0	Magnify by 4
0 0 1 0 0 0	Magnify by 8

### 1-0 Reserved

These bits always return the value of 0 when read.

## **MR34 Playback Line Display Width**

Read / Write at I/O Address 3D3h

7	6	5	4	3	2	1	0
Playback Line Display Width [07:00]							

### **7-0 Playback Line Display Width [07:00]**

If bit 6 of the Playback Control 3 Register (MR0C) is set to 1, then this register specifies the number of quadwords out of a horizontal line's worth of video data that is actually played back, starting at the left-most edge of the video playback window.

This value is calculated as follows:  $(\text{width of line in pixels}) / (\text{number of pixels per quadwords}) - 1$ .

## MR3C Color Key Control

Read / Write at I/O Address 3D3h

Default = 00h

7	6	5	4	3	2	1	0
LSB Disable	16-Bit Overlay	Blank Graphics	Reserved (00) (Writable)		XY Rectangle	Color Key	Video Playback

### 7 LSB (Bit 0) disable

- 0 Normal "Blue bit 0"
- 1 Red, green, and blue bit 0 is forced to 0 at MMUX output (for masking Playback of key when using 16/24 bit overlay key).

### 6 16-bit Overlay Key

- 0 Normal color key
- 1 Color key "Green\_7" is routed to "Blue\_0"

### 5 Blank Graphics

- 0 Normal
- 1 Graphics data blanked to "zero"

### 4-3 Reserved (Writable)

These bits default to 0 after reset. Though these bits may be set to 1, this is not recommended as they are reserved for future use.

### 2 XY Rectangle Enable

- 0 XY Rectangular Region off
- 1 XY Rectangular Region enabled

### 1 Color Key Enable

- 0 Color Key off
- 1 Color Key enabled

### 0 Video Playback Enable

- 0 Graphics only, no video playback
- 1 Video Playback Window enabled



### MR3D-3F Color Keys

Read / Write at I/O Address 3D3h

MR3D: Red, MR3E: Green, MR3F: Blue

7	6	5	4	3	2	1	0
Color Keys							

#### 7-0 Red/Green/Blue Color Keys

- 0 Use the corresponding color key
- 1 Do not use color key

### MR40-42 Color Key Masks

Read / Write at I/O Address 3D3h

MR40: Red Mask, MR41: Green Mask, MR42: Blue Mask

7	6	5	4	3	2	1	0
Color Key Masks							

#### 7-0 Red/Green/Blue Color Key Masks

- 0 Use the corresponding color key
- 1 Do not use color key

The table below describes the bits and values for the color key registers in different graphics modes.

**Table 16-1: Key Value and Key Mask**

Playback Mode	R_Key	G_Key	B_Key	Masks		
				R_Key	G_Key	B_Key
4-Bit Indexed			Blue[3:0]	FF	FF	F0
8-Bit Indexed			Blue[7:0]	FF	FF	00
15-Bit RGB		Green[6:0]	Blue[7:0]	FF	80	00
16-Bit RGB		Green[7:0]	Blue[7:0]	FF	00	00
24-Bit RGB	Red[7:0]	Green[7:0]	Blue[7:0]	00	00	00
16-Bit Key		Green[7]		FF	7F	FF
24-Bit Key			Blue[7:0]	FF	FF	FE

**Note:** Color Key bit assignments:

In 15 Bit RGB (5:5:5) Mode:

- RED[7:3] = G\_Key[6:2]
- GREEN[7:3] = G\_Key[1:0], B\_Key[7:5]
- BLUE[7:3] = B\_Key[4:0]

In 16 Bit RGB (5:6:5) Mode:

- RED[7:3] = G\_Key[7:3]
- GREEN[7:2] = G\_Key[2:0], B\_Key[7:5]
- BLUE[7:3] = B\_Key[4:0]

## MR43 Line Count

Read Only at I/O Address 3D3h

7	6	5	4	3	2	1	0
Line Counter for Graphics Playback							

### 7-0 Line Counter for Graphics Playback (Lower Bits)

## MR44 Line Count

Read Only at I/O Address 3D3h

7	6	5	4	3	2	1	0
Reserved				Line Counter for Graphics Playback [11:08]			

### 7-4 Reserved

### 3-0 Line Counter for Graphics Playback (Upper Bits) [11:08]

This register enables the read back of the playback vertical “line counter”.

# **CHAPTER 17**

## **BITBLT REGISTERS**

**This page intentionally left blank.**

## 17.0 BITBLT REGISTERS

The BitBLT registers are 32-bit memory-mapped registers exist in the upper memory space that the 65555 occupies on the host bus. Refer to Chapter 3 for more information. These registers exist at an offset of 0x400000 (4MB) from the base address of the memory space.

<b>Name</b>	<b>Function</b>	<b>Access</b>	<b>Offset</b>
BR00	Source and Destination Offset Register	Read/Write	0x400000
BR01	Pat/Src Expansion Background Color Reg.	Read/Write	0x400004
BR02	Pat/Src Expansion Foreground Color Reg.	Read/Write	0x400008
BR03	Monochrome Source Control Register	Read/Write	0x40000C
BR04	BitBLT Control Register	Read/Write	0x400010
BR05	Pattern Address Register	Read/Write	0x400014
BR06	Source Address Register	Read/Write	0x400018
BR07	Destination Address Register	Read/Write	0x40001C
BR08	Destination Width & Height Register	Read/Write	0x400020
BR09	Source Expansion Background Color Reg.	Read/Write	0x400024
BR0A	Source Expansion Foreground Color Reg.	Read/Write	0x400028

## BR00 Source and Destination Offset Register

Read/Write at memory space offset 0x400000h

Word or Doubleword accessible

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved (000)								Destination Offset (x:xxxx:xxxx:xxxx)							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved (000)								Source Offset (x:xxxx:xxxx:xxxx)							

### 31-29 Reserved

These bits always return 0 when read.

### 28-16 Destination Offset

When the output destination data is to be written to the frame buffer, these 13 bits specify the offset by which the source address specified in the Destination Address Register (BR07) should be incremented to point to the first byte in the next scan line's worth of output destination data as it is written to the frame buffer.

When the output destination data is to be provided to the host CPU, these 13 bits specify the number of bytes to be counted from the first byte in one scan line's worth of output destination data to the first byte in the next.

If the output destination data is contiguous (i.e., the output destination data is to be created as a single unbroken block of data), then the value of this offset should be set equal to the number of bytes in each scan line's worth of output destination data. If the output destination data is to be discontinuous (i.e., there are to be bytes of something other than output destination data separating the output destination data into sub-blocks of bytes that each represent a scan line's worth of output destination data), then the value of this offset should be set equal to the number of bytes in the interval from the first byte of output destination data in one of these sub-blocks to the first byte of output destination data in the next sub block.

### 15-13 Reserved

These bits always return 0 when read.

### 12-0 Source Offset

These 13 bits are used only when color source data is being used as an input in a BitBLT operation. If monochrome source data or no source data is to be used, then the BitBLT engine will ignore the value carried by these bits.

When color source data is read from the frame buffer, these 13 bits specify the offset by which the source address specified in the Source Address Register (BR06) should be incremented to point to the first byte in the next scan line's worth of color source data to be read from the frame buffer.

When the host CPU provides the color source data, these 13 bits specify the number of bytes to be counted from the first byte in one scan line's worth of color source data to the first byte in the next.

If the color source data is contiguous (i.e., the source data exists as a single unbroken block of data), then the value of this offset should be set equal to the number of bytes in each scan line's worth of source data. If the color source data is discontinuous (i.e., there are bytes of something other than color source data separating it into sub-blocks of bytes that each represent a scan line's worth of source data), then the value of this offset should be set equal to the number of bytes in the interval from the first byte of source data in one of these sub-blocks to the first byte of source data in the next sub block.

## **BR01 Pattern/Source Expansion Background Color Reg.**

Read/Write at memory space offset 0x400004h

Word or Doubleword accessible

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved (0000:0000)								Pat/Src Expansion Background Color Bits 23-16 (xxxx:xxxx)							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Pattern/Source Expansion Background Color Bits 15-0 (xxxx:xxxx:xxxx:xxxx)															

### **31-24 Reserved**

These bits always return 0 when read.

### **23-0 Pattern/Source Expansion Background Color Bits 23-0**

These bits provide the one, two, or three bytes worth of color data that select the background color to be used in the color expansion of either monochrome pattern data only, or both monochrome pattern data and monochrome source data, depending upon the setting of bit 27 of the Monochrome Source Control Register (BR03). When bit 27 of the Monochrome Source Control Register is set so that this register is involved in the color expansion of monochrome pattern data only, then the Source Expansion Background Color Register (BR09) is used to perform the identical function for monochrome source data.

Whether one, two, or three bytes worth of color data is needed depends upon the color depth to which the BitBLT engine has been set through the BitBLT Configuration Register (XR20). For a color depth of 24bpp, 16bpp, and 8bpp, bits 23-0, 15-0, and 7-0, respectively, of this register are used.

## BR02 Pattern/Source Expansion Foreground Color Reg.

Read/Write at memory space offset 0x400008h

Word or Doubleword accessible

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved (0000:0000)								Pat/Src Expansion Foreground Color Bits 23-16 (xxxx:xxxx)							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Pattern/Source Expansion Foreground Color Bits 15-0 (xxxx:xxxx:xxxx:xxxx)															

### 31-24 Reserved

These bits always return 0 when read.

### 23-0 Pattern/Source Expansion Foreground Color Bits 23-0

These bits provide the one, two, or three bytes worth of color data that select the foreground color to be used in the color expansion of either monochrome pattern data, only, or both monochrome pattern data and monochrome source data, depending upon the setting of bit 27 of the Monochrome Source Control Register (BR03). When bit 27 of the Monochrome Source Control Register is set so that this register is involved in the color expansion of monochrome pattern data only, then the Source Expansion Foreground Color Register (BR0A) is used to perform the identical function for monochrome source data.

Whether one, two, or three bytes worth of color data is needed depends upon the color depth to which the BitBLT engine has been set through the BitBLT Configuration Register (XR20). For a color depth of 24bpp, 16bpp, and 8bpp, bits 23-0, 15-0, and 7-0, respectively, of this register are used.



## BR03 Monochrome Source Control Register

Read/Write at memory space offset 0x40000Ch

Word or Doubleword accessible

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved (0000)				Src Exp (x)	Mono Src Align (xxx)			Reserved (00)		Monochrome Source Data Initial Discard (xx:xxxx)					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved (00)		Monochrome Source Data Right Clipping (xx:xxxx)						Reserved (00)		Monochrome Source Data Left Clipping (xx:xxxx)					

### 31-28 Reserved

These bits always return 0 when read.

### 27 Monochrome Source Expansion Color Register Select

- 0 This causes the background and foreground colors used for the color expansion of monochrome source data to be selected by the Pattern/Source Expansion Background Color Register (BR01) and the Pattern/Source Expansion Foreground Color Register (BR02).
- 1 This causes the background and foreground colors used for the color expansion of monochrome source data to be selected by the Source Expansion Background Color Register (BR09) and the Source Expansion Foreground Color Register (BR0A).

### 26-24 Monochrome Source Scan Line Data Alignment

These three bits are used only when the source data is monochrome. They are used to configure the BitBLT engine for the alignment of each scan line's worth of source data as it will be aligned when the BitBLT engine receives it.

Refer to the appendix describing the BitBLT engine for further details concerning the requirements for how the source data must be organized.

Bit 26 25 24	Specified Monochrome Source Data Alignment
0 0 0	Reserved
0 0 1	Bit-Aligned
0 1 0	Byte-Aligned
0 1 1	Word-Aligned
1 0 0	Doubleword-Aligned
1 0 1	Quadword-Aligned
1 1 0	Reserved
1 1 1	Reserved

### **BR03 Monochrome Source Control Register (continued)**

#### **23-22 Reserved**

These bits always return 0 when read.

#### **21-16 Monochrome Source Data Initial Discard**

These six bits are used only when the source data is monochrome. They are used to indicate how many bits (up to 63 bits) of monochrome source data should be discarded in the first quadword of source data to reach the first bit of valid or desired monochrome source data. These bits are normally used to clip one or more of the first scan lines of monochrome source data, (i.e., clipping monochrome source data from the top.)

#### **15-14 Reserved**

These bits always return 0 when read.

#### **13-8 Monochrome Source Data Right Clipping**

These six bits are used only when the source data is monochrome. They are used to indicate how many bits (up to 63 bits) of monochrome source data should be discarded from the end of each scan line's worth of valid or desired monochrome source data. These bits are normally used to clip monochrome source data from the right.

#### **7-6 Reserved**

These bits always return 0 when read.

#### **5-0 Source Data Left Clipping**

These six bits are used only when the source data is monochrome. They are used to indicate how many bits (up to 63 bits) of monochrome source data should be discarded from the beginning of each scan line's worth of valid or desired monochrome source data. These bits are normally used to clip the monochrome source data from the left.

## BR04 BitBLT Control Register

Read/Write at memory space offset 0x400010h

Word or Doubleword accessible

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		
BB Stat (0)	Reserved (000:00)					Color Depth (00)	Dep Ctrl (0)	Pattern Vertical Alignment (000)	Sol Pat (0)	Pat Dep (0)	Pat Mask (0)	Col Mask (0)					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Color-Based Write Mask (0)	Src Mask (0)	Src Dep (0)	Dest Sel (0)	Src Sel (0)	Starting Point Select (00)	Bit-Wise Operation Select (00h)											

### 31 BitBLT Engine Status

**Note:** This bit is read-only -- writes to this bit are ignored.

- 0 Indicates that the BitBLT is idle.
- 1 Indicates that the BitBLT is busy.

### 30-26 Reserved

These bits always return 0 when read.

### 25-24 BitBLT Engine Color Depth

When bit 23 of this register is set to 1, these two bits configure the BitBLT engine for one of three possible color depths. If bit 23 of this register is set to 0, then this function is performed by bits 5 and 4 of the BitBLT Configuration Register (XR20). It is strongly recommended that, when possible, the color depth of the BitBLT engine be set to match the color depth to which the graphics system has been set.

Bit 25 24	BitBLT Engine Color Depth Selected
0 0	8 bits per pixel (1 byte per pixel) This is the default after reset.
0 1	16 bits per pixel (2 bytes per pixel)
1 0	24 bits per pixel (3 bytes per pixel)
1 1	Reserved

The choice of color depth configures the BitBLT engine to work with one, two or three bytes per pixel. This directly affects the number of bytes of graphics data that the BitBLT engine will read and write for a given number of pixels. In the case of monochrome source or pattern data, this setting directly affects the color depth into which such monochrome data will be converted during the color expansion process.

If the graphics system has been set to a color depth that is not supported by the BitBLT engine, then it is strongly recommended that the BitBLT engine not be used. See the chapter on the BitBLT engine for more information.

**BR04 BitBLT Control Register (continued)****23 BitBLT Engine Color Depth Control Select**

- 0 Bits 5 and 4 of the BitBLT Configuration Register (XR20) are used to configure the BitBLT engine for one of three color depths. This is the default after reset.
- 1 Bits 25 and 24 of this register are used to configure the BitBLT engine for one of three color depths.

**22-20 Pattern Vertical Alignment**

Specifies which scan line's worth (which of the 8 horizontal rows) of the 8x8 pattern will appear on the first scan line's worth of the data written to the destination. Depending upon the location of the destination, the upper left corner of the upper left tile of the pattern is usually aligned with the upper left corner of the block of data written to the destination. The BitBLT engine determines the horizontal alignment relative to the destination using the lower order bits of the destination address. However, the vertical alignment relative to the destination must be specified using these bits.

**19 Solid Pattern Select**

This bit applies only when the pattern data is monochrome. Bit 18 of this register specifies whether the pattern data is color or monochrome.

- 0 This causes normal operation with regard to the use of monochrome pattern data. If monochrome pattern data is to be used as an input, then the BitBLT engine proceeds with the process of reading and using monochrome pattern, as usual.
- 1 The BitBLT engine is forced to forgo the process of reading the pattern data. A presumption is made that all of the bits of the pattern data are set to 0, and the pattern operand for all bit-wise operations is forced to the background color specified in the Pattern/Source Expansion Background Color Register (BR01).

**18 Pattern Color Depth**

- 0 Specifies that the pattern data is in color, and therefore, has a color depth of 8, 16, or 24 bits per pixel.
- 1 Specifies that the pattern data is monochrome, and therefore, has a color depth of only 1 bit per pixel.

**17 Monochrome Pattern Write-Masking**

This bit applies only when the pattern data is monochrome. This bit enables a form of per-pixel write-masking in which monochrome pattern data is used a pixel mask that controls which pixels at the destination will be written to by the BitBLT engine. Bit 18 of this register specifies whether the pattern data is color or monochrome.

- 0 This causes normal operation of the BitBLT engine with regard to the use of monochrome pattern data.
- 1 Wherever a bit in monochrome pattern data carries the value of 0, the byte(s) of the corresponding pixel at the destination are not written, thereby preserving any data already carried by those bytes.

## BR04 BitBLT Control Register (continued)

### 16-14 Color-Comparison Write-Masking

These bits select and enable various forms of per-pixel write-masking based on the results of different comparisons between colors.

Note: This function is supported only in 8-bit or 16-bit per pixel modes. This function cannot be used in 24-bit per pixel mode, and so for 24-bit per pixel mode, these bits should be set to 0.

Bit 16 15 14	Form of Per-Pixel Write-Masking Selected
x x 0	No form of per-pixel write-masking based on the results of any comparisons between colors takes place.
0 0 1	The background color specified for use in the color expansion of monochrome source data is compared to the color resulting from the bit-wise operation for the current pixel. If these two colors are NOT the same, then the byte(s) at the destination corresponding to the current pixel are written with the result of the bit-wise operation.
0 1 1	The background color specified for use in the color expansion of monochrome source data is compared to the color specified by the byte(s) at the destination corresponding to the current pixel. If these two colors are NOT the same, then the byte(s) at the destination corresponding to the current pixel are written with the result of the bit-wise operation.
1 0 1	The background color specified for use in the color expansion of monochrome source data is compared to the color resulting from the bit-wise operation for the current pixel. If these two colors are the same, then the byte(s) at the destination corresponding to the current pixel are written with the result of the bit-wise operation.
1 1 1	The background color specified for use in the color expansion of monochrome source data is compared to the color specified by the byte(s) at the destination corresponding to the current pixel. If these two colors are the same, then the byte(s) at the destination corresponding to the current pixel are written with the result of the bit-wise operation.

### 13 Monochrome Source Write-Masking

**Note:** This function is supported only in 8-bit or 16-bit per pixel modes. This function cannot be used in 24-bit per pixel mode, and so for 24-bit per pixel mode, this bit should be set to 0.

This bit applies only when the source data is monochrome. This bit enables a form of per-pixel write-masking in which monochrome source data is used as a pixel mask that controls which pixels at the destination will be written to by the BitBLT engine. Bit 12 of this register specifies whether the source data is color or monochrome.

- 0 This causes normal operation of the BitBLT engine with regard to the use of monochrome source data.
- 1 Wherever a bit in monochrome source data carries the value of 0, the byte(s) of the corresponding pixel at the destination are simply not written, thereby preserving any data already carried by those bytes.

### 12 Source Color Depth

**Note:** This bit should only be set to 1 when bits 7-0 of this register are used to select an operation which requires a source operand.

- 0 Specifies that the source data is in color, and therefore, has a color depth of 8, 16, or 24 bits per pixel.
- 1 Specifies that the source data is monochrome, and therefore, has a color depth of 1 bit per pixel. This setting should be used only if bit 8 of this register is set to 0.

## **BR04 BitBLT Control Register (continued)**

### **11 Destination Select**

- 0 Configures the BitBLT engine to write the output destination data to the frame buffer at the location specified in the Destination Address Register (BR07).
- 1 Configures the BitBLT engine to provide the output destination data to the host CPU. The host CPU receives the output destination data by performing a series of memory read operations from the BitBLT data port.

### **10 Source Select**

- 0 Configures the BitBLT engine to read the source data from the frame buffer at the location specified in the Source Address Register (BR06).
- 1 Configures the BitBLT engine to accept the source data from the host CPU. The host CPU provides the source data by performing a series of memory write operations to the BitBLT data port. The actual addresses within the BitBLT data port to which data is written do not matter -- the BitBLT engine uses source data provided to the BitBLT data port in the ORDER in which it is written to the BitBLT data port by the host CPU.

### **9-8 Starting Point Select**

These two bits are used to select which of the four corners to use as the starting point in reading and writing graphics data in a BitBLT operation. Normally, the upper left corner is used. However, situations involving an overlap of source and destination locations (this usually occurs when the source and destination locations are both on-screen) often require the use of a different corner as a starting point. It should be remembered that the addresses specified for each piece of graphics data used in a BitBLT operation must point to the byte(s) corresponding to whichever pixel is at the selected starting point. If the starting point is changed, then these addresses must also be changed. See the chapter on the BitBLT engine for more information.

<b>Bit 9 8</b>	<b>Corner Selected as the Starting Point</b>
0 0	Upper Left Corner -- This is the default after reset.
0 1	Upper Right Corner
1 0	Lower Left Corner
1 1	Lower Right Corner

### **7-0 Bit-Wise Operation Select**

These 8 bits are meant to be programmed with an 8-bit code that selects which one of 256 possible bit-wise operations is to be performed by the BitBLT engine during a BitBLT operation. These 256 possible bit-wise operations and their corresponding 8-bit codes are designed to be compatible with the manner in which raster operations are specified in the standard BitBLT parameter block normally used in the Microsoft Windows environment, without translation. See the chapter on the BitBLT engine for more information.

## BR05 Pattern Address Register

Read/Write at memory space offset 0x400014h

Word or Doubleword accessible

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved (0000:0000:0)									Pattern Address Bits 22-16 (xxx:xxxx)						
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Pattern Address Bits 15-3 (xxxx:xxxx:xxxx:x)													Reserved (000)		

### 31-23 Reserved

These bits always return 0 when read.

### 22-3 Pattern Address

These 20 bits specify the starting address of the pattern data within the frame buffer as an offset from the beginning of the frame buffer to where the byte(s) for the first pixel of pattern data to be read are located.

The pattern data must be located on a pattern-size boundary within the frame buffer. The pattern is always an 8x8 array of pixels, therefore, its size in bytes is dependent upon its pixel depth. The pixel depth may be 1 bit per pixel if the pattern is monochrome or it may be 8, 16, or 24 bits per pixel if the pattern is in color. The pixel depth of a color pattern must match the pixel depth to which the BitBLT engine has been set. Monochrome patterns require 8 bytes, therefore, the pattern data must start on a quadword boundary. Color patterns of 8, 16, and 24 bits per pixel color depth must start on 64-byte, 128-byte, and 256-byte boundaries, respectively.

**Note:** In the case of 24 bits per pixel, each scan line's worth (each row of 8 pixels) of pattern data takes up 32 consecutive bytes, not 24. The pattern data is formatted so that there is a contiguous block of 8 sets of 3 bytes, each set corresponding to one of the 8 pixels, followed by a contiguous block of the 8 extra bytes. When the BitBLT engine reads 24 bit-per-pixel pattern data, it will read only the first 24 bytes of each scan line's worth of data, picking up the 8 sets of 3 bytes for 8 pixels, and entirely ignoring the remaining 8 bytes.

### 2-0 Reserved

These bits always return 0 when read.

## BR06 Source Address Register

Read/Write at memory space offset 0x400018h

Word or Doubleword accessible

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved (0000:0000:0)									Source Address Bits 22-16 (xxx:xxxx)						
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Source Address Bits 15-0 (xxxx:xxxx:xxxx:xxxx)															

**Important:** This register should never be read while the BitBLT engine is busy.

### 31-23 Reserved

These bits always return 0 when read.

### 22-0 Source Address

When the source data is located within the frame buffer, these 23 bits are used to specify the starting address of the source data within the frame buffer as an offset from the beginning of the frame buffer to where the byte(s) for the first pixel of source data to be read are located.

When the source data is provided by the host CPU through the BitBLT data port, and that source data is in color, only bits 2-0 are used and the upper 20 bits are ignored. These lower 3 bits are used to indicate the position of the first valid byte within the first quadword of the source data.

When the source data is provided by the host CPU through the BitBLT data port, and that source data is monochrome, the BitBLT engine ignores this register.



## **BR07 Destination Address Register**

Read/Write at memory space offset 0x40001Ch

Word or Doubleword accessible

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved (0000:0000:0)									Destination Address Bits 22-16 (xxx:xxxx)						
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Destination Address Bits 15-0 (xxxx:xxxx:xxxx:xxxx)															

**Important:** This register should never be read while the BitBLT engine is busy.

### **31-23 Reserved**

These bits always return 0 when read.

### **22-0 Destination Address**

When the output destination data is to be written to the frame buffer, these 23 bits are used to specify the starting address of the destination location within the frame buffer as an offset from the beginning of the frame buffer to where the byte(s) for the first pixel of output destination data are to be written.

When the output destination data is provided to the host CPU through the BitBLT data port, only bits 2-0 are used, and the upper 20 bits are ignored. These lower 3 bits are used to indicate the position of the first valid byte within the first quadword of the output destination data.

## BR08 Destination Width & Height Register

Read/Write at memory space offset 0x400020h

Word or Doubleword accessible

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved (000)				Destination Scan Line Height (0:0000:0000:0000)											

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved (000)				Destination Byte Width (0:0000:0000:0000)											

**Important:** This register should never be read while the BitBLT engine is busy.

### 31-29 Reserved

These bits always return 0 when read.

### 28-16 Destination Scan Line Height

These 13 bits specify the height of the destination data in terms of the number of scan lines that are covered by the destination data.

### 15-13 Reserved

These bits always return 0 when read.

### 12-0 Destination Byte Width

These 13 bits specify the width of the destination data in terms of the number of bytes per scan line's worth of destination data. The number of pixels per scan line into which this value translates depends upon the color depth to which the BitBLT engine has been set.

## **BR09 Source Expansion Background Color Register**

Read/Write at memory space offset 0x400024h

Word or Doubleword accessible

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved (0000:0000)								Source Expansion Background Color Bits 23-16 (xxxx:xxxx)							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Source Expansion Background Color Bits 15-0 (xxxx:xxxx:xxxx:xxxx)															

### **31-24 Reserved**

These bits always return 0 when read.

### **23-0 Source Expansion Background Color Bits 23-0**

These bits may or may not be used to provide the one, two, or three bytes worth of color data that select the background color to be used in the color expansion of monochrome source data, depending upon the setting of bit 27 of the Monochrome Source Control Register (BR03). When bit 27 of the Monochrome Source Control Register is set so that this register is involved in the color expansion of monochrome source data, then the Pattern/Source Expansion Background Color Register (BR01) is used to perform the identical function for monochrome pattern data, only.

Whether one, two, or three bytes worth of color data is needed depends upon the color depth to which the BitBLT engine has been set through the BitBLT Configuration Register (XR20). For a color depth of 24bpp, 16bpp, and 8bpp, bits 23-0, 15-0, and 7-0, respectively, of this register are used.

## BR0A Source Expansion Foreground Color Register

Read/Write at memory space offset 0x400028h

Word or Doubleword accessible

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved (0000:0000)								Source Expansion Foreground Color Bits 23-16 (xxxx:xxxx)							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Source Expansion Foreground Color Bits 15-0 (xxxx:xxxx:xxxx:xxxx)															

### 31-24 Reserved

These bits always return 0 when read.

### 23-0 Source Expansion Foreground Color Bits 23-0

These bits may or may not be used to provide the one, two, or three bytes worth of color data that select the foreground color to be used in the color expansion of monochrome source data, depending upon the setting of bit 27 of the Monochrome Source Control Register (BR03). When bit 27 of the Monochrome Source Control Register is set so that this register is involved in the color expansion of monochrome source data, then the Pattern/Source Expansion Foreground Color Register (BR02) is used to perform the identical function for monochrome pattern data only.

Whether one, two or three bytes worth of color data is needed depends upon the color depth to which the BitBLT engine has been set through the BitBLT Configuration Register (XR20). For a color depth of 24bpp, 16bpp, and 8bpp, bits 23-0, 15-0, and 7-0, respectively, of this register are used.

# *APPENDIX A*

# *VGA STANDARD MODES*

**This page intentionally left blank.**



This page intentionally left blank



# ***APPENDIX B***

# ***CLOCK GENERATION***

**This page intentionally left blank**

## **B. Clock Generation**

### **B.1 Clock Synthesizer**

The 65555 contains two complete phase-locked loops (PLLs) to synthesize the internal Dot Clock (DCLK) and Memory Clock (MCLK) from an externally supplied reference frequency. Each of the two clock synthesizer phase lock loops may be programmed to output frequencies ranging from 1MHz to the maximum specified operating frequency for that clock in increments not exceeding 0.5%. An external crystal-controlled oscillator (TTL) generates the reference frequency of 14.31818MHz that is driven into the 65555 on pin C3. The 65555 can not generate the 14.31818MHz reference frequency using only an external crystal.

### **B.2 Dot Clock (DCLK)**

Unlike the 65545 and 65548, the 65555 does not require an internal DCLK to be two or three times the theoretical Dot Clock required by the video display resolution and refresh rate. The 65555 has

sufficient internal video data path width to accommodate multiple-byte-per-pixel modes without using extra Dot Clock cycles.

The horizontal and vertical sync frequencies for the CRT display and/or panel are derived by dividing down the DCLK.

DCLK has three sets of registers, CLK0, CLK1, and CLK2 which the MSR (Miscellaneous Output Register) selects. Each CLK has its own registers. The reset values of CLK0 and CLK1 registers pertain to the standard VGA frequencies: 25.175MHz and 28.322MHz. The CLK2 registers are reset to same values as CLK0.

### **B.3 Memory Clock (MCLK)**

MCLK can be adjusted as needed in very small increments to accommodate the exact DRAM speed being used for the display memory. However, for VGA compatibility, the MCLK is set 25.175MHz on reset.

**B.4 PLL Parameters**

Each phase-locked loop consists of the elements shown in the figure below. The reference input frequency (normally 14.31818MHz) can be optionally divided by 4 (which is called Reference Divisor Select) before being further divided by N, a 10-bit programmable value (7-bit for MCLK). The output of the VCO is divided by 4 (or 16 via VCO Loop Divider: VLD) and then further divided by M, another 10-bit programmable value (7-bit for MCLK). The phase detector compares the N and M results and adjusts the VCO frequency as needed to achieve frequency equality.

The settings for normal operation appear in the table below:

VCO Loop Divide (VLD)	=	4
Reference Divisor Select (RDS)	=	1
NTSC Divisor Select (NTSCDS)	=	1

When the loop has stabilized, the VCO frequency ( $F_{VCO}$ ) is related to the reference as follows:

If RDS=1:  $F_{VCO} / 4M = F_{REF} / N$

If RDS=4:  $F_{VCO} / 4M = F_{REF} / 4N$

For RDS =1, the  $F_{VCO}$  can be written as:

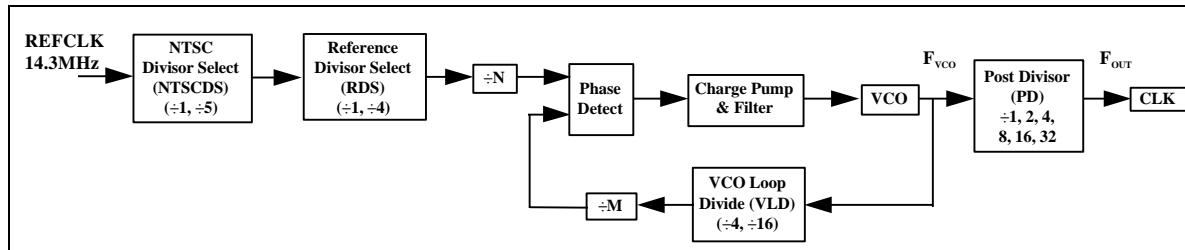
$$F_{VCO} = (F_{REF} * 4M / N)$$

The VCO output can be further divided by 1, 2, 4, 8, 16, or 32 (which is called Post Divisor: PD) to produce the final DCLK or MCLK used for video or memory timing.

Therefore the output frequency is:

$$F_{OUT} = (F_{VCO}) / PD$$

By “fine tuning” the M/N ratio in each PLL, extremely small adjustments in the exact DCLK and MCLK frequencies can be achieved. The VCO itself is designed to operate at frequencies ranging from 48MHz to 220MHz at 3.3V.



M counter = Program value M' +2

$F_{VCO}$ : VCO frequency (before post divisor)

N counter = Program value N' +2

$F_{OUT}$ : Output frequency: (desired frequency)

**Figure B-1: PLL Elements**

### B.5 Programming the Clock Synthesizer

Below are the register tables for CLK0, CLK1, CLK2, and MCLK. Please see the block diagram for M, N, and Post Divide (PD).

	CLK0	CLK1
<b>M</b>	XRC2[1:0]+XRC0	XRC6[1:0]+XRC4
<b>N</b>	XRC2[5:4]+XRC1	XRC6[5:4]+XRC5
<b>PD</b>	XRC3[6:4]	XRC7[6:4]

	CLK2	MCLK
<b>M</b>	XRCA[1:0]+XRC8	XRCC[6:0]
<b>N</b>	XRCA[5:4]+XRC9	XRCD[6:0]
<b>PD</b>	XRCB[6:4]	XECE[6:4]

### B.6 DCLK Programming

The following sequence should be used when programming a new frequency for any of the DCLK VCO's:

- 1) Program M
- 2) Program N
- 3) Program PD  
This will effectively change DCLK into the new frequency

### B.7 MCLK Programming

The following sequence should be used when programming a new frequency for MCLK:

- 1) Reset XRCE[7] to 0 to select MCLK equal to 25.175MHz.
- 2) Program M
- 3) Program N
- 4) Program PD with XRCE[7]=1 to select the programmed frequency.

### B.8 Programming Constraints

The programmer must be aware of the following five programming constraints:

$$1\text{MHz} \leq F_{\text{REF}} \leq 60\text{MHz}$$

$$150\text{KHz} \leq F_{\text{REF}} / (\text{RDS} * \text{N}) \leq 2\text{MHz}$$

$$48\text{MHz} < F_{\text{VCO}} \leq 220\text{MHz}$$

$$3 \leq \text{M} \leq 127 \text{ (1023 for DCLK)}$$

$$3 \leq \text{N} \leq 127 \text{ (1023 for DCLK)}$$

The constraints have to do with trade-offs between optimum speed with lowest noise, VCO stability, and factors affecting the loop equation.

The value of  $F_{\text{VCO}}$  must remain between 48MHz and 220MHz inclusive. Therefore, for output frequencies below 48MHz,  $F_{\text{VCO}}$  must be brought into range by using the post-VCO Divisor.

To avoid crosstalk between the VCOs, the VCO frequencies should not be within 0.5% of each other nor should their harmonics be within 0.5% of the other's fundamental frequency.

The 6555 clock synthesizers will seek the new frequency as soon as it is loaded following a write to the control register. Any change in the post-divisor will take affect immediately. The output may glitch during this transition of post divide values. Therefore, the programmer may wish to hold the post-divisor value constant across a range of frequencies. There is also the consideration of changing from a low frequency VCO value with a post-divide  $\div 1$  (e.g., 50MHz) to a high frequency  $\div 4$  (e.g., 220MHz). Although the beginning and ending frequencies are close together, the intermediate frequencies may cause the 6555 to fail in some environments. In this example, there will be a short-lived time during which the output frequency will be approximately 12.5MHz. The 6555 provides the mux for MCLK so it can select the fixed frequency (25.175MHz) before programming a new frequency. Because of this, the bus interface may not function correctly if the MCLK frequency falls below a certain value. Register and memory accesses synchronized to MCLK may be too slow and violate the bus timing causing a watchdog timer error.

### B.9 Programming Example

The following is an example of the calculations which are performed.

*Derive the proper programming word for a 25.175MHz output frequency using a 14.31818MHz reference frequency.*

Since 25.175MHz < 48MHz, double it to 50.350MHz to get F<sub>VCO</sub> in its valid range. Set the post divide (PD) divide by 2.

Reference Divisor Selector (RDS) = 4

The result:

$$F_{VCO} = 50.350 = (14.31818 \times 4 \times M/4 \times N)$$

$$M/N = 3.51655$$

Several choices for M and N are available:

M	N	F <sub>VCO</sub>	Error
109	31	50.344	-0.00300
102	29	50.360	+0.00500

Choose (M, N) = (109,31) for best accuracy.

Reference Divisor Selector (RDS) = 1

The result:

$$F_{VCO} = 50.350 = (14.31818 \times 4 \times M/1 \times N)$$

M	N	F <sub>VCO</sub>	Error
80	91	50.349	-0.00050

$$F_{REF} / (RDS \times N) = 157.3KHz$$

$$M/N = 0.879127$$

Therefore M/N = 80/91 with RDS = 1 is even better than with RDS = 4.

$$XRC0 = 80 - 2 = 78 (4Eh)$$

$$XRC1 = 91 - 2 = 89 (59h)$$

$$XRC2 = 00h$$

$$XRC3 = 0001 0001b = 11h$$

### B.10 Display Memory Bandwidth

The 65555's ability to support higher resolutions and/or higher color depths may be limited by the maximum DCLK frequency or the available memory bandwidth. The maximum pixel rate that a given MCLK frequency can support depends on the following:

- 1) Pixel depth (number of bytes per pixel): 1 byte for 8 bpp, 2 bytes for 16bpp, 3 bytes for 24bpp.
- 2) Number of additional bytes accessed for STN-DD frame buffering, usually one byte per pixel (independent of pixel depth in main display memory). The next section discusses this effect in more detail. It applies only to STN-DD panels, not to CRT or TFT displays.
- 3) Utilization efficiency. The percentage of peak memory bandwidth needed for RAS overhead (RAS-CAS cycles rather than CAS-only cycles), DRAM refresh, and CPU access. Peak memory bandwidth is the product of MCLK and the number of bytes accessed per MCLK (e.g., 160 MB/sec for 40MHz MCLK). The 65555 needs at least 20% of this peak bandwidth for RAS overhead (higher for STN-DD buffer accesses and CPU accesses due to shorter DRAM bursts). Allow at least an additional 10% bandwidth buffer for CPU accesses and DRAM refresh. This leaves 70% of MCLK cycles available for display refresh.
- 4) Multimedia frame capture. The example calculations do not include this factor. Except where otherwise noted, 65555 mode support estimates do not include provision for frame capture from the video input port.

As an example, suppose MCLK is 40MHz and the pixel depth is 16 bpp. Then the maximum supportable pixel rate for CRT and TFT displays is 40MHz x 70% x 4 ÷ 2 = 56MHz (4 bytes per MCLK, 2 bytes per pixel). Any video mode that uses a 56MHz or lower DCLK can be supported by the 40MHz MCLK. For an STN-DD panel, the maximum supportable pixel rate in 16 bpp modes is 40MHz x 70% x 4 ÷ 3 = 37MHz (4 bytes per MCLK, 3 bytes accessed per pixel). 16 bpp video modes using a 37MHz or lower DCLK can be supported by the 40MHz MCLK with an STN-DD panel.

### **B.11 STN-DD Panel Buffering**

STN-DD panels require the upper and lower halves of the panel to be refreshed simultaneously. In addition, Frame Rate Control (FRC) is needed to achieve more than 8 colors, since the panel itself supports only 3 bits per pixel (one bit each for red, green, and blue). The 65555 implements STN-DD support using either a full frame buffer or a half frame buffer (programmable option). The buffer can be allocated in display memory or in the separate "DRAM C" memory. The buffer holds three bits per pixel, packed in groups of 10 pixels per DWORD. Thus, the buffer requires 0.4 bytes per pixel in addition to the main display memory.

The half frame buffer operates as follows. As each pixel is read out of display memory, the appropriate 3-bit code for the panel is calculated and sent to the panel. In addition, the proper 3-bit code for the same pixel in the NEXT frame is also calculated, with allowance for frame rate control. The second 3-bit code is written into the half frame buffer. During this same pixel time, the previously stored 3-bit code is read out of the half frame buffer and sent to the other half of the panel.

The full frame buffer operates in a similar manner. As each two pixels are read out of display memory, the appropriate 3-bit codes for the panel are calculated and stored in the buffer. During the same two pixel times, previously stored 3-bit codes are read out of the buffer and sent to upper and lower halves of the panel.

There is no difference between a half frame buffer and a full frame buffer in the effect on display memory bandwidth. Both options require 0.4

bytes per pixel to be read and written during each pixel time. If the buffer is located within the main display memory, the total effect is 0.8 extra bytes of memory access per pixel (regardless of pixel depth). In 16 bpp modes, a total of 2.8 bytes of memory access must be performed per pixel – 2 bytes for the 16 original pixel bits, plus 0.8 byte for the buffer bits. The 65555 actually reads and writes one DWORD in the buffer for every 10 pixels, which is the same as 0.8 bytes per pixel. For mode support calculations, it is usually best to assume 1.0 byte per pixel instead of 0.8, since the RAS overhead for STN-DD buffer accesses is somewhat higher than for normal pixel accesses due to shorter DRAM bursts.

The half frame buffer has a timing characteristic for the panel that may be either a problem or an advantage, depending on the application. The panel is refreshed at twice the pixel rate imposed on the display memory. In simultaneous CRT and panel mode, the CRT requirement dictates the pixel rate, and the panel is refreshed at twice that rate. This may exceed panel timing limitations. However, in panel-only mode, the pixel rate from display memory can be reduced to half of what a CRT would need. This change imposes half the burden on display memory bandwidth and allows the available memory bandwidth to support more complex video modes.

The full frame buffer allows the panel refresh rate to be the same as the CRT in simultaneous display mode, but requires the buffer size to be twice as large (full frame instead of half frame, though only 0.4 bytes per pixel).

## B.12 Horizontal and Vertical Clocking

Clocking within a horizontal scan line is generally programmed in units of 8 DCLK cycles (8 pixels), often referred to as “character” clocks (for graphics modes as well as text modes). The “character” clocks are numbered from 0 to n-1, where “n” is the total number of character clocks per horizontal scan (including blanking and border intervals as well as the “addressable video” interval). Character clock #0 corresponds to the start of the “addressable video” interval, also known as the “Display Enable” interval. Starting at character clock #0, the following horizontal timing events occur:

- End of Display Enable
- Start of horizontal blanking (end of right border)
- Horizontal sync pulse start and end
- End of horizontal blanking
- Start of left border. (This border area is actually for the next physical scan line.)
- End of left border area and start of Display Enable (This corresponds to the “Horizontal Total” parameter.)

Similarly, vertical clocking is generally programmed in units of scan lines, numbered from 0 to m-1, where “m” is the total number of scan lines per complete frame and “0” corresponds to the first scan line containing addressable video information. Starting at scan line #0, the following vertical timing events occur:

- End of addressable video
- Start of vertical blanking (end of bottom border)
- Vertical sync pulse start and end
- End of vertical blanking (start of top border) (This border area is actually for the next physical frame.)
- End of top border area and start of addressable video. This corresponds to the “Vertical Total” parameter.

Vertical timing can also be “inter laced,” meaning that even numbered scan lines are displayed during one vertical sweep and odd numbered lines are displayed during the next vertical sweep. This allows more time (two vertical sweeps instead of one) to display a complete frame, which reduces video bandwidth requirements while preserving a reasonably flicker-free image. North American television standards use a 60Hz vertical sync frequency, interlaced for a 30Hz effective frame rate, with 525 scan lines total per frame (even lines plus odd, including blanking). The horizontal sync frequency is  $525 \times 30\text{Hz} = 15.75\text{KHz}$ .

To achieve interlacing, the sweep of odd-numbered lines is offset by half of a scan line relative to the sweep of even-numbered lines. The vertical sync pulse for alternate frames occurs in the middle of a scan line interval (during vertical blanking) instead of at the end. North American television standards sweep 262.5 scan lines on each vertical sweep (60Hz). Each scan line remains full length, but the vertical sync for alternating frames occurs at the middle of the scan line. For optimum centering of odd scan lines between adjacent even scan lines, the exact placement of the half-line vertical sync pulse is designed to be programmable. Computer CRT displays generally need about 25% of the horizontal total for horizontal border and blanking intervals, and at least 5% of the vertical total for vertical border and blanking. Flat panels typically can operate with smaller margins for these “non-addressable” intervals.



## ***APPENDIX C***

# ***MEMORY CONFIGURATIONS***

**This page intentionally left blank.**

## C MEMORY CONFIGURATIONS

The 65555 supports the use of 1, 2, 3 or 4MB of display memory in a wide variety of memory configurations. The 65555 can support 128K, 256K or 512K-deep EDO DRAMs.

The following pages show several of the many possible memory configurations supported by the 65555. Table C-1 provides a quick reference to the major features of the memory configurations shown on the following pages.

All of the of memory configurations show the 65555 using the memory components currently available. This includes EDO DRAM components

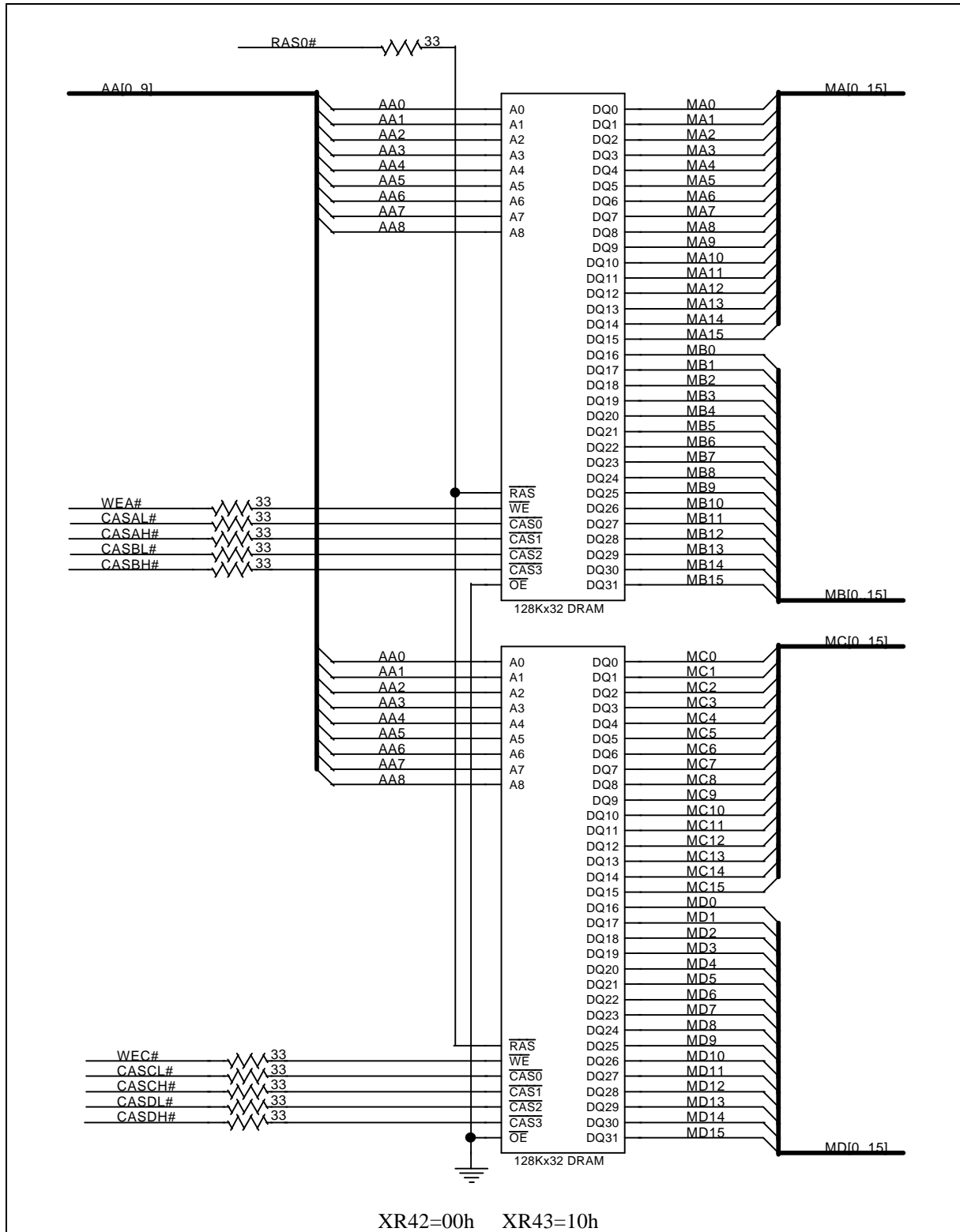
with internal organizations of 128Kx32, 256Kx16, and 512Kx32 with a CAS-per-byte interface.

It is possible to use memory components other than the particular ones those shown in this appendix. For instance, 2-megabit EDO DRAMs with either a 128Kx16 or 256Kx8 internal organization, or 8-megabit EDO DRAMs with either a 256Kx32 or 512Kx16 internal organization may also be used.

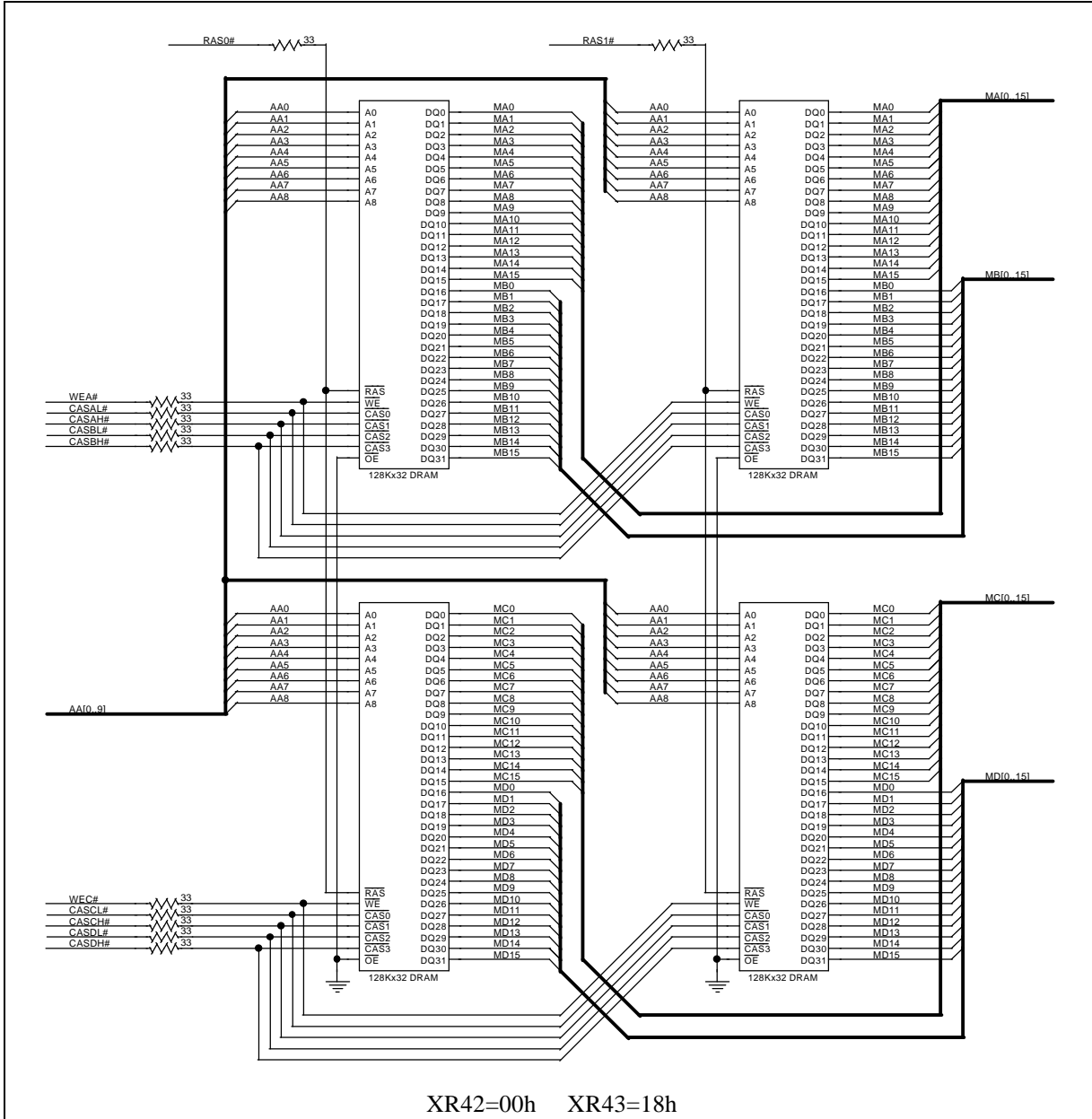
Each memory configuration is also accompanied by the register settings necessary to support that particular configuration.

**Table C-1— Memory Configurations**

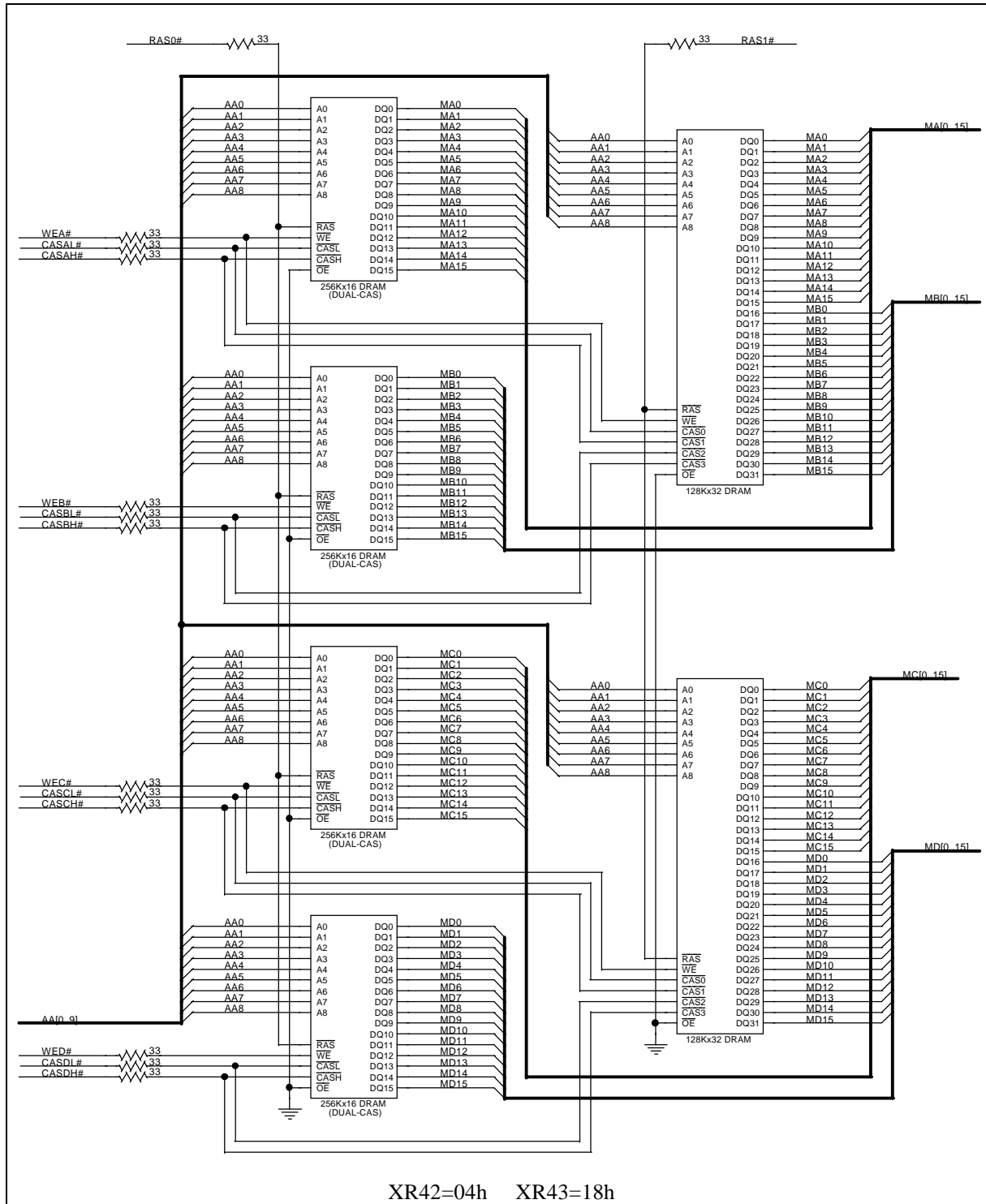
	128K-Deep EDO DRAM	Mixed 128K & 256K-Deep EDO DRAM	256K-Deep EDO DRAM	512K-Deep EDO DRAM
1MB 64-bit Wide Interface	Figure C-1			
2MB 64-bit Wide Interface	Figure C-2		Figure C-4	
3MB 64-bit Wide Interface		Figure C-3		
4MB 64-bit Wide Interface			Figure C-5	Figure C-6



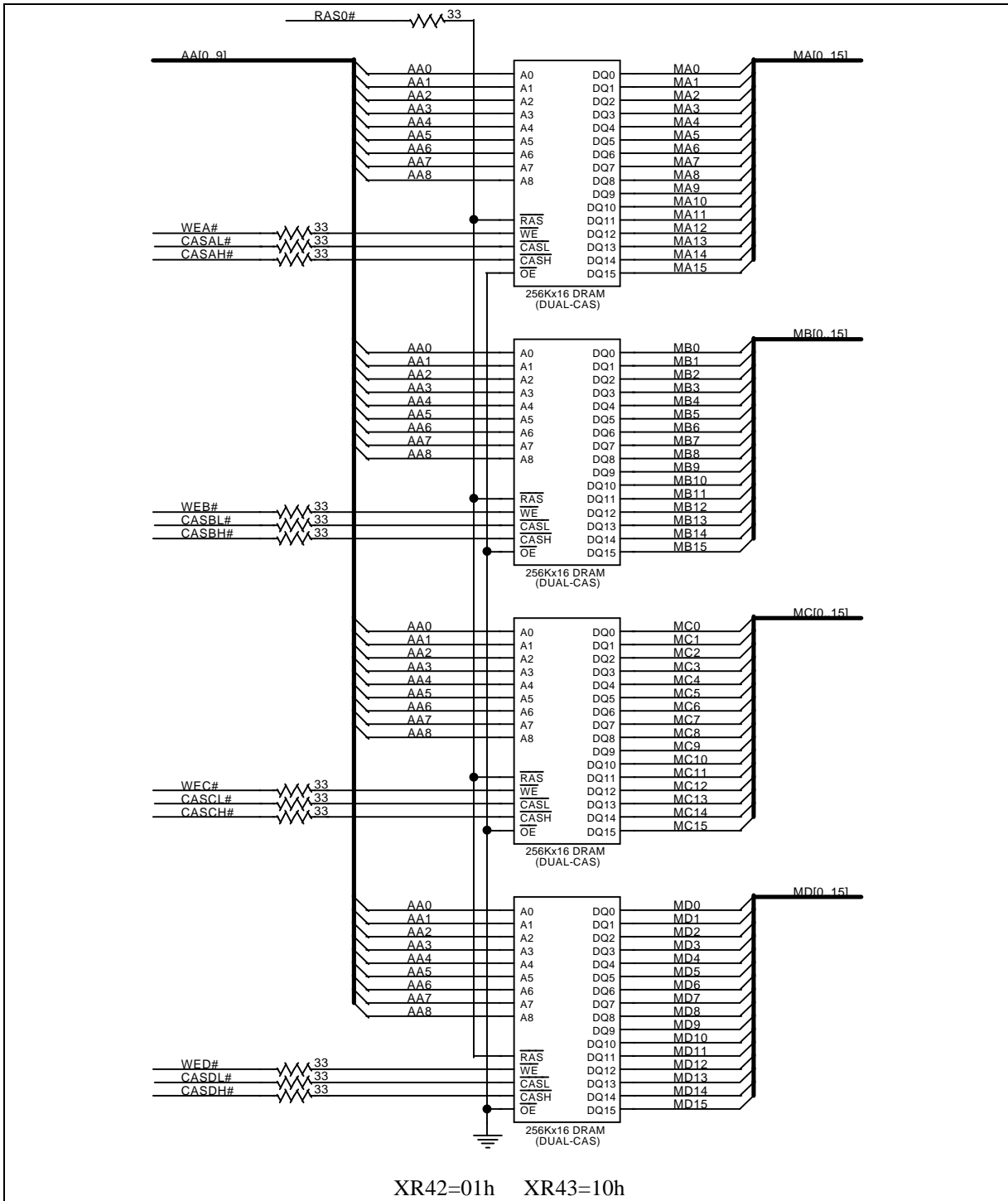
**Figure C-1:**  
**1MB, 64-bit Memory Configuration Using Two 128Kx32 EDO DRAMs**



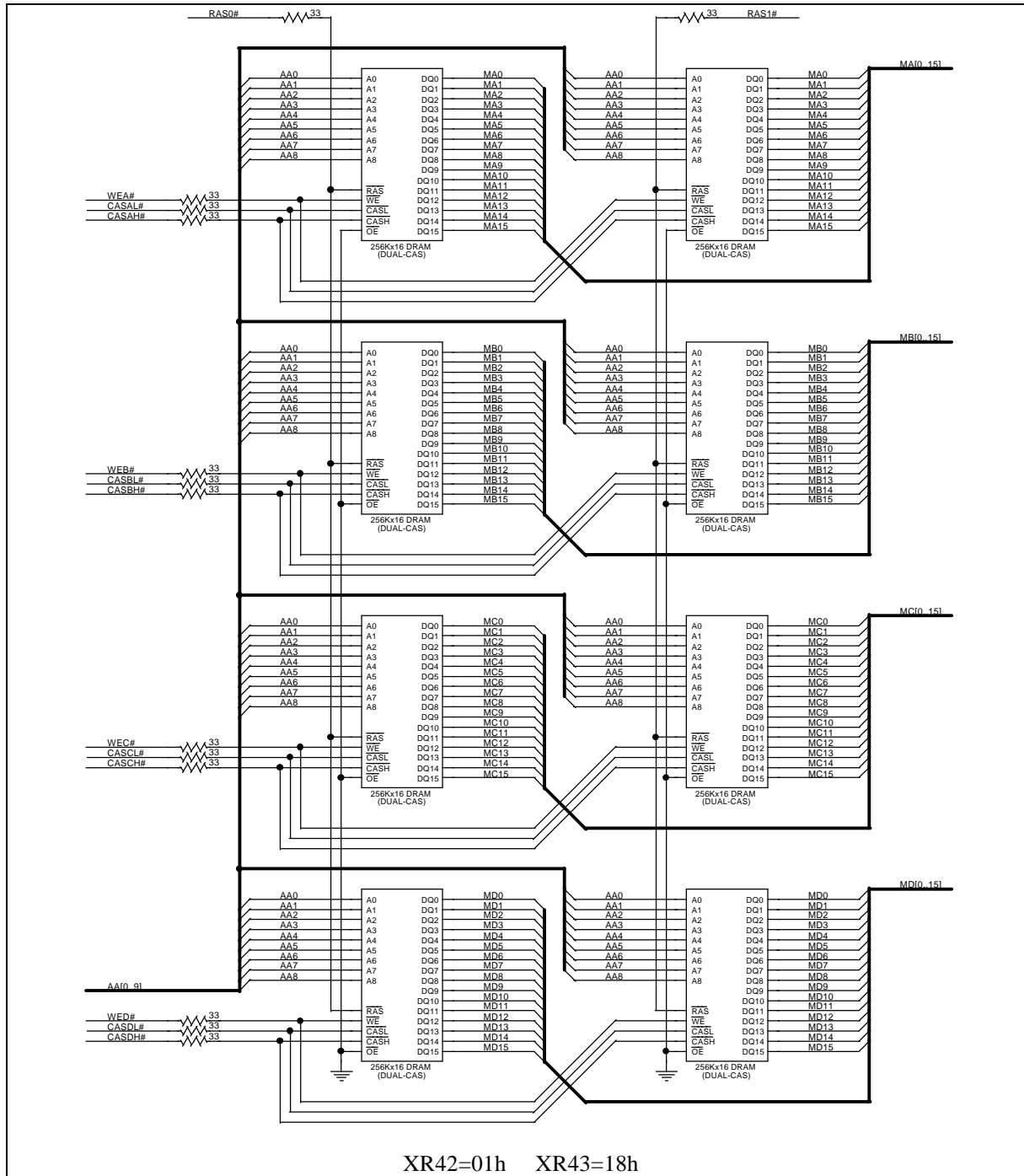
**Figure C-2:**  
**2MB, 64-bit Memory Configuration Using Four 128Kx32 EDO DRAMs**



**Figure C-3:**  
**3MB, 64-bit Memory Configuration Using Four 256Kx16 and Two 128Kx32 EDO DRAMs**

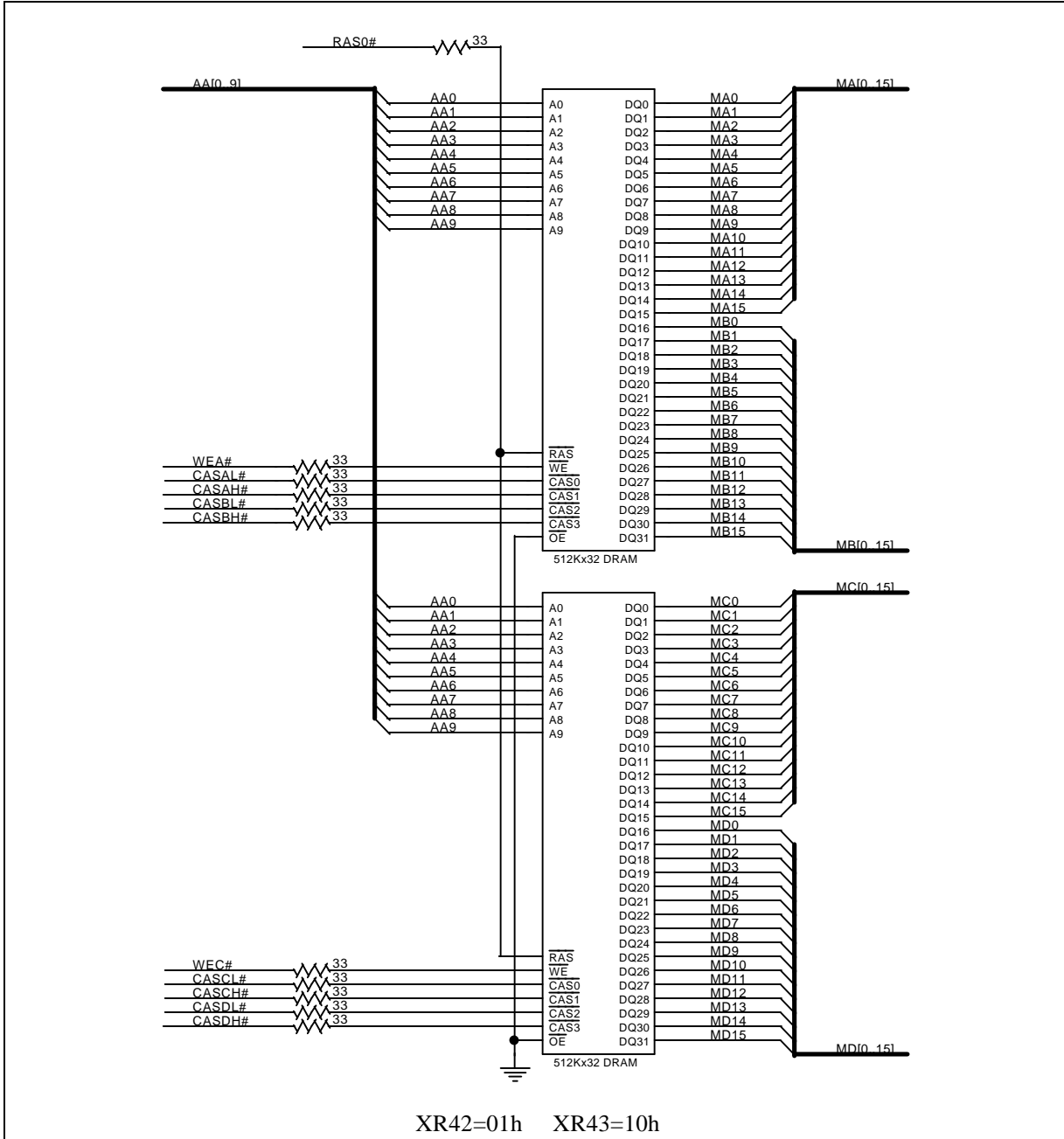


**Figure C-4:**  
**2MB, 64-bit Memory Configuration Using Four 256Kx16 EDO DRAMs**



**Figure C-5:**  
4MB, 64-bit Memory Configuration Using Eight 256Kx16 EDO DRAMs





**Figure C-6:**  
**4MB, 64-bit Memory Configuration Using Two 512Kx32 EDO DRAMs**

**This page intentionally left blank.**

## *APPENDIX D*

# *PANEL POWER SEQUENCING*

**This page intentionally left blank.**

## ***D. Panel Power Sequencing***

Flat panel displays are extremely sensitive to conditions where full biasing voltage VEE is applied to the liquid crystal material without enabling the control and data signals to the panel. This results in severe damage to the panel and may disable the panel permanently.

The graphics controller provides a simple method to provide or remove power to the flat panel display in a sequence of stages when entering various modes of operation to conserve power and provide safe operation to the flat panel.

Three pins called ENAVEE, ENAVDD and ENABKL are provided to regulate the LCD Bias Voltage (VEE), the driver electronics logic voltage

(VDD), and the backlight voltage (BKL) to provide intelligent power sequencing to the panel. The delay between each stage in the sequence is programmable via the Panel Power Sequencing Delay Register (FR04).

The graphics controller performs the 'panel off' sequence when the STNDBY# input becomes low, or if bit 3 of the Power Down Control 1 Register (FR05) is set to 1.

The graphics controller performs the 'panel on' sequence when the STNDBY# input becomes high, or if bit 3 of the Power Down Control 1 Register (FR05) is set to 0.

**This page intentionally left blank.**

***APPENDIX E***

***HARDWARE CURSOR AND POP UP WINDOW***

**This page intentionally left blank.**



## ***E. HARDWARE CURSOR AND POP UP WINDOW***

The graphics controller provides a pair of hardware-based cursors, called “cursor 1” and “cursor 2.” Cursor 1 is normally used to provide the arrow pointer in most GUI applications and operating systems. Cursor 2 has no pre-assigned purpose, however it is usually used to provide some form of pop-up window.

Off-screen memory in the frame buffer provides the locations of the data for both cursor 1 and cursor 2. This allows each cursor to be displayed and used without altering the main image stored in the frame buffer being altered. Each cursor may have multiple patterns stored in these off-screen memory locations. This makes it possible to change each cursor’s appearance simply by switching from one stored image to another.

Two sets of eight registers (XRA0-XRA7 for cursor 1, and XRA8-XRAF for cursor 2) provide the means to configure and position both cursors. In each set of eight registers, two are used to enable, disable, and configure each cursor. Another pair of registers from each set specifies

the base address within the frame buffer memory which stores the cursor data. These registers also provide a way to select one of up to sixteen cursor patterns to be used. The remaining four registers of each set are used to provide the X and Y coordinates to control the current location of each cursor relative to the upper left-hand corner of the display.

Two sets of four alternate color data positions added to the RAMDAC provide places which specify the colors for each of the two cursors (positions 0-3 for colors 0-3 of cursor 2, and positions 4-7 for colors 0-3 of cursor 1). These alternate color data positions are accessed by the same sub-addressing scheme used to access the standard color data positions of the main RAMDAC palette, with the exception that a bit in the Pixel Pipeline Configuration Register 0 (XR80) must be set so that the alternate color data positions are accessible in place of the standard color data positions.

## **E.1 Cursor Configuration**

Registers XRA0-XRA3 and registers XRA8-XRAB are used to enable/disable and configure cursor 1 and cursor 2.

### **E.1.1 Basic Cursor Configuration**

Cursor 1 and cursor 2 can each be independently disabled or configured for one of six possible modes using the Cursor 1 Control Register (XRA0) and the Cursor 2 Control Register (XRA8). Detailed descriptions of each of these six modes appear later in this document.

Horizontal and/or vertical stretching are functions that may be independently enabled or disabled for each cursor using these registers. Similar to the stretching functions used with the main display image, the stretching functions for each of the cursors only apply to flat panel displays. When enabled, the horizontal and vertical stretching functions for each cursor use the same stretching algorithms and parameter settings selected in the registers used to control the horizontal and vertical stretching functions for the main display image. The horizontal and vertical stretching functions for each cursor can be enabled or disabled entirely independent of the horizontal and vertical stretching functions for the main display image.

These same two registers also provide the means to enable or disable blinking for each cursor, and to choose between two possible locations on the screen for the origin of the coordinate system used to specify the cursor location. A bit in each of these registers provides the ability to choose either the upper left-hand corner of the active display area, or the outer-most upper left-hand corner of the display border surrounding the active display area as the exact location of the origin for the coordinate system for each cursor.

Finally, each of these registers allows the vertical extension function to be enabled or disabled for each cursor. The vertical extension function allows the height of the cursor to be specified independently from its width, allowing non-square shaped cursors to be created. This function is discussed in more detail in section D.1.3.

### **E.1.2 Base Address for Cursor Data**

The Cursor 1 Base Address Low Register (XRA2) and the Cursor 1 Base Address High Register (XRA3) are used to program the base address in the frame buffer at which the cursor data for cursor 1 begins. The Cursor 2 Base Address Low Register (XRAA) and the Cursor 2 Base Address High Register (XRAB) provide this function for cursor 2. The base address values stored in these registers actually specify an offset relative to the base address at which the frame buffer begins.

The amount of space allocated for cursor data for each cursor is 4KB. More than one cursor pattern may be stored within this space, depending on the cursor size. The bits in both the high and low base address registers for each of the cursors are combined to provide the base addresses. The upper four bits of each low base address register (XRA2 for cursor 1, and XRAA for cursor 2) are used to select which of the available patterns stored within each space is to be used for each of the cursors. In the 32x32x2bpp AND/XOR pixel plane mode, the 4KB memory space can store up to sixteen 256 byte patterns. All four of the upper bits of the low base address registers are used to select one of these sixteen possible patterns. In all three modes with a cursor resolution of 64x64 pixels, the 4KB memory space can store up to four 1KB patterns. The uppermost two of these four bits are used to select one of these four possible patterns (the other two bits should be set to 0). In both modes with a cursor resolution of 128x128 pixels, a maximum of two 2KB patterns can be stored, and only the uppermost bit of the four bits is used to select between them (the other three bits should be set to 0).

### **E.1.3 Cursor Vertical Extension**

The cursor vertical extension feature allows the vertical size (height) of either cursor in any of the six possible modes to be altered independently of the height normally specified by the choice of cursor mode. The cursor mode still determines the width of the cursor. This feature allows the cursor to have a non-square shape.

This feature is enabled via bit 3 of either the Cursor 1 Control Register (XRA0) for cursor 1 or the Cursor 2 Control Register (XRA8) for cursor 2. Once enabled, the height of the given cursor must be specified -- either in the Cursor 1 Vertical Extension Register (XRA1) for cursor 1, or in the Cursor 2 Vertical Extension Register (XRA9) for cursor 2.

Total size of the cursor data for a given cursor can not exceed the 4KB allotted for the cursor data of each cursor. This places a limit on the height of a cursor of given width and color depth. This also has implications concerning how many patterns may be stored in this space for the given cursor, and the mechanics of selecting which of those patterns is to be displayed using the upper four bits of the low base address register for each cursor.

### **E.1.4 Cursor Colors**

The colors for drawing each of the two cursors are specified in two sets of four alternate color data positions added to the RAMDAC (positions 0-3 for cursor 2 colors 0-3, and positions 4-7 for cursor 1 colors 0-3). These alternate color data positions are accessed using the same sub-addressing scheme used to access the standard color data positions of the main RAMDAC palette, but with bit 0 in the Pixel Pipeline Configuration Register 0 (XR80) set so that the alternate color data positions are made accessible in place of the standard positions.

If the use of a border is enabled, color data positions 6 and 7, which provide colors 2 and 3 for cursor 1, will specify the border colors for the CRT and flat-panel. This will limit cursor 1 to only colors 0 and 1. This limit on cursor 1 will not impact either of the AND/XOR pixel plane modes, or either of the cursor modes with a cursor resolution of 128x128 pixels because none of these four modes use cursor colors 2 or 3.

### **E.2. Cursor Modes**

Each cursor can be independently disabled or set to one of six possible modes. This is done using bits 2-0 in XRA0 for cursor 1 and in XRA8 for cursor 2. The main features which distinguish these modes from each other are the manner in which the cursor data is organized in memory and the meaning of the bits corresponding to each pixel position. The six possible modes are:

- 32x32x2bpp AND/XOR pixel plane mode
- 64x64x2bpp AND/XOR pixel plane mode
- 64x64x2bpp 4-color mode
- 64x64x2bpp 3-color and transparency mode
- 128x128x1bpp 2-color mode
- 128x128x1bpp 1-color and transparency mode

The first two modes are designed to follow the Microsoft Windows 2-plane cursor data structure to ease the work of programming the cursor(s) for that particular GUI environment. The other four modes are intended to improve upon the first two modes by providing additional color options or a larger resolution.

The following pages discuss the various modes in greater detail.

**E.2.1 32x32x2bpp & 64x64x2bpp AND/XOR Pixel Plane Modes**

These two modes are designed to follow the Microsoft Windows cursor data plane structure, which provides two colors that may be used to draw the cursor, a third color for transparency (which allows the main display image behind the cursor to show through), and a fourth color for inverted transparency (which allows the main display image behind the cursor to show through, but with its color inverted). Each pixel position within the cursor is defined by the combination of two bits of data, each of which is stored in planes referred to as the “AND” plane and the “XOR” plane.

In the 32x32x2bpp AND/XOR pixel plane mode, it is possible to have up to 16 different 256byte patterns stored in a 4KB memory space starting at the base address specified in the low and high base address registers for the given cursor. In 64x64x2bpp AND/XOR pixel plane mode, only up to 4 different 1KB patterns may be stored.

The tables that follow show how the cursor data is organized in memory for each of these two modes:

**Table E-1: Memory Organization 32x32x2bpp AND/XOR Pixel Plane Mode**

Offset	Plane	Pixels
000h	AND	31-0 on line 0 of pattern 0
004h	AND	31-0 on line 1 of pattern 0
008h	XOR	31-0 on line 0 of pattern 0
00Ch	XOR	31-0 on line 1 of pattern 0
010h	AND	31-0 on line 2 of pattern 0
014h	AND	31-0 on line 3 of pattern 0
...	...	...
0F0h	AND	31-0 on line 30 of pattern 0
0F4h	AND	31-0 on line 31 of pattern 0
0F8h	XOR	31-0 on line 30 of pattern 0
0FCh	XOR	31-0 on line 31 of pattern 0
100h	AND	31-0 on line 0 of pattern 1
104h	AND	31-0 on line 1 of pattern 1
...	...	...
FF8h	XOR	31-0 of line 30 of pattern 1
FFCh	XOR	31-0 of line 31 of pattern 1

**Table E-2: Memory Organization 64x64x2bpp AND/XOR Pixel Plane Mode**

Offset	Plane	Pixels
000h	AND	31-0 on line 0 of pattern 0
004h	AND	63-32 on line 0 of pattern 0
008h	XOR	31-0 on line 0 of pattern 0
00Ch	XOR	63-32 on line 0 of pattern 0
010h	AND	31-0 on line 1 of pattern 0
014h	AND	63-32 on line 1 of pattern 0
...	...	...
3F0h	AND	31-0 on line 63 of pattern 0
3F4h	AND	63-32 on line 63 of pattern 0
3F8h	XOR	31-0 on line 63 of pattern 0
3FCh	XOR	63-32 on line 63 of pattern 0
400h	AND	31-0 on line 0 of pattern 1
404h	AND	63-32 on line 0 of pattern 1
...	...	...
FF8h	XOR	31-0 on line 63 of pattern 3
FFCh	XOR	63-32 on line 63 of pattern 3

The meaning of the single bit in a given pixel position in the XOR plane changes depending on the bit in the corresponding position in the AND plane. If the value of the bit for a given pixel position in the AND plane is 0, then part of the cursor will be displayed at that pixel position, and the value of the corresponding bit in the XOR plane selects one of the two available cursor colors to be displayed. Otherwise, if the value of the bit in the AND plane is 1, then that pixel position of the cursor will become transparent, allowing a pixel of the main display image behind the cursor to appear. The value of the corresponding bit in the XOR plane chooses whether or not the color of the pixel of the main display image will be inverted. Table D-3 summarizes this information.

**Table E-3: Pixel Data 32x32x2bpp and 64x64x2bpp AND/XOR Pixel Plane Modes**

AND Plane Pixel Data	XOR Plane Pixel Data	Color Displayed at the Corresponding Pixel Position
0	0	Cursor color 0
0	1	Cursor color 1
1	0	Transparent. The pixel of the main display image behind cursor shows through
1	1	Transparent, but inverted. The pixel of the main display image behind cursor shows through with inverted color

### E.2.2 64x64x2bpp 4-Color Mode

This mode provides four colors for drawing the cursor. There is no provision for transparency in the 64x64 pixel space occupied by the cursor, so unless the image behind the cursor is the same color as one of the four colors used to draw the cursor, the cursor will appear to be a 64 x 64 pixel square. Each pixel position within the cursor is defined by the combination of two bits, each of which is stored in planes referred to as plane 0 and plane 1.

In this mode, it is possible to have up to 4 different 1KB patterns stored in a 4KB memory space starting at the base address specified in the low and high base address registers for the given cursor.

The following tables show how the cursor data is organized in memory and the meaning of the two bits for each pixel position.

**Table E-4: Memory Organization  
64x64x2bpp 4-Color Mode**

Offset	Plane	Pixels
000h	0	31-0 on line 0 of pattern 0
004h	0	63-32 on line 0 of pattern 0
008h	1	31-0 on line 0 of pattern 0
00Ch	1	63-32 on line 0 of pattern 0
010h	0	31-0 on line 1 of pattern 0
014h	0	63-32 on line 1 of pattern 0
...	...	...
3F0h	0	31-0 on line 63 of pattern 0
3F4h	0	63-32 on line 63 of pattern 0
3F8h	1	31-0 on line 63 of pattern 0
3FCh	1	63-32 on line 63 of pattern 0
400h	0	31-0 on line 0 of pattern 1
404h	0	63-32 on line 0 of pattern 1
...	...	...
FF8h	1	31-0 on line 63 of pattern 3
FFCh	1	63-32 on line 63 of pattern 3

**Table E-5: Pixel Data  
64x64x2bpp 4-Color Mode**

Plane 0 Pixel Data	Plane 1 Pixel Data	Color Displayed at the Corresponding Pixel Position
0	0	Cursor color 0
0	1	Cursor color 1
1	0	Cursor color 2
1	1	Cursor color 3

### E.2.3 64x64x2bpp 3-Color and Transparency Mode

This mode provides three colors for drawing the cursor, and a fourth color for transparency (which allows the main display image behind the cursor to show through). Each pixel position in the cursor is defined by the combination of two bits, stored in planes called to as plane 0 and plane 1.

In this mode, it is possible to have up to 4 different 1KB patterns stored in a 4KB memory space starting at the base address specified in the low and high base address registers for the given cursor.

The tables that follow show how the cursor data is organized in memory and the meaning of the two bits for each pixel position.

**Table E-6: Memory Organization  
64x64x2bpp 3-Color & Transparency Mode**

Offset	Plane	Pixels
000h	0	31-0 on line 0 of pattern 0
004h	0	63-32 on line 0 of pattern 0
008h	1	31-0 on line 0 of pattern 0
00Ch	1	63-32 on line 0 of pattern 0
010h	0	31-0 on line 1 of pattern 0
014h	0	63-32 on line 1 of pattern 0
...	...	...
3F0h	0	31-0 on line 63 of pattern 0
3F4h	0	63-32 on line 63 of pattern 0
3F8h	1	31-0 on line 63 of pattern 0
3FCh	1	63-32 on line 63 of pattern 0
400h	0	31-0 on line 0 of pattern 1
404h	0	63-32 on line 0 of pattern 1
...	...	...
FF8h	1	31-0 on line 63 of pattern 3
FFCh	1	63-32 on line 63 of pattern 3

**Table E-7: Pixel Data  
64x64x2bpp 3-Color & Transparency Mode**

Plane 0 Pixel Data	Plane 1 Pixel Data	Color Displayed at the Corresponding Pixel Position
0	0	Cursor color 0
0	1	Cursor color 1
1	0	Transparent Pixel of the image behind the cursor shows through
1	1	Cursor color 3

**E.2.4 128x128x1bpp 2-Color Mode**

This mode provides two colors for drawing the cursor. There is no provision for transparency in the 128x128 pixel space occupied by the cursor, so unless the image behind the cursor is the same color as one of the two colors used to draw the cursor, the cursor will appear as a 128x128 pixel square.

In this mode, it is possible to have only up to 2 different 2KB patterns stored in a 4KB memory space starting at the base address specified in the low and high base address registers for the given cursor.

The tables that follow show how the cursor data is organized in memory and the meaning of the bit for each position.

**Table E-8: Memory Organization  
128x128x1bpp 2-Color Mode**

<b>Offset</b>	<b>Pixels</b>
000h	31-0 on line 0 of pattern 0
004h	63-32 on line 0 of pattern 0
008h	95-64 on line 0 of pattern 0
00Ch	127-96 on line 0 of pattern 0
010h	31-0 on line 1 of pattern 0
014h	63-32 on line 1 of pattern 0
...	...
7F0h	31-0 on line 127 of pattern 0
7F4h	63-32 on line 127 of pattern 0
7F8h	95-64 on line 127 of pattern 0
7FCh	127-96 on line 127 of pattern 0
800h	31-0 on line 0 of pattern 1
804h	63-32 on line 0 of pattern 1
...	...
FF8h	95-64 on line 127 of pattern 1
FFCh	127-96 on line 127 of pattern 1

**Table E-9: Pixel Data  
128x128x1bpp 2-Color Mode**

<b>Pixel Data Bit</b>	<b>Color Displayed at the Corresponding Pixel Position</b>
0	Cursor color 2
1	Cursor color 3

### E.2.5 128x128x1bpp 1-Color and Transparency Mode

This mode provides one color for drawing the cursor, and a second color for transparency (which allows the image behind the cursor to show through).

In this mode, it is possible to have only up to 2 different 2KB patterns stored in a 4KB memory space starting at the base address specified in the low and high base address registers for the given cursor. The tables that follow show how the cursor data is organized in memory and the meaning of the bit for each position.

**Table E-10: Memory Organization  
128x128x1bpp 1-Color & Transparency Mode**

Offset	Pixels
000h	31-0 on line 0 of pattern 0
004h	63-32 on line 0 of pattern 0
008h	95-64 on line 0 of pattern 0
00Ch	127-96 on line 0 of pattern 0
010h	31-0 on line 1 of pattern 0
014h	63-32 on line 1 of pattern 0
...	...
7F0h	31-0 on line 127 of pattern 0
7F4h	63-32 on line 127 of pattern 0
7F8h	95-64 on line 127 of pattern 0
7FCh	127-96 on line 127 of pattern 0
800h	31-0 on line 0 of pattern 1
804h	63-32 on line 0 of pattern 1
...	...
FF8h	95-64 on line 127 of pattern 1
FFCh	127-96 on line 127 of pattern 1

**Table E-11: Pixel Bit Definitions  
128x128x1bpp 1-Color & Transparency Mode**

Pixel Data Bit	Color Displayed at the Corresponding Pixel Position
0	Transparent. Pixel of the image behind cursor shows through
1	Cursor color 2



### **E.3 Cursor Positioning**

Registers XRA4-XRA7 and registers XRAC-XRAF are used to position cursor 1 and cursor 2, respectively, on the display. Two registers from each group provide the high and low bytes for the value specifying the horizontal position, and the other two provide the high and low bytes for the value specifying the vertical position.

A bit in one of the configuration registers (XRA0 for cursor 1 and XRA8 for cursor 2) selects whether the values programmed into these registers are interpreted as being relative to the upper left-hand corner of the active display area or to the outer-most upper left-hand corner of the border surrounding the active display area.

The values provided to these registers are signed 12-bit integers. Since the origin of the coordinate

system is generally relative to the upper left corner of the display, the horizontal value is a positive integer, while the vertical value is negative.

These registers are double-buffered and synchronized to VSYNC to ensure that the cursor never appears to come apart in multiple fragments as it is being moved across the screen. To change a cursor position, all four of its position registers must be written, and they must be written in sequence (that is, in order from XRA4 to XRA7 for cursor 1, and in order from XRAC to XRAF for cursor 2.) The hardware will only update the position with the next VSYNC if the registers are written in sequence.

**This page intentionally left blank.**

# *APPENDIX F*

## *BITBLT OPERATION*

**This page intentionally left blank.**

# F BITBLT ENGINE

## F.1 INTRODUCTION

The graphics controller provides a hardware-based BitBLT engine to offload the work of moving blocks of graphics data from the host CPU. Although the BitBLT engine is often used simply to copy a block of graphics data from the source to the destination, it also has the ability to perform more complex functions. The BitBLT engine is capable of receiving three different blocks of graphics data as input as shown in Figure F-1. The source data may exist either in the frame buffer or it may be provided by the host CPU from some other source such as system memory. The pattern data always represents an 8x8 block of pixels that must be located in the frame buffer, usually within the off-screen portion. If the destination is within the frame buffer, then the data already residing at the destination may also be used as an input.

The BitBLT engine may be configured to use various combinations of these three different blocks of graphics data as operands, in both bit-wise logical operations to generate the actual data to be written to the destination, and in per-pixel write-masking to control the writing of data to the destination. It is intended that the BitBLT engine will perform these bit-wise and per-pixel operations on color graphics data that is at a color depth that matches the rest of the graphics system. However, if either the source or pattern data is monochrome, the BitBLT engine has the ability to put either block of graphics data through a process called “color expansion” which converts monochrome graphics data to color. Since the destination is often a location in the on-screen portion of the frame buffer, it is assumed that any data already at the destination will be of the appropriate color depth.

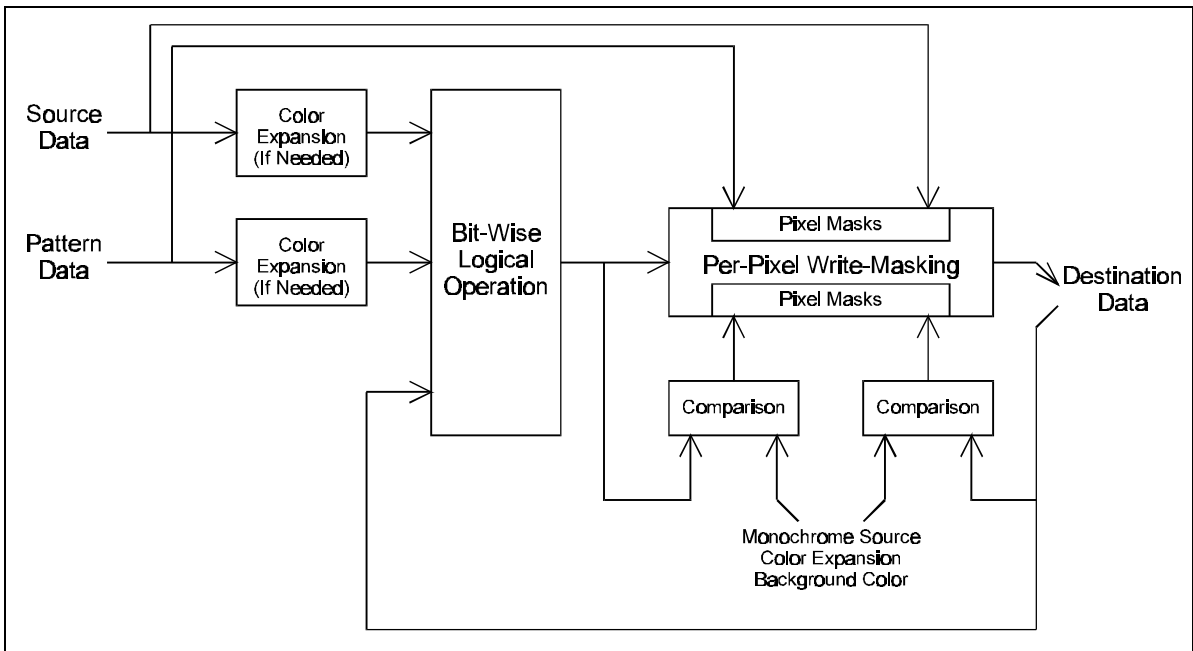


Figure F-1: Block Diagram and Data Paths of the BitBLT Engine

## F.2 BASIC BITBLT FUNCTIONAL CONSIDERATIONS

### F.2.1 COLOR DEPTH CONFIGURATION AND COLOR EXPANSION

The graphics system can be configured for color depths of 1, 2, 4, 8, 16, 24, and 32 bits per pixel, while the BitBLT engine is intended to work with graphics data having a color depth of only 8, 16, or 24 bits per pixel. It is assumed that the BitBLT engine will not be used when the graphics system has been configured for a color depth that the BitBLT engine was not designed to support. In theory, it is possible to configure the BitBLT engine and graphics system for different color depths, but this is not a recommended practice.

The configuration of the BitBLT engine for a given color depth dictates the number of bytes of graphics data that the BitBLT engine will read and write for each pixel while performing a BitBLT operation. When the destination is within the frame buffer, it is assumed that any graphics data already residing at the destination which will be used as an input will already be at the color depth to which the BitBLT engine is configured. Similarly, it is assumed that any source or pattern data used as an input will have this same color depth, unless one or both is monochrome. If either the source or pattern data is monochrome, the BitBLT engine will perform a process called “color expansion” to convert such monochrome data to color at the color depth to which the BitBLT engine has been set.

During “color expansion” the individual bits of monochrome source or pattern data that correspond to individual pixels are converted into 1, 2, or 3 bytes (whichever is appropriate for the color depth to which the BitBLT engine has been

set). If a given bit of monochrome source or pattern data carries a value of 1, then the byte(s) of color data resulting from the conversion process will be set to carry the value of a specified foreground color. If a given bit of monochrome source or pattern data carries a value of 0, the resulting byte(s) will be set to the value of a specified background color.

The BitBLT engine is configured for a color depth of 8, 16, or 24 bits per pixel through either bits 5 and 4 of the BitBLT Configuration Register (XR20) or bits 25 and 24 of the BitBLT Control Register (BR04) depending upon the setting of bit 23 of the BitBLT Control Register (BR04). Whether the source and pattern data are color or monochrome must be specified using bits 12 and 18, respectively, in the BitBLT Control Register (BR04). Foreground and background colors for the color expansion of both monochrome source and pattern data may be specified using the Pattern/Source Expansion Foreground Color Register (BR02) and the Pattern/Source Expansion Background Color Register (BR01). Alternatively, if bit 27 of the Monochrome Source Control Register (BR03) is set to 1, the foreground and background colors used in the color expansion of monochrome source data may be specified independently of those used for the color expansion of monochrome pattern data by using the Source Expansion Foreground Color Register (BR09) and the Source Expansion Background Color Register (BR0A).

### **F.2.2 GRAPHICS DATA SIZE LIMITATIONS**

The BitBLT engine is capable of transferring very large quantities of graphics data. Any graphics data read from and written to the destination is permitted to represent a number of pixels that occupies up to 8191 scan lines and up to 8191 bytes per scan line at the destination. The maximum number of pixels that may be represented per scan line's worth of graphics data depends on the color depth.

Any source data used as an input must represent the same number of pixels as is represented by any data read from or written to the destination, and it must be organized so as to occupy the same number of scan lines and pixels per scan line. Despite these constraints, if the block of source data is received from the host CPU, it may be received as part of a much larger stream of data sent by the host CPU. The BitBLT engine may be programmed to skip over various quantities of bytes in such a stream of data received from the host CPU in order to reach the bytes containing valid source data.

The actual number of scan lines and bytes per scan line required to accommodate data read from or written to the destination are set in the Destination Width & Height Register (BR08). These two values are essential in the programming of the BitBLT engine, because it uses these two values to determine when a given BitBLT operation has been completed. The act of writing a non-zero value for the height into this register is the signal to the BitBLT engine to begin performing the BitBLT operation for which it has been programmed.

### **F.2.3 BIT-WISE OPERATIONS**

The BitBLT engine can perform any one of 256 possible bit-wise operations using various combinations of the three previously described blocks of graphics data that the BitBLT engine can receive as input. These 256 possible bit-wise operations are designed to be compatible with the manner in which raster operations are specified in the standard BitBLT parameter block normally used in the Microsoft<sup>®</sup> Windows<sup>®</sup> environment, without translation.

The choice of bit-wise operation selects which of the three inputs will be used, as well as the particular logical operation to be performed on corresponding bits from each of the selected inputs. The BitBLT engine will automatically forego reading any form of graphics data that has not been specified as an input by the choice of bit-wise operation. An 8-bit code written to the BitBLT Control Register (BR04) chooses the bit-wise operation. It should be noted that if the BitBLT engine is configured to provide the output destination data to the host CPU, instead of writing it to the frame buffer, then bit-wise operations using destination data as an input should not be used. Tables E-1, E-2, and E-3 on the following pages list the available bit-wise operations and their corresponding 8-bit codes.

**Table F-1: Bit-Wise Operations and 8-bit Codes (00 - 5F)**

Code	Value Written to Bits at Destination	Code	Value Written to Bits at Destination
00	writes all 0's	30	P and ( notS )
01	not( D or ( P or S ))	31	not( S or ( D and ( notP )))
02	D and ( not( P or S ))	32	S xor ( D or ( P or S ))
03	not( P or S )	33	notS
04	S and ( not( D or P ))	34	S xor ( P or ( D and S ))
05	not( D or P )	35	S xor ( P or ( not( D xor S )))
06	not( P or ( not( D xor S )))	36	S xor ( D or P )
07	not( P or ( D and S ))	37	not( S and ( D or P ))
08	S and ( D and ( notP ))	38	P xor ( S and ( D or P ))
09	not( P or ( D xor S ))	39	S xor ( P or ( notD ))
0A	D and ( notP )	3A	S xor ( P or ( D xor S ))
0B	not( P or ( S and ( notD )))	3B	not( S and ( P or ( notD )))
0C	S and ( notP )	3C	P xor S
0D	not( P or ( D and ( notS )))	3D	S xor ( P or ( not( D or S )))
0E	not( P or ( not( D or S )))	3E	S xor ( P or ( D and ( notS )))
0F	notP	3F	not( P and S )
10	P and ( not( D or S ))	40	P and ( S and ( notD ))
11	not( D or S )	41	not( D or ( P xor S ))
12	not( S or ( not( D xor P )))	42	( S xor D ) and ( P xor D )
13	not( S or ( D and P ))	43	not( S xor ( P and ( not( D and S )))
14	not( D or ( not( P xor S )))	44	S and ( notD )
15	not( D or ( P and S ))	45	not( D or ( P and ( notS )))
16	P xor ( S xor ( D and ( not( P and S )))	46	D xor ( S or ( P and D ))
17	not( S xor (( S xor P ) and ( D xor S )))	47	not( P xor ( S and ( D xor P )))
18	( S xor P ) and ( P xor D )	48	S and ( D xor P )
19	not( S xor ( D and ( not( P and S )))	49	not( P xor ( D xor ( S or ( P and D )))
1A	P xor ( D or ( S and P ))	4A	D xor ( P and ( S or D ))
1B	not( S xor ( D and ( P xor S )))	4B	P xor ( D or ( notS ))
1C	P xor ( S or ( D and P ))	4C	S and ( not( D and P ))
1D	not( D xor ( S and ( P xor D )))	4D	not( S xor (( S xor P ) or ( D xor S )))
1E	P xor ( D or S )	4E	P xor ( D or ( S xor P ))
1F	not( P and ( D or S ))	4F	not( P and ( D or ( notS )))
20	D and ( P and ( notS ))	50	P and ( notD )
21	not( S or ( D xor P ))	51	not( D or ( S and ( notP )))
22	D and ( notS )	52	D xor ( P or ( S and D ))
23	not( S or ( P and ( notD )))	53	not( S xor ( P and ( D xor S )))
24	( S xor P ) and ( D xor S )	54	not( D or ( not( P or S )))
25	not( P xor ( D and ( not( S and P )))	55	notD
26	S xor ( D or ( P and S ))	56	D xor ( P or S )
27	S xor ( D or ( not( P xor S )))	57	not( D and ( P or S ))
28	D and ( P xor S )	58	P xor ( D and ( S or P ))
29	not( P xor ( S xor ( D or ( P and S )))	59	D xor ( P or ( notS ))
2A	D and ( not( P and S ))	5A	D xor P
2B	not( S xor (( S xor P ) and ( P xor D )))	5B	D xor ( P or ( not( S or D )))
2C	S xor ( P and ( D or S ))	5C	D xor ( P or ( S xor D ))
2D	P xor ( S or ( notD ))	5D	not( D and ( P or ( notS )))
2E	P xor ( S or ( D xor P ))	5E	D xor ( P or ( S and ( notD )))
2F	not( P and ( S or ( notD )))	5F	not( D and P )

**Notes:** S = Source Data  
P = Pattern Data  
D = Data Already Existing at the Destination



**Table F-2: Bit-Wise Operations and 8-bit Codes (60 - BF)**

Code	Value Written to Bits at Destination	Code	Value Written to Bits at Destination
60	$P \text{ and } (D \text{ xor } S)$	90	$P \text{ and } (\text{not}(D \text{ xor } S))$
61	$\text{not}(D \text{ xor } (S \text{ xor } (P \text{ or } (D \text{ and } S))))$	91	$\text{not}(S \text{ xor } (D \text{ and } (P \text{ or } (\text{not}S))))$
62	$D \text{ xor } (S \text{ and } (P \text{ or } D))$	92	$D \text{ xor } (P \text{ xor } (S \text{ and } (D \text{ or } P)))$
63	$S \text{ xor } (D \text{ or } (\text{not}P))$	93	$\text{not}(S \text{ xor } (P \text{ and } D))$
64	$S \text{ xor } (D \text{ and } (P \text{ or } S))$	94	$P \text{ xor } (S \text{ xor } (D \text{ and } (P \text{ or } S)))$
65	$D \text{ xor } (S \text{ or } (\text{not}P))$	95	$\text{not}(D \text{ xor } (P \text{ and } S))$
66	$D \text{ xor } S$	96	$D \text{ xor } (P \text{ xor } S)$
67	$S \text{ xor } (D \text{ or } (\text{not}(P \text{ or } S)))$	97	$P \text{ xor } (S \text{ xor } (D \text{ or } (\text{not}(P \text{ or } S))))$
68	$\text{not}(D \text{ xor } (S \text{ xor } (P \text{ or } (\text{not}(D \text{ or } S))))$	98	$\text{not}(S \text{ xor } (D \text{ or } (\text{not}(P \text{ or } S))))$
69	$\text{not}(P \text{ xor } (D \text{ xor } S))$	99	$\text{not}(D \text{ xor } S)$
6A	$D \text{ xor } (P \text{ and } S)$	9A	$D \text{ xor } (P \text{ and } (\text{not}S))$
6B	$\text{not}(P \text{ xor } (S \text{ xor } (D \text{ and } (P \text{ or } S))))$	9B	$\text{not}(S \text{ xor } (D \text{ and } (P \text{ or } S)))$
6C	$S \text{ xor } (D \text{ and } P)$	9C	$S \text{ xor } (P \text{ and } (\text{not}D))$
6D	$\text{not}(P \text{ xor } (D \text{ xor } (S \text{ and } (P \text{ or } D))))$	9D	$\text{not}(D \text{ xor } (S \text{ and } (P \text{ or } D)))$
6E	$S \text{ xor } (D \text{ and } (P \text{ or } (\text{not}S)))$	9E	$D \text{ xor } (S \text{ xor } (P \text{ or } (D \text{ and } S)))$
6F	$\text{not}(P \text{ and } (\text{not}(D \text{ xor } S)))$	9F	$\text{not}(P \text{ and } (D \text{ xor } S))$
70	$P \text{ and } (\text{not}(D \text{ and } S))$	A0	$D \text{ and } P$
71	$\text{not}(S \text{ xor } ((S \text{ xor } D) \text{ and } (P \text{ xor } D)))$	A1	$\text{not}(P \text{ xor } (D \text{ or } (S \text{ and } (\text{not}P))))$
72	$S \text{ xor } (D \text{ or } (P \text{ xor } S))$	A2	$D \text{ and } (P \text{ or } (\text{not}S))$
73	$\text{not}(S \text{ and } (D \text{ or } (\text{not}P)))$	A3	$\text{not}(D \text{ xor } (P \text{ or } (S \text{ xor } D)))$
74	$D \text{ xor } (S \text{ or } (P \text{ xor } D))$	A4	$\text{not}(P \text{ xor } (D \text{ or } (\text{not}(S \text{ or } P))))$
75	$\text{not}(D \text{ and } (S \text{ or } (\text{not}P)))$	A5	$\text{not}(P \text{ xor } D)$
76	$S \text{ xor } (D \text{ or } (P \text{ and } (\text{not}S)))$	A6	$D \text{ xor } (S \text{ and } (\text{not}P))$
77	$\text{not}(D \text{ and } S)$	A7	$\text{not}(P \text{ xor } (D \text{ and } (S \text{ or } P)))$
78	$P \text{ xor } (D \text{ and } S)$	A8	$D \text{ and } (P \text{ or } S)$
79	$\text{not}(D \text{ xor } (S \text{ xor } (P \text{ and } (D \text{ or } S))))$	A9	$\text{not}(D \text{ xor } (P \text{ or } S))$
7A	$D \text{ xor } (P \text{ and } (S \text{ or } (\text{not}D)))$	AA	$D$
7B	$\text{not}(S \text{ and } (\text{not}(D \text{ xor } P)))$	AB	$D \text{ or } (\text{not}(P \text{ or } S))$
7C	$S \text{ xor } (P \text{ and } (D \text{ or } (\text{not}S)))$	AC	$S \text{ xor } (P \text{ and } (D \text{ xor } S))$
7D	$\text{not}(D \text{ and } (\text{not}(P \text{ xor } S)))$	AD	$\text{not}(D \text{ xor } (P \text{ or } (S \text{ and } D)))$
7E	$(S \text{ xor } P) \text{ or } (D \text{ xor } S)$	AE	$D \text{ or } (S \text{ and } (\text{not}P))$
7F	$\text{not}(D \text{ and } (P \text{ and } S))$	AF	$D \text{ or } (\text{not}P)$
80	$D \text{ and } (P \text{ and } S)$	B0	$P \text{ and } (D \text{ or } (\text{not}S))$
81	$\text{not}((S \text{ xor } P) \text{ or } (D \text{ xor } S))$	B1	$\text{not}(P \text{ xor } (D \text{ or } (S \text{ xor } P)))$
82	$D \text{ and } (\text{not}(P \text{ xor } S))$	B2	$S \text{ xor } ((S \text{ xor } P) \text{ or } (D \text{ xor } S))$
83	$\text{not}(S \text{ xor } (P \text{ and } (D \text{ or } (\text{not}S))))$	B3	$\text{not}(S \text{ and } (\text{not}(D \text{ and } P)))$
84	$S \text{ and } (\text{not}(D \text{ xor } P))$	B4	$P \text{ xor } (S \text{ and } (\text{not}D))$
85	$\text{not}(P \text{ xor } (D \text{ and } (S \text{ or } (\text{not}P))))$	B5	$\text{not}(D \text{ xor } (P \text{ and } (S \text{ or } D)))$
86	$D \text{ xor } (S \text{ xor } (P \text{ and } (D \text{ or } S)))$	B6	$D \text{ xor } (P \text{ xor } (S \text{ or } (D \text{ and } P)))$
87	$\text{not}(P \text{ xor } (D \text{ and } S))$	B7	$\text{not}(S \text{ and } (D \text{ xor } P))$
88	$D \text{ and } S$	B8	$P \text{ xor } (S \text{ and } (D \text{ xor } P))$
89	$\text{not}(S \text{ xor } (D \text{ or } (P \text{ and } (\text{not}S))))$	B9	$\text{not}(D \text{ xor } (S \text{ or } (P \text{ and } D)))$
8A	$D \text{ and } (S \text{ or } (\text{not}P))$	BA	$D \text{ or } (P \text{ and } (\text{not}S))$
8B	$\text{not}(D \text{ xor } (S \text{ or } (P \text{ xor } D)))$	BB	$D \text{ or } (\text{not}S)$
8C	$S \text{ and } (D \text{ or } (\text{not}P))$	BC	$S \text{ xor } (P \text{ and } (\text{not}(D \text{ and } S)))$
8D	$\text{not}(S \text{ xor } (D \text{ or } (P \text{ xor } S)))$	BD	$\text{not}((S \text{ xor } D) \text{ and } (P \text{ xor } D))$
8E	$S \text{ xor } ((S \text{ xor } D) \text{ and } (P \text{ xor } D))$	BE	$D \text{ or } (P \text{ xor } S)$
8F	$\text{not}(P \text{ and } (\text{not}(D \text{ and } S)))$	BF	$D \text{ or } (\text{not}(P \text{ and } S))$

**Notes:** S = Source Data  
P = Pattern Data  
D = Data Already Existing at the Destination

**Table F-3: Bit-Wise Operations and 8-bit Codes (C0 - FF)**

Code	Value Written to Bits at Destination	Code	Value Written to Bits at Destination
C0	P and S	E0	P and ( D or S )
C1	not( S xor ( P or ( D and ( notS ) ) ) )	E1	not( P xor ( D or S ) )
C2	not( S xor ( P or ( not( D or S ) ) ) )	E2	D xor ( S and ( P xor D ) )
C3	not( P xor S )	E3	not( P xor ( S or ( D and P ) ) )
C4	S and ( P or ( notD ) )	E4	S xor ( D and ( P xor S ) )
C5	not( S xor ( P or ( D xor S ) ) )	E5	not( P xor ( D or ( S and P ) ) )
C6	S xor ( D and ( notP ) )	E6	S xor ( D and ( not( P and S ) ) )
C7	not( P xor ( S and ( D or P ) ) )	E7	not(( S xor P ) and ( P xor D ) )
C8	S and ( D or P )	E8	S xor (( S xor P ) and ( D xor S ) )
C9	not( S xor ( P or D ) )	E9	not( D xor ( S xor ( P and ( not( D and S ) ) ) ) )
CA	D xor ( P and ( S xor D ) )	EA	D or ( P and S )
CB	not( S xor ( P or ( D and S ) ) )	EB	D or ( not( P xor S ) )
CC	S	EC	S or ( D and P )
CD	S or ( not( D or P ) )	ED	S or ( not( D xor P ) )
CE	S or ( D and ( notP ) )	EE	D or S
CF	S or ( notP )	EF	S or ( D or ( notP ) )
D0	P and ( S or ( notD ) )	F0	P
D1	not( P xor ( S or ( D xor P ) ) )	F1	P or ( not( D or S ) )
D2	P xor ( D and ( notS ) )	F2	P or ( D and ( notS ) )
D3	not( S xor ( P and ( D or S ) ) )	F3	P or ( notS )
D4	S xor (( S xor P ) and ( P xor D ) )	F4	P or ( S and ( notD ) )
D5	not( D and ( not( P and S ) ) )	F5	P or ( notD )
D6	P xor ( S xor ( D or ( P and S ) ) )	F6	P or ( D xor S )
D7	not( D and ( P xor S ) )	F7	P or ( not( D and S ) )
D8	P xor ( D and ( S xor P ) )	F8	P or ( D and S )
D9	not( S xor ( D or ( P and S ) ) )	F9	P or ( not( D xor S ) )
DA	D xor ( P and ( not( S and D ) ) )	FA	D or P
DB	not(( S xor P ) and ( D xor S ) )	FB	D or ( P or ( notS ) )
DC	S or ( P and ( notD ) )	FC	P or S
DD	S or ( notD )	FD	P or ( S or ( notD ) )
DE	S or ( D xor P )	FE	D or ( P or S )
DF	S or ( not( D and P ) )	FF	writes all 1's

**Notes:** S = Source Data  
P = Pattern Data  
D = Data Already Existing at the Destination

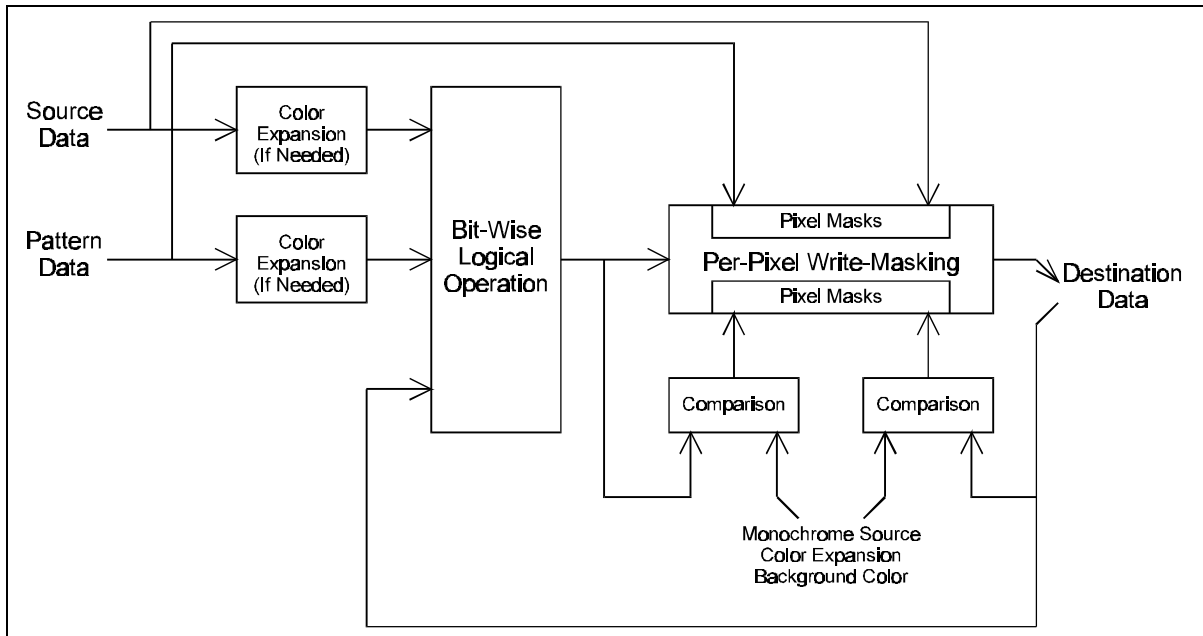
**F.2.4 PER-PIXEL WRITE-MASKING OPERATIONS**

The BitBLT engine is able to perform per-pixel write-masking with various data sources used as pixel masks to constrain which pixels at the destination will actually be written to by the BitBLT engine. As shown in Figure F-2, either monochrome source or monochrome pattern data may be used as pixel masks — neither color source nor color pattern data can be used. Another available pixel mask is derived by comparing a particular color to either the color already specified for a given pixel at the destination or the color that results from the bit-wise operation performed on the data received for a given pixel.

Bits 13 and 17 of the BitBLT Control Register (BR04) are used to select either the monochrome source or the monochrome pattern data as a pixel mask. When this feature is used, the bits in either the monochrome source or the monochrome pattern data that carry a value of 0 cause the bytes of the corresponding pixel at the destination to not be written to by the BitBLT engine, thereby preserving whatever data was originally carried

within those bytes. This feature can be used in writing characters to the display, while also preserving the pre-existing backgrounds behind those characters.

Bits 14 through 16 of the BitBLT Control Register (BR04) can be set to select per-pixel write-masking with a mask based on the results of one of four possible color comparisons. Bit 14 is used to enable this form of write-masking. Bit 15 chooses between two different comparisons of color values. Depending on the setting of bit 15, a comparison is made between the background color specified for use in the color expansion of monochrome source data and either the color already described by the bytes for the pixels at the destination or the color resulting from the bit-wise operation being performed during the BitBLT operation. Bit 16 chooses whether the bytes at the destination will be overwritten when the two compared values chosen by bit 15 are found to be equal or when they are found not to be equal.



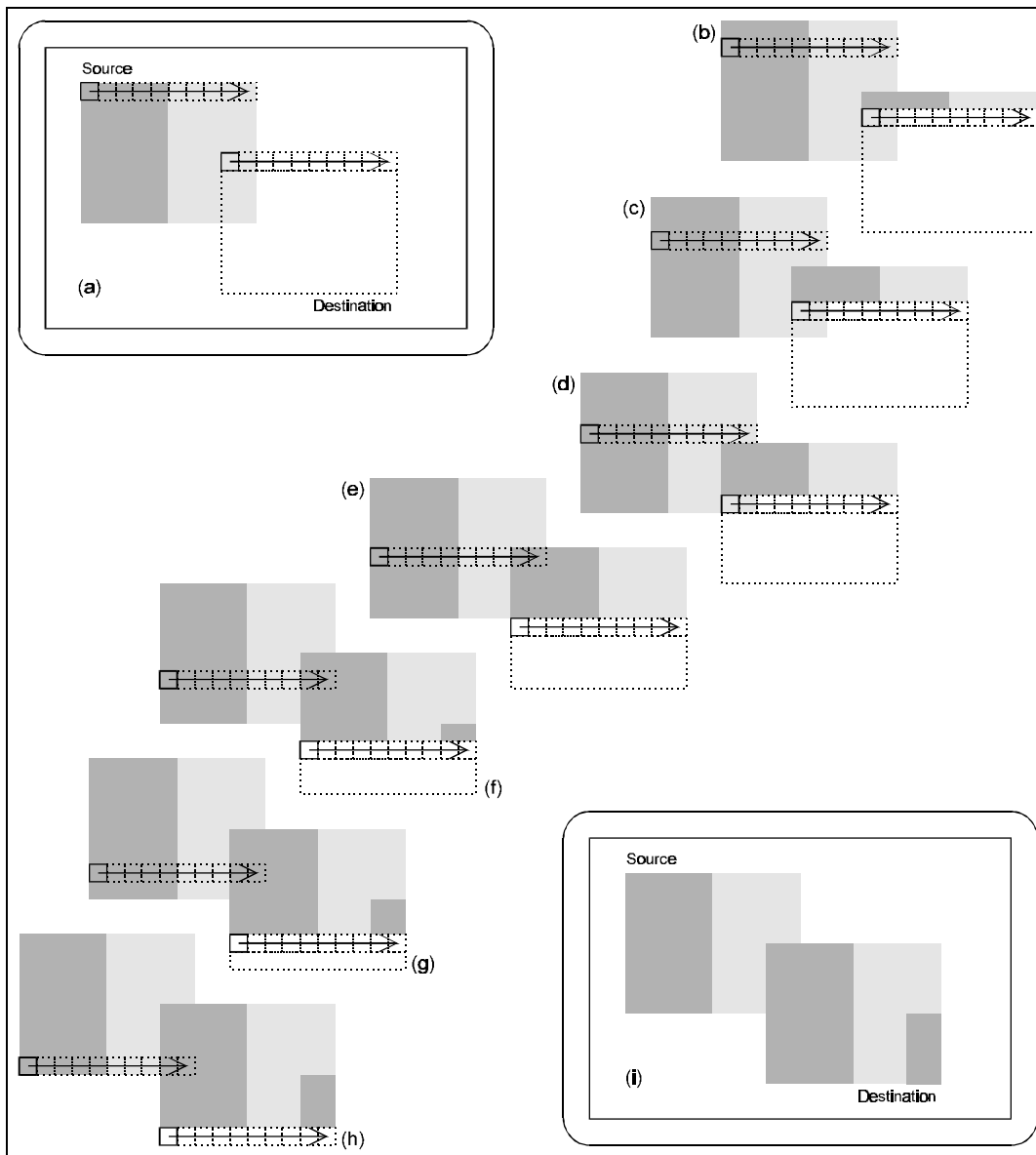
**Figure F-2: Block Diagram and Data Paths of the BitBLT Engine**

**F.2.5 WHEN THE SOURCE AND DESTINATION LOCATIONS OVERLAP**

When the source and destination locations are within the frame buffer, it is possible to have BitBLT operations in which those locations overlap. This frequently occurs in BitBLT operations where a user is shifting the position of a graphical item on the display by only a few pixels. In these situations, the BitBLT engine must be programmed so that destination data is not written into destination locations that overlap with source

locations before the source data at those locations has been read. Otherwise, the source data will become corrupted.

Figure F-3 shows how the source data can be corrupted when a rectangular block is copied from a source location to an overlapping destination location.



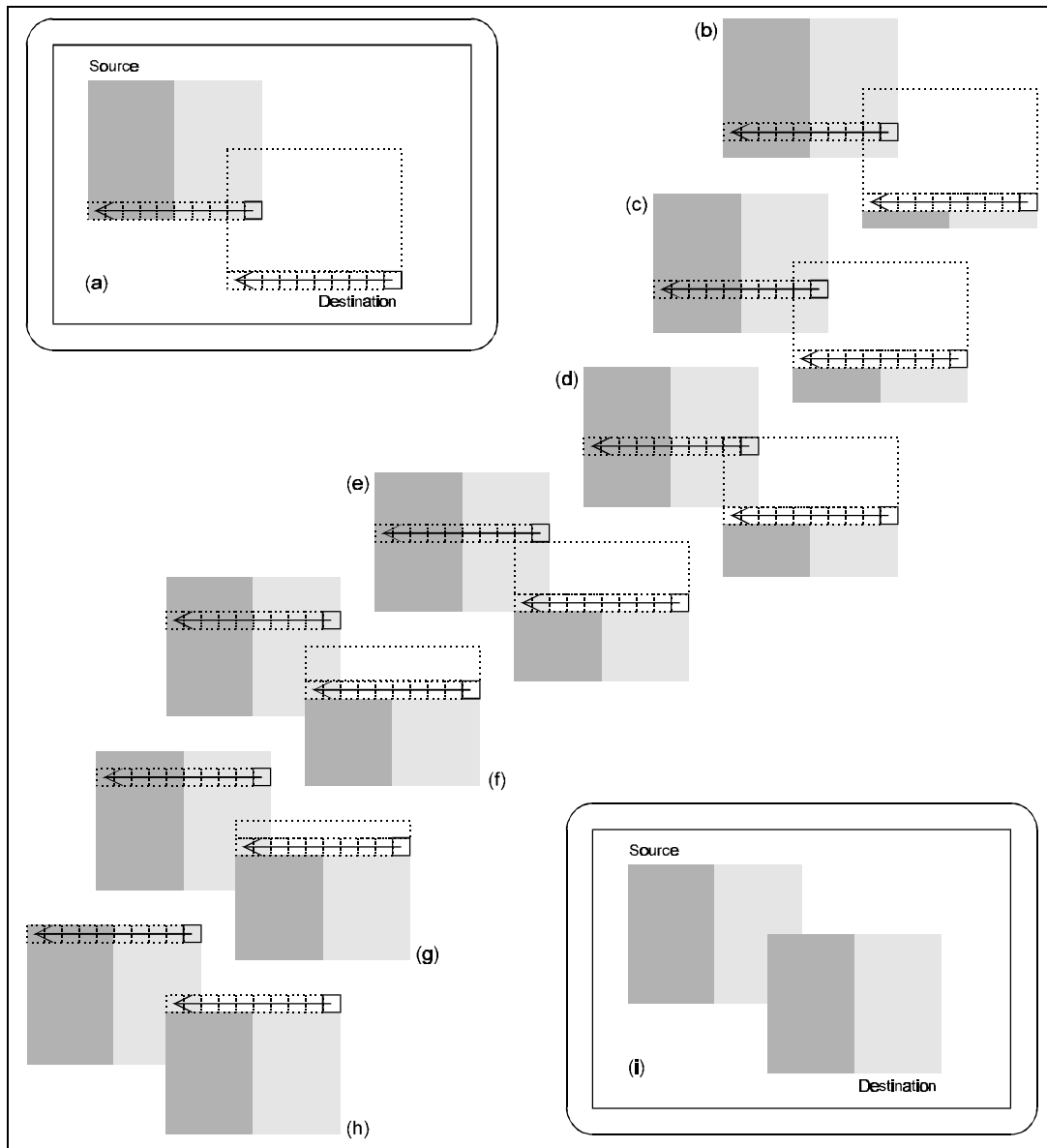
**Figure F-3: Source Corruption in BitBLT with Overlapping Source and Destination Locations**

The BitBLT engine reads from the source location and writes to the destination location starting with the left-most pixel in the top-most line of both, as shown in step (a). As shown in step (b), corruption of the source data has already started with the copying of the top-most line in step (a) — part of the source that originally contained lighter-colored pixels has now been overwritten with darker-colored pixels. More source data corruption occurs as steps (b) through (d) are performed. At step (e), another line of the source data is read, but the two right-most pixels of this line are in the region where the source and destination locations overlap, and where the source has already been overwritten as a result of the copying of the top-most line in step (a). Starting in step (f), darker-colored pixels can be seen in the destination where lighter-colored pixels should be. This errant effect occurs repeatedly throughout the remaining steps in this BitBLT operation. As more lines are copied from the source location to the

destination location, it becomes clear that the end result is not what was originally intended.

The BitBLT engine can be programmed to alter the order in which source data is read and destination data is written when necessary to avoid source data corruption problems when the source and destination locations overlap. Bits 8 and 9 of the BitBLT Control Register (BR04) provide the ability to change the point at which the BitBLT engine begins reading and writing data from the upper left-hand corner (the usual starting point) to one of the other three corners. In other words, through the use of these two bits, the BitBLT engine may be set to read data from the source and write it to the destination starting at any of the four corners of the panel.

Figure F-4 illustrates how this feature of the BitBLT engine can be used to perform the same BitBLT operation as was illustrated in Figure F-3, while avoiding the corruption of source data.



**Figure F-4: Correctly Performed BitBLT with Overlapping Source and Destination Locations**

As shown in Figure F-4, the BitBLT engine reads the source data and writes the data to the destination starting with the right-most pixel of the bottom-most line. By doing this, no pixel existing where the source and destination locations overlap will ever be written to before it is read from by the BitBLT engine. By the time the BitBLT operation has reached step (e) where two pixels existing where the source and destination locations overlap are about to be overwritten, the

source data for those two pixels has already been read.

Figure F-5 on the following page shows the recommended lines and pixels to be used as starting points in each of 8 possible ways in which the source and destination locations may overlap. In general, the starting point should be within the area in which the source and destination overlap.

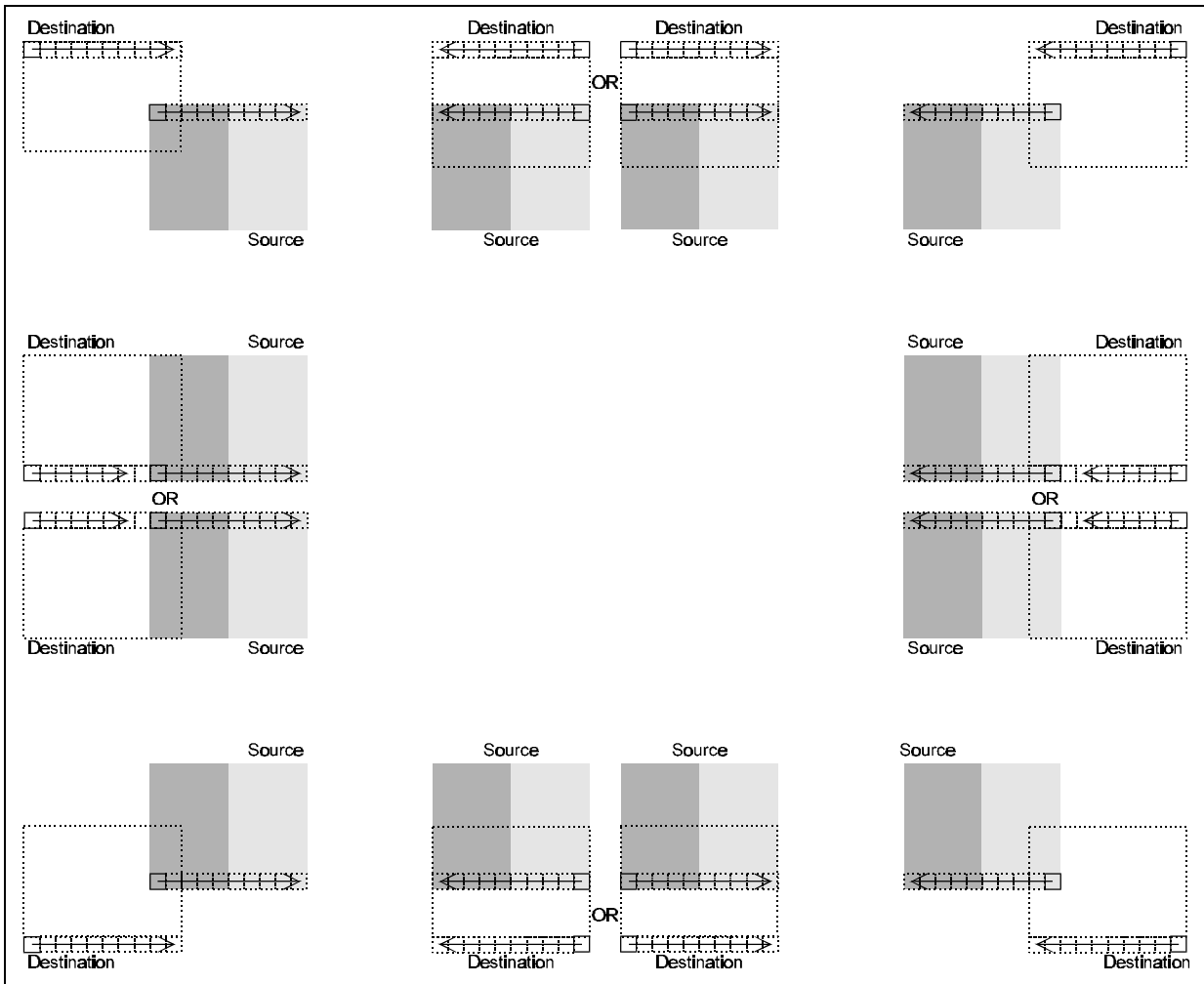


Figure F-5: Suggested Starting Points for Possible Source and Destination Overlap Situations

## F.3.0 BASIC GRAPHICS DATA CONSIDERATIONS

### F.3.1 CONTIGUOUS VS. DISCONTIGUOUS GRAPHICS DATA

Graphics data stored in memory, particularly in the frame buffer of a graphics system, has organizational characteristics that often distinguish it from other varieties of data. The main distinctive feature is the tendency for graphics data to be organized in a discontinuous block of graphics data made up of multiple sub-blocks of bytes, instead of a single contiguous block of bytes.

Figure F-6 shows an example of contiguous graphics data — a horizontal line made up of six adjacent pixels within a single scan line on a display with a resolution of 640x480. Presuming that the graphics system driving this display has been set to 8 bits per pixel, and that the frame

buffer's starting address of 0h corresponds to the upper left-most pixel of this display, then the six pixels that make this horizontal line starting at coordinates (256, 256) would occupy six bytes starting at frame buffer address 28100h, and ending at address 28105h.

In this case, there is only one scan line's worth of graphics data in this single horizontal line, so the block of graphics data for all six of these pixels exists as a single, contiguous block comprised of only these six bytes. The starting address and the number of bytes are the only pieces of information that a BitBLT engine would require to read this block of data.

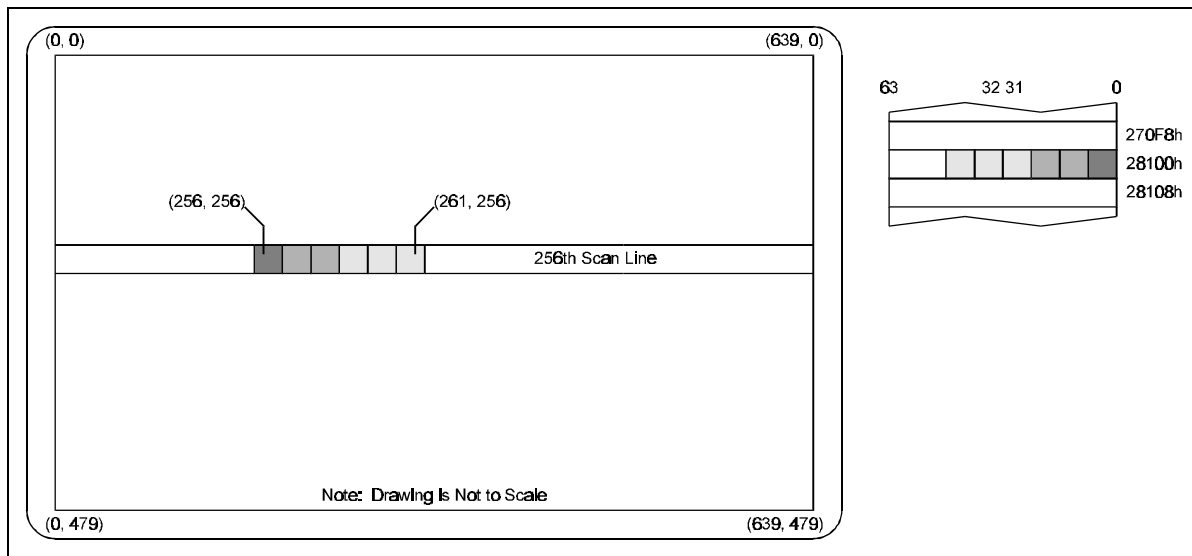


Figure F-6: On-Screen Single 6-Pixel Line in the Frame Buffer



The simplicity of the preceding example of a single horizontal line contrasts sharply to the example of discontinuous graphics data depicted in Figure F-7. The simple six-pixel line of Figure F-6 is now accompanied by three more six-pixel lines placed on subsequent scan lines, resulting in the 6x4 block of pixels shown.

Since there are other pixels on each of the scan lines on which this 6x4 block exists that are not part of this 6x4 block, what appears to be a single 6x4 block of pixels on the display must be represented by a discontinuous block of graphics data made up of 4 separate sub-blocks of six bytes apiece in the frame buffer at addresses 28100h, 28380h, 28600h, and 28880h. This situation makes the task of reading what appears to be a simple 6x4 block of pixels more complex.

Two characteristics of this 6x4 block of pixels help simplify the task of specifying the locations of all 24 bytes of this discontinuous block of graphics data. First, all four of the sub-blocks are of the same length. Second, the four sub-blocks are separated from each other at equal intervals.

The BitBLT engine was designed to make use of these characteristics of graphics data to simplify the programming required to handle discontinuous blocks of graphics data. For such a situation, the BitBLT engine requires only four pieces of information: the starting address of the first sub-block, the length of a sub-block, the offset (in bytes) of the starting address of each subsequent sub-block, and the quantity of sub-blocks.

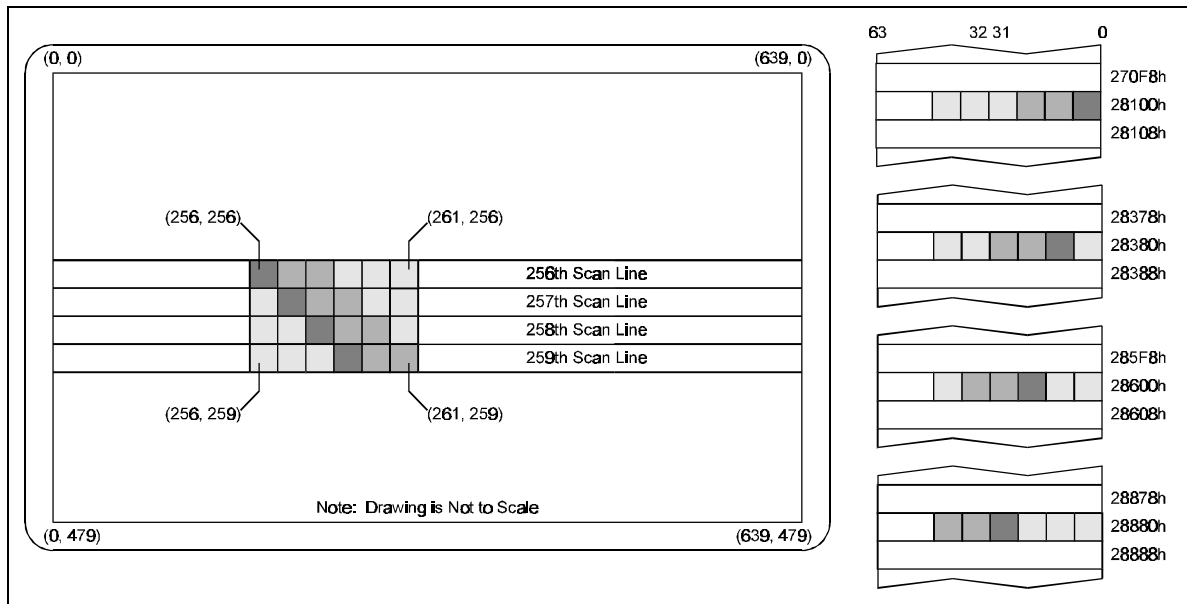


Figure F-7: On-Screen 6x4 Array of Pixels in the Frame Buffer

### F.3.2 SOURCE DATA

The source data may either exist in the frame buffer where the BitBLT engine may read it directly, or it may be provided to the BitBLT engine by the host CPU. The block of source graphics data may be either contiguous or discontinuous, and may be either in color (with a color depth that matches that to which the BitBLT engine has been set) or monochrome.

Bit 10 of the BitBLT Control Register (BR04) specifies whether the source data exists in the frame buffer or is provided by the CPU. Having the source data in the frame buffer will result in increased performance since the BitBLT engine will be able to access it directly without involving the host CPU.

If the source data resides within the frame buffer, then the Source Address Register (BR06) is used to specify the address of the source data as an offset from the beginning of the frame buffer at which the block of source data begins. However, if the host CPU provides the source data, then this register takes on a different function and the three least-significant bits of the Source Address Register (BR06) can be used to specify a number of bytes that must be skipped in the first quadword received from the host CPU to reach the first byte of valid source data.

In cases where the host CPU provides the source data, it does so by writing the source data to the

BitBLT data port, a 64KB memory space on the host bus. There is no actual memory allocated to this memory space, so any data that is written to this location cannot be read back. This memory space is simply a range of memory addresses that the BitBLT engine's address decoder watches for the occurrence of any memory writes. The BitBLT engine loads all data written to any memory address within this memory space in the order in which it is written, regardless of the specific memory address to which it is written and uses that data as the source data in the current BitBLT operation. The block of bytes sent by the host CPU to this data port must be quadword-aligned, although the source data contained within the block of bytes does not need to be aligned. As mentioned earlier, the least significant three bits of the Source Address Register (BR06) are used to specify the number of bytes that must be skipped in the first quadword to reach the first byte of valid source data.

To accommodate discontinuous source data, the Source and Destination Offset Register (BR00) can be used to specify the offset in bytes from the beginning of one scan line's worth source data to the next. Otherwise, if the source data is contiguous, then an offset equal to the length of a scan line's worth of source data should be specified.

### F.3.3 MONOCHROME SOURCE DATA

Bit 12 of the BitBLT Control Register (BR04) specifies whether the source data is color or monochrome. Since monochrome graphics data only uses one bit per pixel, each byte of monochrome source data typically carries data for 8 pixels which hinders the use of byte-oriented parameters when specifying the location and size of valid source data. Some additional parameters must be specified to ensure the proper reading and use of monochrome source data by the BitBLT engine. The BitBLT engine also provides additional options for the manipulation of monochrome source data versus color source data.

The various bit-wise logical operations and per-pixel write-masking operations were designed to work with color data. In order to use monochrome data, the BitBLT engine converts it into color through a process called color expansion, which takes place as a BitBLT operation is performed. In color expansion, the single bits of monochrome source data are converted into one, two, or three bytes (depending on the color depth to which the BitBLT engine has been set) of color data that are set to carry value corresponding to either the foreground or background color that have been specified for use in this conversion process. If a given bit of monochrome source data carries a value of 1, then the byte(s) of color data resulting from the conversion process will be set to carry the value of the foreground color. If a given bit of monochrome source data carries a value of 0, then the resulting byte(s) will be set to the value of the background color.

The foreground and background colors used in the color expansion of monochrome source data can be set in the Pattern/Source Expansion Foreground Color Register (BR02) and the Pattern/Source Expansion Background Color Register (BR01), in which case these colors will be the same colors as those used in the color expansion of monochrome

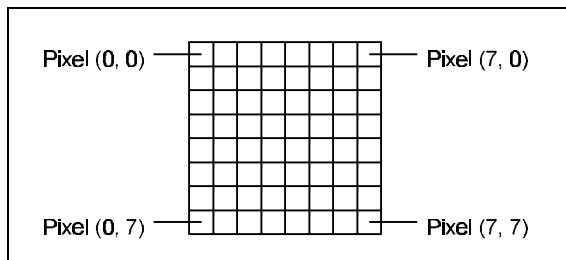
pattern data. However, it is also possible to set the colors for the color expansion of monochrome source data independently of those set for the color expansion of monochrome pattern data by using the Source Expansion Foreground Color Register (BR0A) and the Source Expansion Background Color Register (BR09). Bit 27 in the BitBLT Monochrome Source Control Register (BR03) is used to select between one or the other of these two sets of registers.

The BitBLT engine requires that the alignment of each scan line's worth of monochrome source data be specified. In other words, whether each scan line's worth of monochrome source data can be assumed to start on quadword, doubleword, word, or byte boundaries, or that it cannot be assumed to start on any such boundary must be specified using bits 26-24 of the Monochrome Source Control Register (BR03).

The BitBLT engine also provides various clipping options for use with monochrome source data. Bits 21-16 of the Monochrome Source Control Register (BR03) allow the BitBLT engine to be programmed to skip up to 63 of the 64 bits in the first quadword of a block of monochrome source data to reach the first bit of valid source data. Depending on the width of the block of pixels represented by the monochrome source data, this option can also be used to implement a way of clipping the monochrome source data from the top. Bits 5-0 of this register allow up to 63 of the 64 bits in the first quadword in each scan line's worth of monochrome source data to be skipped to reach the first bit of valid source data in each scan line's worth. This option can be used to implement the clipping of each scan line's worth of monochrome source data from the left. Bits 13-8 of this register provides similar functionality for clipping monochrome source data from the right.

### F.3.4 PATTERN DATA

The pattern data must exist within the frame buffer where the BitBLT engine may read it directly. The host CPU cannot provide the pattern data to the BitBLT engine. As shown in Figure F-8, the block of pattern graphics data always represents a block of 8x8 pixels. The bits or bytes of a block of pattern data may be organized in the frame buffer memory in only one of four ways, depending upon its color depth which may be 8, 16, or 24 bits per pixel (whichever matches the color depth to which the BitBLT engine has been set), or monochrome.

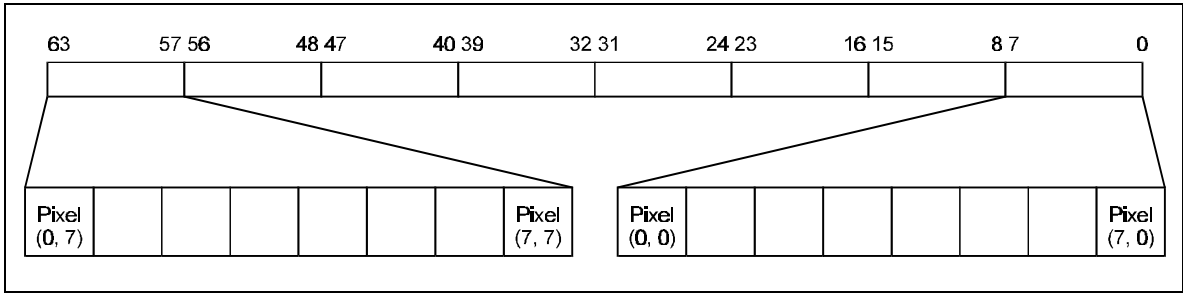


**Figure F-8: Pattern Data**

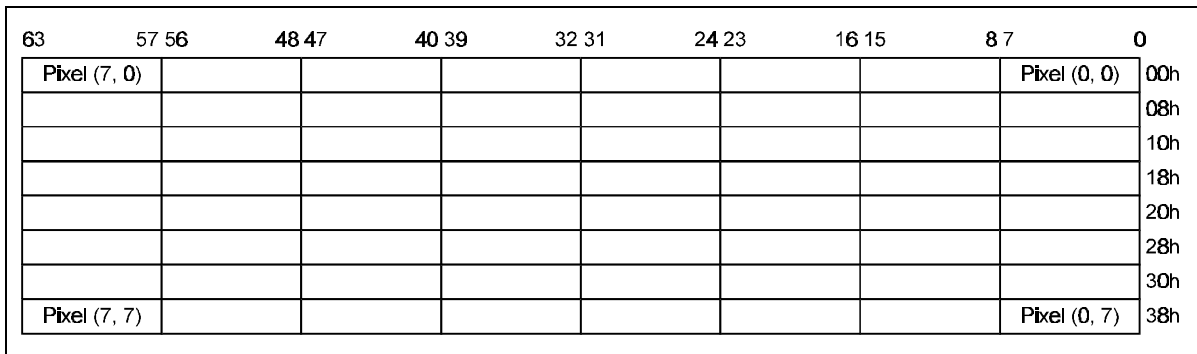
(Always an 8x8 Array of Pixels)

The Pattern Address Register (BR05) is used to specify the address of the pattern data as an offset from the beginning of the frame buffer at which the block of pattern data begins. The three least significant bits of the address written to this register are ignored, because the address must be in terms of quadwords. This is because the pattern must always be located on an address boundary equal to its size. Monochrome patterns take up 8 bytes, or a single quadword of space, and therefore, must be located on a quadword boundary. Similarly, color patterns with color depths of 8 and 16 bits per pixel must start on 64-byte and 128-byte boundaries, respectively. Color patterns with color depths of 24 bits per pixel must start on 256-byte boundaries, despite the fact that the actual color data fills only 3 bytes per pixel.

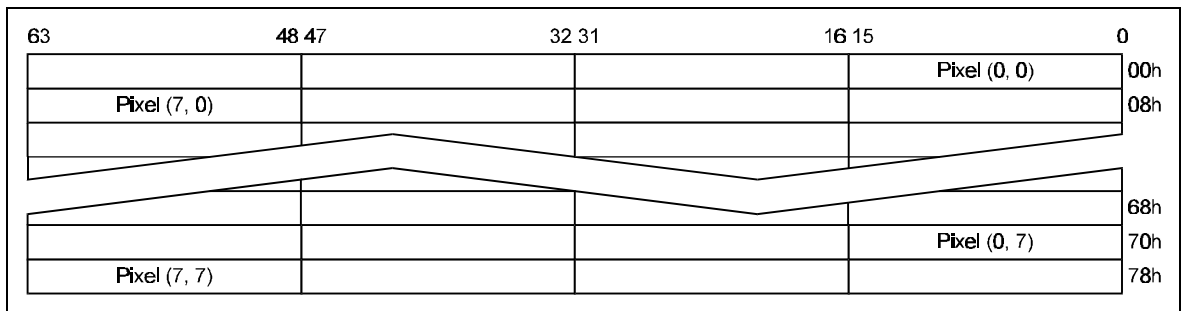
Figures F-9, F-10, F-11, and F-12 show how monochrome, 8bpp, 16bpp, and 24bpp pattern data is organized in memory.



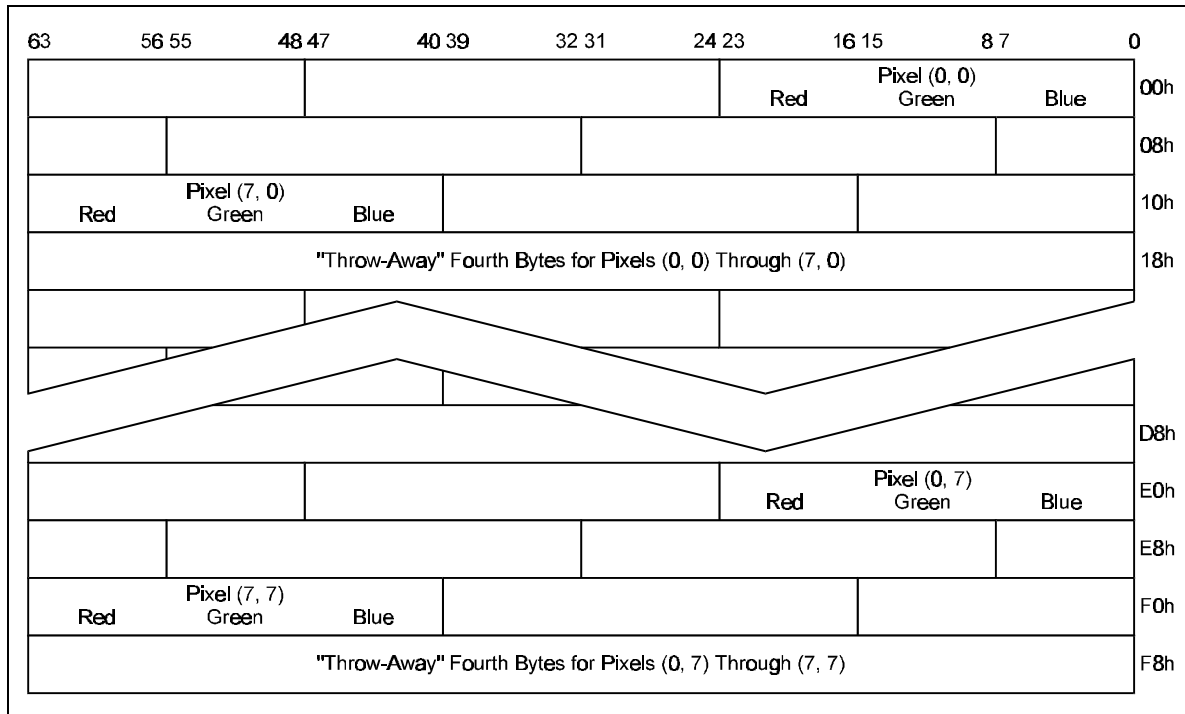
**Figure F-9: Monochrome Pattern Data -- Occupies a Single Quadword**



**Figure F-10: 8bpp Pattern Data -- Occupies 64 Bytes (8 Quadwords)**



**Figure F-11: 16bpp Pattern Data -- Occupies 128 Bytes (16 Quadwords)**



**Figure F-12: 24bpp Pattern Data -- Occupies 256 Bytes (32 Quadwords)**

As is shown in Figure F-12, there are four bytes allocated for each pixel on each scan line's worth of pattern data, which allows each scan line's worth of 24bpp pattern data to begin on a 32-byte boundary. The extra ("fourth") unused bytes of each pixel on a scan line's worth of pattern data are collected together in the last 8 bytes (the last quadword) of each scan line's worth of pattern data.

Bit 18 of the BitBLT Control Register (BR04) specifies whether the pattern data is color or monochrome. The various bit-wise logical operations and per-pixel write-masking operations were designed to work with color data. In order to use monochrome pattern data, the BitBLT engine is designed to convert it into color through a process called "color expansion" which takes place as a BitBLT operation is performed. In color expansion, the single bits of monochrome pattern data are converted into one, two, or three bytes

(depending on the color depth to which the BitBLT engine has been set) of color data that are set to carry values corresponding to either the foreground or background color that have been specified for use in this process. The foreground color is used for pixels corresponding to a bit of monochrome pattern data that carry the value of 1, while the background color is used where the corresponding bit of monochrome pattern data carries the value of 0. The foreground and background colors used in the color expansion of monochrome pattern data can be set in the Pattern/Source Expansion Foreground Color Register (BR02) and Pattern/Source Expansion Background Color Register (BR01). Depending upon the setting of bit 27 in the Monochrome Source Control Register (BR03), these same two registers may also specify the foreground and background colors to be used in the color expansion of the source data.

### F.3.5 DESTINATION DATA

If the destination is within the frame buffer, then there are actually two different types of “destination data”: the graphics data already residing at the location that is designated as the destination, and the data that is to be written into that very same location as a result of a BitBLT operation. If, however, the destination is selected so that the BitBLT engine is to provide its output to the host CPU, then the destination data provided to the host CPU is the only kind there is.

Blocks of destination data to be read from and written to the destination may be either contiguous or discontinuous. All data written to the destination will have the color depth to which the BitBLT engine has been set. It is presumed that any data already existing at the destination which will be read by the BitBLT engine will also be of this same color depth — the BitBLT engine neither reads nor writes monochrome destination data.

Bit 11 of the BitBLT Control Register (BR04) is used to specify whether the destination data is to be written to a location within the frame buffer, or is to be provided to the host CPU. If the destination is within the frame buffer, the Destination Address Register (BR07) is used to specify the address of the destination as an offset from the beginning of the frame buffer at which the destination location begins. Otherwise, only bits 2-0 of the Destination Address Register (BR07) are used, and their purpose is to specify which byte in the first quadword of destination data provided to the host CPU is the first byte of actual destination data.

To accommodate discontinuous destination data, the Source and Destination Offset Register (BR00) can be used to specify the offset in bytes from the beginning of one scan line’s worth of destination data to the next. Otherwise, if the destination data is contiguous, then an offset equal to the length of a scan line’s worth of destination data should be specified.

## F.4.0 BITBLT PROGRAMMING EXAMPLES

### F.4.1 PATTERN FILL -- A VERY SIMPLE BITBLT

In this example, a rectangular area on the screen is to be filled with a color pattern stored as pattern data in off-screen memory. The screen has a resolution of 1024x768 and the graphics system has been set to a color depth of 8 bits per pixel.

As shown in Figure F-13, the rectangular area to be filled has its upper left-hand corner at coordinates (128, 128) and its lower right-hand

corner at coordinates (191, 191). These coordinates define a rectangle covering 64 scan lines, each scan line's worth of which is 64 pixels in length — in other words, an array of 64x64 pixels. Presuming that the pixel at coordinates (0, 0) corresponds to the byte at address 00h in the frame buffer memory, the pixel at (128, 128) corresponds to the byte at address 20080h.

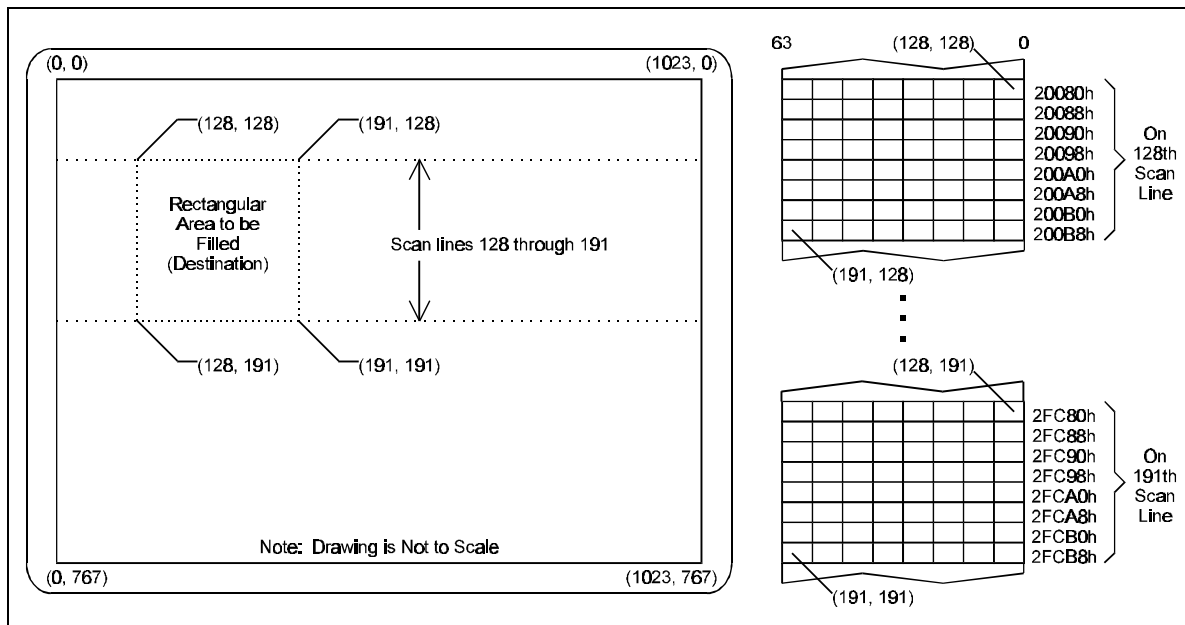


Figure F-13: On-Screen Destination for Example Pattern Fill BitBLT



As shown in Figure F-14, the pattern data occupies 64 bytes starting at address 100000h. As always, the pattern data represents an 8x8 array of pixels.

Before programming the BitBLT engine in any way, bit 0 of the BitBLT Configuration Register (XR20) or bit 31 of the BitBLT Control Register (BR04) should be checked to see if the BitBLT engine is currently busy. The BitBLT engine should not be programmed in any way until all BitBLT operations are complete and the BitBLT engine is idle. Once the BitBLT engine is idle, programming the BitBLT engine for the operation in this example should begin by making sure that the BitBLT Configuration Register (XR20) is set to 00h, in order to specify a color depth of 8 bits per pixel and enable normal operation. Alternatively, if bit 23 of the BitBLT Control Register (BR04) is set to 1, then the color depth of the BitBLT engine may be set to 8 bits per pixel by setting bits 25 and 24 of the same register to 0, although it is still necessary to ensure that at least bit 1 of the BitBLT Configuration Register is set to 0 to enable normal operation.

The BitBLT Control Register (BR04) is used to select the features to be used in this BitBLT operation, and must be programmed carefully. Bits 22-20 should be set to 0 to select the top-most horizontal row of the pattern as the starting row used in drawing the pattern starting with the top-most scan line covered by the destination. Since actual pattern data will be used, bit 19 should be set to 0. The pattern data is in color with a color depth of 8 bits per pixel, so bits 18 and 17 should also be set to 0. Since this BitBLT operation does not use per-pixel write-masking, bits 16-13 should be set to 0. Bit 12 should be set to 0 to ensure that the settings in the Monochrome Source Control Register (BR03) will have no effect on this BitBLT operation. Bit 11 should be set to 0 to configure

the BitBLT engine for a destination within the frame buffer. The setting of bits 10-8 do not affect this BitBLT operation, since source data is not used. Therefore, these bits might as well be set to zero as a default. Finally, bits 7-0 should be programmed with the 8-bit value of F0h to select the bit-wise logical operation in which a simple copy of the pattern data to the destination takes place. Selecting this bit-wise operation in which no source data is used as an input causes the BitBLT engine to automatically forego either reading source data from the frame buffer or waiting for the host CPU to provide it.

Bits 28-16 of the Source and Destination Offset Register (BR00) must be programmed with number of bytes in the interval from the start of one scan line's worth of destination data to the next. Since the color depth is 8 bits per pixel and the horizontal resolution of the display is 1024, the value to be programmed into these bits is 400h, which is equal to the decimal value of 1024. Since this BitBLT operation does not use source data, the BitBLT engine ignores bits 12-0.

Bits 22-3 of the Pattern Address Register (BR05) must be programmed with the address of the pattern data. This address is specified as an offset from the beginning of the frame buffer where the pattern data begins. In this case, the address is 100000h.

Similarly, bits 22-0 of the Destination Address Register (BR07) must be programmed with the address of the destination, i.e., the offset from the beginning of the frame buffer of the byte at the destination that will be written to first. In this case, the address is 20080h, which corresponds to the byte representing the pixel at coordinates (128, 128).

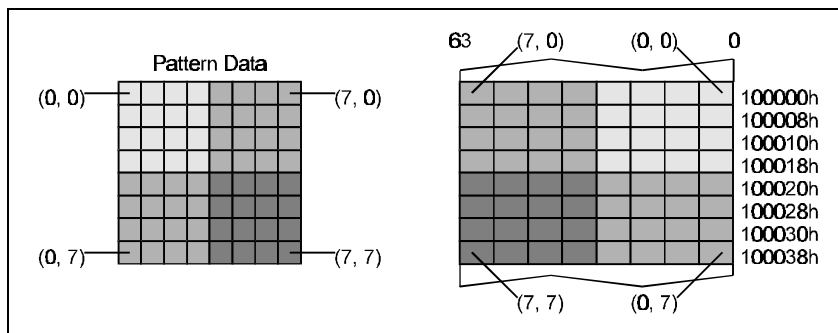


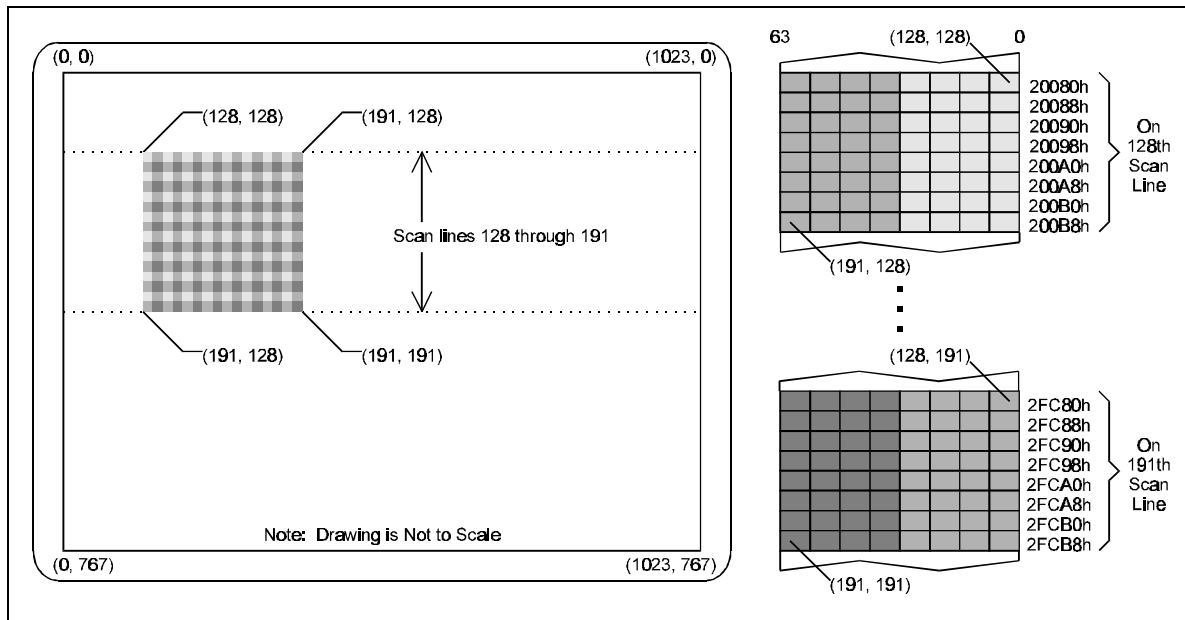
Figure F-14: Pattern Data for Example Pattern Fill BitBLT

This BitBLT operation does not use the values in the Pattern/Source Expansion Background Color Register (BR01), the Pattern/Source Expansion Foreground Color Register (BR02), the Monochrome Source Control Register (BR03), the Source Address Register (BR06), the Source Expansion Background Color Register (BR09), or the Source Expansion Foreground Color Register (BR0A).

The Destination Width and Height Register (BR08) must be programmed with values that describe to the BitBLT engine the 64x64 pixel size of the destination location. Bits 28-16 should be set to carry the value of 40h, indicating that the

destination location covers 64 scan lines. Bits 12-0 should be set to carry the value of 40h, indicating that each scan line's worth of destination data occupies 64 bytes. The act of writing a non-zero value for the height to the Destination Width and Height Register (BR08) is what signals the BitBLT engine to begin performing this BitBLT operation. Therefore, it is important that all other programming of the BitBLT registers be completed before this is done.

Figure F-15 shows the end result of performing this BitBLT operation. The 8x8 pattern has been repeatedly copied ("tiled") into the entire 64x64 area at the destination.

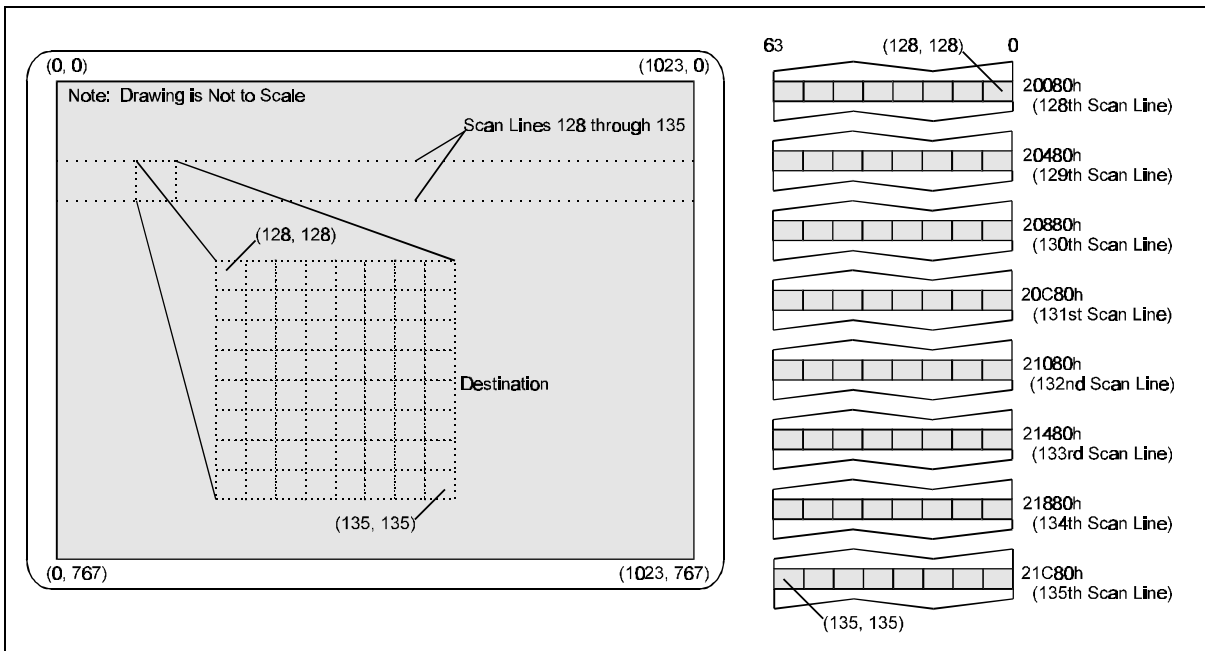


**Figure F-15: Results of Example Pattern Fill BitBLT**

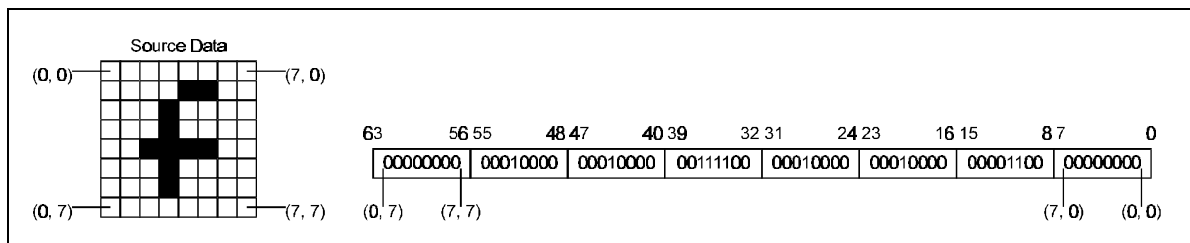
**F.4.2 DRAWING CHARACTERS USING A FONT STORED IN SYSTEM MEMORY**

In this example BitBLT operation, a lowercase letter “f” is to be drawn in black on a display with a gray background. The resolution of the display is 1024x768, and the graphics system has been set to a color depth of 8 bits per pixel. Figure F-16 shows the display on which this letter “f” is to be drawn. As shown in this figure, the entire display is shaded gray. The letter “f” is to be drawn into an 8x8 region on the display with the upper left-hand corner at the coordinates (128, 128).

Figure F-17 shows both the 8x8 pattern making up the letter “f” and how it is represented somewhere in the host’s system memory — the actual address in system memory is not important. The letter “f” is represented in system memory by a block of monochrome graphics data that occupies 8 bytes. Each byte carries the 8 bits needed to represent the 8 pixels in each scan line’s worth of this graphics data. This type of pattern is often used to store character fonts in system memory.



**Figure F-16: On-Screen Destination for Example Character Drawing BitBLT**



**Figure F-17: Source Data in System Memory for Example Character Drawing BitBLT**

During this BitBLT operation, the host CPU will read this representation of the letter “f” from system memory, and write it to the BitBLT engine by performing memory writes to the BitBLT data port. The BitBLT engine will receive this data from the host CPU and use it as the source data for this BitBLT operation. The BitBLT engine will be set to the same color depth as the graphics system — 8 bits per pixel, in this case. Since the source data in this BitBLT operation is monochrome, color expansion must be used to convert it to an 8 bpp color depth. To ensure that the gray background behind this letter “f” is preserved, per-pixel write masking will be performed, using the monochrome source data as the pixel mask.

As in the example of the pattern fill BitBLT operation, the first step before programming the BitBLT engine in any way is to check either bit 0 of the BitBLT Configuration Register (XR20) or bit 31 of the BitBLT Control Register (BR04) to see if the BitBLT engine is currently busy. After waiting until the BitBLT engine is idle, programming the BitBLT engine should begin by making sure that the BitBLT Configuration Register (XR20) is set to 00h, to specify a color depth of 8 bits per pixel and to enable normal operation. Alternatively, if bit 23 of the BitBLT Control Register (BR04) is set to 1, then the color depth of the BitBLT engine may be set to 8 bits per pixel by setting bits 25 and 24 of the same register to 0, although it is still necessary to ensure that at least bit 1 of the BitBLT Configuration Register is set to 0 to enable normal operation.

The BitBLT Control Register (BR04) is used to select the features to be used in this BitBLT operation. Since pattern data is not required for this operation, the BitBLT engine will ignore bits 22-17, however as a default, these bits can be set to 0. Since monochrome source data will be used as the pixel mask for the per-pixel write-masking operation used in this BitBLT operation, bits 16-14 must be set to 0, while bit 13 should be set to 1. Bit 12 should be set to 1, to specify that the data source is monochrome. Bit 11 should be set to 0 to configure the BitBLT engine for a destination within the frame buffer. Bit 10 should be set to 1, to indicate that the source data will be provided by the host CPU. Presuming that the host CPU will provide the source data starting with the byte that carries the left-most pixel on the top-most scan line’s worth of the source data, bits 9 and 8 should

both be set to 0. Finally, bits 7-0 should be programmed with the 8-bit value CCh to select the bit-wise logical operation that simply copies the source data to the destination. Selecting this bit-wise operation in which no pattern data is used as an input, causes the BitBLT engine to automatically forego reading pattern data from the frame buffer.

Unlike the earlier example of a pattern fill BitBLT operation where the Monochrome Source Control Register (BR03) was entirely ignored, several features of this register will be used in this BitBLT operation. Bit 27 of this register will be set to 0, thereby selecting the Pattern/Source Expansion Foreground Color Register (BR02) to specify the color with which the letter “f” will be drawn. This example assumes that the source data will be sent in one quadword that will be quadword-aligned. Therefore, bits 26, 25, and 24, which specify alignment should be set to 1, 0, and 1, respectively. Since clipping will not be performed in this BitBLT operation, bits 21-16, 13-8, and 5-0 should all be set to 0.

Bits 28-16 of the Source and Destination Offset Register (BR00) must be programmed with a value equal to number of bytes in the interval between the first bytes of each adjacent scan line’s worth of destination data. Since the color depth is 8 bits per pixel and the horizontal resolution of the display is 1024 pixels, the value to be programmed into these bits is 400h, which is equal to the decimal value of 1024. Since the source data used in this BitBLT operation is monochrome, the BitBLT engine will not use a byte-oriented offset value for the source data. Therefore, bits 12-0 will be ignored.

Since the source data is monochrome, color expansion is required to convert it to color with a color depth of 8 bits per pixel. Since the Pattern/Source Expansion Foreground Color Register (BR02) was selected to specify the foreground color of black to be used in drawing the letter “f”, this register must be programmed with the value for that color. With the graphics system set for a color depth of 8 bits per pixel, the actual colors are specified in the RAMDAC palette, and the 8 bits stored in the frame buffer for each pixel actually specify the index used to select a color from that palette. This example assumes that the color specified at index 00h in the palette is black, and therefore bits 7-0 of this register

should be set to 00h to select black as the foreground color. The BitBLT engine ignores bits 23-8 of this register because the selected color depth is 8 bits per pixel. Even though the color expansion being performed on the source data normally requires that both the foreground and background colors be specified, the value used to specify the background color is not important in this example. Per-pixel write-masking is being performed with the monochrome source data as the pixel mask, which means that none of the pixels in the source data that will be converted to the background color will ever be written to the destination. Since these pixels will never be seen, the value programmed into the Pattern/Source Expansion Background Color Register (BR01) to specify a background color is not important.

Since the CPU is providing the source data, and this source data is monochrome, the BitBLT engine ignores all of bits 22-0 of the Source Address Register (BR06).

Bits 22-0 of the Destination Address Register (BR07) must be programmed with the address of the destination data. This address is specified as an offset from the start of the frame buffer of the pixel at the destination that will be written to first. In this case, the address is 20080h, which corresponds to the byte representing the pixel at coordinates (128, 128).

This BitBLT operation does not use the values in the Pattern Address Register (BR05), the Source Expansion Background Color Register (BR09), or the Source Expansion Foreground Color Register (BR0A).

The Destination Width and Height Register (BR08) must be programmed with values that describe to the BitBLT engine the 8x8 pixel size of the destination location. Bits 28-16 should be set to carry the value of 8h, indicating that the destination location covers 8 scan lines. Bits 12-0 should be set to carry the value of 8h, indicating that each scan line's worth of destination data occupies 8 bytes. As mentioned in the previous example, the act of writing a non-zero value for the height to the Destination Width and Height Register (BR08) provides the BitBLT engine with the signal to begin performing this BitBLT operation. Therefore, it is important that all other programming of the BitBLT engine registers be completed before this is done.

Figure F-18 shows the end result of performing this BitBLT operation. Only the pixels that form part of the actual letter "F" have been drawn into the 8x8 destination location on the display, leaving the other pixels within the destination with their original gray color.

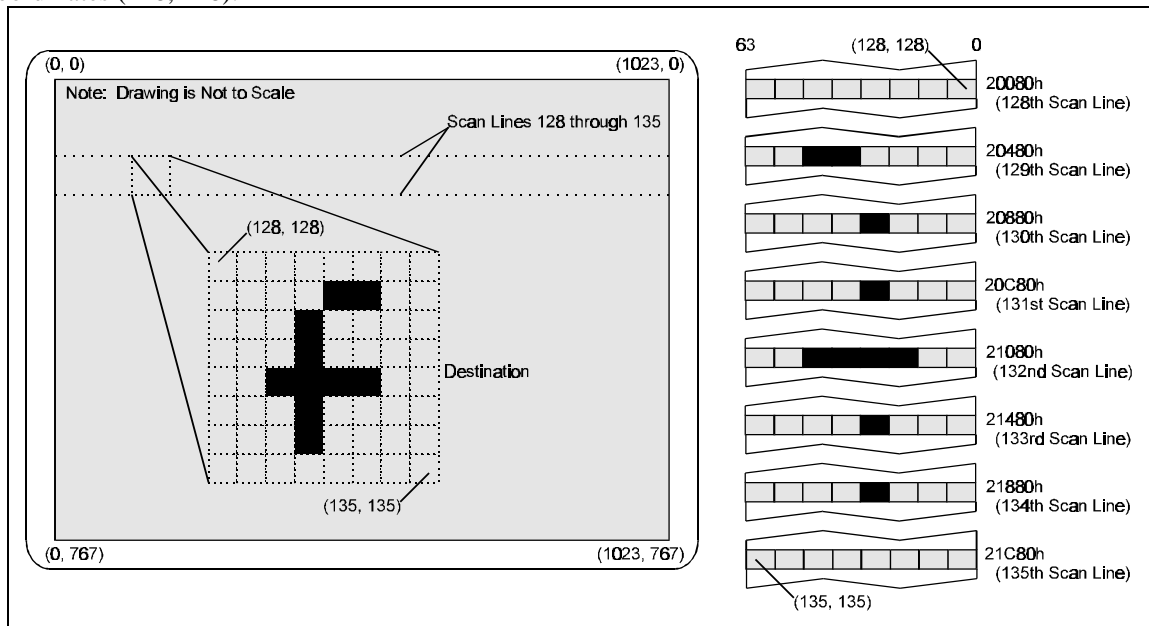


Figure F-18: Results of Example Character Drawing BitBLT

This page intentionally left blank



Chips and Technologies, Inc.  
2950 Zanker Road  
San Jose, California 95134  
Phone: 408-434-0600  
FAX: 408-894-2080

Title: 65555 HiQVPro™ High Performance  
Flat Panel / CRT GUI Controllers  
Publication No.: DS179.1  
Stock No.: 010-179-001  
Revision No.1.1  
Date: 09/19/97