ProMotion®-AT3D

Integrated 3D graphics
MultiMedia User Interface
Accelerator

Technical manual

PRELIMINARY

NDA Confidential

ALLIANCE SEMICONDUCTOR

Document ID 11-50002 Rev. **-06. Printed 6/97.

Document ID 11-50002 Rev. **-06. Printed 6/97.

Manual researched and written by M. Gadzikowski and M. Shanmugasundaram. Contributors include S. Greene, A. Daniel, G. Tsztoo, J. Washburn, N. Taneja, C. Nguyen, V. Reddy, B. Schnaitter. L. Poitras.

Table of contents

Preface

Statement of intent

Alliance provides this document for development partners, not as an invitation for reverse engineering Alliance proprietary and confidential information.

Intended audience

Material presented in Section 1 through Section 9 of this manual appears in the *ProMotion-AT3D Databook*. This material, intended for the video board (hardware) developer, has been updated to reflect changes made since that databook went to press, including addition of a NAND tree as Section 10. Section 11 through Section 17 contain material intended for software driver developers, providing an introduction to ProMotion® features. Section 15 through Section 16 provide a detailed reference to the ProMotion register set. Section 18 is an appendix including a glossary and late breaking material.

Reserved registers and bits

To prevent unexpected operation, all reserved register bits should be written with 0s and masked off if read back. Future compatability is jeopardized if this procedure is not followed.

Writing conventions

Register addresses and indices appear in hexadecimal; bits are indicated in decimal.

XXX.YY	indicates an I/O mapped index/data accessible register with index at XXXh and index value of YYh.
Mnnn	indicates memory mapped offset nnn (hex) from register base address.
[ZZ:zz]	indicates bits ZZh through zzh.

Critical notes and important warnings are set apart from other material with horizontal lines and have this symbol in the margin.

Hints and tips have this symbol in the margin.

NDA Confidential information

Material included in this manual includes confidential information which could compromise Alliance's position if disclosed. Refer to your non-disclosure agreement with Alliance Semiconductor Corporation for further details of fair use.

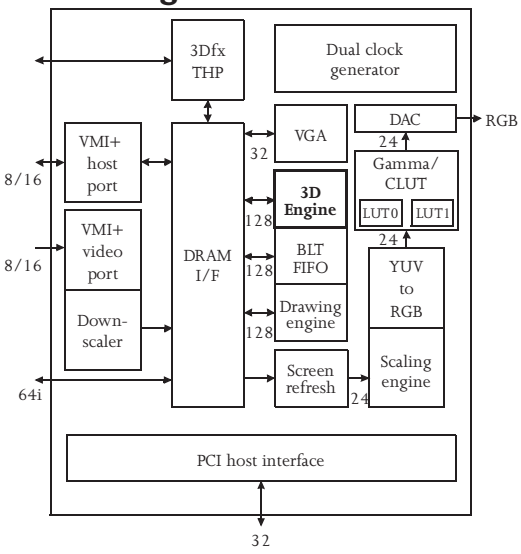
Do not show this manual to anyone without an NDA from Alliance.

1. Introduction

Features

- Fast 3D rendering
  - Full hardware setup
  - On-chip D3D texture palette
  - Fast DMA palette load
  - True divide-per-pixel perspective
  - Modulated texture mapping
  - Full lighting and fog
  - Full and compressed Z-buffer support
  - Precomputed MIPMap filtering
  - Gouraud shading
- Advanced upgrade port
  - Glueless 3*Dfx Interactive* THP interface
  - Upgrade to arcade quality
- Fast 128-bit GUI engine
  - Optimized 24- and 32-bit truecolor
  - Source and destination transparency
- No-cost motion video™
  - Smooth scaling and color conversion
  - 64 step bi-linear filter with full line buffer
  - DirectDraw/DirectVideo/DCI support
  - Multiple independent video windows
  - Hardware occlusion without loss of quality
- VMI compatible TV/video port
  - Glueless 7110/7111 support
  - Hardware MPEG support
  - Interpolated downscaling filter
- Integrated 175 MHz DAC & clockgen
  - Programmable HW gamma & color correction
  - Full 256×24-bit CLUT RAM
- VESA® standards: VAFC, DPMS, DDC 2.0B
- High throughput PCI v2.1 interface
- Programmable bi-endian support
- Flexible EDO DRAM interface
  - 1, 1.5, 2, 3, 4 MB configurations
  - Single-cycle (1–2 MB) and interleaved (3–4 MB)
- Programmable resolution to 1600×1200

Block diagram



Overview

ProMotion-AT3D is a high-performance integrated 3D, 2D, and video accelerator. It incorporates an advanced 3D rendering engine with a powerful Windows graphical user interface accelerator engine, unique motion video acceleration hardware and a high-precision DAC + clock generator, all in a single integrated 208-pin PQFP package. The AT3D is fully pin-compatible with previous-generation ProMotion-6422 and AT24 controllers.

The chip's 128-bit internal architecture and ultra high performance memory interface give the AT3D superior performance in a low-cost mainstream Windows accelerator. Hardware gamma correction and a full 256∞24-bit CLUT ensure optimum color quality.

ProMotion-AT3D acts as the central media hub in a feature rich multimedia subsystem. AT3D's video input port implements a superset of the Video Module Interface (VMI) standard. AT3D supports glueless 8/16-bit connection to live video decoders such as 7110 & 7111 with filtered downscaling, and an 8/16-bit host port connects to ISA or Motorola style devices, including audio, MPEG, and video-conferencing codecs. A hardware scaler with bilinear filter and full line buffer smoothly scales playback or capture windows from native size up to full screen at full speed.

ProMotion-AT3D also drives a proprietary 'THP' upgrade port jointly developed with 3Dfx Interactive, Inc., for high-performance seamless upgradability to arcade-quality 3D hardware acceleration.

Software drivers and BIOS

- ProMotion Director's Chair™ for Windows 95™
  - Direct3D, DirectDraw™ & DirectVideo™
- Windows NT™ 4.0, 3.5x
  - Direct3D, DirectDraw™
  - OpenGL MCD
- Major 3D APIs
  - 3Dfx interactive GLIDE™
  - Argonaut Brender
  - Criterion Renderware
- ProMotion Director's Chair™ for Windows™ 3.X
  - Display control interface (DCI)
  - Resolution switching on the fly
  - Virtual desktop to 1600×1200
- Microsoft Video for Windows
- AutoDesk® ADI
- WordPerfect® 6.0
- OS/2™ Warp, 2.11
- SCO Open Desktop™
- Linux
- Industry standard Phoenix® VGA BIOS
  - VESA DPMS power management
  - DDC 2.0B
  - VESA BIOS extensions

⇨ Complete, high-performance, robust

Alliance supports the ProMotion family with high-quality flat-model optimized driver software. ProMotion drivers take full advantage of ProMotion-AT3D hardware and the latest software technology to accelerate real performance of real applications, from word processing and spreadsheets to the most demanding CAD programs and multimedia software.

The same driver set supports all register-compatible ProMotion controllers: 32xx, 64xx, ATxx, EDxx.

The ProMotion driver set accelerates all major operating environments, graphics-intensive software, and motion video applications. With 100% VGA and VESA compatibility, ProMotion controllers can also run standard DOS and VBE-compatible applications directly without driver software.

Source code for ProMotion drivers and BIOS is available to permit customization and differentiation.

Flexible memory interface

	256 colors	32K/64K colors	16M colors
1 M	1152×864	800×600	640×480
2 M	1600×1200	1152×864	800×600
4 M	1600×1200	1600×1200	1280×1024

ProMotion's optimized memory interface delivers high-quality non-interlaced truecolor display at up to 1280×1024 resolution, using economical EDO DRAM.

Manufacturing package

- Reference PCB designs
- OEM software utilities
- Customer software utilities

⇨ *Full customer support*

ProMotion reference designs, OEM tools, and application notes reduce time-to-market. Alliance's OEM support and quality standards, developed over years as a high-volume system supplier to the PC industry, meet the strictest requirements.

3Dfx Interactive THP interface

- Voodoo Graphics 'VG-96' module
  - 40+ MPixel/second fill rate
  - Bilinear filtered and advanced filtered textures
  - Polygon anti-aliasing
  - Alpha blending
  - Programmable fog table
  - GLIDE™ low-level software API
  - Strong developer support

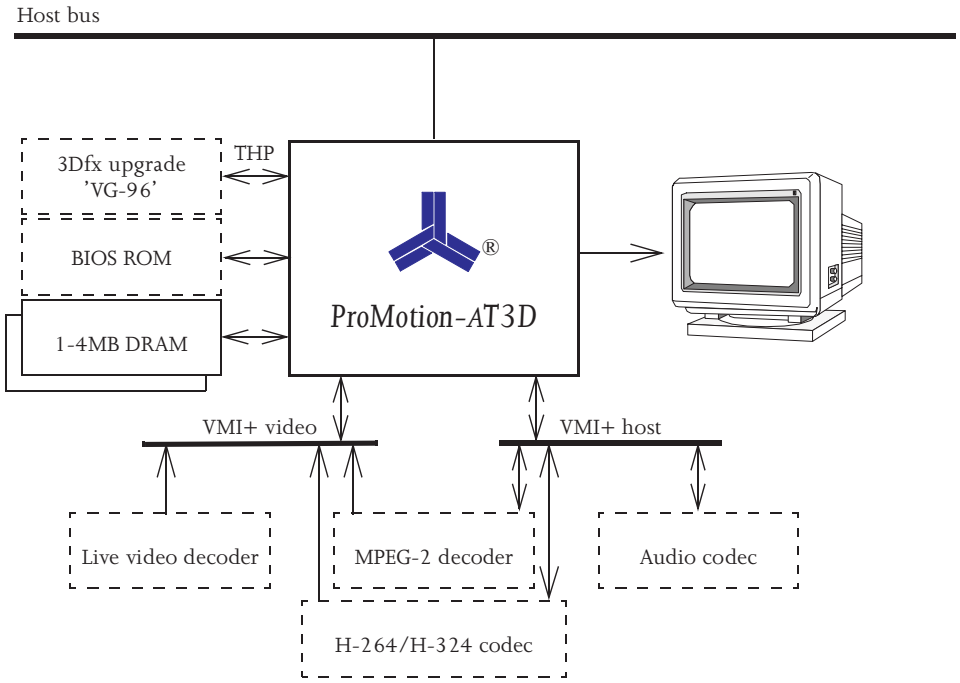
⇨ *Upgrade path from mass-market to leading-edge*

Thanks to Alliance Semiconductor's unique partnership with 3Dfx Interactive, Inc., the ProMotion accelerator family offers unmatched flexibility and upgradability. ProMotion-AT3D's proprietary 'THP' upgrade port, jointly developed by Alliance and 3Dfx, supports glueless interface to the 3Dfx 'VG-96' chipset, for super-advanced PC rendering performance and special effects.

Motherboards and adapters designed with ProMotion-AT3D and outfitted with the inexpensive THP upgrade connector can be upgraded using a daughtercard module based on the VG-96 chipset. Direct3D™ and GLIDE™ based software titles for AT3D can transparently take advantage of VG-96 capabilities.

The VG-96 upgrade represents just one possible use of the THP port. THP is an extensible platform suitable for new Alliance and 3Dfx offerings, ensuring that manufacturers' designs and users' systems will not face early obsolescence.

System block diagram



2. Functional description

2.1 3D rendering accelerator

ProMotion-AT3D features a high speed **3D rendering engine** to accelerate texture-mapped 3D polygons. The AT3D rendering engine is designed for speed to make game development easy. High triangle and fill rate performance enable all supported features and effects across the 640×480×16 display, at 30 frames per second and higher. Thanks to its high speed rendering engine, ProMotion-AT3D enables developers to focus on content rather than the micro-optimizations required by lower-performance hardware.

Fully perspective correct, ProMotion-AT3D texture mapping implements true **divide-per-pixel perspective** with no performance penalty. The on-chip **setup engine** computes edge deltas, off-loading expensive divides from the host CPU, and—more importantly—reducing PCI traffic to less than 40 bytes per vertex. **Hardware tri-strip support** further reduces PCI traffic for common triangle mesh constructs.

An on-chip **texture lookup table** (TLUT), separate from the DAC CLUT, is optimized for Direct3D and other indexed texture environments, reducing texture memory requirements to 4 or 8 bits/texel. Off-screen **palette caching** with a fast palette load instruction permits a separate full 256-entry palette per texture, or even per polygon. In texture mapping, the indexed texel passes through the TLUT for conversion to a direct color value, which then passes through interpolated lighting modulation and fog blending stages before being written to the back buffer. TLUT bypass allows for **RGB textures** as well.

Hardware MIPMapping allows software to store pre-filtered textures, avoiding the substantial performance hit of on-the-fly filtering. Texture memory for multiple map levels comes from the savings associated with palettized textures.

A **Z-buffer stage** in the rendering pipeline provides accurate hidden surface removal. Proprietary techniques permit Z-buffering in as little as 8KB off screen memory, leaving more memory for richer textures.

2.2 THP coprocessor interface

ProMotion's proprietary **THP coprocessor interface** jointly developed with 3Dfx Interactive, Inc., provides a high-bandwidth control and data interface for coprocessors like the 3Dfx 'VG-96.' Refer to "Recommended 3Dfx THP interface," described on page 317.

2.3 2D graphics accelerator

The ProMotion-AT3D integrated MMUI accelerator includes a high performance **128-bit graphics accelerator** designed for demanding truecolor, hi-color, and 256-color GUI and CAD applications. An optimized **BLT engine** maximizes performance of host-to-screen and screen-to-screen operations. A separate **drawing engine** efficiently handles pattern fills, text rendering, lines and polygons. Advanced features include:

- 256 raster operations
- Color DitherFill™
- Source and destination transparency
- Programmable BLT stride
- Line draw
- Strip draw
- Quick-start and auto-update capability
- Linear memory access
- Mono-to-color expansion
- Clipping
- Hardware cursor

2.4 Motion video accelerator

An on-chip **motion video accelerator** enables software codecs to achieve 30 fps full-screen playback with no additional hardware. ProMotion-AT3D accomplishes this feat by off-loading the CPU-intensive tasks of scaling and color space conversion, and by minimizing the memory bandwidth required for display of decompressed video data.

The chip manages a hardware motion video window, the vWindow™. When displaying the vWindow, the controller stretches by programmable X and Y factors ranging from 1.01 to 255.0. High-precision **bilinear interpolation filter** and **on-chip line buffer** circuitry enhance the quality of scaled low-resolution images. **Hardware occlusion support** in up to 1280∞1024 means full quality video even in the most demanding system resolutions.

Motion video data may be in pseudo-color, RGB, or YUV format (4:2:2, 4:1:1, or 4:0:0). ProMotion-AT3D converts YUV data to RGB "on the fly" for display in photorealistic color using ProMotion's onboard DAC. The advanced ProMotion architecture permits full 24-bit color for motion video data, even when the graphics desktop uses lower color depth. With ProMotion-AT3D, even a 1MB graphics system can display 8-bit graphics up to 1024×768 resolution, along with 24-bit full-screen motion video.

The ProMotion architecture maximizes motion video performance as well. Because YUV format is more compact than truecolor RGB, and because each motion video frame is sent across the host bus at its unscaled resolution, the host sends a minimum of data across the system bus. Because ProMotion-AT3D does scaling on the fly, it reads only the minimum required data for each screen update, making the best possible use of available bandwidth. ProMotion's innovative architecture removes bandwidth bottlenecks to display multimedia data at its full speed.

2.5 VGA controller

A fully register-compatible Super VGA controller in the ProMotion-AT3D chip supports all monochrome and 4-bit packed and planar modes. Super VGA modes conform to VESA standards. Refer to Table 2.6.1 for extended modes.

Table 2.5 VGA modes

VGA mode	*Screen format*	*Supported display mode*
0, 1	360 × 400	text
2, 3	720 × 400	text
4, 5	320 × 200	graphics
6	640 × 200	graphics
7	720 × 400	text
D	320 × 200	graphics
E	640 × 200	graphics
F	640 × 350	graphics
10	640 × 350	graphics
11	640 × 480	graphics
12	640 × 480	graphics
13	320 × 200	graphics

2.6 Clock generator and DAC

ProMotion-AT3D's high-frequency clock generator and integrated palette DAC give high-quality, high-resolution display. Table 2.6.1 details ProMotion resolutions available with standard BIOS. Analog biasing circuitry appears in Figure 2.6.2 and Figure 2.6.3.

Hardware gamma correction in 16- and 24-bit modes—including **separate gamma tables** for desktop and video areas—permits software color matching and brightness/tint control.

Table 2.6.1 ProMotion–AT3D extended graphics modes

Display resolution	Bits per pixel	VESA mode (hex)	Mem. req. (MB)	Vert.freq. (Hz)	Horiz. freq. (KHz)	Pixel freq. (MHz)	VCLK freq. (MHz)
640×400	8	100	1.0	70	31.5	25.175	25.175
	32		1.0	70	31.5	25.175	25.175
640×480	4		1.0	60	31.5	25.175	25.175
	8	101	1.0	60	31.5	25.175	25.175
	15, 16	110, 111	1.0	60	31.5	25.175	25.175
	24	112	1.0	60	31.5	25.175	25.175
	32	112	2.0	60	31.5	25.175	25.175
	4		1.0	72	37.9	31.5	31.5
	8		1.0	72	37.9	31.5	31.5
	15, 16		1.0	72	37.9	31.5	31.5
	24		1.0	72	37.9	31.5	31.5
	32		2.0	72	37.9	31.5	31.5
	4		1.0	75	37.5	31.5	31.5
	8		1.0	75	37.5	31.5	31.5
	15, 16		1.0	75	37.5	31.5	31.5
	24		1.0	75	37.5	31.5	31.5
	32		2.0	75	37.5	31.5	31.5
	8		1.0	85	43.3	36.0	36.0
	15, 16		1.0	85	43.3	36.0	36.0
	32		2.0	85	43.3	36.0	36.0
	8		1.0	100	50.95	41.165	41.165
	15, 16		1.0	100	50.95	41.165	41.165
	8		1.0	120	63.92	53.69	53.69
	15, 16		1.0	120	63.92	53.69	53.69

Notes for Table 2.6.1:

1 Modes supported through BIOS are independent of drivers.
2 Implementation of refresh rates is driver-dependant.
3 Refresh rates shown may require high MCLK and/or non-fast page DRAM.

Table 2.6.1 ProMotion-AT3D extended graphics modes

Display resolution	Bits per pixel	VESA mode (hex)	Mem. req. (MB)	Vert.freq. (Hz)	Horiz. freq. (KHz)	Pixel freq. (MHz)	VCLK freq. (MHz)
800×600	4	102	1.0	56	35.2	36	36
	8	103	1.0	56	35.2	36	36
	15, 16	113, 114	1.0	56	35.2	36	36
	24		2.0	56	35.2	36	36
	32	115	1.5†	56	35.2	36	36
	8		1.0	60	37.9	40	40
	15, 16		1.0	60	37.9	40	40
	24		2.0	60	37.9	40	40
	32		1.5†	60	37.9	40	40
	8		1.0	72	48.1	50	50
	15, 16		1.0	72	48.1	50	50
	24		2.0	72	48.1	50	50
	32		1.5†	72	48.1	50	50
	8		1.0	75	46.9	50	50
	15, 16		1.0	75	46.9	50	50
	24		2.0	75	46.9	50	50
	32		1.5†	75	46.9	50	50
	8		1.0	85	53.7	56.3	56.3
	15, 16		1.0	85	53.7	56.3	56.3
	32		1.5†	85	53.7	56.3	56.3
	8		1.0	100	64.0	65.0	65.0
	15, 16		1.0	100	64.0	65.0	65.0
	8		1.0	120	75.2	76.96	76.96
	15, 16		1.0	120	75.2	76.96	76.96

Notes for Table 2.6.1:

1 Modes supported through BIOS are independent of drivers.
2 Implementation of refresh rates is driver-dependant.
3 Refresh rates shown may require high MCLK and/or non-fast page DRAM.

Table 2.6.1 ProMotion-AT3D extended graphics modes

Display resolution	Bits per pixel	VESA mode (hex)	Mem. req. (MB)	Vert.freq. (Hz)	Horiz. freq. (KHz)	Pixel freq. (MHz)	VCLK freq. (MHz)
1024×768	8		1.0	43(86i)	35.52	44.9	44.9
	15, 16		2.0	43(86i)	35.52	44.9	44.9
	4	104	1.0	60	48.3	65	65
	8	105	1.0	60	48.3	65	65
	15, 16	117	2.0	60	48.3	65	65
	32	118	3.0†	60	48.3	65	65
	4		1.0	70	56.5	75	75
	8		1.0	70	56.5	75	75
	15, 16		2.0	70	56.5	75	75
	32		4.0	70	56.5	75	75
	4		1.0	75	60	80	80
	8		1.0	75	60	80	80
	15, 16		2.0	75	60	80	80
	32		3.0†	75	60	80	80
	4		1.0	85	68.6	94.5	94.5
	8		1.0	85	68.6	94.5	94.5
	32		3.0†	85	68.6	94.5	94.5
	4		1.0	100	80.8	108	108
	8		1.0	100	80.8	108	108
	15, 16		2.0	100	80.8	108	108
1152×864	8		1.0	60	54.1	80	80
	15, 16		2.0	60	54.1	80	80
	32		4.0	60	54.1	80	80
	8		1.0	70	53.9	94.5	94.5
	15, 16		2.0	70	53.9	94.5	94.5
	32		4.0	70	53.9	94.5	94.5
	8		1.0	75	67.5	100	100
	15, 16		2.0	75	67.5	100	100
	8		1.0	85	77.09	121.5	121.5
	15, 16		2.0	85	77.09	121.5	121.5

Notes for Table 2.6.1:

1 Modes supported through BIOS are independent of drivers.
2 Implementation of refresh rates is driver-dependant.
3 Refresh rates shown may require high MCLK and/or non-fast page DRAM.

Table 2.6.1 ProMotion-AT3D extended graphics modes

Display resolution	Bits per pixel	VESA mode (hex)	Mem. req. (MB)	Vert.freq. (Hz)	Horiz. freq. (KHz)	Pixel freq. (MHz)	VCLK freq. (MHz)
1280×1024	8		2.0	43(86i)	96.4	78.75	78.75
	4	106	1.0	60	64	110	110
	8	107	2.0	60	64	100	100
	15, 16	119, 11A	4.0	60	64	110	110
	24	11B	4.0	60	72	75	75
	4		1.0	75	80.0	135	135
	8		2.0	75	80.5	144	144
	15, 16		4.0	75	80.5	144	144
	4		1.0	85	91.146	157	157
1600×1200	8		2.0	48(96i)	62.5	135	135
	8		2.0	60	75	160	160
	15, 16		4.0	60	70	70	70
	8		2.0	65	81.25	175.5	175.5

Notes for Table 2.6.1:

1 Modes supported through BIOS are independent of drivers.
2 Implementation of refresh rates is driver-dependant.
3 Refresh rates shown may require high MCLK and/or non-fast page DRAM.

Table 2.6.2 AT3D memory requirements for VESA modes

		VESA/VBE vertical refresh						
Resolution	Color depth	43Hz (86i)	56Hz	60Hz	70Hz	72Hz	75Hz	85Hz
640×480	4-bit			1 MB		1 MB	1 MB	1 MB
	8-bit			1 MB		1 MB	1 MB	1 MB
	15/16-bit			1 MB		1 MB	1 MB	1 MB
	32-bit			2 MB		2 MB	2 MB	2 MB
800×600	4-bit			1 MB		1 MB	1 MB	
	8-bit		1 MB	1 MB		1 MB	1 MB	
	15/16-bit			1 MB		1 MB	1 MB	
	32-bit			2 MB	2 MB	2 MB	2 MB	2 MB
1024×768	4-bit			1 MB		1 MB	1 MB	
	8-bit	1 MB		1 MB	1 MB	1 MB	1 MB	1 MB
	15/16-bit			2 MB		2 MB	2 MB	2 MB
	32 MB-bit			4 MB		4 MB	4 MB	
1154×864	4-bit			1 MB		1 MB	1 MB	
	8-bit			1 MB		1 MB	1 MB	
	15/16-bit			2 MB		2 MB	2 MB	
1280×1024	4-bit			1 MB		1 MB		
	8-bit	2 MB		2 MB		2 MB	2 MB	2 MB
	15/16-bit			4 MB			4 MB	
	32-bit			4 MB				
1600×1200	8-bit	2 MB*		2 MB	2 MB			
	15/16-bit			4 MB				

* 1600×1200 8-bit VESA interlaced mode is 48Hz (96i).

NOTE: All AT3D refresh rates comply with VESA tolerances, ±0.5% PCLK.

Figure 2.6.2 ProMotion analog interface

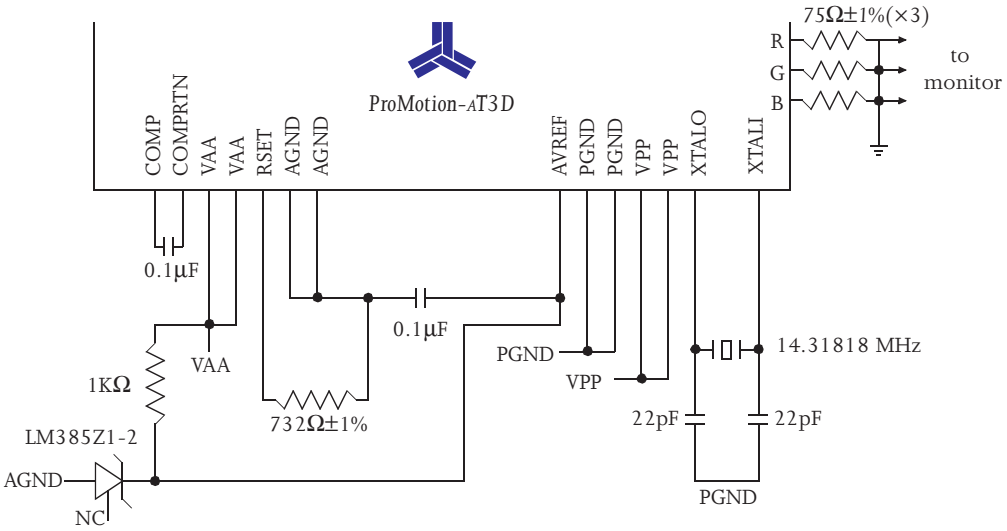
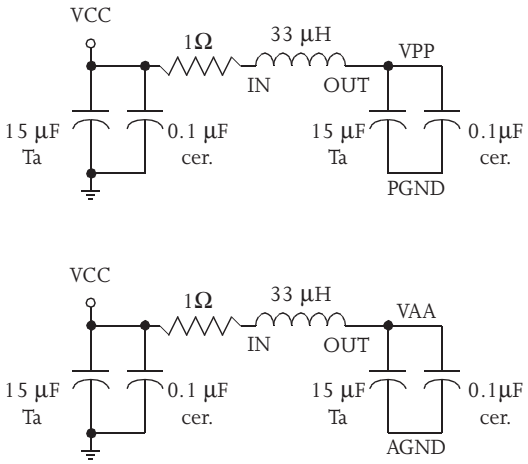


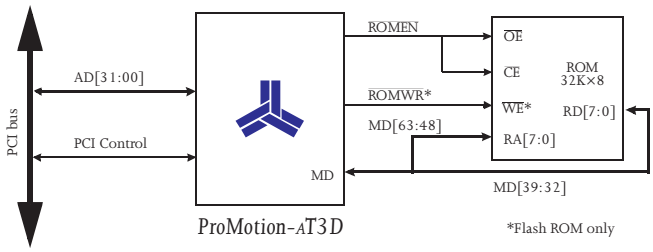
Figure 2.6.3 Suggested analog power filter



2.7 PCI host interface

ProMotion-AT3D interfaces directly to a PCI bus. The controller supports **zero-wait-state bursts** of successive dwords into the chip's inbound command FIFO. After dispatching commands and data to the AT3D, the host CPU can continue execution. Configuration strap MD[27] selects PCI bus operation; refer to "ProMotion-AT3D configuration straps," on page 61.

Figure 2.7. Glueless PCI/ROM interface



2.8 ROM BIOS interface

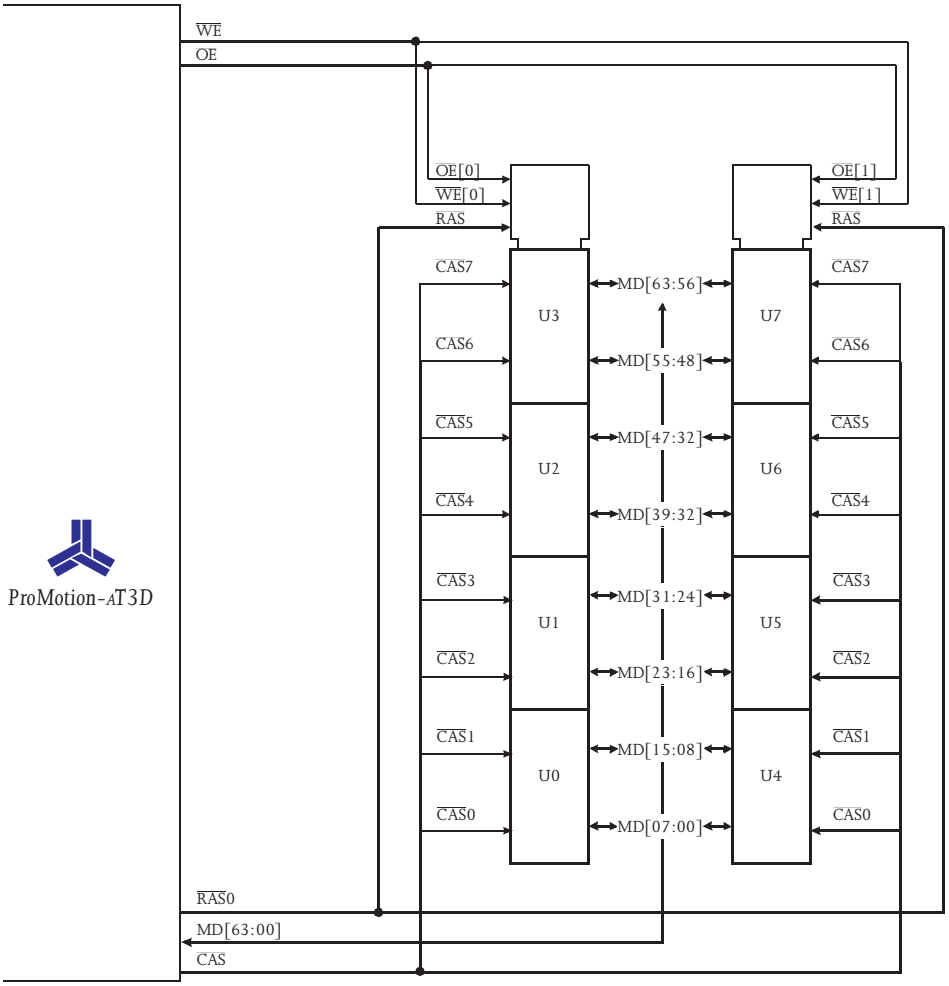
ProMotion-AT3D supports address, data, and flash write control interface for ROM BIOS or Flash EEPROM as shown in Figure 2.7, "Glueless PCI/ROM interface."

2.9 DRAM interface

ProMotion-AT3D controls 1, 2, or 4 megabytes of DRAM frame buffer memory. For 1MB and 2MB systems 256K×4, ×8, or ×16 parts may be used. For 4MB systems 256K×8 or ×16 may be used. Dual-$\overline{CAS}$ EDO and fast-page memories are supported.

Single cycle EDO timing permits high memory efficiency even in 1–2 MB configurations. Programmable memory timing allows ProMotion-AT3D to use standard speed DRAM or take advantage of high-speed DRAMs.

Figure 2.9.1 Memory interface: 1–4M (default mode/multiple $\overline{CAS}$)



2.10 Monitor and feature connector interface

For interoperability with video capture and other multimedia cards, ProMotion-AT3D offers two feature connector options, selectable by configuration strap MD[26]. In VSVPC mode, ProMotion-AT3D connects to an industry-standard 8-bit VGA pass-through connector; refer to Figure 2.10.1, "Glueless VSVPC feature connector." In VAFC mode, the chip supports the VESA Advanced Feature Connector standard, including 16-bit input and output. With the circuit shown in Figure 2.10.2, VAFC can be implemented without an expensive multiport DAC.

Figure 2.10.1: Glueless VSVPC feature connector

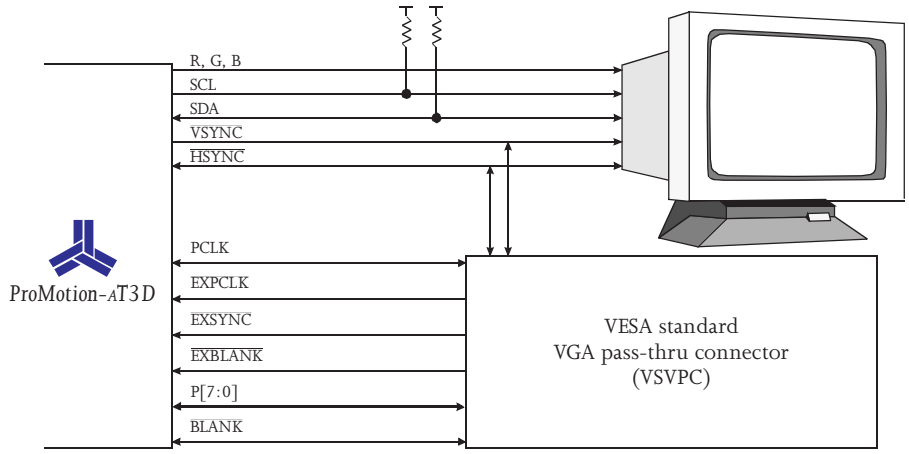
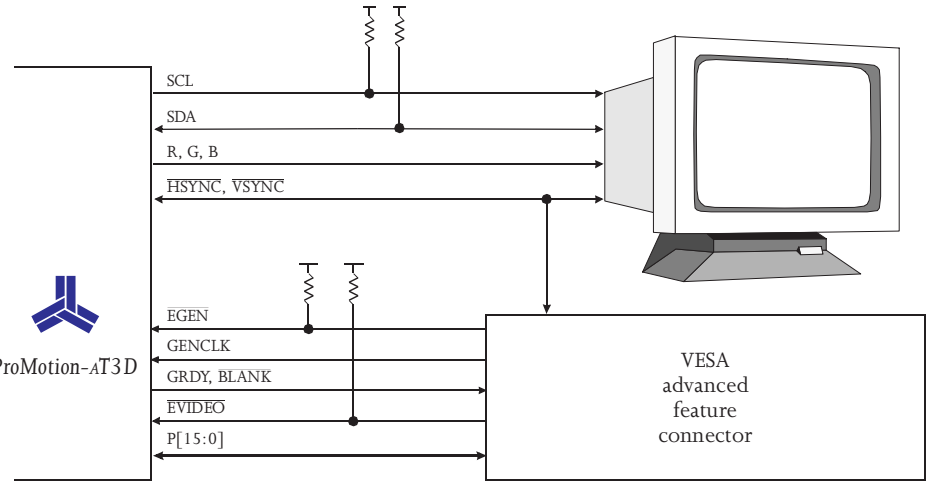


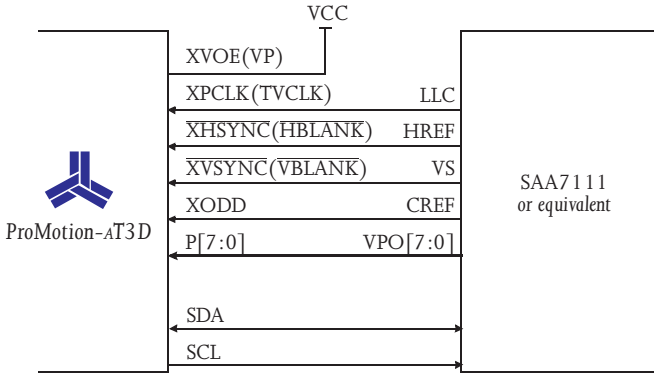
Figure 2.10.2: Glueless VAFC feature connector



2.11 VMI+ video interface

ProMotion-AT3D provides a VMI-compatible interface port for live video and hardware codec input. The controller's VMI+ video input port supports 8-bit and 16-bit digital video. Refer to ProMotion implementation notes for details.

Figure 2.11. VMI+ interface and decoder



2.12 VMI+ Host interface

ProMotion-AT3D supports host modes A and B, corresponding to Intel- and Motorola-style peripheral interface, to drive control and data inputs of MPEG coprocessors of other devices. Refer to ProMotion VMI implementation notes for details.

2.13 DDC 2.0B support

ProMotion-AT3D includes dedicated I/O pins for bi-directional DDC monitor connections. Using industry standard protocols, software can use DDC to read status and write configurations to compliant monitors. The same serial interface can control serial devices such as EEPROM for nonvolatile configuration strap.

3. VGA registers

Table 3.1 VGA attribute controller registers

I/O mapped port (hex)	Index (hex)	Register	Bits	r/w
3C0	-	"Index," described on page 119	6	r/w
3C0	00–0F	"Palette registers 0-15," described on page 119	6	r/w
3C0	10	"Mode control," described on page 119	8	r/w
3C0	11	"Overscan color," described on page 121	8	r/w
3C0	12	"Color plane enable," described on page 121	6	r/w
3C0	13	"Horizontal pixel panning," described on page 122	4	r/w
3C0	14	"Color select," described on page 123	4	r/w

Table 3.2 VGA general registers

I/O mapped port (hex)	Index (hex)	Register	Bits	r/w
3C2	-	"Item select/miscellaneous output," described on page 124	8	w
3CC	-		8	r
3BA/3DA		"Feature control/vertical enable," described on page 125	4	w
3CA	-		4	r
3C2	-	"Input status 0," described on page 126	8	r
3BA/3DA		"Input status 1," described on page 126	6	r

Table 3.3 VGA sequencer registers

I/O mapped port (hex)	Index (hex)	Register	Bits	r/w
3C4	-	"Sequencer index," described on page 128	4	r/w
3C5	00	"Reset," described on page 128	2	r/w
3C5	01	"Clocking mode," described on page 128	6	r/w
3C5	02	"Map mask," described on page 129	4	r/w
3C5	03	"Character map select," described on page 130	6	r/w
3C5	04	"Memory mode," described on page 131	4	r/w

Table 3.4 VGA graphics controller registers

I/O mapped port (hex)	Index (hex)	Register	Bits	r/w
3CE	-	"Graphics index," described on page 132	4	r/w
3CF	00	"Set/reset," described on page 132	4	r/w
3CF	01	"Enable set/reset," described on page 132	4	r/w
3CF	02	"Color compare," described on page 133	4	r/w
3CF	03	"Data rotate," described on page 133	5	r/w
3CF	04	"Read map select," described on page 134	2	r/w

Table 3.4 VGA graphics controller registers

I/O mapped port (hex)	Index (hex)	Register	Bits	r/w
3CF	05	"Graphics mode," described on page 134	5	r/w
3CF	06	"Miscellaneous," described on page 135	4	r/w
3CF	07	"Color don't care," described on page 136	4	r/w
3CF	08	"Bit mask," described on page 137	8	r/w

Table 3.5 VGA CRTC registers

I/O mapped port (hex)	Index (hex)	Register	Bits	r/w
3D4	-	"CRTC index," described on page 138	6	r/w
3D5	00	"Horizontal total," described on page 138	8	r/w
3D5	01	"Horizontal display enable end," described on page 140	8	r/w
3D5	02	"Horizontal blank start," described on page 141	8	r/w
3D5	03	"Horizontal blank end," described on page 142	8	r/w
3D5	04	"Horizontal retrace start," described on page 143	8	r/w
3D5	05	"Horizontal retrace end," described on page 144	8	r/w
3D5	06	"Vertical total," described on page 145	8	r/w
3D5	07	"Vertical overflow," described on page 146	8	r/w
3D5	08	"Preset row scan," described on page 146	7	r/w
3D5	09	"Maximum scan line," described on page 147	8	r/w
3D5	0A	"Block cursor start," described on page 148	6	r/w
3D5	0B	"Block cursor end," described on page 149	7	r/w
3D5	0C	"Serial start address," described on page 150	16	r/w
3D5	0E	"Block cursor location," described on page 151	16	r/w
3D5	10	"Vertical retrace end," described on page 153	8	r/w
3D5	11	"Vertical retrace end," described on page 153	8	r/w
3D5	12	"Vertical display enable end," described on page 154	8	r/w
3D5	13	"Serial offset," described on page 155	8	r/w
3D5	14	"Underline location/dword mode," on page 156	7	r/w
3D5	15	"Vertical blank start," described on page 157	8	r/w
3D5	16	"Vertical blank end," described on page 158	8	r/w
3D5	17	"CRTC mode control register," described on page 159	8	r/w
3D5	18	"Line compare," described on page 160	8	r/w
3D5	22	"Readback latch data," described on page 161	8	r
3D5	24	"Attribute index data," described on page 161	8	r

Table 3.6 VGA palette DAC registers

I/O mapped port (hex)	Index (hex)	Memory mapped offset (hex)	Register	Bits	r/w
3C6	-	-	"Palette RAM pel mask," described on page 163	8	r/w
3C7	-	-	"Palette RAM state/read address," described on page 163	8	w
3C7	-	-		2	r
3C8	-	-	"Palette RAM write address," described on page 164	8	r/w
3C9	-	-	"Palette RAM data," described on page 164	8	r/w
3C9	000–0FF	-	"Primary palette registers 0–255," described on page 165	24	r/w
3C9	100–11F	-	"Secondary palette registers 0–31," described on page 165†	24	r/w

†3C9.100–11F are extended registers, included in this group for completeness.

4. ProMotion-AT3D extended registers

Table 4.1 Extended setup registers

I/O mapped port (hex)	Index (hex)	Memory mapped offset (hex)	Register	Bits	r/w	Reset
3C5	10	-	"Unlock extended registers," on page 167 3C5.10[7:0] unlock Writing 12h unlocks I/O registers. ProMotion memory mapped registers cannot be locked. Writes to regosters 000–13F pass through the command FIFO. Writes to registers 140–1FF do not pass through the command FIFO.	8	r/w	7:0
3C5	11–19	-	"Chip ID," described on page 167 3C5.11[72:0] ASCII string	72	r	-
3C5	1A	-	"Flat model base address," described on page 168 3C5.1A[7:0] base address	8	r/w	[7:0]
3C5	1B	-	"Remap control," on page 168 3C5.1B[5:3] remap host BLT port 3C5.1B[2:0] remap ProMotion registers	6	r/w	[5:0]
3C5	1C	-	"Flat model control," on page 169 3C5.1C[0] flat model access 3C5.1C[2:1] flat model aperture 3C5.1C[3] disable VGA memory access 3C5.1C[4] VGA aperture addressing 3C5.1C[5] simultanous linear/drawing engine access	6	r/w	[5:0]
3C5	1D	-	"Alternate access space pointer LOW," described on page 171 3C5.1D[7:0] PMPOINTER [9:2]	8	r/w	-
3C5	1E–1F	-	"Alternate access space decode," described on page 172 3C5.1E[15:0] PMDECODE	16	r/w	[15:0]
3C5	20–27	-	"Scratchpad," described on page 174† 3C5.20[64:0] scratchpad	64	r/w	-
3C5	28	-	"Alternate access space pointer HIGH," described on page 177 3C5.28[7:0] PMPOINTER [17:10]	8	r/w	-
3C5	29	-	Reserved	-	-	-
3C5	30	-	"BIOS Paging," on page 177 3C5.30[4:0] BIOS page 3C5.30[6:5] BIOS page memory mapping 3C5.30[7] local	8	r/w	-

† ProMotion-3210™ memory mapped scratchpad is not supported. ProMotion-AT3D scratchpad registers are I/O mapped and reverse compatible with ProMotion-6410™, 6422, AT24, and AT3D.

Table 4.1 Extended setup registers

I/O mapped port (hex)	Index (hex)	Memory mapped offset (hex)	Register		Bits	r/w	Reset
-	-	1FC–1FE	"Extended/DAC status," on page 178		22	r	-
			1FC[3:0]	command FIFO entries available			
			1FC[4]	DAC threshold red			
			1FC[5]	DAC threshold green			
			1FC[6]	DAC threshold blue			
			1FC[7]	signature analyzer busy			
			1FC[8]	host BLT in progress			
			1FC[9]	host BLT read data available			
			1FC[10]	drawing engine busy			
			1FC[11]	vertical display active			
			1FC[12]	EXVID pin 66 input			
			1FC[13]	EXPCLK pin 67 input			
			1FC[14]	EXSYNC pin 68 input			
			1FC[15]	feature connector input			
			1FC[16]	SDA input, equivalent to 0D0[4]			
-	-	1FF	"Abort," described on page 180		0	w	-
			1FF[ ]	any write aborts drawing operation			

† ProMotion-3210™ memory mapped scratchpad is not supported. ProMotion-AT3D scratchpad registers are I/O mapped and reverse compatible with ProMotion-6410™, 6422, AT24, and AT3D.

Table 4.2 Extended CRTC registers

I/O mapped port (hex)	Index (hex)	Memory mapped offset (hex)	Register		Bits	r/w	Reset
3D5	19	-	"Horizontal interlaced start," on page 181		8	r/w	[7:0]†
			3D5.19[7:0]	horizontal interlaced start [7:0] of [8:0]			
3D5	1A	-	"Vertical extended overflow," on page 181		5	r/w	[4:0]†
			3D5.1A[0]	vertical total [10] of [10:0]. "Vertical total," described on page 145.			
			3D5.1A[1]	vertical display enable end [10] of [10:0]. "Vertical display enable end," described on page 154.			
			3D5.1A[2]	vertical blank start [10] of [10:0]. "Vertical blank start," described on page 157.			
			3D5.1A[3]	vertical retrace start [10] of [10:0]. "Vertical retrace start," described on page 152			
			3D5.1A[4]	line compare [10] of [10:0]. "Line compare," described on page 160.			

† Writing to 3D5.00 resets this register unless autoreset is disabled with 3D5.1E.

Table 4.2 Extended CRTC registers

I/O mapped port (hex)	Index (hex)	Memory mapped offset (hex)	Register	Bits	r/w	Reset
3D5	1B	-	"Horizontal overflow," on page 182	5	r/w	[4:0]†
			3D5.1B[0] horrizontal total [8] of [8:0]. "Horizontal total," described on page 138.			
			3D5.1B[1] horizontal display enable end [8] of [8:0]. "Horizontal display enable end," described on page 140.			
			3D5.1B[2] horizontal blank start [8] of [8:0]. "Horizontal blank start," described on page 141.			
			3D5.1B[3] horizontal retrace start [8] of [8:0]. "Horizontal retrace start," described on page 143.			
			3D5.1B[4] horizontal interlaced start [8] of [8:0]. "Horizontal interlaced start," described on page 181.			
3D5	1C	-	"Serial overflow," on page 182	8	r/w	[7:0]†
			3D5.1C[3:0] serial start address [19:16] of [19:0]. "Serial start address," described on page 150.			
			3D5.1C[7:4] serial offset [11:8] of 12. "Serial offset," described on page 155.			
3D5	1D	-	"Character clock adjust," on page 183	3	r/w	[2:0]†
			3D5.1D[2:0] character clock adjust			
3D5	1E	-	"Extended CRTC autoreset," on page 183	1	r/w	[0]
			3D5.1E[0] disable automatic CRTC reset			
-	-	1FA–1FB	"Vertical current position," on page 184	11	r	-
			1FA[10:0] vertical current position			

† Writing to 3D5.00 resets this register unless autoreset is disabled with 3D5.1E.

Table 4.3 2D Drawing engine registers

I/O mapped port (hex)	Index (hex)	Memory mapped offset (hex)	Register	Bits	r/w	Reset
-	-	030	"Clipping control," on page 186	3	r/w	[0]
			030[0] clipping enable			
			030[1] clipping polarity			
			030[2] clipping abort			
-	-	031–037	Reserved	-	-	-
-	-	038–039	"Clipping boundary left," described on page 187	12	r/w	-
			038[11:0] clipping boundary left			
-	-	03A–03B	"Clipping boundary top," described on page 187	12	r/w	-
			03A[11:0] clipping boundary top			

Table 4.3 2D Drawing engine registers

I/O mapped port (hex)	Index (hex)	Memory mapped offset (hex)	Register		Bits	r/w	Reset
-	-	03C–03D	"Clipping boundary right," described on page 187		12	r/w	-
			03C[11:0]	clipping boundary right			
-	-	03E–03F	"Clipping boundary bottom," described on page 188		12	r/w	-
			03E[11:0]	clipping boundary bottom			
-	-	040–043	"Drawing engine control," on page 188		32	r/w	[31:0]
			040[3:0]	drawing engine command			
			040[5:4]	reserved			
			040[6]	direction x			
			040[7]	direction y			
			040[8]	major axis			
			040[9]	source address XY/linear			
			040[10]	source pattern			
			040[11]	source rectangular/contiguous			
			040[12]	source color/monochrome			
			040[13]	source transparent			
			040[16:14]	pixel depth			
			040[17]	reserved			
			040[18]	destination address XY/linear			
			040[19]	destination rectangular/contiguous			
			040[20]	destination transparent			
			040[21]	destination transparency polarity			
			040[23:22]	pattern format			
			040[26:24]	address model			
			040[28:27]	destination update			
			040[30:29]	quick start			
			040[31]	drawing engine start			
-	-	044–045	Reserved.		-	-	-
-	-	046	"Raster operation," described on page 192		4	r/w	-
			046[3:0]	raster operation			
-	-	047	"Byte mask," described on page 193		4	r/w	-
			047[3:0]	byte mask			

Table 4.3 2D Drawing engine registers

I/O mapped port (hex)	Index (hex)	Memory mapped offset (hex)	Register	Bits	r/w	Reset
-	-	048–04F	"Pattern," on page 193 8×8 monochrome 048[7:0] top row of pattern 048[15:8] second row 048[23:18] third row 048[31:24] fourth row 048[39:32] fifth row 048[47:40] sixth row 048[55;48] seventh row 048[63:56] bottom row of pattern 4×4 16-color 048[3:0] row 1 column 1 of pattern 048[7:4] row 1 column 2 of pattern 048[11:8] row 1 column 3 of pattern 048[15:12] row 1 column 4 of pattern 048[19:16] row 2 column 1 of pattern 048[23:20] row 2 column 2 of pattern 048[27:24] row 2 column 3 of pattern 048[31:28] row 2 column 4 of pattern 048[35:32] row 3 column 1 of pattern 048[39:36] row 3 column 2 of pattern 048[43:40] row 3 column 3 of pattern 048[47:44] row 3 column 4 of pattern 048[51:48] row 4 column 1 of pattern 048[55:52] row 4 column 2 of pattern 048[59:56] row 4 column 3 of pattern 048[63:60] row 4 column 4 of pattern	64	r/w	-
-	-	050–051	"Source location X/low," described on page 196 XY addressing mode 050[11:0] source location x Linear addressing mode 050[11:0] source linear pixel address [11:00]	12	r/w	-
-	-	052–053	"Source location Y/high," described on page 196 XY addressing mode 052[11:0] source location y Linear addressing mode 052[11:0] source linear pixel address [23:12]	12	r/w	-
-	-	054–055	"Destination location X/low," described on page 197 XY addressing mode 054[11:0] destination location x Linear addressing mode 054[11:0] destination linear pixel address	12	r/w	-

Table 4.3 2D Drawing engine registers

I/O mapped port (hex)	Index (hex)	Memory mapped offset (hex)	Register	Bits	r/w	Reset
-	-	056–057	"Destination location Y/high," described on page 197 XY addressing mode 056[11:0] destination location y Linear addressing mode 056[11:0] destination linear pixel address	12	r/w	-
-	-	058–059	"Source size X/vector pixel count," described on page 198 058[11:0] dimension x pixel count	12	r/w	-
-	-	05A–05B	"Source size Y," described on page 198 05A[11:0] dimension y pixel count	12	r/w	-
-	-	05C–05D	"Destination row pitch," described on page 199 05C[12:0] destination row pitch	13	r/w	-
-	-	05E–05F	"Source row pitch," described on page 199 05E[12:0] source row pitch	13	r/w	-
-	-	060–063	"Foreground color," on page 200 4-bit packed mode 060[3:0] foreground color 060[7:4] foreground color 060[31:28] reserved 8-bit mode 060[7:0] foreground color 060[31:8] reserved 16-bit mode 060[15:0] foreground color 060[31:16] reserved 32-bit mode 060[31:0] foreground color	32	r/w	-

Table 4.3 2D Drawing engine registers

I/O mapped port (hex)	Index (hex)	Memory mapped offset (hex)	Register	Bits	r/w	Reset
-	-	064–067	"Background color/source transparency," on page 201 4-bit packed mode 064[3:0] foreground color 064[7:4] background color 064[31:28] reserved 8-bit mode 064[7:0] background color 064[31:8] reserved 16-bit mode 064[15:0] background color 064[31:16] reserved 32-bit mode 064[31:0] background color	32	r/w	-
-	-	068–06B	Reserved	-	-	-
-	-	06C–06E	"Destination transparency color," described on page 202 8-bit mode 06C[7:0] destination transparency color 06C[31:8] reserved 16-bit mode 06C[15:0] destination transparency color 06C[31:16] reserved 32-bit mode 06C[31:0] destination transparency color	24	r/w	-
-	-	06F	"Destination transparency mask," described on page 203 06F[0] compare bits [7:0] 06F[1] compare bits [14:8] 06F[2] compare bit [15] 06F[3] compare bits [23:16] 06F[4] compare bits [31:24] against color [7:0]	5	r/w	-
-	-	070–071	"DDA axial step constant," described on page 203 070[15:0] DDA axial step constant	16	r/w	-
-	-	072–073	"DDA diagonal step constant," described on page 203 072[15:0] DDA diagonal step constant	16	r/w	-

Table 4.3 2D Drawing engine registers

I/O mapped port (hex)	Index (hex)	Memory mapped offset (hex)	Register	Bits	r/w	Reset
-	-	074–075	"DDA error term," described on page 204 074[15:0] error term	16	r/w	-
-	-	076–07F	Reserved	-	-	-

Table 4.4 Motion video registers

I/O mapped port (hex)	Index (hex)	Memory mapped offset (hex)	Register	Bits	r/w	Reset
-	-	080	See "Extended configuration registers" on page 33.	8	r/w	[6:0]†
-	-	081	Reserved	-	-	-
-	-	082–083	"vWindow group 0 control," on page 205 082[0] enable vWindow 082[3:1] vWindow pixel bit-depth 082[6:4] vWindow format 082[7] reserved 082[8] YUV to RGB conversion enable 082[9] motion video stretch 082[10] stretch video interpolation horizontal 082[11] stretch video interpolation vertical 082[12] Smoothing filter enable 082[13] reserved 082[14] chromakey enable	15	r/w	[14:0]†
-	-	084–085	"vWindow group 0 data pitch," on page 207 084[11:0] base address	12	r/w	-
-	-	086–087	"vWindow group 0 scale factor horizontal," described on page 208 088[11:0] motion video scale factor horizontal	12	r/w	-
-	-	088–089	"vWindow group 0 scale offset horizontal," described on page 208 086[11:0] motion video group 0 scale offset horizontal	12	r/w	
-	-	08A–08B	"vWindow group 0 scale factor vertical," described on page 209 08A[11:0] motion video group 0 scale factor vertical 1	12	r/w	-
-	-	08C–08D	"vWindow group 0 stretch offset vertical," described on page 209 08C[11:0] motion video group 0 stretch offset vertical	12	r/w	-

† Writing to 3D5.00 resets this register unless autoreset is disabled with 3D5.1E.

Table 4.4 Motion video registers

I/O mapped port (hex)	Index (hex)	Memory mapped offset (hex)	Register		Bits	r/w	Reset
-	-	08E–08F	"Tile sequence control," described on page 210		16	r/w	[9:8], [3:0]
			08E[3:0]	sequence base			
			08E[7:4]	sequence length			
			08E[8]	VMI+ swap			
			08E[9]	pin swap			
			08E[11:10]	number of buffers			
			08E[12]	stereo			
			08E[15:10]	reserved			
-	-	090	"Chromakey color," described on page 210		24	r/w	-
			direct color				
			090[23:0]	chromakey color			
			indexed 8-bit color				
			090[7:0]	chromakey color			
			090[23:8]	reserved			
-	-	092	"vWindow group 1 control," on page 211		15	r/w	[14:0]†
			092[0]	enable vWindow			
			092[3:1]	vWindow pixel bit-depth			
			092[6:4]	vWindow format			
			092[7]	reserved			
			092[8]	YUV to RGB conversion enable			
			092[9]	motion video stretch			
			092[10]	stretch video interpolation horizontal			
			092[11]	stretch video interpolation vertical			
			092[12]	Smoothing filter enable			
			092[13]	reserved			
			092[14]	chromakey enable			
-	-	094	"vWindow group 1 data pitch," described on page 213		24	r/w	-
			094[23:0]	data pitch			
-	-	096	"vWindow group 1 scale factor horizontal," described on page 214		12	r/w	-
			096[11:0]	motion video stretch factor horizontal 1			
-	-	098	"vWindow group 1 scale offset horizontal," described on page 215		16	r/w	-
			096[15:0]	motion video stretch factor horizontal 2			
-	-	09A	"vWindow group 1 scale factor vertical," described on page 215		12	r/w	-
			09A[11:0]	motion video stretch factor vertical 1			
-	-	09C	"vWindow group 1 stretch offset vertical," described on page 216		16	r/w	-
			09A[15:0]	motion video stretch factor vertical 2			

† Writing to 3D5.00 resets this register unless autoreset is disabled with 3D5.1E.

Table 4.5 Video tile buffer registers

I/O mapped port (hex)	Index (hex)	Memory mapped offset (hex)	Register	Bits	r/w	Reset
-	-	200	"Tile 0 control register," on page 217 200[2:0] tile vWindow select 200[3] reserved 200[4] tile rightmost	4	r/w	-
-	-	202–203	"Tile 0 display position left," on page 217 202[10:0] display position left	11	r/w	-
-	-	204–205	"Tile 0 display position right," on page 218 204[10:0] display position right	11	r/w	-
-	-	206–207	"Tile 0 display position bottom," on page 218 206[10:0] display position bottom	11	r/w	-
-	-	208–209	"Tile 0 data width," on page 219 208[10:0] data width	11	r/w	-
-	-	20A–20C	"Tile 0 data location," on page 219 20A[21:0] data location	22	r/w	-
-	-	210–21F	Tile 1 register group	-	r/w	-
-	-	220–22F	Tile 2 register group	-	r/w	-
-	-	230–23F	Tile 3 register group	-	r/w	-
-	-	240–24F	Tile 4 register group	-	r/w	-
-	-	250–25F	Tile 5 register group	-	r/w	-
-	-	260–26F	Tile 6 register group	-	r/w	-
-	-	270–27F	Tile 7 register group	-	r/w	-
-	-	280–28F	Tile 8 register group	-	r/w	-
-	-	290–29F	Tile 9 register group	-	r/w	-
-	-	2A0–2AF	Tile 10 register group	-	r/w	-
-	-	2B0–2BF	Tile 11 register group	-	r/w	-

Table 4.6 Extended configuration registers

I/O mapped port (hex)	Index (hex)	Memory mapped offset (hex)	Register		Bits	r/w	Reset
-	-	080	"Serial control," on page 221		8	r/w	[6:0]†
			080[2:0]	desktop pixel bit-depth			
			080[4:3]	desktop pixel format			
			080[5]	enable double index			
			080[6]	enable extended VGA modes			
			080[7]	nibble swap mode			
-	-	0C0	"Page offset," on page 222		10	r/w	[9:0]†
			0C0[9:0]	page offset			
-	-	0C2	"Aperture control," on page 223		11	r/w	[6:4]
			0C2[0]	video subsystem select			
			0C2[1]	reserved			
			0C2[3:2]	ROM access			
			0C2[4]	flash ROM enable			
			0C2[6:5]	palette DAC access			
			0C2[7]	host XY addressing enable			
			0C2[10:8]	reserved			
-	-	0C3	Reserved		7	r/w	-
-	-	0C4	"Display memory configuration," on page 224		12	r/w	[5]
			0C4[0]	MD[30] interleaved memory			
			0C4[1]	MD[22] fast RAS disable			
			0C4[2]	reserved.			
			0C4[3]	fast RMW disable; dual bank drive (Rev. C)			
			0C4[4]	MD[29] dual WE select			
			0C4[5]	128-bit graphics engine enable ☞ must be set to 1			
			0C4[6]	MD[23] dual RAS drive enable			
			0C4[7]	MD[17] BetterHalf™ enable			
			0C4[8]	MD[16] 64-bit memory bus enable			
			0C4[9]	MD[13] EDO DRAM disable			
			0C4[10]	MD[12] single cycle page mode enable			
			0C4[11]	slow DRAM refresh enable			
-	-	0C6	Reserved		-	-	-
-	-	0C7	"DRAM timing adjust," on page 225		4	-	-
			0C6[0]	DRAM read timing adjust			
			0C6[3:1]	DRAM timing delay select			

† Writing to 3D5.00 resets this register unless autoreset is disabled with 3D5.1E.

Table 4.6 Extended configuration registers

I/O mapped port (hex)	Index (hex)	Memory mapped offset (hex)	Register		Bits	r/w	Reset
-	-	0C8	"VGA override," on page 226		14	r/w	[13:0]
			0C8[0]	lock VGA sequencer register 3C5			
			0C8[1]	lock VGA CRTC registers 3D5.00–24h			
			0C8[2]	lock VGA graphics controller registers			
			0C8[3]	lock VGA attribute controller registers			
			0C8[4]	lock VGA general registers			
			0C8[5]	force CRTC 0–7 unlock			
			0C8[6]	force 8-dot clock			
			0C8[7]	force graphics mode			
			0C8[8]	force DCLK=VCLK			
			0C8[9]	force 3C2[3:2] = 11b			
			0C8[10]	cursor blink disable			
			0C8[11]	DRAM refresh disable			
			0C8[12]	VGA I/O disable			
			0C8[13]	reserved			
-	-	0CA	"Host interface," on page 227		4	r/w	-
			0CA[0]	MD[27] PCI host interface enable			
			0CA[1]	MD[26] double edge feature connector			
			0CA[2]	MD[25] tri-state LDEV			
			0CA[3]	MD[10] PCI 66 MHz enable			
-	-	0CB	"PCI STOP latency," on page 228		8	r/w	-
			0CB[7:0]	clock cycles			
-	-	0CC	"Feature connector control," on page 228		8	r/w	[7], [3:1]
			0CC[0]	MD[15] VAFC feature connector enable			
			0CC[1]	feature connector direction			
			0CC[2]	feature connector disable			
			0CC[3]	genlock enable			
			0CC[4]	genlock reset			
			0CC[6:5]	genlock interlaced control			
			0CC[7]	generic feature connector enable			
-	-	0CD	"Generic feature connector control," on page 229		8	r/w	-
			0CD[7:0]	generic feature connector outputs			
-	-	0CE	"VAFC control," on page 230		5	r/w	-
			0CE[0]	DCLK control			
			0CE[1]	GRDY control			
			0CE[2]	chromakey enable			
			0CE[3]	feature connector format direct			
			0CE[4]	reserved			
-	-	0CF	"Genlock control," on page 231		8	r/w	-
			0CF[3:0]	vertical skew			
			0CF[7:4]	horizontal skew			

† Writing to 3D5.00 resets this register unless autoreset is disabled with 3D5.1E.

Table 4.6 Extended configuration registers

I/O mapped port (hex)	Index (hex)	Memory mapped offset (hex)	Register	Bits	r/w	Reset
		0D0	"DPMS/sync control," on page 232 0D0[0] DPMS $\overline{\text{HSYNC}}$ suspend 0D0[1] DPMS VSYNC suspend 0D0[2] DDC tri state HSYNC 0D0[3] SCL level [0] of [1:0] 0D0[5:4] SDA level 0D0[6] SCL level [1] of [1:0]	7	r/w	[6:0]
		0D2	"Monitor interlace control," on page 232 0D2[0] enable interlace 0D2[1] Drive XODD pin	1	r/w	[0]†
		0D4	"Pixel FIFO request point," on page 233 0D4[7:0] high priority request point–page break 0D4[15:8] high priority request point–no break 0D4[23:16] low priority request point	24	r/w	[23:16] = 14h [15:8] = 14h [7:0] = 14h
		0D8	"FIFO underflow," on page 234 0D7[0] FIFO underflow	1	r/w	
		0D9	"External signal timing," on page 234 0D9[3:0] EPROM access timing 0D9[5:4] LDEV wait states 0D9[6] disable PCI STOP 0D9[7] disable PCI LOCK	8	r/w	[7:6], [5:4] = 2h if PCI 66 MHz [5:4] = 1h if PCI 33 MHz [3:0] = 8h
-	-	0DA	Reserved	-	-	-
-	-	0DB	Enable Extended registers 0DB[0] enable extended register - DOS space 0DB[1] enable extended registers - linear space 0DB[2] enable coprocessor aperture 0DB[3] enable second linear aperture	4	r/w	-
-	-	0DC–0DD	Bi-endian control	12	r/w	-

† Writing to 3D5.00 resets this register unless autoreset is disabled with 3D5.1E.

Table 4.7 Hardware cursor registers

I/O mapped port (hex)	Index (hex)	Memory mapped offset (hex)	Register	Bits	r/w	Reset
-	-	140	"Hardware cursor control," on page 237	3	r/w	‡
			140[0] hw cursor enable 140[1] hw 3-color mode 140[2] hw cursor full-color enable			
-	-	141	"Hardware cursor color 1," on page 238	8	r/w	-
			141[7:0] hw cursor color 1			
-	-	142	"Hardware cursor color 2," on page 239	8	r/w	-
			142[7:0] hw cursor color 2			
-	-	143	"Hardware cursor color 3," on page 239	8	r/w	-
			143[7:0] hw cursor color 3			
-	-	144–145	"Hardware cursor pattern base address," on page 239	12	r/w	-
			144[11:0] hw cursor pattern location			
-	-	147	Reserved.	-	-	-
-	-	148–149	"Hardware cursor display position X," on page 240	12	r/w	-
			148[11:0] hw cursor position x			
-	-	14A–14B	"Hardware cursor display position Y," on page 240	12	r/w	-
			14A[11:0] hw cursor position y			
-	-	14C	"Hardware cursor display offset X," on page 241	6	r/w	-
			14C[5:0] hw cursor offset x			
-	-	14E	"Hardware cursor display offset Y," on page 242	6	r/w	-
			14E[5:0] hw cursor offset y			

‡ Any write to VGA CRTC registers index 0–17 resets bits [1:0] to 0.

Table 4.8 PCI configuration registers

PCI I/O (hex)	Index (hex)	Offset (hex)	Register	Bits	r/w	Reset
00	-	180–181	"PCI vendor ID," on page 244	16	r	1142h
			180[15:0] vendor ID: 1142h			
02	-	182–183	"PCI device ID," on page 244	16	r	643Dh
			182[15:0] device ID: 643Dh			
04	-	184	"PCI command," on page 244	6	r/w	[5:0]
			184[0] I/O space enable 184[1] memory space enable 184[4:2] reserved; always zero 184[5] VGA palette snooping			
05	-	185	Reserved.	-	-	-

† PCI specification defines these registers as r/w but ProMotion implements them read-only, returning 0.

Table 4.8 PCI configuration registers

PCI I/O (hex)	Index (hex)	Offset (hex)	Register	Bits	r/w	Reset
06–07	-	186–187	"PCI status," on page 245 186[8:0] reserved 186[10:9] DEVSEL timing 186[14:11] reserved 186[15] detected parity error	16	r	40h
08	-	188	"PCI revision ID," on page 246 188[7:0] revision ID	8	r	-
09	-	189	"Class code," on page 246 189[7:0] class code: 300h 189[15:8] reserved	16	r	300h
0A–0B	-	18A–18B	Reserved.	-	-	-
0C	-	18C	"Cache line size," on page 246 18C[0:0] cache line size	-	r‡	0
0D	-	18D	"Latency timer," on page 247 18D[0:0] latency timer	-	r‡	0
0E	-	18E	"Header type," on page 247 18E[0:0] header type	-	r	0
0F	-	18F	"BIST," on page 247 18C[0:0] BIST	-	r‡	0
10–1B	-	190–193	"PCI memory base address," on page 248 190[0] memory space indicator 190[23:1] reserved 190[31:24] base address	32	r/w	[31:0]
14–17	-	194–197	"PCI I/O base address," on page 248 194[0] I/O space indicator, default=1 194[3:1] reserved 194[31:4] base address	32	r/w	[3:1]
28–2B	-	1A8–1AB	Reserved.	-	-	-
2C–2D	-	1AC–1AD	"Subsystem vendor ID," on page 249	16	r	[15:0]
2E–2F	-	1AE–1AF	"Subsystem ID," on page 249	16	r	[15:0]
30–3C	-	1B0–1BB	"Expansion ROM base address," on page 249 1B0[0] ROM address enable 1B0[15:1] reserved 1B0[31:16] ROM base address	32	r/w	[0]
3C	-	1BC	"Interrupt line," on page 250 1BC[7:0] interrupt line	8	r/w	[7:0]
3D	-	1BD	"Interrupt pin," on page 250 1BD[0] set by MD[11] cfg. strap	1	r†	1h

† PCI specification defines these registers as r/w but ProMotion implements them read-only, returning 0.

Table 4.8 PCI configuration registers

PCI I/O (hex)	Index (hex)	Offset (hex)	Register		Bits	r/w	Reset
3E	-	1BE	"Minimum grant," on page 251		8	r†	[7:0]
			1BE[7:0]	minimum grant			
3F	-	1BF	"Maximum grant," on page 251		8	r†	[7:0]
			1BF[7:0]	maximum grant			
	-	1C0	"Enable write subsystem ID," on page 251		3	r	[2:0]
			1C0[0]	subsystem vendor id			
			1C0[1]	subsystem device id			
			1C0[2]	dual PCI id. reset: cnf MD[24]			

† PCI specification defines these registers as r/w but ProMotion implements them read-only, returning 0.

Table 4.9 DAC registers

I/O mapped port (hex)	Index (hex)	Memory mapped offset (hex)	Register		Bits	r/w	Reset
-	-	0E0	"Color correction," on page 253		7	r/w	†
			0E0[1:0]	desktop color correction			
			0E0[3:2]	vWindow color correction			
			0E0[4]	host RAM data width			
			0E0[5]	host palette select			
			0E0[6]	reserved			
-	-	0E4	"DAC control," on page 254		4	r/w	†
			0E4[0]	blanking pedestal enable			
			0E4[1]	reserved			
			0E4[2]	overcurrent boost			
			0E4[3]	DAC power			
-	-	0E5	"Overcurrent red," on page 254		6	r/w	-
			0E5[2:0]	reserved			
			0E5[5:3]	red vWindow boost			
-	-	0E6	"Overcurrent green," on page 255		6	r/w	-
			0E6[2:0]	reserved			
			0E6[5:3]	green vWindow boost			
-	-	0E7	"Overcurrent blue," on page 255		6	r/w	-
			0E7[2:0]	reserved			
			0E7[5:3]	blue vWindow boost			

† Writing to 3D5.00 resets this register unless autoreset is disabled with 3D5.1E.

Table 4.10 Clock registers

I/O mapped port (hex)	Index (hex)	Memory mapped offset (hex)	Register	Bits	r/w	Reset
-	-	0E8	"MCLK control," on page 258 0E8[0] MCLK bypass 0E8[1] MCLK power off 0E8[3:2] MCLK postscaler 0E8[6:4] MCLK frequency range 0E8[7] MCLK high speed	8	r/w	[0] = MD[21] [1] = MD[21]
-	-	0E9	"MCLK denominator," on page 259 0E8[6:0] MCLK denominator (M) 0E8[7] reserved	8	r/w	-
-	-	0EA	"MCLK numerator," on page 259 0E8[6:0] MCLK numerator (N) 0E8[7] reserved	8	r/w	-
-	-	0EB	Reserved	3	r/w	MD[20:18]
-	-	0EC	"VCLK control," on page 259 0EC[0] VCLK bypass 0EC[1] VCLK power off 0EC[3:2] VCLK postscaler 0EC[6:4] VCLK frequency range 0EC[7] VCLK high speed	8	r/w	[0] = MD[21] [1] = MD[21]
-	-	0ED	"VCLK denominator," on page 260 0EC[6:0] VCLK denominator (M) 0EC[7] reserved	8	r/w	-
-	-	0EE	"VCLK numerator," on page 261 0EC[6:0] VCLK numerator (N) 0EC[7] reserved	8	r/w	-
-	-	0EF	Reserved	-	-	-
-	-	0F0	"VCLK default 0 control," on page 261 0F0[1:0] reserved 0F0[3:2] VCLK default 0 postscaler 0F0[6:4] VCLK default 0 frequency range 0F0[7] reserved	8	r/w	
-	-	0F1	"VCLK default 0 denominator," on page 262 0F0[14:8] VCLK default 0 denominator (M) 0F0[15] reserved	8	r/w	-
-	-	0F2	"VCLK default 0 numerator," on page 262 0F0[6:0] VCLK default 0 numerator (N) 0F0[7] reserved	8	r/w	-
-	-	0F3	Reserved.	8	r/w	-

Table 4.10 Clock registers

I/O mapped port (hex)	Index (hex)	Memory mapped offset (hex)	Register		Bits	r/w	Reset
-	-	0F4	"VCLK default 1 control," on page 263		8	r/w	-
			0F4[1:0]	reserved			
			0F4[3:2]	VCLK default 1 postscaler			
			0F4[6:4]	VCLK default 1 frequency range			
			0F4[7]	reserved			
-	-	0F5	"VCLK default 1 denominator," on page 263		8	r/w	-
			0F4[14:8]	VCLK default 1 denominator (M)			
			0F4[15]	reserved			
-	-	0F6	"VCLK default 1 numerator," on page 264		8	r/w	-
			0F4[22:16]	VCLK default 1 numerator (N)			
			0F4[31:23]	reserved			
-	-	0F7	Reserved.		8	r/w	-

Table 4.11General purpose I/O registers

I/O mapped port (hex)	Index (hex)	Memory mapped offset (hex)	Register		Bits	r/w	Reset
-	-	1F0	"GPIO control," on page 265		8	r/w	[7:0]
			1F0[0]	GPIO pin 0 enable			
			1F0[1]	GPIO pin 1 enable			
			1F0[2]	GPIO pin 2 enable			
			1F0[3]	GPIO pin 3 enable			
			1F0[4]	GPIO pin 4 enable			
			1F0[5]	GPIO pin 5 enable			
			1F0[6]	GPIO pin 6 enable			
			1F0[7]	GPIO pin 7 enable			
-	-	1F1	"GPIO direction," on page 265		8	r/w	-
			1F1[0]	GPIO pin 0 output enable			
			1F1[1]	GPIO pin 1 output enable			
			1F1[2]	GPIO pin 2 output enable			
			1F1[3]	GPIO pin 3 output enable			
			1F1[4]	GPIO pin 4 output enable			
			1F1[5]	GPIO pin 5 enable			
			1F1[6]	GPIO pin 6 output enable			
			1F1[7]	GPIO pin 7 output enable			

Table 4.11 General purpose I/O registers

I/O mapped port (hex)	Index (hex)	Memory mapped offset (hex)	Register	Bits	r/w	Reset
-	-	1F2	"GPIO level," on page 266	8	r/w	-
			1F2[0] GPIO pin 0 level			
			1F2[1] GPIO pin 1 level			
			1F2[2] GPIO pin 2 level			
			1F2[3] GPIO pin 3 level			
			1F2[4] GPIO pin 4 level			
			1F2[5] GPIO pin 5 level			
			1F2[6] GPIO pin 6 level			
			1F2[7] GPIO pin 7 level			
-	-	1F3	"GPIO readback," on page 267	8	r	-
			1F3[0] GPIO pin 0 status			
			1F3[1] GPIO pin 1 status			
			1F3[2] GPIO pin 2 status			
			1F3[3] GPIO pin 3 status			
			1F3[4] GPIO pin 4 status			
			1F3[5] GPIO pin 5 status			
			1F3[6] GPIO pin 6 status			
			1F3[7] GPIO pin 7 status			

Table 4.12 VMI+ host port registers

I/O mapped port (hex)	Index (hex)	Memory mapped offset (hex)	Register	Bits	r/w	Reset
-	-	100	"VMI+ host port 0 control," on page 269	6	r/w	[7:0]
			100[0] port 0 enable			
			100[1] port 0 repeat			
			100[2] port 0 access type			
			100[3] port 0 retry			
			100[5:4] port 0 width			
-	-	101	"VMI+ host port 0 timing," on page 270	8	r/w	-
			101[3:0] port 0 command pulse width			
			101[7:4] port 0 time-out			
-	-	102	"VMI+ host port 0 index offset," on page 270	16	r/w	-
			102[15:0] port 0 index offset			
-	-	104	"VMI+ host port 1 control," on page 270	6	r/w	[7:0]
			104[0] port 1 enable			
			104[1] port 1 repeat			
			104[2] port 1 access type			
			104[3] port 1 retry			
			104[5:4] port 1 width			

Table 4.12 VMI+ host port registers

I/O mapped port (hex)	Index (hex)	Memory mapped offset (hex)	Register	Bits	r/w	Reset
-	-	105	"VMI+ host port 1 timing," on page 271 105[3:0] port 1 command pulse width 105[7:4] port 1 time-out	8	r/w	-
-	-	106	"VMI+ host port 1 index offset," on page 272 106[15:0] port 1 index offset	16	r/w	-

Table 4.13 THP interface registers

I/O mapped port (hex)	Index (hex)	Memory mapped offset (hex)	Register	Bits	r/w	Reset
-	-	110	THP control 110[1:0] 3Dfx THP interface mode	2	r/w	[1:0]
-	-	111	Reserved.	-	-	-
-	-	112	Slave request high timing 112[7:0] slave request high timing	8	r/w	-
-	-	113	Slave grant high timing 113[7:0] slave grant high timing	8	r/w	-
-	-	1F4–1F5	"Serial input," on page 274 1F4[15:0]serial input	16	r	-

Table 4.14VMI+ video port registers

I/O mapped port (hex)	Index (hex)	Memory mapped offset (hex)	Register		Bits	r/w	Reset
-	-	120–123	"VMI+ video port control," on page 275		32	r/w	-
			120[31]	invert TVCLK input			
			120[30]	internal active			
			120[29]	swap odd and even lines in memory (top even)			
			120[28]	swap U & V memory			
			120[27]	reset line counter on falling VSYNC			
			120[26]	increment line counter on rising HSYNC			
			120[25]	reset pixel counter on falling HSYNC			
			120[24]	sample ODD on falling VSYNC			
			120[23]	internal vertical blank			
			120[22]	internal horizontal blank			
			120[21]	internal odd field			
			120[20]	XVSYNC pin positive (active LOW)			
			120[19]	XHSYNC pin positive (active LOW)			
			120[18]	XODD pin positive (active LOW)			
			120[17]	source interlace			
			120[16]	divide clock by 2			
			120[15]	invert pixel qualifier			
			120[14:13]	reserved			
			120[12:9]	FIFO trip point			
			120[8:6]	decimation vertical			
			120[5:3]	decimation horizontal			
			120[2]	input averaging			
			120[1]	double buffering			
			120[0]	VMI+ video port enable			
-	-	124–125	"VMI+ video input port pitch," on page 277		32	r/w	-
			124[11:0]	VMI+ video port pitch ☞ bits 2:0 must be 0			
			124[31:12]	VMI+ video port pitch			
-	-	126	Reserved.		-	-	-
-	-	127	"VMI+ FIFO status," described on page 278				
			127[6:0]	reserved			
			127[7]	VMI+ FIFO status ☞ reset on read			
-	-	128–12A	"VMI+ video port base address 0," on page 278		19	r/w	-
			128[18:0]	base address 0 ☞ bits 2:0 must be 0			
-	-	12B	Reserved		-	-	-
-	-	12C–12E	"VMI+ video port base address 1," on page 279		19	r/w	-
			12C[18:0]	base address 0 ☞ bits 2:0 must be 0			
-	-	12F	Reserved		-	-	-
-	-	130–131	"Video input cropping boundary left," on page 279		10	r/w	-
			130[9:0]	left boundary, in TVCLKs			

Table 4.14VMI+ video port registers

I/O mapped port (hex)	Index (hex)	Memory mapped offset (hex)	Register	Bits	r/w	Reset
-	-	132–133	"Video input cropping boundary top," on page 280 132[9:0] top boundary, in TVCLKs	10	r/w	-
-	-	134–135	"Video input cropping boundary right," on page 280 134[9:0] right boundary, in TVCLKs	10	r/w	-
-	-	136–137	"Video input cropping boundary bottom," on page 281 136[9:0] bottom boundary in TVCLKs	10	r/w	-

Table 4.15 3D rendering engine registers

I/O mapped port (hex)	Index (hex)	Memory mapped offset (hex)	Register		Bits	r/w	Reset
-	-	300–303	"Polygon engine control 0," on page 282		32	r/w	[27:26] = reset on read
			300[2:1]	Quick start enable			
			300[3]	TLUT load cycle			
			300[4]	Texture enable			
			300[5]	Texture address jitter enable			
			300[6]	Texture data dithering enable			
			300[7]	Texture feed-forward dither			
			300[8]	Texture transparency enable			
			300[9]	Gouraud shading enable			
			300[10]	Lighting enable			
			300[11]	Fog enable			
			300[14:12]	Destination format			
			300[15]	Vertex alpha enable			
			300[16]	Z-buffer read			
			300[17]	Z-buffer write			
			300[18]	Z-buffer tiled			
			300[19]	MIPMap enable			
			300[20]	3D Clipping enable			
			300[21]	Disable low-angle line correction			
			300[23]	Gradient re-interpolation enable			
			300[25:24]	Bounding box control			
			300[27:26]	Bounding box check results			
			300[30:28]	Back buffer width			
			300[31]	Polygon start			
-	-	304–307	"Polygon engine control 1," on page 284		32	r/w	[27:26] = reset on read
			304[5]	Enable programmable gradient re-interpolation			
			304[6]	Texture mirror			
			304[7]	Texture clamp			
			304[8]	Texture address rounding disable			
			304[9]	Texture source alpha enable ☞ You must set both 304[9] and 30C[3] to enable source texture alpha			
			304[10]	Alpha polarity			
			304[11]	Vertex stack disable			
			304[12]	Disable 128 bit access			
			304[15:13]	Z compare mode			
			304[18:16]	Gouraud overlap timing - 8bpp			
			304[21:19]	Gouraud overlap timing - 16bpp			
			304[25:23]	Gouraud overlap timing - 32 bpp			
			304[26]	Disable hidden spanlet skip			
			304[27]	Disable U/V monotonicity clamp			
			304[30:28]	3D FIFO watermark			
			304[31]	Overlap Gouraud interpolate & write ☞ Must be 0 when textures are enabled.			
-	-	308–30A	"Texture map base address," on page 286		24	r/w	-
			308[23:0]	texture map base address			
-	-	30B	Reserved.		-	-	-

Table 4.153D rendering engine registers

I/O mapped port (hex)	Index (hex)	Memory mapped offset (hex)	Register	Bits	r/w	Reset
-	-	30C	"Texture format," on page 286 30C[2:0] Texel format 30C[3] Source texture alpha enable ☞ You must set both 30430C[9] and 30C30C[3] to enable source texture alpha 30C[7:5] Texture width 30C[10:8] Texture wrap height	11	r/w	-
-	-	30E	"Texel index offset," on page 288 30E[7:0] texel index format	8	r/w	-
-	-	30F	Reserved.	-	-	-
-	-	310	"3D internal register index," on page 288 310[2:0] Minor index 310[7:4] Major index	8	r/w	-
-	-	311	Reserved.	-	-	-
-	-	312	"Disable span delta calculation," on page 289 312[0] dY/dx 312[2] dZ/dx 312[3] dW/dx 312[4] dL/dx 312[5] dF/dx 312[6] dUW/dx 312[7] dWV/dx	8	-	-
-	-	313	Reserved.	-	-	-
-	-	314–317	"3D internal register data," on page 290 To read or write internal 3D engine registers, write the corresponding internal register index into the Index register and read or write the register value from the Data register.	32	r	-
-	-	318–31A	"Z buffer base pointer," on page 290	24	r/w	-
-	-	31B	Reserved.	-	-	-
-	-	31C–31D	"Z buffer front clipping plane," on page 291	16	r/w	-
-	-	31E–31F	"Z-buffer back clipping plane," on page 291	16	r/w	-
-	-	320–322	"Texel transparency color," on page 291	24	r/w	-
-	-	323	Reserved.	-	-	-
-	-	324–326	"Fog color," on page 292	24	r/w	-
-	-	327	Reserved.	-	-	-
-	-	328–32A	"Back buffer base address," on page 292	24	r/w	-

Table 4.15 3D rendering engine registers

I/O mapped port (hex)	Index (hex)	Memory mapped offset (hex)	Register	Bits	r/w	Reset
-	-	32B–32F	Reserved.	-	-	-
-	-	330–331	"3D clipping left," on page 293	12	-	-
-	-	332–333	"3D clipping top," on page 293	12	-	-
-	-	334–335	"3D clipping right," on page 293	12	-	-
-	-	336–337	"3D clipping bottom," on page 293	12	-	-
-	-	338–341	Reserved.	-	-	-

Table 4.16 Polygon vertex stack registers

I/O mapped port (hex)	Index (hex)	Memory mapped offset (hex)	Register	Bits	r/w	Reset
-	-	342–343	"Destination vertex X stack 0," on page 295	16	r/w	-
-	-	344–345	Reserved.	-	-	-
-	-	346–347	"Destination vertex Y stack 0," on page 296	16	r/w	-
-	-	348–349	Reserved.	-	-	-
-	-	34A–34B	"Destination vertex Z stack 0," on page 296	16	r/w	-
-	-	34C	Reserved.	-	-	-
-	-	34D	"Destination vertex W stack 0," on page 296	8	r/w	-
-	-	34E–34F	Reserved.	-	-	-
-	-	350	"Destination vertex L (lighting) stack 0," on page 297	8	r/w	-
-	-	351–352	Reserved.	-	-	-
-	-	353	"Destination vertex A (alpha) stack 0," on page 297	8	r/w	-
-	-	354	"Destination vertex F (fog) stack 0," on page 297	8	r/w	-
-	-	355–359	Reserved.	-	-	-
-	-	35A	"Source vertex U stack 0," on page 298	16	r/w	-

Table 4.16 Polygon vertex stack registers

I/O mapped port (hex)	Index (hex)	Memory mapped offset (hex)	Register	Bits	r/w	Reset
-	-	35E	"Source vertex V stack 0," on page 298	16	r/w	-
-	-	35F	Reserved.	-	-	-
-	-	360–37F	Polygon stack 1 registers	16	r	-
-	-	380–39F	Polygon stack 2 registers	16	r	-

Table 4.17 Texture scale registers

I/O mapped port (hex)	Index (hex)	Memory mapped offset (hex)	Register	Bits	r/w	Reset
-	-	3C0–3C1	"U factor," on page 299	16	-	-
-	-	3C2–3C3	"U offset," on page 299	9	-	-
-	-	3C4–3C5	"V factor," on page 299	16	-	-
-	-	3C6–3C7	"V offset," on page 300	9	-	-
-	-	3C8	"Gradient re-interpolation count," on page 300	4	-	Fh

Table 4.18 Test Registers

I/O mapped port (hex)	Index (hex)	Memory mapped offset (hex)	Register	Bits	r/w	Reset
-	-	0B4	"Signature analyzer control," described on page 301 0B4[3:2] signature select 0B4[0] signature start/clear	3	r/w	-
-	-	0B5	"Signature value," described on page 302 0B5[23:0] signature value	24	r/w	-

5. Pin description

Table 5.1 PCI bus host interface

Signal name	Pin #	I/O	Drive	Description
IDSEL	95	I		Host address high byte is zero.
$\overline{\text{STOP}}$	96	O	12 mA TS	Asserted by ProMotion to retry or abort a cycle.
AD[31:00]‡	120-124, 126-136, 139-149, 151-155	I/O	12 mA TS	Host address/data bus.
C/$\overline{\text{BE}}$[3:0]	91-94	I	-	Command/byte enable.
$\overline{\text{RST}}$	62	I	-	System reset.
CLK	105	I	-	PCI clock.
$\overline{\text{LOCK}}$	90	I	-	Locked access. Asserted by initiator to lock ProMotion-AT3D.
PAR	89	I/O	12 mA	Parity. ProMotion computes and drives parity for all host reads.
FRAME	88	I	-	Cycle frame. Asserted by the host for the duration of an access.
$\overline{\text{IRDY}}$	87	I	-	Initiator ready. Asserted by the host when it is ready to transmit or receive data.
$\overline{\text{TRDY}}$	86	I/O	12 mA TS	Target ready. Asserted by ProMotion when it is ready to transmit or receive data.
DEVSEL	85	O	12 mA	Local device. Asserted by ProMotion when it identifies itself as target of a PCI bus cycle.
$\overline{\text{INTA}}$	84	O	8 mA	Interrupt request.

‡ Shared function pins. Refer to Table 5.13, "ProMotion-AT3D multi-function pins, PCI bus," on page 54. Some pins may not be available when certain features are implemented.

Table 5.2 DRAM interface

Signal name	Pin #	I/O	Drive	Description
MD[63:0]‡	156-165, 168-183, 186-187, 189-206, 1-18	I	4 mA TS	DRAM data.
MA[8, 0, 6:1, 7]	35-39, 41-44	O	8 mA TS	DRAM row and column address.
$\overline{\text{RAS}}$[1:0]	24	O	12 mA TS	Row address strobe.
$\overline{\text{CAS}}$[7:0]	25-29, 32-34	O	12 mA TS	Byte-wise $\overline{\text{CAS}}$ control. Drives per-byte $\overline{\text{CAS}}$ lines. In 1MB configurations use only $\overline{\text{CAS}}$[3:0].

‡ Shared function pins. Refer to Table 5.13, "ProMotion-AT3D multi-function pins, PCI bus," on page 54. Some pins may not be available when certain features are implemented.

Table 5.2 DRAM interface

Signal name	Pin #	I/O	Drive	Description
OE[1:0]	20-21	O	12 mA TS	Output enable. OE[0] selects the first 2MB bank of DRAM and $\overline{OE}$[1] selects the second 2MB bank of DRAM, if any.
$\overline{WE}$[1:0]	22-23	O	12 mA TS	Bank-wise $\overline{WE}$ control. Drives $\overline{WE}$. $\overline{WE}$[0] selects the first 2MB bank of DRAM and WE[1] selects the second 2MB bank of DRAM, if any.

‡ Shared function pins. Refer to Table 5.13, "ProMotion-AT3D multi-function pins, PCI bus," on page 54. Some pins may not be available when certain features are implemented.

Table 5.3 THP arbitration

Signal name	Pin #	I/O	Drive	Description
$\overline{3REQ}$‡	108	I/O	8 mA TS	THP access request. *AT3D slave*: O, request from AT3D to host for THP access. *AT3D host:* I, request from external device to AT3D for THP access.
$\overline{3GNT}$	106	I/O	8 mA TS	THP grant. *AT3D slave:* I, host grants AT3D THP access. *AT3D host*: O, AT3D grants external device THP access.
3CLK	107	O	8 mA TS	THP clock. Synchronizes $\overline{3REQ}$ and $\overline{3GNT}$ signals.
SERIAL_IN‡	119	I	-	Serial input from THP device.
SWAP‡	118	I	-	Buffer swap input from THP device.
The THP data bus connects to MD[31:0].				

‡ Shared function pins. Refer to Table 5.13, "ProMotion-AT3D multi-function pins, PCI bus," on page 54. Some pins may not be available when certain features are implemented.

Table 5.4 Monitor/display interface

Signal name	Pin #	I/O	Drive	Description
HSYNC	64	O	12 ma TS	Horizontal sync to monitor and feature connector (programmable).
$\overline{VSYNC}$	65	I/O	12 ma TS	Vertical sync to monitor and from/to feature connector (programmable).
R, G, B	51, 49, 47	O	-	Red, green, and blue analog outputs to monitor.
LEFT‡	117	O	4 ma TS	Stereo glasses driver (HIGH = display left eye).
SDA	98	I/O	8 ma TS	DDC/I^2C channel to/from monitor.
SCL	99	I/O	8 ma TS	DDC/I^2C clock to monitor.

‡ Shared function pins. Refer to Table 5.13, "ProMotion-AT3D multi-function pins, PCI bus," on page 54. Some pins may not be available when certain features are implemented.

Table 5.5 Feature connector interface: VSVPC mode

Signal name	*Pin #*	*I/O*	*Drive*	*Description*
$\overline{\text{EXVID}}$‡	66	I	-	External video. Places ProMotion-AT3D P[15:0] lines in high-impedance mode, so external device can drive DAC pixel data bus.
EXPCLK‡	67	I	-	External clock. Places ProMotion-AT3D PCLK in high-impedance mode, so external device can drive DAC pixel clock.
$\overline{\text{EXSYNC}}$‡	68	I	-	External sync. Places ProMotion-AT3D $\overline{\text{HSYNC}}$, VSYNC, and $\overline{\text{BLANK}}$ signals in high-impedance mode, so external devices can drive them.
PCLK‡	70	I/O	4 mA TS	Pixel clock to/from feature connector.
P[15:0]‡	112-119, 73-78, 81-82	I/O	4 mA TS	Pixel data to/from feature connector.
$\overline{\text{BLANK}}$‡	72	I/O	4 mA TS	Blank signal to/from feature connector.

‡ Shared function pins. Refer to Table 5.13, "ProMotion-AT3D multi-function pins, PCI bus," on page 54. Some pins may not be available when certain features are implemented.

Table 5.6 Feature connector interface: VAFC mode

Signal name	*Pin #*	*I/O*	*Drive*	*Description*
DCLK‡	70	O	4 mA TS	Dot clock. Equal to PCLK or PCLK/2 depending on state of VAFC control register.
GRDY‡	68	O	4 mA TS	Graphics ready. Signals that external pixel has been accepted.
$\overline{\text{EGEN}}$‡	67	I	-	Enable GENCLK to drive in place of VCLK.
GENCLK‡	66	I	-	Genlock clock from feature connector.
P[15:0]	112-119, 73-78, 81-82	I/O	4 mA TS	Pixel data to/from feature connector.
$\overline{\text{BLANK}}$‡	72	O	4 mA TS	Blank signal to feature connector.
$\overline{\text{EVIDEO}}$	111	I	-	External video enable. Inputs P signal and merges into RAMDAC.

‡ Shared function pins. Refer to Table 5.13, "ProMotion-AT3D multi-function pins, PCI bus," on page 54. Some pins may not be available when certain features are implemented.

Table 5.7 VMI+ video input port

Signal name	*Pin #*	*I/O*	*Drive*	*Description*
$\overline{\text{VREF}}$‡	67	I	-	External VSYNC. Generated by external device.
$\overline{\text{HREF}}$‡	72	I	-	External HSYNC. Generated by external device.
XODD	68	I/O	4 mA TS	Odd field. Used for interlaced input.
PIXCLK‡	70	I	-	Pixel clock from external device.

‡ Shared function pins. Refer to Table 5.13, "ProMotion-AT3D multi-function pins, PCI bus," on page 54. Some pins may not be available when certain features are implemented.

Table 5.7 VMI+ video input port

Signal name	Pin #	I/O	Drive	Description
VACTIVE‡	66	I	-	Blank signal / active pixel qualifier from external device.
VID[15:0]‡	112-119, 73-78, 81-82	I	4 mA TS	Video from external device. High byte contains Y component and low byte contains U/V components, with U component transmitted first.

‡ Shared function pins. Refer to Table 5.13, "ProMotion-AT3D multi-function pins, PCI bus," on page 54. Some pins may not be available when certain features are implemented.

Table 5.8 VMI+ host interface

Signal name	Pin #	I/O	Drive	Description
$\overline{CS}$[2:1]‡	108, 100	O	4 mA TS	External chip select for 2 separate devices.
XHA[15:0]‡	168–173 156–165	O	12 mA TS	VMI address bus.
XHD[15:0]‡	3–18	I/O	4 mA TS	VMI+ data bus.
RESET	62	I	-	System reset.
READY	103	I	-	External device ready.
$\overline{WR}$	102	O	4 mA TS	External I/O Write. Equivalent to ISA $\overline{IOW}$ signal or non-ISA R/$\overline{W}$ signal.
$\overline{RD}$	101	O	4 mA TS	External I/O Read. Equivalent to ISA $\overline{IOR}$ signal.
$\overline{XBUF}$	104	O	4 mA TS	External buffer enable. Drives external '244 buffers to read data.

The VMI+ host address bus connects to MA[8:0] and VMI+ host data bus connects to MD[31:16].

‡ Shared function pins. Refer to Table 5.13, "ProMotion-AT3D multi-function pins, PCI bus," on page 54. Some pins may not be available when certain features are implemented.

Table 5.9 ROM BIOS interface

Signal name	Pin #	I/O	Drive	Description
$\overline{ROMEN}$	63	O	4 mA	External ROM enable.
$\overline{ROMWR}$	97	O	4 mA TS	ROM write. Used to write a Flash EPROM. Figure 2.8, "ROM BIOS interface," on page 14, for more information on Flash EPROM.
ROMADD[15:0]‡	156-165, 168-173	I/O	4 mA TS	BIOS ROM address.
ROMDAT[7:0]‡	182-183, 186-187, 189-192	I/O	4 mA TS	BIOS ROM data.

‡ Shared function pins. Refer to Table 5.13, "ProMotion-AT3D multi-function pins, PCI bus," on page 54. Some pins may not be available when certain features are implemented.

Table 5.10General purpose I/O interface

Signal name	*Pin #*	*I/O*	*Drive*	*Description*
GPIO[7:0]‡	112-119	I/O	4 mA TS	General-purpose I/O.
Each GPIO pin independently may be input or an output, may be driven high, low, or high impedance.				
‡ Shared function pins. Refer to Table 5.13, "ProMotion-AT3D multi-function pins, PCI bus," on page 54. Some pins may not be available when certain features are implemented.				

Table 5.11Analog interface

Signal name	*Pin #*	*I/O*	*Drive*	*Description*
RSET	55	I	-	Full scale adjust. Connect a resistor between this pin and AGND to set full-scale intensity of the DACs.
AVREF	45	-	-	Analog voltage reference.
COMP, COMPRTN	53, 54	I	-	Compensation pins. Connect capacitor between these pins.
XTALI	58	I	-	XTALamp. Connect a 14.31818 MHz crystal between XTALI and XTALO.
XTALO	60	O	-	XTALamp. Connect a 14.31818 MHz crystal between XTALI and XTALO.
VAA	48, 52	-	-	Power to DAC.
AGND	46, 50	-	-	Ground to DAC.
VPP	56, 57	-	-	Power to PLL.
PGND	59, 61	-	-	Ground to PLL.
‡ Shared function pin. Refer to Table 5.13, "ProMotion-AT3D multi-function pins, PCI bus," on page 54. Some pins may not be available when certain features are implemented.				

Table 5.12Power/ground pins

Signal Name	*Pin #*	*I/O*	*Drive*	*Description*
VCC	30, 71, 80, 137, 166, 184, 207	-	-	Power.
GND	19, 31, 40, 69, 79, 83, 125, 138, 150, 167, 185, 188, 208	-	-	Ground.

Table 5.13 ProMotion-AT3D multi-function pins, PCI bus

AT3D pin number	THP connector	Feature connector		VMI+				Other shared function
		VSVPC	VAFC	input port	host interface	BIOS ROM	General I/O	
1–2	MD[17:16]							MD[17:16]
3–18	$\overline{MD}$[15:0]				XHD[15:0]			$\overline{MD}$[15:0]
20–21	$\overline{OE}$[1:0]							$\overline{OE}$[1:0]
22–23	$\overline{WE}$[1:0]							$\overline{WE}$[1:0]
24	$\overline{RAS}$[0]							$\overline{RAS}$[0]
25–29	$\overline{CAS}$[7:0]							$\overline{CAS}$[7:0]
62	$\overline{HRESET}$				$\overline{RESET}$			$\overline{RST}$
64	$\overline{HSYNC}$			$\overline{HSYNC}$				$\overline{HSYNC}$
65	VSYNC			VSYNC				VSYNC
66		$\overline{EXVID}$	GENCLK	VACTIVE				
67		$\overline{EXPCLK}$	$\overline{EGEN}$	VREF				
68		$\overline{EXSYNC}$	GRDY					XODD
70		PCLK	DCLK	PIXCLK				
72		$\overline{BLANK}$		HREF				
73		P[7]		VID[7]				
74		P[6]		VID[6]				
75		P[5]		VID[5]				
76		P[4]		VID[4]				
77		P[3]		VID[3]				
78		P[2]		VID[2]				
81		P[1]		VID[1]				
82		P[0]		VID[0]				
100					$\overline{CS}$[1]			
101					$\overline{RD}$			
102					$\overline{WR}$			
103					READY			
104					XBUF			
108	3$\overline{REQ}$				$\overline{CS}$[2]			
109	$\overline{RAS}$[1]							$\overline{RAS}$[1]
112		P[8]					GPIO[0]	
113		P[9]					GPIO[1]	
114		P[10]					GPIO[2]	
115		P[11]					GPIO[3]	
116	SRESET	P[12]					GPIO[4]	
117		P[13]					GPIO[5]	LEFT

Note: This table does not display single-function AT3D pins. Refer to Table 5.15, "AT3D pin numbers, PCI bus," on page 58 for a complete pin listing.

Table 5.13 ProMotion–AT3D multi–function pins, PCI bus

AT3D pin number	THP connector	Feature connector		VMI+				Other shared function
		VSVPC	VAFC	input port	host interface	BIOS ROM	General I/O	
118	SWAP	P[14]					GPIO[6]	
119	SERIAL_IN	P[15]					GPIO[7]	
156–165 168–173	MD[63:48]				XHA[15:0]	ROMADD [15:0]		MD[63:48]
182–183 186–187 189–192	MD[39:32]					ROMDAT [7:0]		MD[39:32]
193–206	MD[31:18]							MD[31:18]

Note: This table does not display single-function AT3D pins. Refer to Table 5.15, "AT3D pin numbers, PCI bus," on page 58 for a complete pin listing.

Table 5.14ProMotion-AT3D/AT24/6422 pin deltas (PCI bus)

Pin #	AT3D	AT24	6422
3	MD[15]/XHD[15]	MD[15]	
4	MD[14]/XHD[14]	MD[14]	
4	MD[13]/XHD[13]	MD[13]	
6	MD[12]/XHD[12]	MD[12]	
7	MD[11]/XHD[11]	MD[11]	
8	MD[10]/XHD[10]	MD[10]	
9	MD[09]/XHD[09]	MD[09]	
10	MD[08]/XHD[08]	MD[08]	
11	MD[07]/XHD[07]	MD[07]	
12	MD[06]/XHD[06]	MD[06]	
13	MD[05]/XHD[05]	MD[05]	
14	MD[04]/XHD[04]	MD[04]	
15	MD[03]/XHD[03]	MD[03]	
16	MD[02]/XHD[02]	MD[02]	
17	MD[01]/XHD[01]	MD[01]	
18	MD[00]/XHD[00]	MD[00]	
24	$\overline{RAS}$[0]	$\overline{RAS}$	
66	GENCLK/$\overline{XBLANK}$	$\overline{EXVID}$	
67	EXVID/GENCLK/VACTIVE	EXPCLK	
68	$\overline{EXPCLK}$/EGEN/VREF	$\overline{EXSYNC}$	
70	PCLK/DCLK/PIXCLK	PCLK	
72	$\overline{BLANK}$/$\overline{HREF}$	$\overline{BLANK}$	
100	$\overline{CS}$[1]	NC	
101	$\overline{RD}$	NC	
102	WR	NC	
103	READY	NC	
104	$\overline{XBUF}$	NC	
106	$\overline{3GNT}$	NC	
107	$\overline{3CLK}$	NC	
108	$\overline{3REQ}$/$\overline{CS}$[2]	NC	
109	RAS[1]	NC	
111	$\overline{EVIDEO}$		NC
112	P[8]/GPIO[0]		NC
113	P[9]/GPIO[1]		NC
114	P[10]/GPIO[5]		NC
115	P[11]/GPIO[3]		NC
116	P[12]/GPIO[4]/SRESET	P[12]/GPIO[4]	NC

Refer to Table 5.15 for a complete list of AT3D pins.

Table 5.14ProMotion-AT3D/AT24/6422 pin deltas (PCI bus)

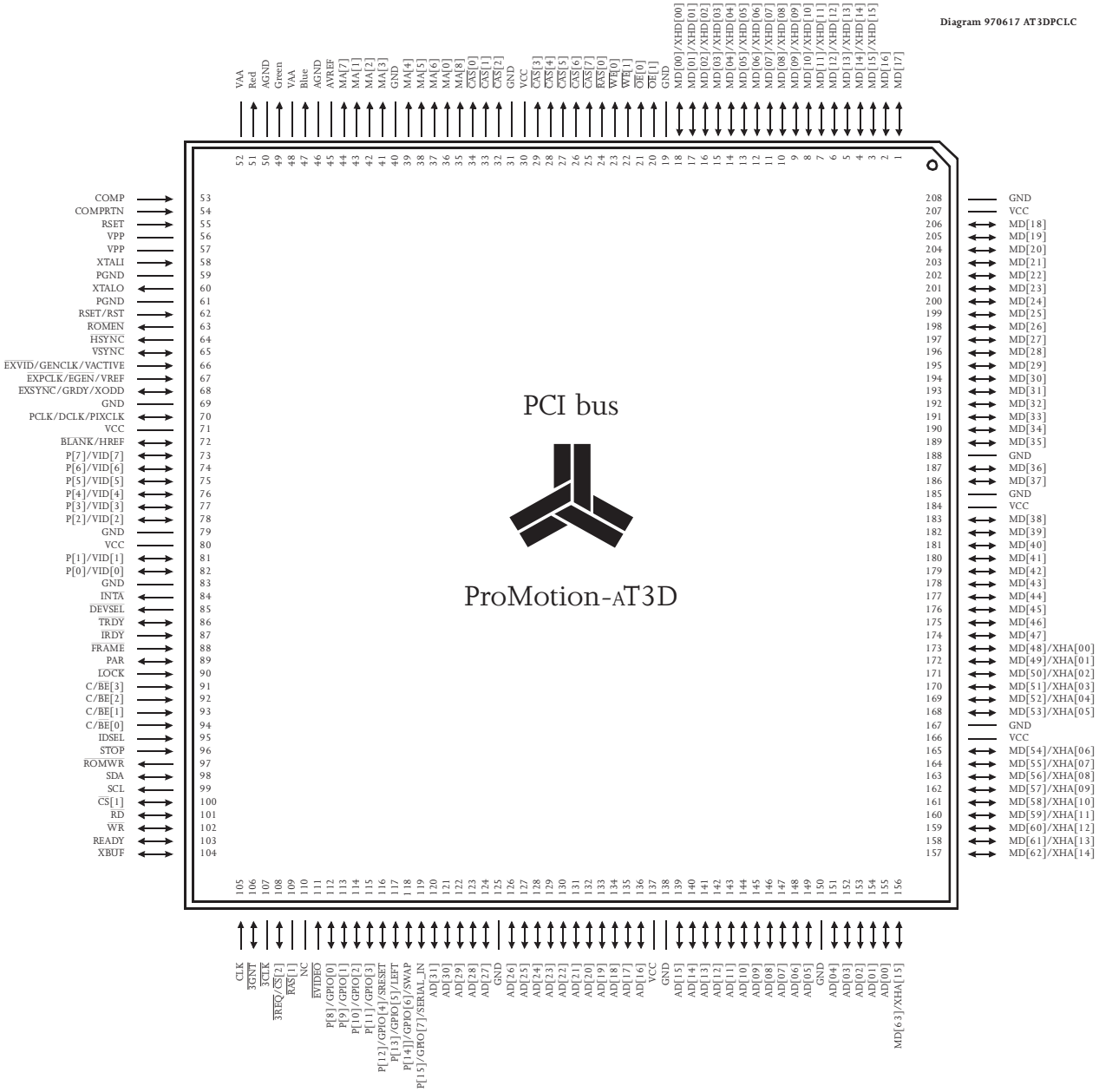
Pin #	AT3D	AT24	6422
117	P[13]/GPIO[5]/LEFT	P[13]	NC
118	P[14]/GPIO[6]/SWAP	P[14]	NC
119	P[15]/GPIO[7]/SERIAL_IN	P[15]	NC
156	MD[63]/ROMADD[15]/XHA[15]	MD[63]/ROMADD[15]	MD[63]
157	MD[62]/ROMADD[14]/XHA[14]	MD[62]/ROMADD[14]	MD[62]
158	MD[61]/ROMADD[13]/XHA[13]	MD[61]/ROMADD[13]	MD[61]
159	MD[60]/ROMADD[12]/XHA[12]	MD[60]/ROMADD[12]	MD[60]
160	MD[59]/ROMADD[11]/XHA[11]	MD[59]/ROMADD[11]	MD[59]
161	MD[58]/ROMADD[10]/XHA[10]	MD[58]/ROMADD[10]	MD[58]
162	MD[57]/ROMADD[09]/XHA[09]	MD[57]/ROMADD[09]	MD[57]
163	MD[56]/ROMADD[08]/XHA[06]	MD[56]/ROMADD[08]	MD[56]
164	MD[55]/ROMADD[07]/XHA[07]	MD[55]/ROMADD[07]	MD[55]
165	MD[54]/ROMADD[06]/XHA[06]	MD[54]/ROMADD[06]	MD[54]
166	MD[53]/ROMADD[05]/XHA[05]	MD[53]/ROMADD[05]	MD[53]
169	MD[52]/ROMADD[04]/XHA[04]	MD[52]/ROMADD[04]	MD[52]
170	MD[51]/ROMADD[03]/XHA[03]	MD[51]/ROMADD[03]	MD[51]
171	MD[50]/ROMADD[02]/XHA[02]	MD[50]/ROMADD[02]	MD[50]
172	MD[49]/ROMADD[01]/XHA[01]	MD[49]/ROMADD[01]	MD[49]
173	MD[48]/ROMADD[00]/XHA[00]	MD[48]/ROMADD[00]	MD[48]
182	MD[39]/ROMDAT[7]		MD[39]
183	MD[38]/ROMDAT[6]		MD[39]
186	MD[37]/ROMDAT[5]		MD[39]
187	MD[36]/ROMDAT[4]		MD[39]
189	MD[35]/ROMDAT[3]		MD[39]
190	MD[34]/ROMDAT[2]		MD[39]
191	MD[33]/ROMDAT[1]		MD[39]
192	MD[32]/ROMDAT[0]		MD[39]

Refer to Table 5.15 for a complete list of AT3D pins.

Table 5.15 AT3D pin numbers, PCI bus

Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal
1	MD[17]	53	COMP	105	CLK	157	MD[62]/ROMADD[14]/XHA[14]
2	MD[16]	54	COMPRTN	106	3GNT	158	MD[61]/ROMADD[13]/XHA[13]
3	MD[15]/XHD[15]	55	RSET	107	3CLK	159	MD[60]/ROMADD[12]/XHA[12]
4	MD[14]/XHD[14]	56	VPP	108	3REQ/CS[2]	160	MD[59]/ROMADD[11]/XHA[11]
5	MD[13]/XHD[13]	57	VPP	109	RAS[1]	161	MD[58]/ROMADD[10]/XHA[10]
6	MD[12]/XHD[12]	58	XTALI	110	NC	162	MD[57]/ROMADD[09]/XHA[09]
7	MD[11]]/XHD[11]	59	PGND	111	EVIDEO	163	MD[56]/ROMADD[08]/XHA[08]
8	MD[10]/XHD[10]	60	XTALO	112	P[8]/GPIO[0]	164	MD[55]/ROMADD[07]/XHA[07]
9	MD[09]/XHD[09]	61	PGND	113	P[9]/GPIO[1]	165	MD[54]/ROMADD[06]/XHA[06]
10	MD[08]/XHD[08]	62	RESET/RST	114	P[10]/GPIO[2]	166	VCC
11	MD[07]/XHD[07]	63	ROMEN	115	P[11]/GPIO[3]	167	GND
12	MD[06]/XHD[06]	64	HSYNC	116	P[12]/GPIO[4]/SRESET	168	MD[53]/ROMADD[05]/XHA[05]
13	MD[05]/XHD[05]	65	VSYNC	117	P[13]/GPIO[5]/LEFT	169	MD[52]/ROMADD[04]/XHA[04]
14	MD[04]/XHD[04]	66	EVIDEO/GENCLK/VACTIVE	118	P[14]/GPIO[6]/SWAP	170	MD[51]/ROMADD[03]/XHA[03]
15	MD[03]/XHD[03]	67	EDCLK/EGEN/VREF	119	P[15]/GPIO[7]/SERIAL_IN	171	MD[50]/ROMADD[02]/XHA[02]
16	MD[02]/XHD[02]	68	ESYNC/GRDY/XODD	120	AD[31]	172	MD[49]/ROMADD[01]/XHA[01]
17	MD[01]/XHD[01]	69	GND	121	AD[30]	173	MD[48]/ROMADD[00]/XHA[00]
18	MD[00]/XHD[00]	70	PCLK/DCLK/PIXCLK	122	AD[29]	174	MD[47]
19	GND	71	VCC	123	AD[28]	175	MD[46]
20	OE[1]	72	BLANK/HREF	124	AD[27]	176	MD[45]
21	OE[0]	73	P[7]/VID[7]	125	GND	177	MD[44]
22	WE[1]	74	P[6]/VID[6]	126	AD[26]	178	MD[43]
23	WE[0]	75	P[5]/VID[5]	127	AD[25]	179	MD[42]
24	RAS[0]	76	P[4]/VID[4]	128	AD[24]	180	MD[41]
25	CAS[7]	77	P[3]/VID[3]	129	AD[23]	181	MD[40]
26	CAS[6]	78	P[2]/VID[2]	130	AD[22]	182	MD[39]
27	CAS[5]	79	GND	131	AD[21]	183	MD[38]
28	CAS[4]	80	VCC	132	AD[20]	184	VCC
29	CAS[3]	81	P[1]/VID[1]	133	AD[19]	185	GND
30	VCC	82	P[0]/VID[0]	134	AD[18]	186	MD[37]
31	GND	83	GND	135	AD[17]	187	MD[36]
32	CAS[2]	84	INTA	136	AD[16]	188	GND
33	CAS[1]	85	DEVSEL	137	VCC	189	MD[35]
34	CAS[0]	86	TRDY	138	GND	190	MD[34]
35	MA[8]	87	IRDY	139	AD[15]	191	MD[33]
36	MA[0]	88	FRAME	140	AD[14]	192	MD[32]
37	MA[6]	89	PAR	141	AD[13]	193	MD[31]
38	MA[5]	90	LOCK	142	AD[12]	194	MD[30]
39	MA[4]	91	C/BE[3]	143	AD[11]	195	MD[29]
40	GND	92	C/BE[2]	144	AD[10]	196	MD[28]
41	MA[3]	93	C/BE[1]	145	AD[09]	197	MD[27]
42	MA[2]	94	C/BE[0]	146	AD[08]	198	MD[26]
43	MA[1]	95	IDSEL	147	AD[07]	199	MD[25]
44	MA[7]	96	STOP	148	AD[06]	200	MD[24]
45	REFV	97	ROMWR	149	AD[05]	201	MD[23]
46	AGND	98	SDA	150	GND	202	MD[22]
47	Blue	99	SCL	151	AD[04]	203	MD[21]
48	VAA	100	CS[1]	152	AD[03]	204	MD[20]
49	Green	101	RD	153	AD[02]	205	MD[19]
50	AGND	102	WR	154	AD[01]	206	MD[18]
51	Red	103	READY	155	AD[00]	207	VCC
52	VAA	104	XBUF	156	MD[63]/ROMADD[00]/XHA[00]	208	GND

Figure 5.15. Pin diagram, PCI bus



6. Configuration straps

At release of power-on RESET ProMotion-AT3D configuration bits are latched from MD lines. These I/O pins are internally pulled to a weak HIGH state. To override default configuration add a weak pulldown resistor (5.1KΩ recommended).

Table 6.1 ProMotion-AT3D configuration straps

Signal name	MD	Description	Offset [bit]	r/w[1]
MAPOUT	31	Pull down to map out ROM for motherboard applications.	0C2[3], 3C5.30[7] (inverted)	r/w
INTLV	30	Pull down for interleaved memory in 4 MB configurations (non EDO). This strap normally overridden by BIOS.	0C4[0]	r/w
MULTWE	29	Default multiple-$\overline{CAS}$ array. Do not pull down.	0C4[4]	r/w
$\overline{PCI}$	27	PCI configuration. This strap must be pulled down.	0CA[0]	r
LDEVTS	25	Pull down for wired-OR LDEV. This strap must be pulled down.	0CA[2]	r
ALTPCI	24	Pull down for alternate PCI device ID.	1C0[2]	r/w
FASTRAS	22	Pull down for extended RAS (fast RAS disable). Default is fast RAS enabled.[2]	0C4[1]	r/w
BYPASS	21	Pull down to bypass on-chip clock generators.	0EC[1:0]	r/w
MCLK	20:18	MCLK speed select. Status of this field is available to software.	0EB[2:0]	r/w
VAFC	15	Pull down for VAFC feature connector.	0CC[0]	r/w
16FC	14	Pull down for 16-bit feature connector.	0CE[4]	r/w
MEMTYPE	13	Pull down for non-EDO DRAM. This strap normally overridden by BIOS.	0C4[9]	r/w
SCPM	12	Pull down for single-cycle page mode for EDO DRAM. This strap normally overridden by BIOS.	0C4[10]	r/w
INTPIN	11	Pull down to set PCI interrupt pin configuration register to read back value of 1. Default reads value of 0.	1BD[0]	r
PCI33	10	Pull down for PCI = 66 MHz.	0CA[3]	r
UMA	9	Pull down for UMA / PUMA.	110[1]	r/w
SFB	8	Pull down for shared frame buffer / PUMA.	110[0]	r/w
FCDIS	7	Pull down for feature connector disable/TV input enable.	0CC[2]	r/w
INPUTS[6:0]	6:0	OEM configurable inputs.	3C5.20[6:0]	r/w

[1] W in this column indicates strap may be overridden by BIOS software.

[2] Alliance recommends extended RAS for MCLK rates >50 MHz. Fast RAS is recommended for MCLK rates =<50 MHz, with DRAM access 70 ns or faster. Refer to "Page mode DRAM: read/write," on page 72.

Table 6.2 Reserved AT3D configuration straps

Signal name	MD	Description	Offset [bit]	r/w[1]
-	28	Reserved.	0C6[2]	-
-	26	Reserved.	0CA[1]]	-
-	17:16	Reserved.	0C4[8:7]	r/w

[1] W in this column indicates strap may be overridden by BIOS software.

Table 6.3 ProMotion family configuration strap deltas

MD	AT3D+AGP	AT3D	AT24	6422	6410	3210
MD31	MAPOUT					
MD30	Reserved	INTLV				
MD29	Reserved	MULTWE				
MD28	Reserved				DUALPCLK	
MD27	Reserved	VL/$\overline{\text{PCI}}$				
MD26	Reserved				DAC16	
MD25	Reserved	LDEVTS				
MD24	ALTPCI		SEL3C3			
MD23	Reserved	DUALRAS				
MD22	FASTRAS					
MD21	BYPASS				Reserved	
MD20:18	Reserved	MCLK			Reserved	
MD17	Reserved		BHALF		Reserved	
MD16	MEM64				Reserved	
MD15	VAFC				Reserved	
MD14	Reserved	16FC			Reserved	
MD13	MEMTYPE			Reserved		
MD12	SCPM			Reserved		
MD11	INTPIN					Reserved
MD10	PCI33			Reserved		
MD9	UMA		Reserved			
MD8	SFB		Reserved			
MD7	FCDIS/INPUT[7]		INPUT[7]	Reserved		
MD6:0	INPUT[6:0]			Reserved		

7. Electrical characteristics

Table 7.1 Absolute maximum ratings

Symbol	Parameter	Rating	Unit
T_{stg}	Storage temperature	-65 to 150	° C
V_{IN}	Voltage on any pin	-0.5 to +6.5	Volts
P_D	Operating power dissipation	1.5	Watts
V_a	Power supply voltage	7.0	Volts
I_{OUT}	DC output current (per pin)	20	mA
-	Injection current (latch up testing)	100	mA

NOTE: Stresses greater than those listed under *Absolute Maximum Ratings* may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions outside those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect reliability.

Table 7.2 Recommended operating conditions

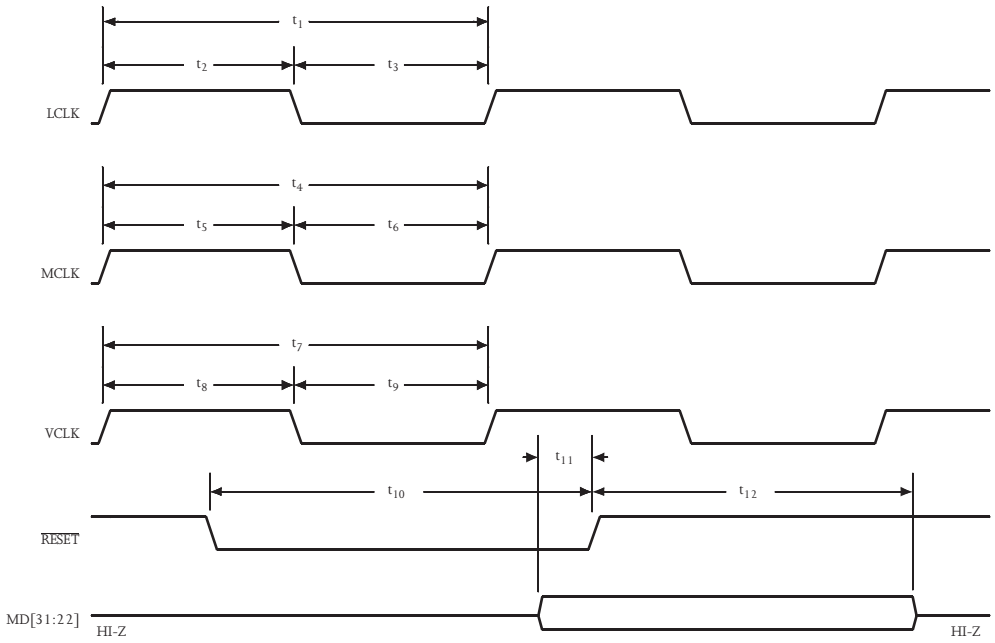
Symbol	Parameter	Test conditions	Min	Max	Unit
T_a	Ambient temperature	Normal operation	0	70	° C
V_{CC}	Power supply voltage	Normal operation	4.75	5.25	Volts
V_{IL}	Input low voltage		0	0.8	Volts
V_{IH}	Input high voltage		2.0	V_{CC} +0.5	Volts
V_{OL}	Output low voltage	I_{OL} = 4 mA	-	0.4	Volts
V_{OH}	Output high voltage	I_{OH} = 400 µA	2.4	-	Volts
I_{CC}	Supply current	V_{CC} = MAX	-	300	mA
I_{IH}	Input high current	$V_{IL} = V_{CC}$	-	10	µA
I_{IL}	Input low current	V_{CC} = 5.25 V, V_{IL} = -0.5 V	-10	-	µA
I_{OL}	Input leakage	$0 < V_{IN} < V_{CC}$	-10	10	µA
C_{IN}	Input capacitance		-	10	pF
C_{OUT}	Output capacitance		-	10	pF

8. AC timing

8.1 Clock and reset timing

Waveform 8.1.1. Clock and reset timing

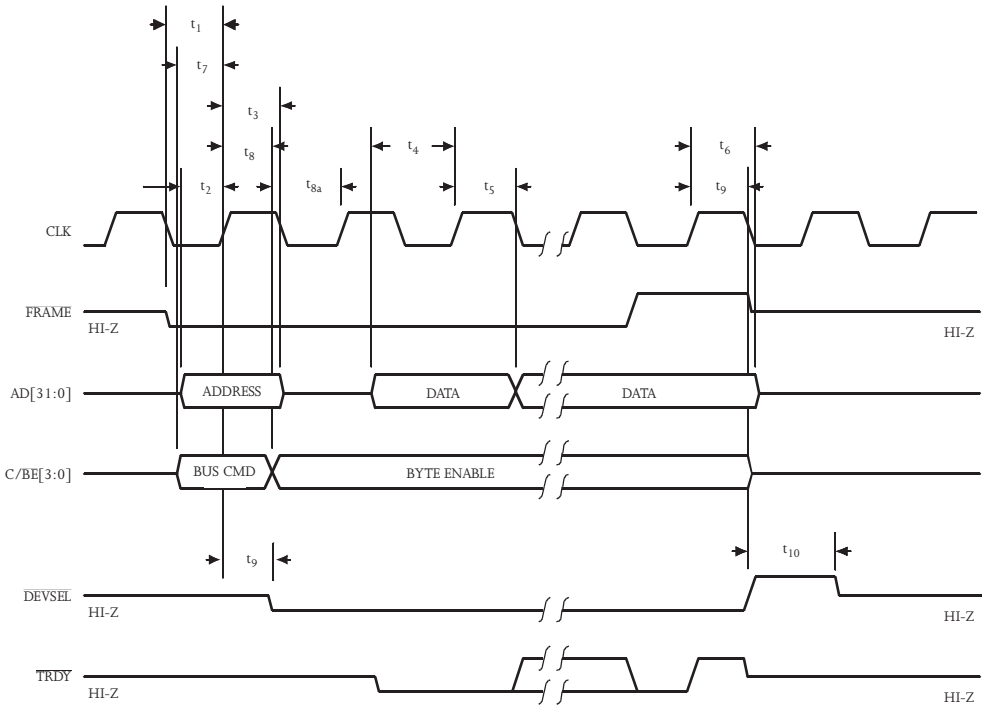
Symbol	Parameter	Min	Max	Unit
t_1	LCLK period	20	-	ns
t_2	LCLK high period	8	-	ns
t_3	LCLK low period	8	-	ns
t_4	MCLK period	18	-	ns
t_5	MCLK high period	7	-	ns
t_6	MCLK low period	7	-	ns
t_7	VCLK period	9	-	ns
t_8	VCLK high period	4	-	ns
t_9	VCLK low period	4	-	ns
t_{10}	$\overline{\text{RESET}}$ pulse width	400	-	ns
t_{11}	MD strap setup to $\overline{\text{RESET}}$ inactive	10	-	ns
t_{12}	MD strap hold from $\overline{\text{RESET}}$ inactive	5	-	ns



8.2 Host interface timing

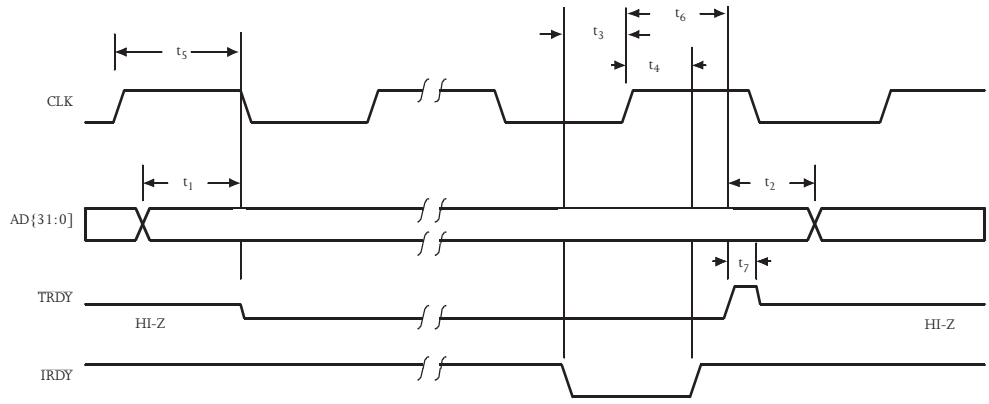
Waveform 8.2.1. PCI timing: FRAME, DEVSEL, AD[31:0], C/BE[3:0]

Symbol	Parameter	66 MHz Min	66MHz Max	33 MHz Min	33 MHz Max	Unit
t_1	FRAME setup to CLK	3	-	7	-	ns
t_2	AD[31:0] (address) setup to CLK	3	-	7	-	ns
t_3	AD[31:0] (address) hold from CLK	0	-	0	-	ns
t_4	AD[31:0] (data) setup to CLK	3	-	7	-	ns
t_5	AD[31:0] (data) hold from CLK	0	-	0	-	ns
t_6	AD[31:0] C/BE[3:0] HI-Z from CLK	0	14	0	28	ns
t_7	C/BE[3:0] (bus CMD) setup to CLK	3	-	7	-	ns
t_8	C/BE[3:0] (bus CMD) hold from CLK	0	-	0	-	ns
t_{8a}	C/BE[3:0] (byte enable) setup to CLK	3	-	7	-	ns
t_9	DEVSEL delay from CLK	-	6	-	11	ns
t_{10}	DEVSEL high before HI-Z	1 CLK	-	1 CLK	-	ns



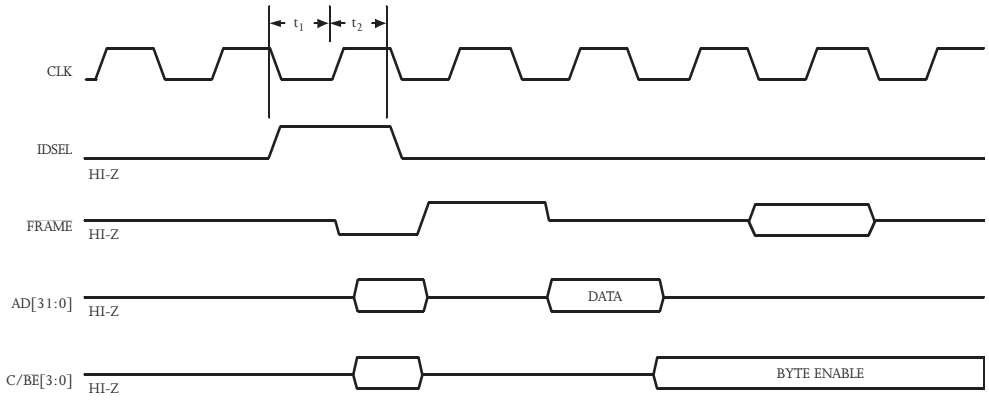
Waveform 8.2.2. PCI timing: TRDY, IRDY, read data

Symbol	Parameter	Min	Max	Unit
t_1	Read data setup to TRDY active	7	-	ns
t_2	Read data hold from TRDY inactive	0	-	ns
t_3	IRDY setup to CLK	7	-	ns
t_4	IRDY hold from CLK	0	-	ns
t_5	TRDY active delay from CLK	-	15	ns
t_6	TRDY inactive delay from CLK	-	15	ns
t_7	TRDY high before HI-Z	1 CLK	-	ns



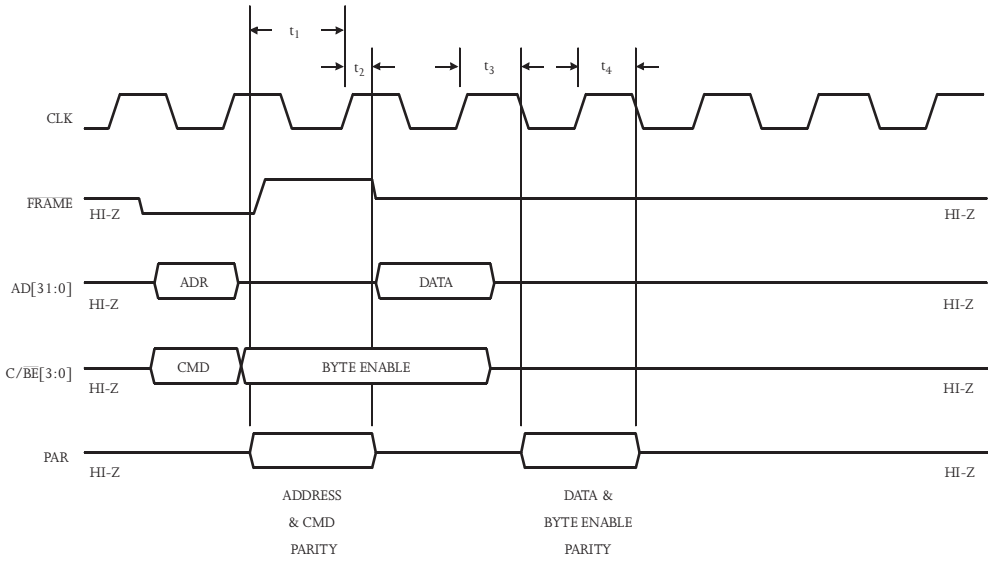
Waveform 8.2.3. PCI timing: IDSEL

Symbol	Parameter	Min	Max	Unit
t_1	IDSEL setup to CLK	-	15	ns
t_2	IDSEL hold from CLK	-	15	ns



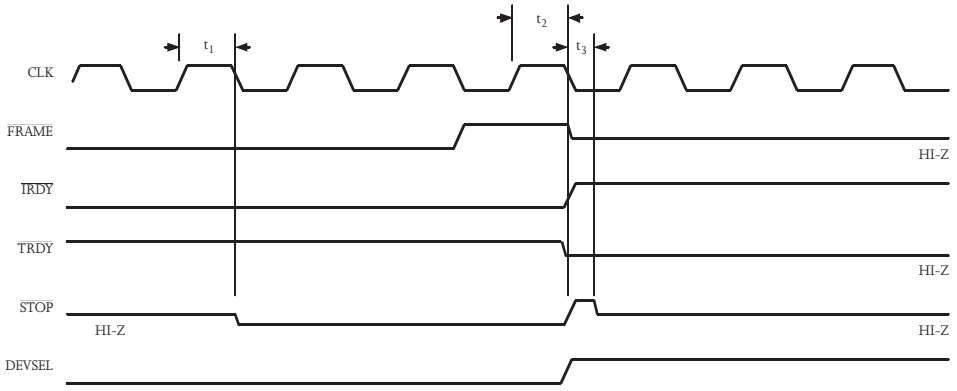
Waveform 8.2.4. PCI timing: $\overline{PAR}$

Symbol	Parameter	Min	Max	Unit
t_1	PAR setup to CLK as input	7	-	ns
t_2	PAR hold from CLK as input	0	-	ns
t_3	PAR delay from CLK as output	7	-	ns
t_4	PAR hold from CLK as output	0	-	ns



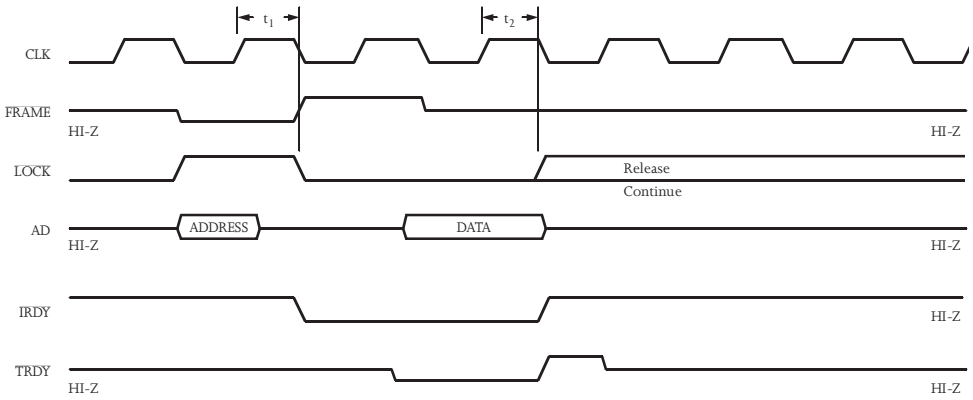
Waveform 8.2.5. PCI timing: $\overline{\text{STOP}}$

Symbol	*Parameter*	*Min*	*Max*	*Unit*
t_1	$\overline{\text{STOP}}$ active delay from CLK	2	10	ns
t_2	$\overline{\text{STOP}}$ inactive delay from CLK	2	10	ns
t_3	$\overline{\text{STOP}}$ HIGH before HI-Z	17	33	ns



Waveform 8.2.6. PCI timing: $\overline{\text{LOCK}}$

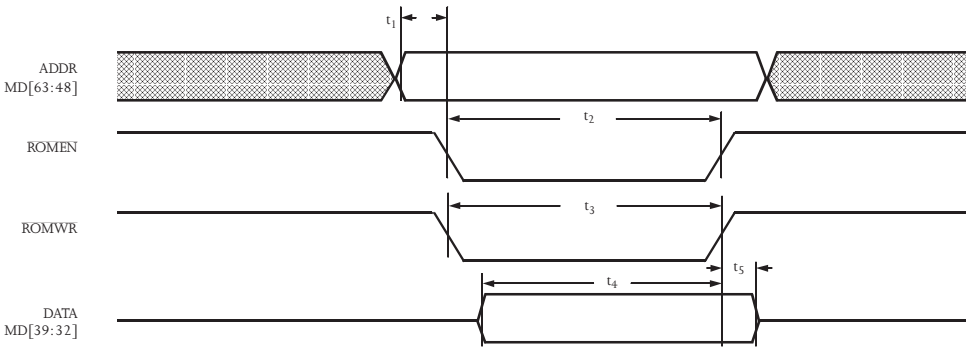
Symbol	*Parameter*	*Min*	*Max*	*Unit*
t_1	$\overline{\text{LOCK}}$ input setup time to CLK	7	-	ns
t_2	$\overline{\text{LOCK}}$ hold time from CLK	0	-	ns



Waveform 8.2.7. BIOS ROM/Flash: write timing

Symbol	Parameter	Min	Max	Unit
t_1	Address to $\overline{\text{ROMEN}}$ setup	1.5 CLK	-	ns
t_2	$\overline{\text{ROMEN}}$ pulse width	4†	-	ns
t_3	$\overline{\text{ROMWR}}$ pulse width	4†	-	ns
t_4	Data setup time to $\overline{\text{ROMWR}}$ high	t_3-0.5 CLK	-	ns
t_5	Data setup hold time from $\overline{\text{ROMWR}}$ high	-	2 CLK	ns

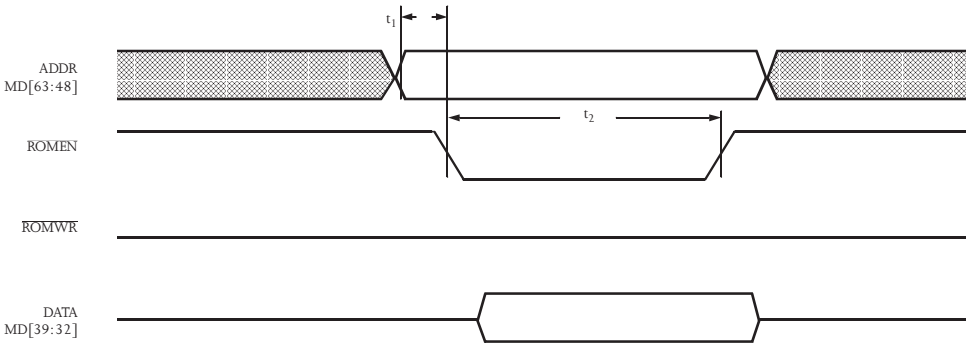
† Only when EPROM_WAIT = 0. If EPROM_WAIT ≥1 this value = [5+(EPROM_WAIT–1)×2]. Since default for EPROM_WAIT = 8, the default number of clocks is [5+(8–1)×2], or 19 MCLK.



Waveform 8.2.8. BIOS ROM/Flash: read timing

Symbol	Parameter	Min	Max	Unit
t_1	Address to $\overline{\text{ROMEN}}$ setup	1.5		ns
t_2	$\overline{\text{ROMEN}}$ pulse width	4†		ns

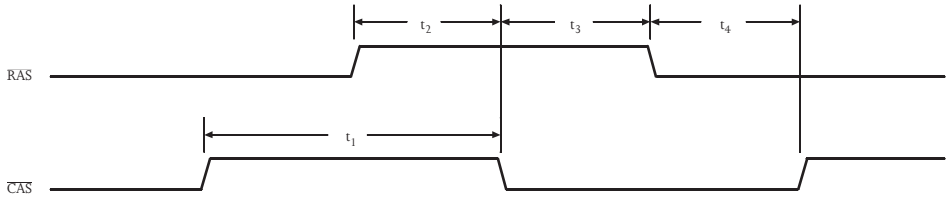
† Only when EPROM_WAIT = 0. If EPROM_WAIT ≥1 this value = [5+(EPROM_WAIT–1)×2]. Since default for EPROM_WAIT = 8, the default number of clocks is [5+(8–1)×2], or 19 MCLK.



8.3 Display memory timing

Waveform 8.3.1. Display memory timing: $\overline{CAS}$-before-$\overline{RAS}$ refresh

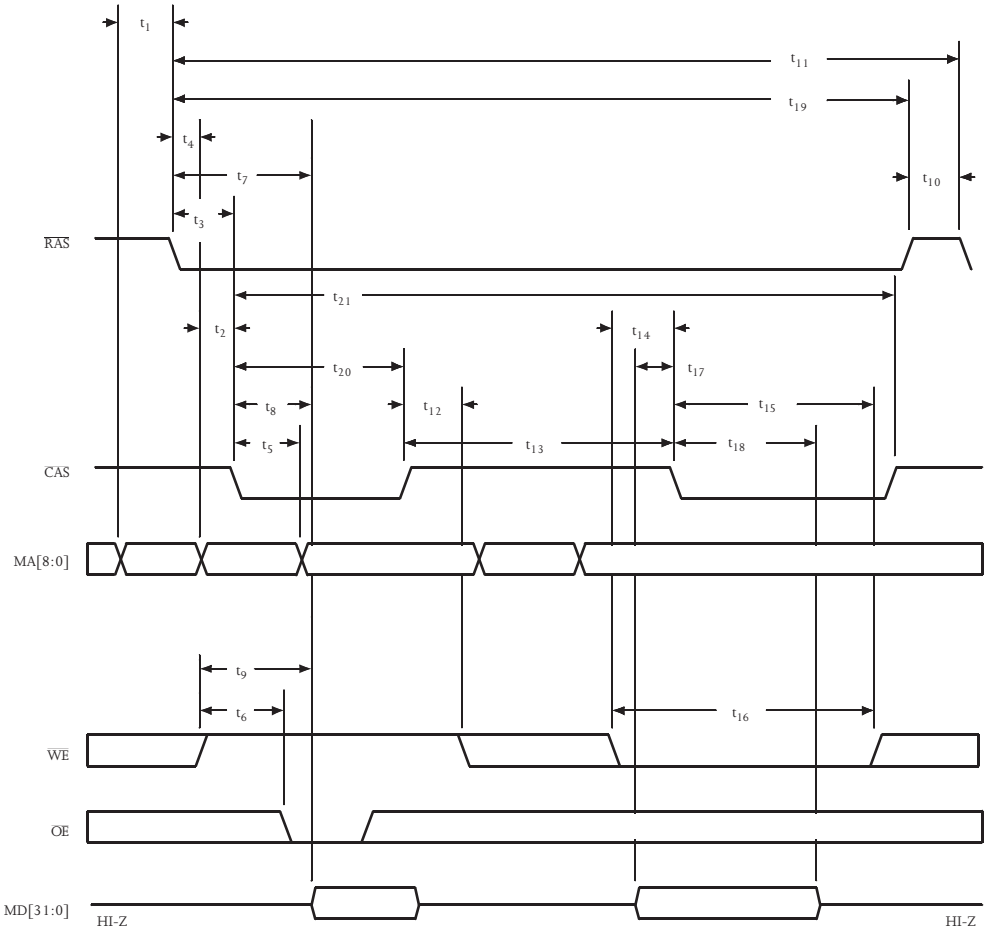
Symbol	Std. Symbol	Parameter	Min	Max	Unit
t_1	t_{CPN}	$\overline{CAS}$ precharge time	1 MCLK	-	ns
t_2	t_{RPC}	$\overline{RAS}$ high to $\overline{CAS}$ low precharge time	1 MCLK	-	ns
t_3	t_{CSR}	$\overline{CAS}$ before $\overline{RAS}$ setup time	1.5 MCLK	-	ns
t_4	t_{CHR}	$\overline{CAS}$ before $\overline{RAS}$ hold time	3.5 MCLK	-	ns



Waveform 8.3.2. Page mode DRAM: read/write

Symbol	Std. Symbol	Parameter	Min	Max	Unit
t_1	t_{ASR}	MA setup to $\overline{RAS}$ active (fast $\overline{RAS}$)	1.5 MCLK	-	ns
		MA setup to $\overline{RAS}$ active (ext. $\overline{RAS}$)	2 MCLK	-	ns
t_2	t_{ASC}	MA setup to $\overline{CAS}$ active	1 MCLK	-	ns
t_3	t_{RCD}	$\overline{RAS}$ to $\overline{CAS}$ delay (fast $\overline{RAS}$)	2.5 MCLK	-	ns
		$\overline{RAS}$ to $\overline{CAS}$ delay (ext. $\overline{RAS}$)	3 MCLK	-	ns
t_4	t_{RAH}	Row address hold from $\overline{RAS}$ active (fast $\overline{RAS}$)	1.5 MCLK	-	ns
		Row address hold from $\overline{RAS}$ active (ext. $\overline{RAS}$)	2 MCLK	-	ns
t_5	t_{CAH}	Column address hold from $\overline{CAS}$ active	1 MCLK	-	ns
t_6	-	$\overline{WE}$ inactive to $\overline{OE}$ active	1 MCLK	-	ns
t_7	t_{RAC}	Data valid from $\overline{RAS}$ (fast $\overline{RAS}$)	-	3.5 MCLK	ns
		Data valid from $\overline{RAS}$ (ext. $\overline{RAS}$)	-	4 MCLK	ns
t_8	t_{CAC}	Data valid from $\overline{CAS}$ active	-	1 MCLK	ns
t_9	t_{AA}	Data valid from column address valid	-	2 MCLK	ns
t_{10}	t_{RP}	$\overline{RAS}$ precharge (fast $\overline{RAS}$)	2.5 MCLK	-	ns
		$\overline{RAS}$ precharge (ext. $\overline{RAS}$)	3 MCLK	-	ns
t_{11}	t_{RC}	Random cycle (fast $\overline{RAS}$)	6 MCLK	-	ns
		Random cycle (ext. $\overline{RAS}$)	7 MCLK	-	ns
t_{12}	t_{RCH}	Read command hold from $\overline{CAS}$ high	1 MCLK	-	ns
t_{13}	t_{CP}	$\overline{CAS}$ precharge	1 MCLK	-	ns
t_{14}	t_{CWL}	$\overline{WE}$ active setup to $\overline{CAS}$ active	0 MCLK	-	ns
t_{15}	t_{WCH}	$\overline{WE}$ active hold from CAS active	1 MCLK	-	ns
t_{16}	t_{WP}	$\overline{WE}$ active pulse width	1 MCLK	-	ns
t_{17}	t_{DS}	Write data setup to $\overline{CAS}$ active	0.5 MCLK	-	ns
t_{18}	t_{DH}	Write data hold from $\overline{CAS}$ active	0.5 MCLK	-	ns
t_{19}	t_{RAS}	$\overline{RAS}$ pulse width low (fast $\overline{RAS}$)	3.5 MCLK	-	ns
		$\overline{RAS}$ pulse width low (ext. $\overline{RAS}$)	4 MCLK	-	ns
t_{20}	t_{CAS}	$\overline{CAS}$ pulse width low	1 MCLK	-	ns
t_{21}	t_{PC}	Page mode cycle time	2 MCLK	-	ns

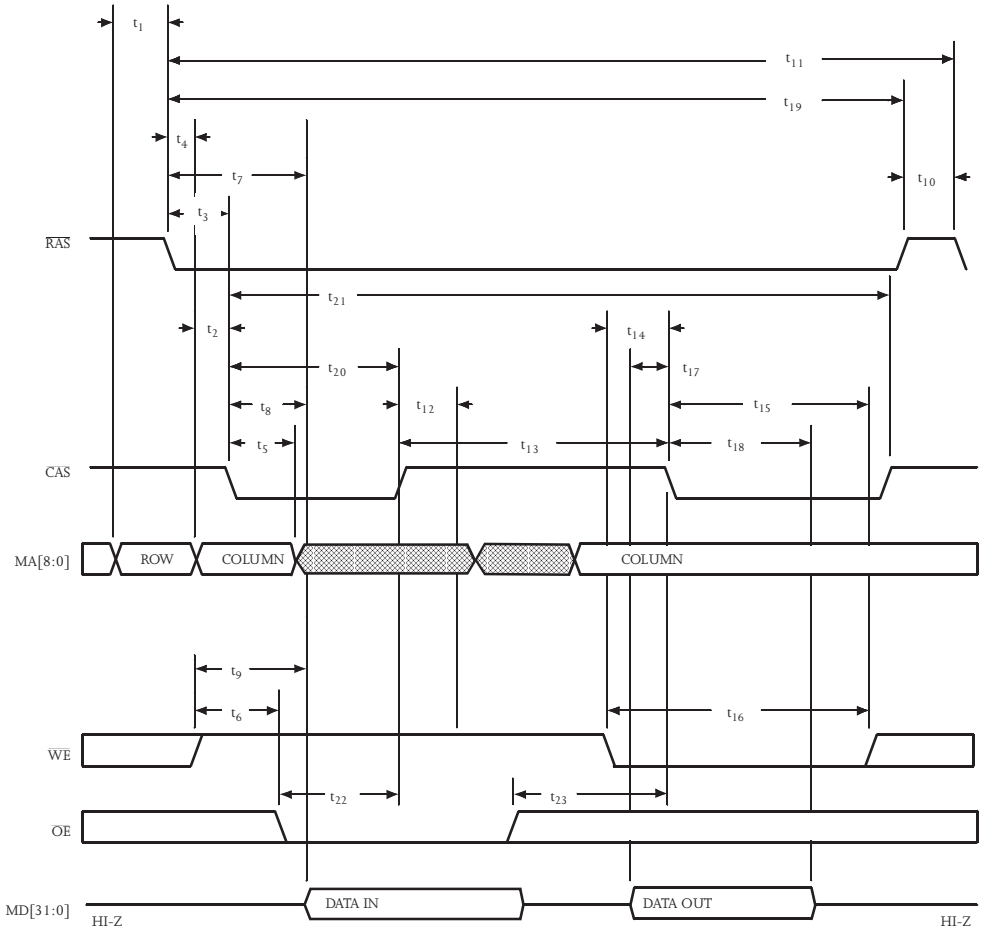
Waveform 8.3.2 Page mode DRAM: read/write



Waveform 8.3.3. Single cycle EDO DRAM: read/write

Symbol	Std. Symbol	Parameter	Min	Max	Unit
t_1	t_{ASR}	MA setup to $\overline{RAS}$ active (fast $\overline{RAS}$)	0.5 MCLK	-	ns
		MA setup to $\overline{RAS}$ active (ext. $\overline{RAS}$)	1 MCLK	-	ns
t_2	t_{ASC}	MA setup to $\overline{CAS}$ active	1 MCLK	-	ns
t_3	t_{RCD}	$\overline{RAS}$ to $\overline{CAS}$ delay (fast $\overline{RAS}$)	2 MCLK	-	ns
		$\overline{RAS}$ to $\overline{CAS}$ delay (ext. $\overline{RAS}$)	2.5 MCLK	-	ns
t_4	t_{RAH}	Row address hold from $\overline{RAS}$ active (fast $\overline{RAS}$)	1 MCLK	-	ns
		Row address hold from $\overline{RAS}$ active (ext. $\overline{RAS}$)	1.5 MCLK	-	ns
t_5	t_{CAH}	Column address hold from $\overline{CAS}$ active	0.5 MCLK	-	ns
t_6	-	$\overline{WE}$ inactive to $\overline{OE}$ active	2.5 MCLK	-	ns
t_7	t_{RAC}	Data valid from $\overline{RAS}$ (fast $\overline{RAS}$)	-	3 MCLK	ns
		Data valid from $\overline{RAS}$ (ext. $\overline{RAS}$)	-	3.5 MCLK	ns
t_8	t_{CAC}	Data valid from $\overline{CAS}$ active	-	1 MCLK	ns
t_9	t_{AA}	Data valid from column address valid (fast $\overline{RAS}$)	-	1.5 MCLK	ns
		Data valid from column address valid (ext. $\overline{RAS}$)	-	2.0 MCLK	ns
t_{10}	t_{RP}	$\overline{RAS}$ precharge (fast $\overline{RAS}$)	3 MCLK	-	ns
		$\overline{RAS}$ precharge (ext. $\overline{RAS}$)	2.5 MCLK	-	ns
t_{11}	t_{RC}	Random cycle (fast $\overline{RAS}$)	6 MCLK	-	ns
		Random cycle (ext. $\overline{RAS}$)	6 MCLK	-	ns
t_{12}	t_{RCH}	Read command hold from $\overline{CAS}$ high	1 MCLK	-	ns
t_{13}	t_{CP}	$\overline{CAS}$ precharge	0.5 MCLK	-	ns
t_{14}	t_{CWL}	$\overline{WE}$ active setup to $\overline{CAS}$ active	0 MCLK	-	ns
t_{15}	t_{WCH}	$\overline{WE}$ active hold from $\overline{CAS}$ active	1 MCLK	-	ns
t_{16}	t_{WP}	$\overline{WE}$ active pulse width	1 MCLK	-	ns
t_{17}	t_{DS}	Write data setup to $\overline{CAS}$ active	0.5 MCLK	-	ns
t_{18}	t_{DH}	Write data hold from $\overline{CAS}$ active	0.5 MCLK	-	ns
t_{19}	t_{RAS}	$\overline{RAS}$ pulse width low (fast $\overline{RAS}$) read	4 MCLK	-	ns
		$\overline{RAS}$ pulse width low (ext. $\overline{RAS}$) read	4.5 MCLK	-	ns
		$\overline{RAS}$ pulse width low (fast $\overline{RAS}$) write	3 MCLK	-	ns
		$\overline{RAS}$ pulse width low (ext. $\overline{RAS}$) write	3.5 MCLK	-	ns
t_{20}	t_{CAS}	$\overline{CAS}$ pulse width low	0.5 MCLK	-	ns
t_{21}	t_{PC}	Page mode cycle time	1 MCLK	-	ns
t_{22}	t_{OES}	$\overline{OE}$ low to $\overline{CAS}$ high setup	0.5 MCLK	-	ns
t_{23}	t_{OEP}	$\overline{OE}$ pulse width high	1 MCLK	-	ns

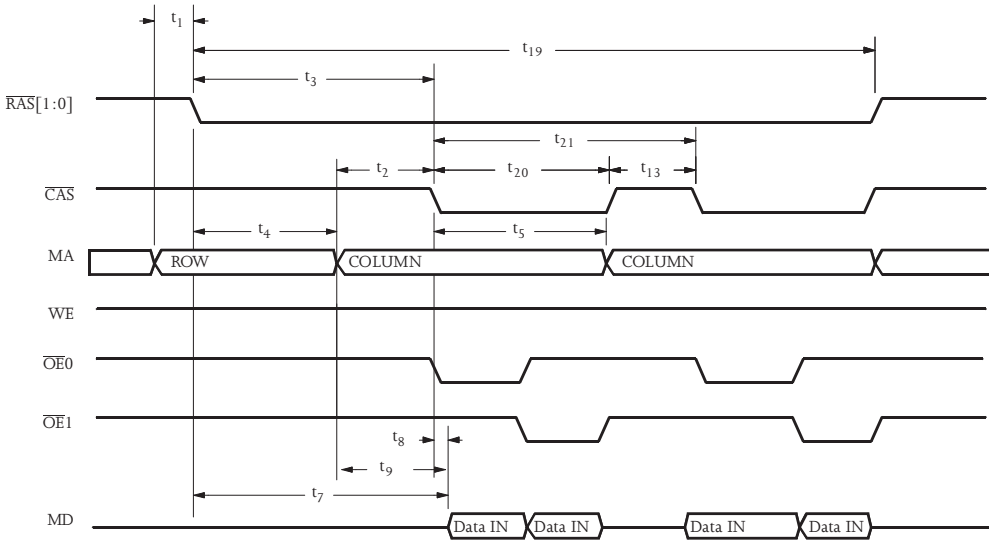
Waveform 8.3.3 Single cycle EDO DRAM: read/write



Waveform 8.3.4. Interleaved DRAM: read

Symbol	Std. Symbol	Parameter	Min	Max	Unit
t_1	t_{ASR}	MA setup to $\overline{RAS}$ active (fast $\overline{RAS}$)	1.5 MCLK	-	ns
		MA setup to $\overline{RAS}$ active (ext. $\overline{RAS}$)	2 MCLK	-	ns
t_2	t_{ASC}	MA setup to $\overline{CAS}$ active	1 MCLK	-	ns
t_3	t_{RCD}	$\overline{RAS}$ to $\overline{CAS}$ delay (fast $\overline{RAS}$)	2.5 MCLK	-	ns
		$\overline{RAS}$ to $\overline{CAS}$ delay (ext. $\overline{RAS}$)	3 MCLK	-	ns
t_4	t_{RAH}	Row address hold from $\overline{RAS}$ active (fast $\overline{RAS}$)	1.5 MCLK	-	ns
		Row address hold from $\overline{RAS}$ active (ext. $\overline{RAS}$)	2 MCLK	-	ns
t_5	t_{CAH}	Column address hold from $\overline{CAS}$ active	2 MCLK	-	ns
t_7	t_{RAC}	Access time from $\overline{RAS}$ (fast $\overline{RAS}$)	-	3.5 MCLK	ns
		Access time from $\overline{RAS}$ (ext. $\overline{RAS}$)	-	4 MCLK	ns
t_8	t_{CAC}	Access time from $\overline{CAS}$ active	-	1 MCLK	ns
t_9	t_{AA}	Access time from column address valid	-	2 MCLK	ns
t_{13}	t_{CP}	$\overline{CAS}$ precharge	1 MCLK	-	ns
t_{19}	t_{RAS}	$\overline{RAS}$ pulse width low (fast $\overline{RAS}$)	4.5 MCLK	-	ns
		$\overline{RAS}$ pulse width low (ext. $\overline{RAS}$)	5 MCLK	-	ns
t_{20}	t_{CAS}	$\overline{CAS}$ pulse width low	2 MCLK	-	ns
t_{21}	t_{PC}	Page mode cycle time	3 MCLK	-	ns

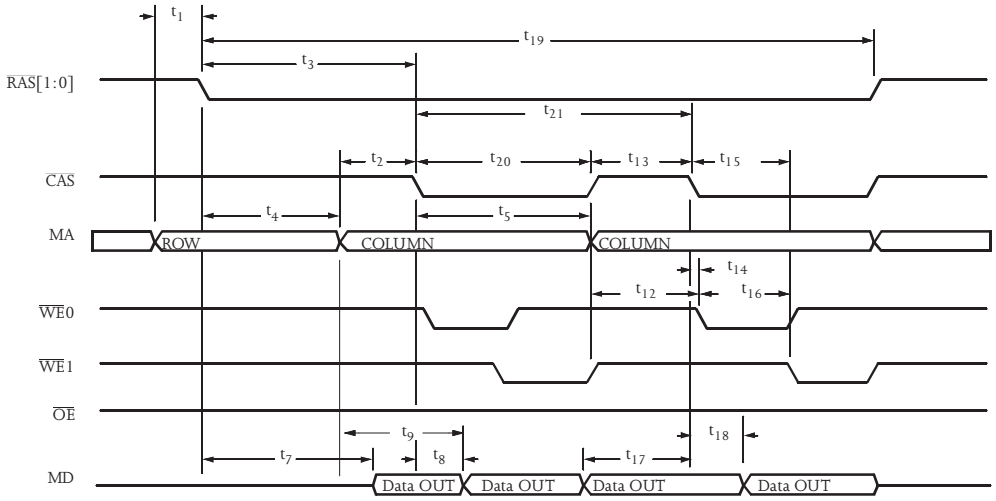
Waveform 8.3.4 Interleaved DRAM: read



Waveform 8.3.5. Interleaved DRAM: write

Symbol	Std. Symbol	Parameter	Min	Max	Unit
t_1	t_{ASR}	MA setup to $\overline{RAS}$ active (fast $\overline{RAS}$)	1.5 MCLK	-	ns
		MA setup to $\overline{RAS}$ active (ext. $\overline{RAS}$)	2 MCLK	-	ns
t_2	t_{ASC}	MA setup to $\overline{CAS}$ active	1 MCLK	-	ns
t_3	t_{RCD}	$\overline{RAS}$ to $\overline{CAS}$ delay (fast $\overline{RAS}$)	2.5 MCLK	-	ns
		$\overline{RAS}$ to $\overline{CAS}$ delay (ext. $\overline{RAS}$)	3 MCLK	-	ns
t_4	t_{RAH}	Row address hold from $\overline{RAS}$ active (fast $\overline{RAS}$)	1.5 MCLK	-	ns
		Row address hold from $\overline{RAS}$ active (ext. $\overline{RAS}$)	2 MCLK	-	ns
t_5	t_{CAH}	Column address hold from $\overline{CAS}$ active	2 MCLK	-	ns
t_7	t_{RAC}	Access time from $\overline{RAS}$ (fast $\overline{RAS}$)	-	3.5 MCLK	ns
		Access time from $\overline{RAS}$ (ext. $\overline{RAS}$)	-	4 MCLK	ns
t_8	t_{CAC}	Access time from $\overline{CAS}$ active	-	1 MCLK	ns
t_9	t_{AA}	Access time from column address valid	-	2 MCLK	ns
t_{12}	t_{RCH}	Read command hold from $\overline{CAS}$ high	1 MCLK	-	ns
t_{13}	t_{CP}	$\overline{CAS}$ precharge	1 MCLK	-	ns
t_{14}	t_{CWL}	$\overline{WE}$ active setup to $\overline{CAS}$ active	0 MCLK	-	ns
t_{15}	t_{WCH}	$\overline{WE}$ active hold from CAS active	1 MCLK	-	ns
t_{16}	t_{WP}	$\overline{WE}$ active pulse width	1 MCLK	-	ns
t_{17}	t_{DS}	Write data setup to $\overline{WE}$ active	0.5 MCLK	-	ns
t_{18}	t_{DH}	Write data hold from $\overline{WE}$ active	0.5 MCLK	-	ns
t_{19}	t_{RAS}	$\overline{RAS}$ pulse width low (fast $\overline{RAS}$)	4.5 MCLK	-	ns
		$\overline{RAS}$ pulse width low (ext. $\overline{RAS}$)	5 MCLK	-	ns
t_{20}	t_{CAS}	$\overline{CAS}$ pulse width low	2 MCLK	-	ns
t_{21}	t_{PC}	Page mode cycle time	3 MCLK	-	ns

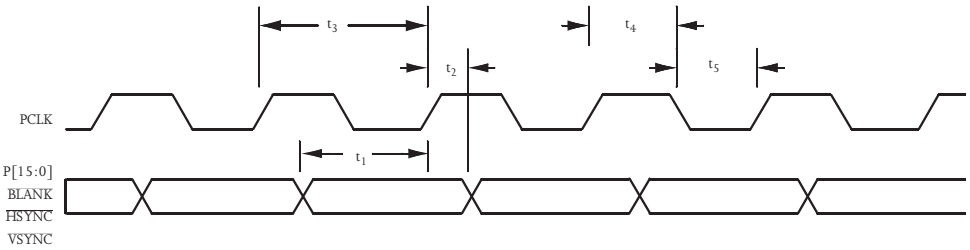
Waveform 8.3.5 Interleaved DRAM: write



8.4 Feature connector timing

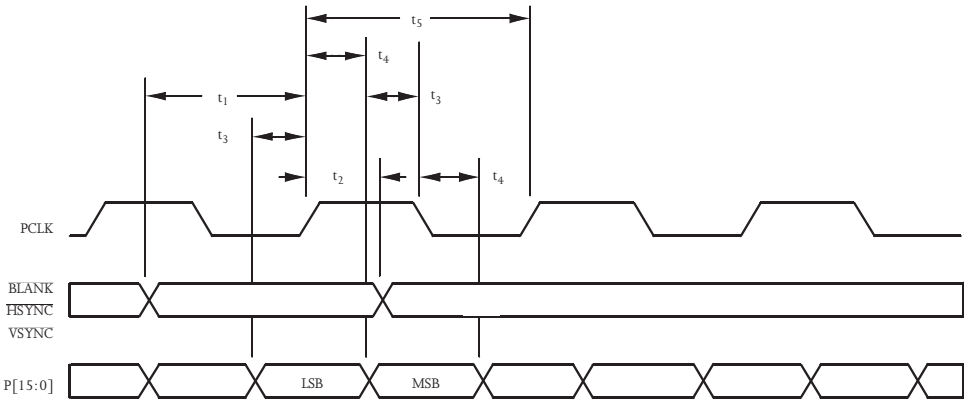
Waveform 8.4.1. Feature connector timing: single edge clocking mode

Symbol	Parameter	Min	Max	Unit
t_1	P[15:0], BLANK, HSYNC, VSYNC setup time	4	-	ns
t_2	P[15:0], BLANK, HSYNC, VSYNC hold time	4	-	ns
t_3	PCLK period	12	-	ns
t_4	PCLK high time	5	-	ns
t_5	PCLK low time	5	-	ns



Waveform 8.4.2. Feature connector timing: double edge clocking mode

Symbol	*Parameter*	*Min*	*Max*	*Unit*
t_1	$\overline{\text{BLANK}}$, $\overline{\text{HSYNC}}$, $\overline{\text{VSYNC}}$ setup time	4	-	ns
t_2	$\overline{\text{BLANK}}$, $\overline{\text{HSYNC}}$, $\overline{\text{VSYNC}}$ hold time	4	-	ns
t_3	P[15:0] setup time	10	-	ns
t_4	P[15:0] hold time	10	-	ns
t_5	PCLK period	24	-	ns



8.5 VMI+ and THP timing

Refer to Alliance publication "*VMI+ Implementation Notes*" for VMI+ timing waveforms and additional information. Refer to Alliance publication "*THP Implementation Notes*" for details on THP timing.

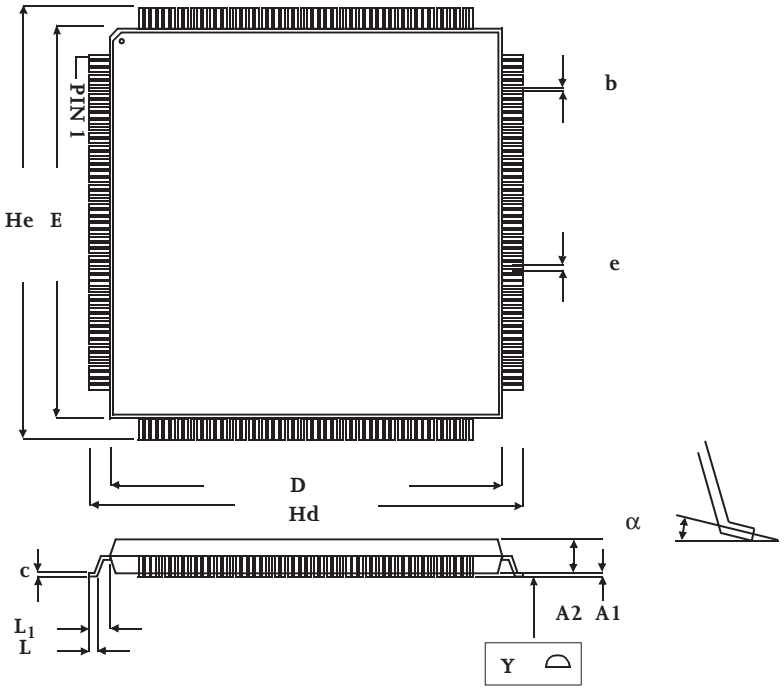
8.6 Test conditions

Pin name	*Capacitive load*	*Unit*
$\overline{\text{BLANK}}$, $\overline{\text{HSYNC}}$, $\overline{\text{VSYNC}}$, PCLK	30	pF
P[15:0]	40	pF
MA[8:0]	60	pF
$\overline{\text{CAS}}$[7:0]	20	pF
$\overline{\text{RAS}}$[1:0]	80	pF
$\overline{\text{WE}}$, $\overline{\text{OE}}$	40	pF
MD[63:00]	30	pF
DAT[31:00], IRQ, M/$\overline{\text{IO}}$, W/$\overline{\text{R}}$	75	pF
ROMEN	40	pF

9. Physical dimensions

Figure 9.1. 208-pin plastic quad flat pack (PQFP)

Diagram 960822 QFP208.C



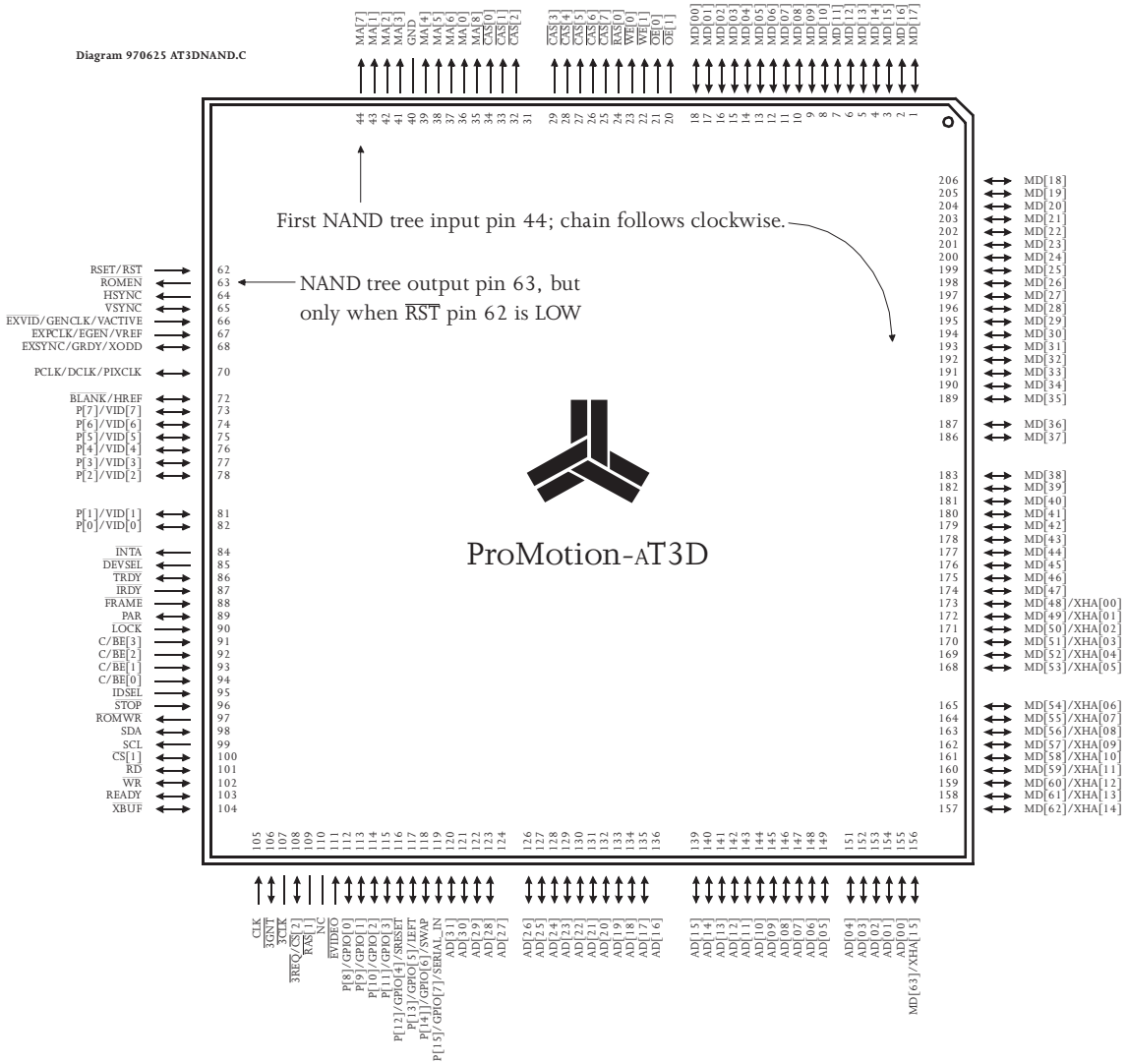
	2.6 mm		3.2 mm	
	Min	Max	Min	Max
A1	0.05	0.50	0.05	0.50
A2	3.17	3.47	3.15	3.47
b	0.10	0.30	0.10	0.30
c	0.10	0.20	0.10	0.23
D	27.87	28.10	27.87	28.10
E	27.87	28.10	27.87	28.10
e	0.50		0.50	
Hd	30.35	30.85	30.95	31.45
He	30.35	30.85	30.95	31.45
L	0.40	0.75	0.65	0.95
L1	1.30		1.60	
α	0°	7°	0°	7°
Y	-	0.15	-	0.15

10. AT3D NAND tree

The first input pin in the NAND tree chain is MA[7], pin 44, with subsequent NAND tree inputs following clockwise around the controller, omitting all VCC and GND pins, RST pin [62], and analog pins [45–61].

NAND tree output is pin [63], but only while pin [62] is LOW.

Figure 10a. AT3D NAND tree reference diagram, PCI bus



10.2 ProMotion-AT3D NAND tree diagram

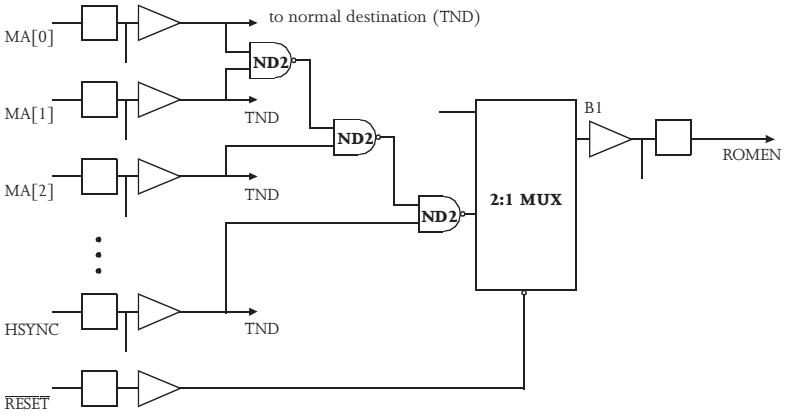


Table 10.2 NAND tree truth table

MA[0]	MA[1]	. . .	EXVID	VSYNC	HSYNC	ROMEN
0	0	. . .	0	0	1	0
0	0	. . .	0	0	0	1
0	0	. . .	0	0	1	0
0	0	. . .	0	1	1	1
0	0	. . .	0	0	1	0
0	0	. . .	0	1	1	1
0	0	. . .	1	1	1	0
0	0	. . .	1	1	1	1
0	0	. . .	1	1	1	0
0	1	. . .	1	1	1	1
0	0	. . .	1	1	1	0
0	1	. . .	1	1	1	1
1	1	. . .	1	1	1	0
0	1	. . .	1	1	1	1
1	1	. . .	1	1	1	0

11. Graphics programming notes

11.1 ProMotion registers

The ProMotion-aT3D includes I/O mapped and memory mapped registers. Many I/O registers are mapped as index/data pairs: a write to the index sets the pointer which selects the register accessed at the data port. All I/O mapped registers are accessible unless locked. Registers are always unlocked except for 3D5.11[7] "Vertical retrace end," described on page 153, and extended 0C8–0C9, "VGA override," described on page 226.

ProMotion extended registers are memory mapped for fast access. Each register has a unique address; there are no index/data pairs. Memory mapped registers are accessible when unlocked by 3C5.10, "Unlock extended registers," described on page 167. All memory space addresses are expressed as offsets from the ProMotion register base address, which is selected in I/O space using 3C5.1A, "Flat model base address," described on page 168, and 3C5.1B, "Remap control," described on page 168.

Most register writes pass through the ProMotion command FIFO. Therefore, operators which change chip state do not execute until pending graphics commands are complete. Registers which do not pass through the command FIFO are noted in their descriptions. Most register reads execute only after the command FIFO is fully drained. Exceptions are noted in the register descriptions.

11.2 Reserved bits

To prevent unexpected operation, all reserved register bits should be written with 0s and masked off if read back. Future compatability is jeopardized if this procedure is not followed.

11.3 Clipping

Clipping may apply to any drawing engine operation specified by M040–043h, "Drawing engine control," on page 188. There are five ProMotion clipping registers.

- M030h, "Clipping control," described on page 186.
- M038h, "Clipping boundary left," described on page 187.
- M03Ah, "Clipping boundary top," described on page 187.
- M03Ch, "Clipping boundary right," described on page 187.
- M03Eh, "Clipping boundary bottom," described on page 188.

To enable clipping, load the four clipping boundary registers with pixel coordinates and set the clipping enable bit M030[0]. Clipping has pixel granularity. The single-pixel wide outline defined by the four coordinates is considered inside the rectangle.

11.4 Drawing engine control

11.4.1. Drawing engine command

Eight operations are performed by the drawing engine: NOP; screen-to-screen BLT; rectangle; strip draw; host BLT write; host BLT read; vector, draw endpoint; vector, don't draw endpoint. The first four operations require little explanation and are summarized below. Host BLT read/writes and vector line draws are each described in detail in following sections.

- NOP loads command register bit fields only and does not start any operation.
- Screen-to-screen BLT copies pixels from one region of display memory to another. BLT directions may be set horizontally and vertically. Source and destination transparency, byte masking and raster operations are available. The source may be monochrome, in which case source data must be aligned to a 4-byte boundary. All source and destination regions using linear rather than X/Y addressing must be contiguous.
- Rectangle fills the destination region with the foreground color or pattern. This operation has no source region. Destination transparency, byte masking, and raster operations are available.
- Strip draw is similar to Rectangle, but with a height of one pixel.

11.4.1.1 Host BLT read/write

Host BLT read and write functions accelerate non-aligned block transfers between the host and display memory. The ProMotion graphics engine efficiently shifts any source alignment to the specified destination alignment, offloading this operation from the host processor.

For aligned transfers, simple linear memory writes and reads are usually faster than host BLT operations.

Host BLT write is similar to screen-to-screen BLT except that the source region is not in display memory. After this operation is started, the source region is passed to destination from the host BLT write queue as it is written by the host. The host BLT write port is a contiguous memory mapped region in the host address space, determined by 3C5.1B, "Remap control," on page 168. Any data written to this region is written into a single queue from which the source pixels are read, regardless of what address within the region is used for the write.

Access to the host BLT write port must be in 32-bit doublewords only. The source region may have any alignment. The low-order bits of the source location register point to the pixel within the first dword corresponding to the first pixel of the destination, which also may have any alignment. The host must write exactly the correct number of dwords to the host BLT write port. It must write all source dwords of which at least one pixel fall within the destination region. Host BLT write access must be left-to-right, top-to-bottom.

Monochrome-to-color expansion on host BLT write is supported during host BLT writes only in ProMotion-aT3D revision D and later. All other logical operations are available during host BLT write.

Device drivers MUST poll 1FC and wait until bit [8] is high before writing to the host BLT port. Writing to the host BLT port while 1FC[8] is low may cause unexpected results. Refer to "Extended/DAC status," on page 178, for a description of 1FC[8].

Host BLT read is a screen-to-memory host BLT, similar to screen-to-screen BLT except that the destination region is not in display memory. After this operation is started, the source region is read from display memory and placed in a queue to be read by the host. The host BLT read port is a contiguous memory mapped region in the host address space, determined by 3C5.1B, "Remap control," on page 168. Any address read within this region returns the next pixel from the queue supplied by the host BLT read engine.

The host BLT read port and write port use the same addresses, but they have separate functions and should not be thought of as a storage area.

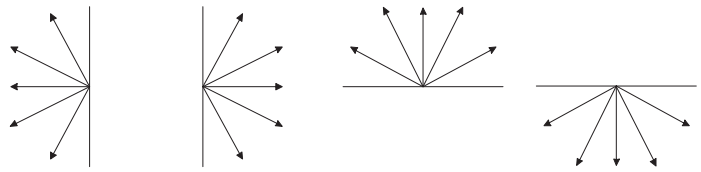
Access to the host BLT read port must be in doublewords only. The destination region may have any alignment. The low-order bits of the destination location register point to the pixel within the first dword corresponding to the first pixel of the source, which also may have any alignment. The host must read exactly the correct number of dwords from the host BLT port; it must read all destination dwords of which at least one pixel fall within the source region. Host BLT read access must be left-to-right, top-to-bottom.

Logical operations are not available during host BLT read, although host BLT read may be interrupted.

11.4.1.2 Vector line draw

The destination location registers define the start point for line draw operations. Directions X and Y are displayed in the following diagram.

Vectors



Dir X = 1 Dir X = 0 Dir Y = 1 Dir Y = 0

Vector, draw endpoint command draws a vector between any two points in display memory, whether or not these points are on the visible screen. The vector is drawn using the foreground color. The DDA (Digital Differential Analyzer) registers are used for line draw.

For a vector from (x1,y1) to (x2,y2), we define the following:

- dx = abs(x1-x2).
- dy = abs(y1-y2).
- dmin = min(dx,dy).
- dmax = max(dx,dy).

For vector line draw, load these registers.

Table 11.4.1.2a Vector draw register setup

Register		Use
M050–051, M052–053	"Source location X/low," described on page 196 and "Source location Y/high," described on page 196	Not used.
M054–055, M056–057	"Destination location X/low," described on page 197 and "Destination location Y/high," described on page 197	Start point of the vector (x1, y1).
M058–059	"Source size X/vector pixel count," described on page 198	Dimension X: dmax + 1.
M05A–05B	"Source size Y," described on page 198	Dimension Y: not used.
M060–063	"Foreground color," described on page 200	Color of vector
M064–067	"Background color/source transparency," described on page 201	Background color not used. Source transparency must be disabled.
M040[21:20]	"Drawing engine control," described on page 188	Destination transparency/mask: available as in BITBLT.
M070–071	"DDA axial step constant," described on page 203	2*dmin.
M072–073	"DDA diagonal step constant," described on page 203	(2*dmin) – (2*dmax).
M074–075	"DDA error term," described on page 204	(2*dmin) – dmax.
M046	"Raster operation," described on page 192	Available as in BITBLT.
M047	"Byte mask," described on page 193	Available.
M048	"Pattern," described on page 193	Not used.
M040	"Drawing engine control," described on page 188	☞ Must be loaded last, unless you use "Quick start, M040[30:29]," described on page 93

Setup Vector line draw as shown below with "Drawing engine control," described on page 188.

Table 11.4.1.2b Drawing engine control setup for vector draw

Bits	Description	Use
M040[3:0]	Drawing engine command	Set to 1100 (draw endpoint) or 1101 (don't draw endpoint)
M040[6]	Direction X	Set to 1 if x2 < x1, set to 0 otherwise
M040[7]	Direction Y	Set to 1 if y2 < y1, set to 0 otherwise
M040[8]	Major axis	Set to 1 if dy > dx, set to 0 otherwise
M040[13:9]	Source format	Not used
M040[18]	Destination address XY/linear	XY addressing only
M040[19]	Destination address rectangular	Rectangular only
M040[20]	Destination transparency	Available
M040[26:24]	Address model	Same as for BITBLT
M040[28:27]	Destination update	Set to 00 (do not update destination) or 11 (set destination to last pixel)
M040[30:29]	Quick start	Available for line draw. It is useful for chained vectors

- Vector, don't draw endpoint is similar to Vector, draw endpoint, above, except that endpoint is interpolated but not drawn. Start point is always drawn.
- Vectors are subject to clipping if clipping is enabled. If destination update is enabled and set to '11', destination location registers are set to the last pixel interpolated even though it is not drawn.
- The combination of destination update to last pixel and vector, don't draw endpoint is very useful when using the XOR raster operation to prevent vertices in a chain of vectors from being drawn twice and thus cleared.

11.4.1.3 BITBLT directions

The Source or Destination registers (pointing to the starting point of the rectangle) define BITBLT source and destination rectangles. Normally, the direction X and direction Y bits are set to 0 to indicate that these registers point to the top-left corner, and BLT proceeds left-to-right and top-to-bottom. Alternately, other starting corners may be specified by pointing to those locations, and by setting the Direction register.

The conventional left-to-right, top-to-bottom procedure is appropriate for most BITBLT operations.

When the source and destination regions overlap, it is necessary to examine the BITBLT operation and start from the proper corner in order to avoid overwriting source pixels before they are copied.

The following is a simple general solution which does not require complex overlap calculations.

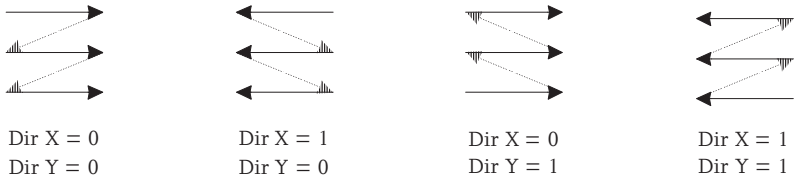
- If the top-left corner of the source region is at a higher address in display memory than the destination region, or if there is no source region, then the BITBLT Source and

Destination registers should point to the top-left corner of the respective regions and the direction X and Y bits should be set to 0.

- If the top-left corner of the source region is at a lower address in display memory than the destination region, then the BITBLT Source and Destination registers should point to the bottom-right corner of the respective regions and the Direction X and Y bits should be set to 1.

When the source region is rectangular and the destination is linear, or vice versa, the two regions cannot be permitted to overlap.

BITBLT directions



Dir X = 0 Dir Y = 0	Dir X = 1 Dir Y = 0	Dir X = 0 Dir Y = 1	Dir X = 1 Dir Y = 1

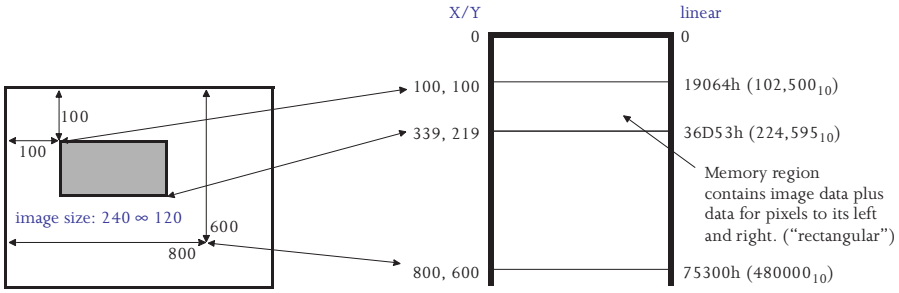
11.4.2. Source formats, M040[13:10]

ProMotion uses two addressing formats, XY and linear, selected by M040[26:24], "Drawing engine control," on page 188.

- In **XY addressing** mode address is interpreted as a row/column pair. ProMotion uses row width of 640, 800, 1024, 1152, 1280, or 1600 pixels-per-line to compute an X/Y address, also selected by M040[26:24].
- In **linear addressing** mode the address is interpreted as a pixel address starting from the top left corner of the screen, with a fixed 4096 pixels-per-line.

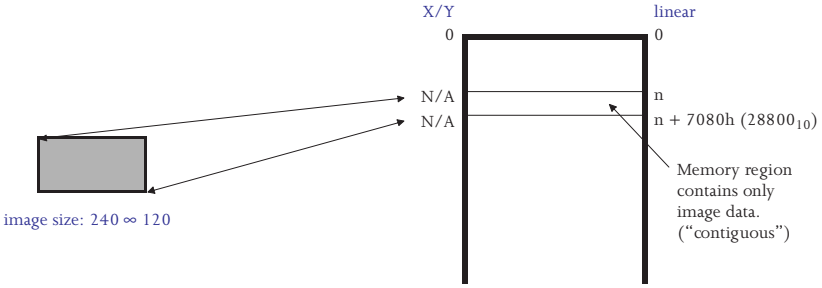
The following diagram illustrates an example of XY vs. linear addressing for an on-screen image. Generally, you would use XY addressing for on-screen rectangular regions and linear addressing for off-screen contiguous regions.

Figure 11.4.2a: XY vs linear addressing



Rectangular/contiguous - *Rectangular* means adjacent rows are offset by some amount. *Contiguous* means adjacent rows are stored one after the other. Contiguous is generally used for off-screen source regions. When a region is specified as linear it must also be specified as contiguous, as shown in the following diagram.

Figure 11.4.2b: Rectangular vs. contiguous regions



Color/monochrome - Monochrome regions are expanded to depth of display memory by replacing source 0s with background color and 1s with foreground color. To expand a monochrome region to foreground/transparent, the host should set both the source monochrome and source transparent bits, and set the background color to a different color than the foreground color register. The beginning of a monochrome source region must be 64-bit aligned. Subsequent rows of the source are byte aligned, with the remainder of any partially-used source byte from the previous row discarded.

11.4.3. Destination formats, M040[21:18]

Destination format is similar to source format. There is no concept of pattern for the destination, but a destination transparency polarity bit is added.

11.4.4. Quick start, M040[30:29]

The ProMotion-aT3D provides a **Quick start** feature to reduce the number of register writes required to invoke frequently-used operations. With Quick start enabled, writing a designated register automatically starts a new graphics engine operation with the same command as the previous operation. This allows efficient execution of a series of similar operations with similar parameters.

You may set quick start to trigger on writes to Dimension X, Source, or Destination registers. Quick start and destination updates combine effectively for chained operations. For example, with a polyline, using destination update = last pixel, the start point is automatically set to the end point of the previous segment. The driver then loads any other draw parameters which change for the next segment, writing endpoint X value last. With Quick start set appropriately this causes the draw command to begin. Other examples are shown in the following table.

Operation	Command	Quick start	Destination update
trapezoidal fills	strip draw	start on Dimension X	below bottom left corner
RLE decodes	strip draw	start on source	right of top right corner
chained vector operations	vector don't draw endpoint	start on destination	last pixel

Drawing engine operations may also be started in quick start mode. The Drawing Engine Command register may be written with Drawing Engine Start reset in order to set up other register bits without starting an operation.

11.4.5. Source location registers

These notes apply to M050–051, "Source location X/low," described on page 196 and M052–053, "Source location Y/high," described on page 196. Specify source values in either (X,Y) format, or linear format. The direction bits in the BITBLT control register determine which corner of the source rectangle is specified by these registers.

Linear address is in pixels rather than bytes. A particular byte address in the frame buffer has a different linear address depending on current pixel depth mode.

When a BITBLT is specified as being right-to-left, the source location and destination location registers specify the right edge, using the top-right or bottom-right corner of the source and destination regions, respectively. When a BITBLT is specified as being bottom-to-top, the source location and destination location registers specify the bottom edge, using the bottom-left or bottom-right corner of the source and destination regions respectively. The right-to-left and bottom-to-top bits may be set independently. Right-to-left and bottom-to-top operations should not be used for drawing engine operations other than screen-to-screen BLT.

11.4.6. Destination location registers

These notes apply to M054–055, "Destination location X/low," described on page 197 and M056–057, "Destination location Y/high," described on page 197. Unlike the source location registers, the destination location registers can be automatically updated to reflect the endpoint of a drawing engine operation.

Values are specified in a manner analogous to the source location registers.

Drawing engine operations such as filled rectangle and patterns have only a destination but no source.

11.5 BIOS extended initialization routine

Execute the following steps to initialize ProMotion BIOS.

1. unlock registers.
2. disable VGA memory.
3. set the extended register IO port base address from PCI address space
4. set bit to disable vertical interrupts.
5. turn screen off.

6. clear betterhalf bit.
7. reset eight 8-bit scratchpad registers
8. reset overdrive registers
9. initialize default VGA video clocks
10. enable FIFO command registers
11. reset interlace Control Register
12. initialize the pixel FIFO request points
13. disable PCI VGA palette snooping (not applicable to all AT24 revs.)
14. save DAC type in scratchpad register 1
15. initialize memory clock, 2MB + 4 MB
16. get memory size
17. save memory size in scratchpad register 1
18. set memory clock according to memory: EDO, fast page, fast page interleaved
19. enable VGA memory
20. turn screen on

11.6 Setting extended modes

Execute the following steps to set ProMotion extended graphic modes.

1. set standard VGA registers.
2. set extended CRTC registers (if required).
3. load RAMDAC.
4. reset chip to VGA compatible.
5. map ProMotion registers to desired location (if required)
6. reset flat model.
7. setup VCLK.
8. set up interlaced (if required).
9. setup FIFO.
10. load font.
11. set cursor.

Use the following example SetMode function to set various video display characteristics.

When SetMode is called, the new mode value that is to be set is passed in AL register. A test for MDA/CGA is performed first to determine if the correct hardware (a VGA or MDA/CGA card) is present.

After the correct software support for hardware is loaded, SetMode goes on and checks if the mode is available. If the mode is not available a default mode(mode 0 or 7) is set and exits the SetMode function. If mode is available the new mode is then checked to see if it is supported by the system; checks for VGA, mono, EGA emulation, single video, etc are made.

The function goes on to initialize the following common parameters: pointer to video parameter table, number of columns and rows, page size, cursor start and end scans, etc. The color palette is subsequently initialized unless default palette load is disabled; an application can use it's own palette.

```
SetMode:
   IF VGA is the only adapter
      Jump to ModeChk. This is most likely to occur.
   ELSE check VGA
      IF VGA running mono and mono is currently running.
       Jump to ModeChk.
```



```
      ELSE IF not vga and mono not current
       Prevent snow if present.
       Indicate VGA inactive.
       Set font to 8x8, if mode 7 not currently present.
       Load screen byte per character.
       Load graphics fonts pointer.
       Revector INT42(EGA default video driver)
      ELSE
       Jump to ModeChk.
ModeChk:
   Check if mode is available
   IF NOT available
      Default mode(previous mode) is set and
      exits SetMode functions.
   ELSE it is available
      New mode is check to see if it is supported by system.
      The following tests are made; check for VGA, mono, EGA
      emulation, single video adapter, etc.
      SetCommonData.

SetCommonData:
   Initialize common parameters per mode. The following common
   parameters are loaded: pointer to video parameter table,
   number of columns and rows, page size, cursor start and
   end scans, etc.
   Initialize the hardware registers in the following order:
      Init Sequence reg
      Init Misc output reg
      Disable video at sequencer
      Init CRT controller reg
      Init attribute controller reg
      Init graphic controller reg
   Color palette is then initialized unless default palette
   load is disabled. An application can use it's own palette.
   Call ExtSetMode.

ExtSetMode:
;-----------------------------------------------------------------------------
; Procedure:    ExtSetMode
;
; Desc: Set Mode calls this routine to set up the extended registers.
;
; In:   None.
;
; Out:  None.
;     All registers preserved.
;-----------------------------------------------------------------------------
      public  ExtSetMode
ExtSetMode      proc    near
      push    ax                ; Save scratch registers.
      push    bx
      push    cx
      push    dx
      push    si
      push    di

      call    UnlockRegs        ; Unlock extended registers.
      call    ResetVGA          ; Write to CRT0 to reset to VGA.
      call    ResetFlatMode ; Reset the flat mode access bit.

                                ;
                                ; Initially reset extended
                                ; CRTC registers 19h to 1Dh.
                                ;
      mov     dx, 03d4h
      mov     al, 019h          ; Start at index 019h
      mov     ah, 000h
```



```
esmRstCRTC:                         ; Loop to reset registers 19h to 1Dh.
      out     dx, ax                ; Initialize register to zero.
      inc     al                    ; Increment register.
      cmp     al, 01Eh              ; IF last CRTC register exit!, else
      jb      esmRstCRTC            ; Reset next extended CRTC register.

      mov     bx, 0d2h              ; Setup Interlace control register.
      mov     al, 0                 ; Initially reset interlace register.
      call    WriteIOMappedReg

                                    ;
                                    ; Set the Pixel FIFO Request Points
                                    ;
      mov     al, 01Bh              ; Load 1Bh into registers D4,D5,and D6.
      mov     bx, 0D4h              ; High priority request point-pg brk
      call    WriteIOMappedReg
      mov     bx, 0D5h              ; High priority request point-no pg brk
      call    WriteIOMappedReg
      mov     bx, 0D6h              ; Low priority request point
      call    WriteIOMappedReg
                                    ;
                                    ; Check if requested mode is an extended mode.
                                    ;
      mov     al, crtmode           ; Check if it's an extended mode.
      call    SearchPmodeTbl        ; in: al=mode; out: al,ah,di.
      or      ah, ah                ; Is it a Good mode?
      jnz     CannotHandle          ; IF not, EXIT!
ESMExtMode:                         ; ELSE  do the following.
      call    DisableVGAMem         ; Disable VGA memory.
      call    GXXInterlace          ; Set up the CRTC horizontal interlace
                                    ; start register.

                                    ;
                                    ; Set up a pointer to point to the mapped regs.
                                    ;
      mov     al, cs:[di].clock_int; AL contains the desired freq code.
      call    ClockSetUp            ; Program pixel clock rate
                                    ;
                                    ; This part of the code checks the pclk value if
                                    ; it is greater than 157 then it enables double
                                    ; index otherwise it enables single index.
                                    ;
      push    bx
      mov     al, cs:[di].clkval
      cmp     al, 0157              ; Is the pclk value greater than 157?
      jl      ESMlclk               ; NO!
      mov     bx, 080h              ; Set up register.
      call    ReadIOMappedReg
      or      al, Bit5
      call    WriteIOMappedReg      ; Enable double index.
      pop     bx
      jmp     ESMclkend
ESMlclk:                            ; PCLK is less than 157.
      mov     bx, 080h              ; Set up register
      call    ReadIOMappedReg
      and     al, (NOT Bit5)
      call    WriteIOMappedReg      ; Enable single index.
      pop     bx

ESMclkend:
                                    ;
                                    ; Check if requested mode is an interlaced mode.
                                    ;
      mov     al, byte ptr cs:[di].inter
      cmp     al, GXX_INTER         ; Is it in Interlace mode?
      jne     @f
      mov     bx, 0d2h              ; Yes, Interlace control register.
```



```
      mov     al, 1               ; Enable interlace.
      call     WriteIOMappedReg
@@:
      mov     al, crtmode
      call    SetUpOverFlow       ; Noninterlaced, Set extended CRTC regs.

                                  ;
                                  ; Check if requested mode is a 24-bit mode.
                                  ;
ifdef ADJ_24_FIFO
      mov     al, byte ptr cs:[di].pixels
      cmp     al, GXX_PPP_024     ; Is it a 24-bit mode?
      jne     @f                  ; IF not skip the next line, ELSE
      call    Load24FIFO          ; Load FIFO parameters for 24-bit modes.
@@:
endif                             ; ADJ_24_FIFO
                                  ;
                                  ; Check memory on video board.
                                  ;
      call    GetVidMemSize       ; Check on board memory size.
      mov     cl, al              ; Save memory size.
      cmp     cl, 018h            ; Is 1.5 MB present?
      jne     Not1_5MB            ; NO.
      call    Handle1_5MB         ; YES, BetterHalf 1.5 MB.
      jmp     Not3MB
Not1_5MB:
      cmp     cl, 030h            ; Is 3 MB present?
      jne     Not3MB              ; NO.
      call    Handle3MB           ; YES, BetterHalf 3 MB.
Not3MB:
                                  ;
                                  ; The following does a check
                                  ; for Banked or Planar modes.
                                  ;
      mov     al, byte ptr cs:[di].linear
      cmp     al, GXX_BANKED      ; Is it Banked?
      je      CannotHandle        ; Yes, but can not handle it. Exit!
      cmp     al, GXX_PLANAR      ; Not Banked, but is it Planar?
      jne     @f                  ; No.
      xor     ax, ax              ; Yes, go on and set PPPmode.
      mov     al, byte ptr cs:[di].pixels
      call    SetPPPMode          ; writes the Serial Control register.
      jmp      CannotHandle       ; Done, now EXIT!
@@:
      xor     ax, ax              ; Set up PPPmode, SWmode, and Flatmode.
      mov     al, byte ptr cs:[di].pixels
      call    SetPPPMode          ; writes the Serial Control register.
      mov     ax, word ptr cs:[di].scanw
      call    SetSWMode           ; Set screen width for mode.
      mov     al, byte ptr cs:[di].linear
      call    SetFlatMode         ; Set the chip for flat mode access.

                                  ;
                                  ; EXIT the procedure call.
                                  ;
CannotHandle:
      call    EnableVGAMem        ; Set up memory so that it wraps at
                                  ; 64k boundary.

      pop     di                  ; Restore registers.
      pop     si
      pop     dx
      pop     cx
      pop     bx
      pop     ax
      ret                         ; EXIT!
ExtSetMode      endp
```



```
After the call to ExtSetMode is made, a font corresponding to the
mode is then loaded.
Set mode is now ready to terminate. Go on to SetModeEnd.

SetModeEnd:
```

Before exiting the setmode function, the following occurs:

- The screen is Cleared if AL bit-7=0(zero)
- Video is enabled at attribute controller
- The cursor type is set. Registers are restored
- from stack, and finally exits.

11.7 AT24/3D changes affecting software development

This section is intended for ProMotion-AT24/3D developers familiar with previous ProMotion family controllers.

11.7.1. Command FIFO depth deltas

The command FIFO, which queues CPU write accesses from the PCI bus, has been expanded in size. From a software point of view the change is as follows: In the 6422, the bits [3:0] of the Status register (offset 0x1FC), can range from 0 to 4 to indicate the number of command FIFO entries available for filling from the CPU side. In ProMotion-AT24/3D, this value can range from 0 to 8. Bit 3 in ProMotion-6422 is hardwired to 0.

For a burst transfer, in which the address is incremented internally to the AT24/3D, the effective depth of the AT24/3D FIFO can be up to 16. However the effective depth is 8 for the individual register accesses which take place in programming the AT24/3D.

11.7.2. Pixel depth control deltas

In the 6422, there is no register which specifies the pixel depth for a BLT operation independently from the pixel depth of the video refresh logic. In ProMotion-AT24/3D, the pixel depth can be independently programmed by means of M040[16:14] "Drawing engine control," on page 188. First of all, backward compatibility is maintained in the following way. If these bits all = 0, the 6422 behavior is maintained. That is, the serial control register (offset 0x80) determines the pixel depth.

For this to work, this depends on current drivers having written these bits with zeros beforehand. All reserved register bits should be written with 0s and masked off if read back.

If bits [16:14] are not = 0, the pixel depth is determined in the following way:

Value (binary)	*Pixel depth*
000	Determined by "Serial control," described on page 221
100	24
x01	8
x10	16
x11	32

11.7.3. Raster operation deltas

ProMotion-AT24/3D supports ternary operations, while previous ProMotion family members support binary raster ops.

Raster operations for the ProMotion-6422 and earlier correspond to the binary raster op codes as defined by the Microsoft Windows GDI ROP2. (Refer to the Windows Programmer's Reference.) There is a 4-bit field, where for example 0 = black, 0xC = copy, etc. In the chip there is a 4-bit register which holds the OP-code, the upper 4 bits being reserved.

ProMotion-AT24/3D support ternary operations. The AT24/3D register "Raster operation," described on page 192, is 8 bits wide. The three elements involved in the 3-element ROP are referrred to as source, destination and pattern. In terms of the pattern formats available as defined in register bits M040[23:22], "Drawing engine control," on page 188, the ROP3 is only available in pattern format 2 (8x8x1, monochrome). The pattern operand is not an 8-bit color. Instead individual bits in the Pattern register (offset 0x4F:0x48) are replicated eight times to form bytes which enter the internal ALU.

Select ternary operation by the 8-bit code in the ROP register. A truth table of each ROP can be constructed with P, S, and D as inputs and in that order. The ROP code of a specific logical operation is equal to the bits in the result column read from bottom to top.

Table 11.7.3a Raster operation: R = P*S*D

PSD	R
000	0
001	0
010	0
011	0
100	0
101	0
110	0
111	1

ROP = 80

Table 11.7.3b Raster operation: R = S XOR D

PSD	R
000	0
001	1
010	1
011	0
100	0
101	1
110	1
111	0

ROP = 66

Table 11.7.3c Raster operation: R=S

PSD	R
000	0
001	0
010	1
011	1
100	0
101	0
110	1
111	1

ROP = CC

The limitation is that the pattern bits are not color expanded but are only sign extended before the ROP. This means that the pattern data cannot provide color but only masking.

Table 11.7.3d Raster operation R=P

PSD	R
000	0
001	0
010	0
011	0
100	1
101	1
110	1
111	1

ROP = F0

This ROP will result in a pixel of all ones if pattern bit is 1 or a pixel of all zeros if pattern bit is 0, not the foreground or background colors as you would expect with monochrome pattern expansion.

Backward compatibility: There is no backward compatibility mechanism in the hardware. In software, to recover the previous behavior for a 2-operand ROP, duplicate the data which was written into the low-order nibble, into the high-order nibble as well.

11.7.4. Byte mask

For the 6422, this register at offset 0x47 was 4 bits wide. It enables individual byte lanes in the Graphics engine. The bits are normally set to all 1s at init time.

For ProMotion-AT24/3D, the register "Byte mask," described on page 193, is now 8 bits wide. **All 8 bits now must be set to 1s.** There is no backward compatibility mechanism in the hardware.

11.7.5. 24-bit per pixel modes

The graphics engine can be used in the new 24 bit per pixel modes in a limited way. That is, in terms of the Drawing engine command field, Screen to Screen Blit and rectangle fill operations will work in 24 bit per pixel modes, except that transparency is not supported.

For all 24 bit packed modes set:

- pixel depth(24bpp) = 100
- destination_linear = 1, destination_contiguous=0
- source_linear = 1, source_contiguous=0

Dimension calculations:

- dim_y = y_dimension(in pixels)
- dim_x = x_dimension(in pixels) * 3

11.7.5.1 How to program source and destination location registers in 24-bit mode

The registers are programmed in bytes, so that given a source screen coordinate X,Y in pixels,

- Source Location X = (X*3 + Y*(horizontal resolution, in pixels)*3) / 0x1000
- Source Location Y = (X*3 + Y*(horizontal resolution, in pixels)*3) & 0xFFF

Program Destination Location X,Y the same manner.

11.7.5.2 How to program row pitch register

There is an additional register to be programmed in 24-bit mode, the Destination Row Pitch register, at 0x5C:5D. The value to be programmed is a function of the the resolution, x dimension, and direction of BLIT as follows:

T2B	L2R	Row Pitch Register
1	1	HRES - DIMX
0	1	HRES + DIMX
1	0	HRES + DIMX
0	0	HRES - DIMX

HRES = horizontal resolution, in pixels
DIMX = x dimension, in bytes
T2B = BLIT direction top to bottom,
L2R = BLIT direction left to right.

11.7.5.3 For screen to screen set:

- operation = 0001
- sample command =1000_0001_0000_0101_0000 _0000_0000_0010=81050201

11.7.5.4 For rectangle fill set:

- operation = 0010
- sample command =1000_0001_0000_0101_0000 _0000_0000_0010=81050002

The driver must rotate the fill color in register "Foreground color," described on page 200, according to the destination alignment. fg is one of 3 values in the table below.

- foreground_color = 00543210, R=10, G=32, B=54
- fg24={R,B.G.R,B,G,R}
- rot24= ((address in bytes)/8)%3. Foreground_color = 0x00ffffff & (fg24>> 16*rot24)

Alignment	7	6	5	4	3	2	1	0
2	B	G	R	B	G	R	B	G
1	R	B	G	R	B	G	R	B
0	G	R	B	G	R	B	G	R

11.7.6. Mono-to-color expansion

ProMotion-6422 hardware does not perform mono-to-color expansion when the source data is originating from the CPU (host write). However, in ProMotion-aT3D, mono-to-color expansion can be performed during host writes. To program for this, set bits [9], [11] and [12] (Linear, Contiguous, and Mono) of register "Drawing engine control," described on page 188. Command must be set to Host BLIT write.

The following demonstrates writing a character of width 10 pixels and height 9, with an 8-bit desktop.

- Set Dimension register = 10.
- Do a 32-bit write setting bits [9:0] to mono data bits for top line of char, and bits [16:25] to mono data for line 2. Do this write at address offset 0 of host blit port.
- At address offset 8 of host blit port, do next 32-bit write setting bits[9:0] to mono data bits for line 3 of char, and bits [16:25] to mono data for line 4.
- Continue on in this way until address offset 32, where lines 8 and 9 are written. Then do another write access at offset M040 with arbitrary data; this causes the chip to complete the write to lines 8 and 9.

Other notes for mono-to-color expansion:

- Host address increment is doublewords.
- AT24/3D Monochrome source data must be aligned on 64 bit boundaries.
- AT24/3D the Source Location X register must be written with a value of zero when mono-to-color operations are taking place.

11.7.6.1 Patterns

In the AT24/3D, there is an additional pattern type available, 8 pixels by 8 pixels by 8 bits. . This pattern type can be used in 2-operand ROPS but cannot be used as the pattern part of the three-operand ROPS. The Source Location register is used to define the position of the 8x8x8 pattern.

The command type for which patterns are supported is screen-to-screen BLTs.

The patterns are programmed differently between ProMotion-6422 and AT24/3D. The following table show the register setting for bits [23,22 and 10] in the command register:

Pattern	6422	AT24 /3D
none	000	000
8x8x1	001	100
4x4x4	011	010
8x8x8	-	110

11.7.7. Alignment issues: source and destination addresses

- The AT24/3D data path is twice as wide as the 6422's and therefore data can be rotated by twice as many pixels.
- The GE rotates data by a number of pixels indicated by the LSBs of the difference of source and destination addresses.
- The number of LSBs of difference used to specify data rotation for 8, 16, 32 bpp is shown below for AT24/3D and 6422:

bit depth (BPP)	AT24/3D	6422
8	3 LSBs in AT24	2 LSBs in 6422
16	2 LSBs in AT24	1 LSBs in 6422
32	1 LSBs in AT24	0 LSBs (no rotation possible) in 6422

- ProMotion-6422 measures alignment with respect to 32 bit boundary whereas AT24 measures alignment with repect to a 64 bit boundary.
- Data aligned to an odd 32 bit address is considered aligned by 6422 but will be rotated by 4 pixels (8bpp) in AT24/3D. Conversely, if the data is quad word aligned (address is multiple of 8 bytes), the data will be rotated to match the destination address. In terms of device driver programming what this means is that the source address must be aligned on a 64-bit boundaries. This is only an issue for host-screen blit operations.

11.7.8. Motion Video deltas between ProMotion-6422 and AT24/3D

Unlike ProMotion-6422, the AT24/3D supports two motion video windows, with occlusion. There is no backward compatibility mode in the hardware. Refer to "Video tile buffer registers," on page 217, for a description of AT24/3D video tiling.

12. Motion video notes

12.1 Software standards

The motion video feature set supports multiple software and codec approaches. In particular it gives hardware support for Microsoft's DirectDraw, Intel's Display Control Interface (DCI), and higher level APIs and codecs built on these, including Direct Video, Video for Windows, Indeo video, and others.

12.2 Hardware features

12.2.1. vWindow™, the ProMotion hardware window

The ProMotion-AT3D provides a window of video (or graphics) data, the **vWindow**, up to the size of the full display.

Previous ProMotion family members stored video data in a video memory area pointed to by "Base Address" register at M090–092, which is implementsd differently in ProMotion-aT3D.

Pointers are stored to the screen position where the vWindow is to be displayed. When the controller is refreshing the screen area inside the vWindow, it fetches from the vWindow memory area rather than the on-screen frame buffer memory.

12.2.2. Scaling

Video data are stored in memory in native resolution, e.g. 320×240 pixels. On-chip horizontal and vertical scaling registers permit the data to be displayed at native size or any upscaled size up to full-screen.

12.2.3. Color space conversion

Control bits indicate the format of the vWindow data: RGB (8/15/16/24-bit) or YUV (4:2:2/4:1:1/grayscale). Data format of the vWindow is independent of the desktop or RAMDAC graphics format, which may be 8-bit indexed or 15/16/24-bit direct RGB.

The controller is capable of color space conversion on the fly for data stored in YUV format. An on-chip color space converter accepts 16 equivalent bits per pixel in either CCIR 4:2:2 or upsampled 4:1:1 format, which is easy to generate from standard 4:1:1 formats. Refer to the figures below.

Figure 12.2.3a: CCIR–601 4:2:2 32–bit word

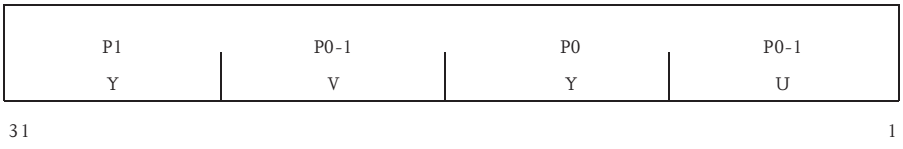


Figure 12.2.3b: Upsampled 4:1:1 32-bit words

The color space converter takes YUV input in one of these formats and outputs RGB data at the precision of the RAMDAC output mode up to 24-bit photorealistic truecolor. The R, G, B values are computed according to CCIR coefficients, as follows:

R	Y + 1.402 V
G	Y – 0.344 U – 0.714 V
B	Y + 1.772 V

12.2.4. Stretch minimum replication

This feature is intended to attenuate blockiness in images with large scale factors. The recommended values for this field are as follows.

horizontal stretch factor <=3	set minimum replication =0
horizontal stretch factor >3, <=5	set minimum replication =1
horizontal stretch factor >5, <=7	set minimum replication =2
horizontal stretch factor >7	set minimum replication =3

12.3 Motion video in the vWindow

12.3.1. In–place video data

Normally, video data are stored in off-screen memory in a contiguous region. Where this is not possible or desired, you may store video data in place by writing it into the on-screen display memory region that is occluded by the vWindow.

In this case, each row of video data should be stored in a different row of on-screen memory. The video surface width should match the VGA offset register rather than the actual width, as adjusted for pixel depth. See "vWindow group 0 data pitch," on page 207 and "vWindow group 0 control," on page 205, and vWindow group 1 counterparts, for register descriptions.

vWindow data is not included in desktop screen captures, such as Print Screen.

12.3.2. vWindow straddling screen boundaries

In some cases a vWindow may straddle one or more edges of the screen. At such times the motion video registers reflect only the visible portion of the vWindow as if it were the entire vWindow. Regardless whether the vWindow straddles the edge of the screen, the left and right edges of the vWindow must be 32-bit aligned on the screen, and the source data displayed in that window must be 32-bit aligned in display memory.

For example, in 8-bit desktop mode both left and right edges of the vWindow must be aligned to 4-pixel boundaries.

There are cases where a codec or other driver always writes the entire frame's source data but only a portion is to be displayed. Straddle is accomplished by setting vWindow data base address to point to the first displayed pixel (32-bit alignment requirement still applies) and setting vWindow data pitch to describe only that portion of the vWindow that is displayed. Data pitch contains a larger value than vWindow data width when straddling the left or right edges of vWindow; the difference represents the position of the video data hidden beyond the left or right edges of the screen.

12.3.3. vWindow alignment restrictions

There are no alignment restrictions for the position on screen where the vWindow may be displayed; or the memory address to store video data. Video data maybe on any byte boundary.

12.4 Video tile allocation procedure

12.4.1. Suggested tile allocation procedure

1) Determine all the unique destination Y coordinates for all the rectangles in the cliplists of all the active windows.
2) Create horizontal "Strip" rectangles from the set of Y coordinates.
3) For each rectangle in the cliplist of each window, clip the rectangle against the set if Strip rectangles. If the result has non-zero area, allocate a Tile for this rectangle. At the same time:
   - If there is no overlap between any of the rectangles and any of the Strip rectangles, allocate a Space tile for that Strip.
   - Maintain the rightmost Tile associated with each Strip, in order to set the Tile Rightmost bit, 2x0[4]
4) Tiles must be allocated in sequential order across all active windows.

You must zero out tile register values for tiles which are not used in a given screen configuration.

12.4.2. Space tiles

Where there is a vertical gap on the desktop with no motion video data, including at the top, but not the bottom), you must allocate special "space tiles."

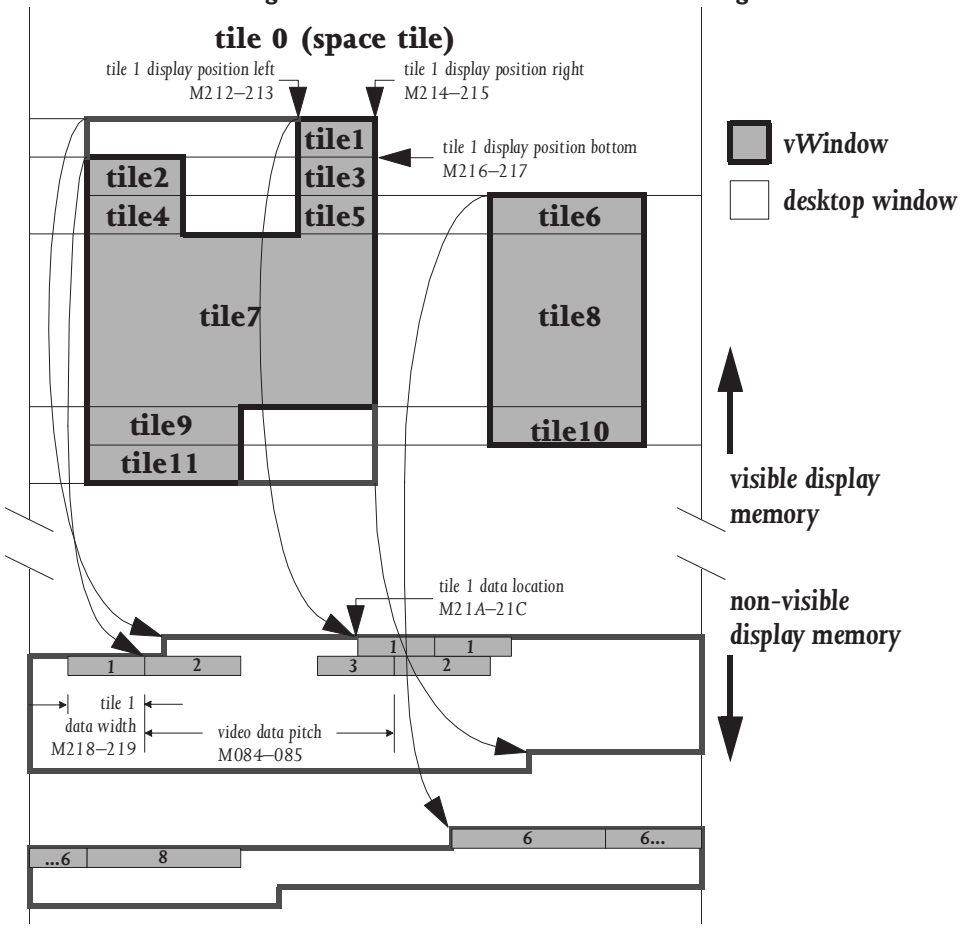
- Set Tile Left = 0xFFF.
- Set Tile Rightmost = 1.
- Set Tile Bottom as for normal tiles.
- Set other tile register values = 0.

Refer to Figure 12.4.3, "ProMotion-AT3D video tiling," for an example using a space tile,

12.4.3. Horizontal video tiling

- When two vWindows overlap horizontally, they must be broken up into tiles according to the vertical geometry, as shown in Figure 12.4.3, "ProMotion-AT3D video tiling."
- ProMotion-aT3D does not support tiles being programmed such that they adjoin each other horizontally. There must be a horizontal gap of at least 8 pixels between tiles on the same scan line.

Figure 12.4.3 ProMotion-AT3D video tiling



13. ProMotion-AT3D bi-endian support

Bi-endian support goals are two-fold: frame buffer data must appear to the Power PC processor like conventional memory, under all possible combinations of reads, writes, sizes and alignments; and frame buffer data as stored must be usable by all the modules within the Promotion. Modifying memory-mapped register accesses for ProMotion bi-endian hardware since that can be handled in the driver software level.

13.1 Overview

There are several modules within ProMotion which independently request data accesses from the frame buffer. However all memory accesses pass through the memory interface (MI) module. Therefore the byte swapping logic necessary for bi-endian support can be centralized in this module. The other modules can dynamically request different types of byte swapping, depending on the size of the data types which they are accessing, and the bi-endian configuration. The PCI interface module can also request the required swapping, depending on which of two PCI memory space apertures is being accessed.

The bi-endian support logic allows the frame buffer data to be stored in either big-endian or little-endian format, as desired. The double PCI apertures also allow a mixed format whereby the frame buffer data accessed by big-endian processors can be in big-endian format, and at the same time frame buffer data accessed by little-endian processors can be little-endian.

13.2 CPU register interface

13.2.1. Bi-endian control register

This register determines the bi-endian configuration of the Promotion.The size is two bytes. It appears in memory-mapped register space at offsets 0xDD:DC.

Bits	Description	Notes
1:0	Host Memory Aperture 0 transform code	Memory offset 0-8MB
3:2	Host Memory Aperture 1 transform code	Memory offset 8-16MB
5:4	Pixel data module transform control	
7:6	GE module transform control	
9:8	3D module transform control	
11:10	TV module transform control	

13.3 Data transform codes

Code	Description
0	no transform (default)
1	16-bit transforms
2	32-bit transforms
3	reserved

13.3.1. Little-endian module to big-endian memory: 16-bit transforms

Access by little-endian module	Example data	$\overline{BE}$ on internal memory bus	Data on internal memory bus	$\overline{BE}$ to frame buffer	Data to/from big-endian frame buffer [63:0]
byte , address 0	0x10	11111110	XXXXXXX10	11111101	XXXXXX10X
byte , address 1	0x11	11111101	XXXXXX11X	11111110	XXXXXXX11
byte , address 2	0x12	11111011	XXXXX12XX	11110111	XXXX12XXX
byte , address 3	0x13	11110111	XXXX13XXX	11111011	XXXXX13XX
byte , address 4	0x14	11101111	XXX14XXXX	11011111	XX14XXXXX
byte , address 5	0x15	11011111	XX15XXXXX	11101111	XXX15XXXX
byte , address 6	0x16	10111111	X16XXXXXX	01111111	16XXXXXXX
byte , address 7	0x17	01111111	17XXXXXXX	10111111	X17XXXXXX
half word, address 0	0x1110	11111100	XXXXXX1110	11111100	XXXXXX1011
half word, address 2	0x1312	11110011	XXXX1312XX	11110011	XXXX1213XX
half word, address 4	0x1514	11001111	XX1514XXXX	11001111	XX1415XXXX
half word, address 6	0x1716	00111111	1716XXXXXX	00111111	1617XXXXXX
word , address 0	0x13121110	11110000	XXXX13121110	11110000	XXXX12131011
word , address 4	0x17161514	00001111	17161514XXXX	00001111	16171415XXXX
doubleword	0x1716151413121110	00000000	1716151413121110	00000000	1617141512131011

13.3.2. Little-endian module to big-endian memory: 32-bit transforms

Access by little-endian module	Example data	$\overline{BE}$ on internal memory bus	Data on internal memory bus	$\overline{BE}$ to frame buffer	Data to/from big-endian frame buffer [63:0]
byte , address 0	0x10	11111110	XXXXXXX10	11110111	XXXX10XXX
byte , address 1	0x11	11111101	XXXXXX11X	11111011	XXXXX11XX
byte , address 2	0x12	11111011	XXXXX12XX	11111101	XXXXXX12X
byte , address 3	0x13	11110111	XXXX13XXX	11111110	XXXXXXX13
byte , address 4	0x14	11101111	XXX14XXXX	01111111	14XXXXXXX
byte , address 5	0x15	11011111	XX15XXXXX	10111111	X15XXXXXX
byte , address 6	0x16	10111111	X16XXXXXX	11011111	XX16XXXXX
byte , address 7	0x17	01111111	17XXXXXXX	11101111	XXX17XXXX
half word, address 0	0x1110	11111100	XXXXXX1110	11110011	XXXX1011XX
half word, address 2	0x1312	11110011	XXXX1312XX	11111100	XXXXXX1213
half word, address 4	0x1514	11001111	XX1514XXXX	00111111	1415XXXXXX
half word, address 6	0x1716	00111111	1716XXXXXX	11001111	XX1617XXXX
word , address 0	0x13121110	11110000	XXXX13121110	11110000	XXXX10111213
word , address 4	0x17161514	00001111	17161514XXXX	00001111	14151617XXXX
doubleword	0x1716151413121110	00000000	1716151413121110	00000000	1415161710111213

13.3.3. Big-endian processor to little-endian memory: 16-bit transforms

Access by big-endian processor	Example data	$\overline{BE}$ on PCI	Data on PCI AD[31:0]	$\overline{BE}$ to frame buffer	Data in little-endian frame buffer [31:0]
byte , address 0	0x10	1110	XXXXXX10	1101	XXXX10XX
byte , address 1	0x11	1101	XXXX11XX	1110	XXXXXX11

Access by big-endian processor	Example data	$\overline{BE}$ on PCI	Data on PCI AD[31:0]	$\overline{BE}$ to frame buffer	Data in little - endian frame buffer [31:0]
byte , address 2	0x12	1011	XX12XXXX	0111	12XXXXXX
byte , address 3	0x13	0111	13XXXXXX	1011	XX13XXXX
word, address 0	0x1011	1100	XXXX1110	1100	XXXX1011
word, address 2	0x1213	0011	1312XXXX	0011	1213XXXX
double word	0x10111213	0000	131211110	0000	12131011

13.3.4. Big-endian processor to little-endian memory: 32-bit transforms

Access by big-endian processor	Example data	$\overline{BE}$ on PCI	Data on PCI AD[31:0]	$\overline{BE}$ to frame buffer	Data in little - endian frame buffer [31:0]
byte , address 0	0x10	1110	XXXXXX10	0111	10XXXXXX
byte , address 1	0x11	1101	XXXX11XX	1011	XX11XXXX
byte , address 2	0x12	1011	XX12XXXX	1101	XXXX12XX
byte , address 3	0x13	0111	13XXXXXX	1110	XXXXXX13
word, address 0	0x1011	1100	XXXX1110	0011	1011XXXX
word, address 2	0x1213	0011	1312XXXX	1100	XXXX1213
double word	0x10111213	0000	131211110	0000	10111213

13.4 Bi-endian implementation notes

Each module presents two additional signals to the MI at the time of a memory request. The signals contain the transform control code as defined above. The MI dynamically performs swapping, for reads and writes, according to the transform code for each module. The transform code coming from a particular module can also change dynamically, depending on the type of access that it being performed by that module. The following tables indicate how each type of module asserts the transform code signals.

13.4.1. Graphics engine, 3D modules

Transform control[0]	Data accessed	Transform requested
0	X	none
1	monochrome data	none
1	4 or 8 bit pixels or texels	none
1	15,16 bit pixels or texels	16
1	32 bit pixels or texels	32

13.4.2. PD, TV modules

Transform control[0]	Data accessed	Transform requested
0	X	none
1	Hardware cursor data	none
1	8 bit pixels	none
1	15, 16 bit RGB, YUV 4:2:2, any other 16-bit data type	16
1	32 bit pixels	32

HOST: uses codes in bi-endian control register, depending on aperture.

14. 3D programming notes

14.1 Texture mapping

Use these steps for rendering a textured object from a series of triangles.

1. Load an 8-bit texture in off-screen display memory.
2. Load M308, "Texture map base address," described on page 286, to point to the first byte of the texture.
3. Load M30C, "Texel index offset," described on page 288. with the texture height width and format.
   - 3.a If the texture is indexed, load the TLUT with 24-bit RGB values for each color in the texture.
   - 3.b If the texture is a 4-bit texture and you aren't starting with TLUT address 0, load M30E, "Texel index offset," described on page 288, to specify which TLUT location corresponds to texel color 0.
4. Load M300, "Polygon engine control 0," described on page 282, with parameters.

15. Standard VGA register descriptions

15.1 VGA attribute controller registers

Writing to port 3C0 writes index and data alternately. Reading port 3C0 returns index and reading port 3C1 returns data. Reading port 3DA or 3BA causes next 3C0 write to update the index, depending on the mode set at 3C2[0], "Item select/miscellaneous output," on page 124.

15.1.1. Index

Use this register to specify the register accessed with the next read to 3C1 or the next write to 3C0.

Read/write:	r/w	*Address:*	3C0h
Default:	Undefined.	*Address index:*	-

7	6	5	4	3	2	1	0
		palette access	attribute index				

Bits	Description
[5]	Palette access by CPU. 1 = normal display. 0 = entire screen displays border color specified in 3C0.11.
[4:0]	Attribute index.

15.1.2. Palette registers 0–15

Use these registers to define the EGA color palette.

Read/write:	r/w	*Address:*	3C0h
Default:	Undefined.	*Address index:*	00–0Fh

7	6	5	4	3	2	1	0
		palette output					

Bits	Description
[5:0]	Palette output.

15.1.3. Mode control

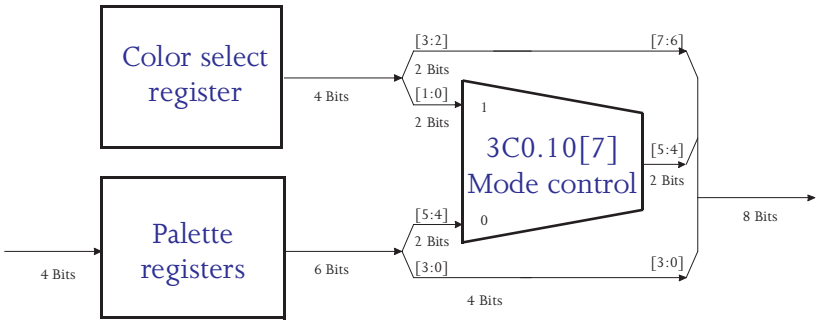
Use this register to determine attribute control bits.

Read/write:	r/w	*Address:*	3C0h
Default:	0h	*Address index:*	10h

7	6	5	4	3	2	1	0
select source	256 col mode	pix pan enable		char blink enable	line graphics	mono text	graphics/alpha

Bits	Description
[0]	Graphics/Alpha mode. 1 = graphics mode 0 = alphanumeric mode
[1]	Monochrome text attribute (for alphanumeric mode if blinking is enabled). 1 = Monochrome blinking 0 = Color blinking
[2]	Line graphics codes 1 = 9th dot copies 8th dot for chars C0-DFh 0 = 9th dot is background color
[3]	Character blink enable. 1 = blink at vertical refresh frequency ÷ 32. 0 = blink disabled.
[4]	Reserved.
[5]	Pixel panning enable. 1 = pan split screens separately. 0 = pan split screens together.
[6]	256 color mode. 1 = 256 color mode on. 0 = 256 color mode off.
[7]	Select source for lookup table address [5:4]. 1 = use 3C0.14 [1:0], "Color select," on page 123. 0 = use 3C0.00–0F [5:4], "Palette registers 0-15," on page 119.

Figure 15.1.3 Mode control



15.1.4. Overscan color

Use this register to determine the border color. For direct color modes the overscan color is applied as a RGB332 value. See "vWindow data formats in display memory" on page 303.

Read/write:	r/w	*Address:*	3C0h
Default:	0h	*Address index:*	11h

7	6	5	4	3	2	1	0
border color							

Bits	Description
[7:0]	Border color.

15.1.5. Color plane enable

Use this register to enable the four VGA color planes, and to determine input for register 3DA/3BA, "Input status 1," on page 126.

Read/write:	r/w	*Address:*	3C0h
Default:	0h	*Address index:*	12h

7	6	5	4	3	2	1	0
		display pixel readback mux		color plane 3	color plane 2	color plane 1	color plane 0

Bits	Description
[0]	Color plane 0 enable. 1 = plane enabled. 0 = plane disabled.
[1]	Color plane 1 enable. 1 = plane enabled. 0 = plane disabled.
[2]	Color plane 2 enable. 1 = plane enabled. 0 = plane disabled.
[3]	Color plane 3 enable. 1 = plane enabled. 0 = plane disabled.
[5:4]	Display pixel readback mux control. 11 = 3DA [5:4] returns {VID7, VID6}. 10 = 3DA [5:4] returns {VID3, VID1}. 01 = 3DA [5:4] returns {VID5, VID4}. 00 = 3DA [5:4] returns {VID2, VID0}.

15.1.6. Horizontal pixel panning

Use this register to specify the number of pixels to shift the display horizontally to the left. A maximum pan of seven pixels can be shifted except for these modes:

- 9-bit character mode, the output can be shifted a maximum of eight pixels.
- 256 color modes, the output can be shifted a maximum of four pixels.

Therefore in 8-bit character mode, and all graphics modes but 256-color mode, the output can be shifted a maximum of seven pixels.

Read/write:	r/w	*Address:*	3C0h
Default:	0h	*Address index:*	13h

7	6	5	4	3	2	1	0
				horizontal pan			

Bits	*Description (9-bit characters)*
[3:0]	Horizontal pan. 1xxx = no shift. 0111 = 8 bits left. 0110 = 7 bits left. 0101 = 6 bits left. 0100 = 5 bits left. 0011 = 4 bits left. 0010 = 3 bits left. 0001 = 2 bits left. 0000 = 1 bit left.

Bits	*Description (8-bit characters)*
[3:0]	Horizontal pan. 1xxx = 1 bit right. 0111 = 7 bits left. 0110 = 6 bits left. 0101 = 5 bits left. 0100 = 4 bits left. 0011 = 3 bits left. 0010 = 2 bits left. 0001 = 1 bits left. 0000 = no shift.

Bits	Description (Mode 13)
[3:0]	Horizontal pan. 1xxx = no shift. 0111 = no shift. 0110 = 6 bits left. 0101 = no shift. 0100 = 2 bits left. 0011 = no shift. 0010 = 1 bit left. 0001 = no shift. 0000 = no shift.

15.1.7. Color select

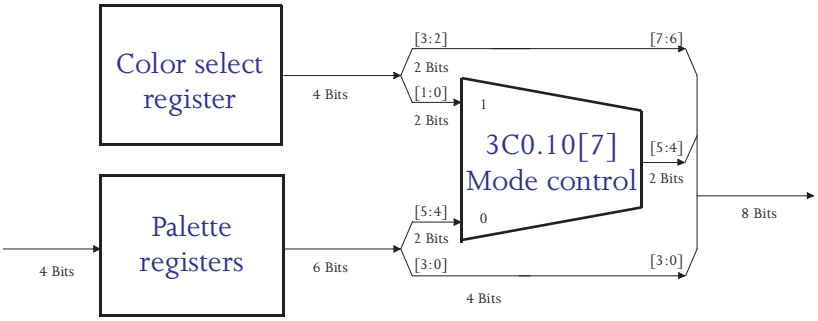
Use this register to expand 4 bit color information to drive 8 bits. This register is used in conjunction with 3C0.10[7], "Mode control," on page 119, and 3C0.00–0F [5:0], "Palette registers 0-15," on page 119. This register is used only in 4-bit planar mode.

Read/write:	r/w	*Address:*	3C0h
Default:	0h	*Address index:*	14h

7	6	5	4	3	2	1	0
				color select 1		color select 0	

Bits	Description
[1:0]	Color select 0. If 3C0.10 [7] is 1, then these bits combine with 3C0.00–0F [3:0] and 3C0.14 [3:2] to provide 8 bits of color information. These bits are ignored if 3C0.10 [7] is 0.
[3:2]	Color select 1. If 3C0.10 [7] is 1, then these bits combine with bits 3C0.00–0F [3:0] and bits 3C0.14 [1:0] to provide 8 bits of color information. If Mode control 3C0.10 [7] is set to 1 then these color select bits combine with bits from 3C0.00–0F [5:0] to provide 8 bits of color information.

Figure 15.1.7 Color select



15.2 VGA general registers

15.2.1. Item select/miscellaneous output

Refer to "Input status 0," on page 126, for information on reading 3C2.

	r	*Address:*	3CCh
Read/write:	w	*Address:*	3C2h
Default:	0h	*Address index:*	-

7	6	5	4	3	2	1	0
VSYNC polarity	*HSYNC*	*mem page*		*VCLK select*		*enable host DMA*	*color/mono*

Bits	Description
[0]	Select mode. 1 = color mode: 3DA, 3D4, 3D5 enabled. 0 = monochrome: 3BA, 3B4, 3B5 enabled.
[1]	Enable host display memory access. 1 = access enabled. 0 = access disabled. When disabled, the device does not respond to reads or writes to display memory. Reads return random data.
[3:2]	Clock select. Refer to "Clock registers and formulas," on page 257 for information on VCLK. 1x = programmable VCLK. 01 = VCLK default 1. 00 = VCLK default 0. ☞ Setting 0C8[9], "VGA override," on page 226, overrides [3:2] with a value of 11b.
[4]	Reserved.
[5]	Select high 64K memory page. 1 = select even memory locations. 0 = select odd memory locations. In some configurations there are two 64K pages, both at A000:0–AFFF.F; this bit determines which one is mapped into that space.
[6]	HSYNC polarity. 1 = negative. 0 = positive.
[7]	VSYNC polarity. 1 = negative. 0 = positive.

15.2.2. Feature control/vertical enable

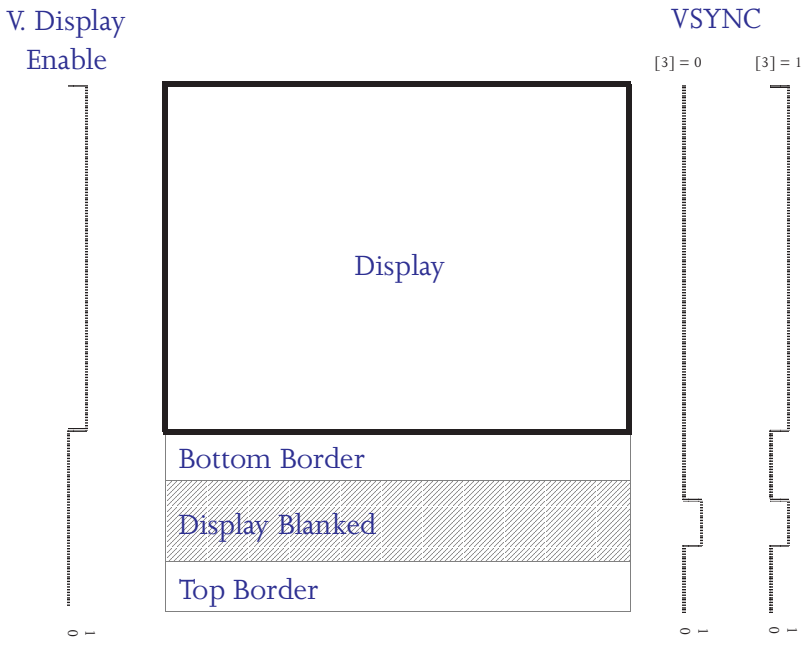
Use this register to control VSYNC.

	r	*Address:*	3CAh
Read/write:	w	*Address:*	3BA, 3DA
Default:	0h	*Address index:*	-

7	6	5	4	3	2	1	0
				VSYNC control			

Bits	Description
[2:0]	Reserved.
[3]	VSYNC control. 1 = VSYNC OR'd with vertical display enable. 0 = normal VSYNC.

Figure 15.2.2 Vertical enable



15.2.3. Input status 0

Use this read-only register to obtain the state of the vertical interrupt and sense pin.

Refer to "Item select/miscellaneous output," on page 124, for information on writing 3C2.

Read/write:	r	*Address:*	3C2h
Default:	Undefined.	*Address index:*	-

7	6	5	4	3	2	1	0
vertical interrupt			inverted sense pin				

Bits	Description
[3:0]	Reserved.
[4]	Inverted sense pin. 1 = pin LOW—typically means $\overline{\text{MDET}}$ active and monitor present. 0 = pin HIGH—typically means $\overline{\text{MDET}}$ inactive with no monitor present.
[6:5]	Reserved.
[7]	Vertical interrupt. 1 = active. 0 = inactive.

15.2.4. Input status 1

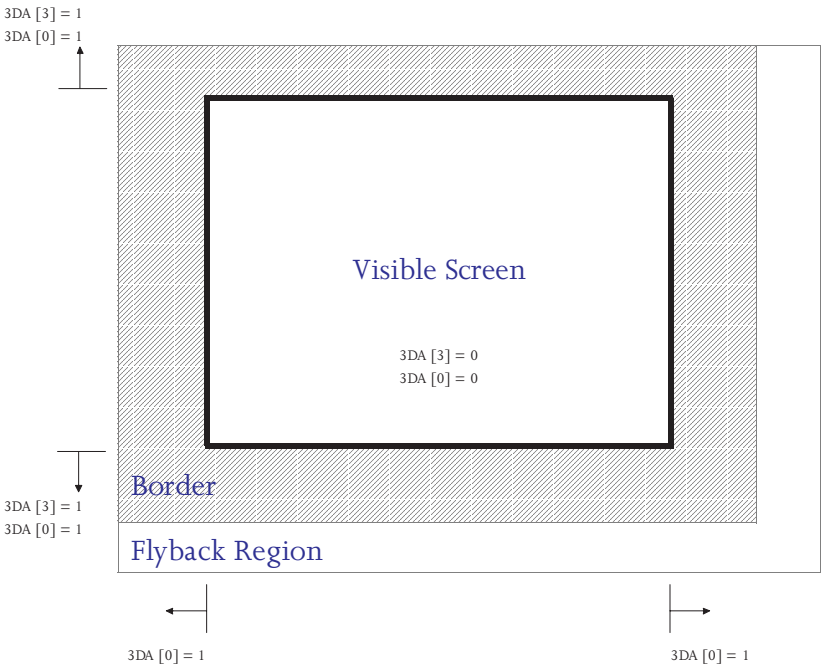
Use this register to read current monitor control values.

Read/write:	r	*Address:*	3BA, 3DAh
Default:	Undefined.	*Address index:*	-

7	6	5	4	3	2	1	0
		display pixel readback MUX control		vertical retrace			pixel display

Bits	Description
[0]	Pixel display. 1 = inactive. 0 = active.
[2:1]	Reserved.
[3]	Vertical retrace. 1 = active. 0 = inactive.
[5:4]	Display pixel readback mux control. Returns bits specified by 3C0.12 [5:4], "Color plane enable," on page 121.

Figure 15.2.4 Pixel status



15.3 VGA sequencer registers

15.3.1. Sequencer index

Use this register to specify the register accessed with the next read/write to 3C5.

Read/write:	r/w	*Address:*	3C4h
Default:	Undefined.	*Address index:*	-

7	6	5	4	3	2	1	0
				sequencer read/write index			

Bits	Description
[3:0]	Index for read/write.

15.3.2. Reset

Read/write:	r/w	*Address:*	3C5h
Default:	0h	*Address index:*	00h

7	6	5	4	3	2	1	0
						reset	

Bits	Description
[1:0]	This register has no effect on the operation of the ProMotion-aT3D. Reads return the value written.

15.3.3. Clocking mode

Use this register to alter clock rates. 3C5.01 [5] disables display refresh, allowing the host direct access to display memory. 3C5.01 [3] divides clock by 2, providing compatibility with low-resolution modes. 3C5.01 [0] determines clock for 8- or 9-dot character.

Read/write:	r/w	*Address:*	3C5h
Default:	0h	*Address index:*	01h

7	6	5	4	3	2	1	0
		screen off		dot clock			character clock

Bits	Description	
[0]	Character clock.	
	1 =	8 dot wide characters.
	0 =	9 dot wide characters.
[2:1]	Reserved.	
[3]	Dot clock.	
	1 =	dot clock/2.
	0 =	normal dot clock.
[4]	Reserved.	
[5]	Screen off.	
	1 =	screen disabled.
	0 =	normal operation.

15.3.4. Map mask

Use this register to control writes to each plane in planar mode.

Read/write:	r/w	*Address:*	3C5h
Default:	0h	*Address index:*	02h

7	6	5	4	3	2	1	0
				map plane 3	*map plane 2*	*map plane 1*	*map plane 0*

Bits	Description	
[0]	Map plane 0.	
	1 =	write enabled.
	0 =	write disabled.
[1]	Map plane 1.	
	1 =	write enabled.
	0 =	write disabled.
[2]	Map plane 2.	
	1 =	write enabled.
	0 =	write disabled.
[3]	Map plane 3.	
	1 =	write enabled.
	0 =	write disabled.

15.3.5. Character map select

Use this register to specify the primary and secondary character sets for text modes.

Read/write:	r/w	*Address:*	3C5h
Default:	Undefined.	*Address index:*	03h

7	6	5	4	3	2	1	0
		font *b*	font *a*	font *b*		font *a*	

Bits	Description
[4, 1:0]	Select font A. 111 = 56K offset: character map 0. 110 = 40K offset: character map 1. 101 = 24K offset: character map 2. 100 = 8K offset: character map 3. 011 = 48K offset: character map 4. 010 = 32K offset: character map 5. 001 = 16K offset: character map 6. 000 = 0K offset: character map 7.
[5, 3:2]	Select font B. 111 = 56K offset: character map 0. 110 = 40K offset: character map 1. 101 = 24K offset: character map 2. 100 = 8K offset: character map 3. 011 = 48K offset: character map 4. 010 = 32K offset: character map 5. 001 = 16K offset: character map 6. 000 = 0K offset: character map 7.

15.3.6. Memory mode

Read/write:	r/w	*Address:*	3C5h
Default:	0h	*Address index:*	04h

7	6	5	4	3	2	1	0
				sequencer odd/even		extended memory	

Bits	Description
[0]	Reserved.
[1]	Extended memory. 1 = >64K present. 0 = 64K present. This bit provided for compatibility only, and should always be set.
[3:2]	Odd/even sequencer mode. 1x = chain 4 mode. 01 = unchained mode. 00 = chain 2 mode. Bit [2] should be set opposite to 3CF.5 [4], "Graphics mode," on page 134.

15.4 VGA graphics controller registers

15.4.1. Graphics index

Use this register to determine the register accessed with the next read/write to 3CF.

Read/write:	r/w	*Address:*	3CEh
Default:	Undefined.	*Address index:*	-

7	6	5	4	3	2	1	0
				graphics index			

Bits	Description
[3:0]	Graphics index.

15.4.2. Set/reset

Use this register to specify the value to be written into the display memory planes in planar mode. It operates in conjunction with 3CF.01 [3:0], "Enable set/reset," on page 132.

Read/write:	r/w	*Address:*	3CFh
Default:	Undefined.	*Address index:*	00h

7	6	5	4	3	2	1	0
				plane 3	plane 2	plane 1	plane 0

Bits	Description
[0]	Value for plane 0.
[1]	Value for plane 1.
[2]	Value for plane 2.
[3]	Value for plane 3.

15.4.3. Enable set/reset

Use this register to enable the values in 3CF.00 [3:0], "Set/reset," on page 132, to be written to the appropriate display memory planes.

Read/write:	r/w	*Address:*	3CFh
Default:	Undefined.	*Address index:*	01h

7	6	5	4	3	2	1	0
				enable plane 3	enable plane 2	enable plane 1	enable plane 0

Bits	Description
[0]	Enable value in 3CF.00 [0] to be written into display memory plane 0. 1 = write enabled. 0 = write disabled.
[1]	Enable value in 3CF.00 [1] to be written into display memory plane 1. 1 = write enabled. 0 = write disabled.
[2]	Enable value in 3CF.00 [2] to be written into display memory plane 2. 1 = write enabled. 0 = write disabled.
[3]	Enable value in 3CF.00 [3] to be written into display memory plane 3. 1 = write enabled. 0 = write disabled.

15.4.4. Color compare

Use this register specifies the color against which pixels in display memory are compared during a VGA read operation.

Read/write:	r/w	*Address:*	3CFh
Default:	Undefined.	*Address index:*	02h

7	6	5	4	3	2	1	0
				color compare			

Bits	Description
[3:0]	Color compare.

15.4.5. Data rotate

Use this register for determining operations in write modes 0 and 3. Bits [4:3] of this register set a logical output between source and readback latch for writing data to display memory. Bits [2:0] of this register rotate data up to seven positions prior to alteration by the set/reset register.

Read/write:	r/w	*Address:*	3CFh
Default:	Undefined.	*Address index:*	03h

7	6	5	4	3	2	1	0
			function select		rotate count		

Bits	Description
[2:0]	Rotate count.
[4:3]	Function select. Valid only for write mode 0. 11 = Source XOR readback latch. 10 = Source OR readback latch. 01 = Source AND readback latch. 00 = Source copy.

15.4.6. Read map select

Use this register to select the display memory plane.

Read/write:	r/w	*Address:*	3CFh
Default:	Undefined.	*Address index:*	04h

7	6	5	4	3	2	1	0
						read map select	

Bits	Description
[1:0]	Read map select. Used for read mode 0. 11 = plane 3. 10 = plane 2. 01 = plane 1. 00 = plane 0.

15.4.7. Graphics mode

Use this register to configure data shift registers and to specify read/write mode.

Read/write:	r/w	*Address:*	3CFh
Default:	Undefined.	*Address index:*	05h

7	6	5	4	3	2	1	0
			odd/even mode	read mode		write mode 0–3	

Bits	Description
[1:0]	Write mode. Determines how data is written into four display memory planes. 11 = write mode 3. Set/reset, 8 bits host rotated & bit mask. 10 = write mode 2. 4 bits host, bit mask. 01 = write mode 1. Data written into each planes' readback latches. 00 = write mode 0. 8 bits host, rotate, enable set/reset, set/reset, raster operation, bit mask. Write modes specify how the 8-bit host byte is translated into the 32-bit display memory region addressed by the host address This translation also includes several other VGA graphics controller registers. Each byte address refers to one byte per plane map.
[2]	Reserved.
[3]	Read mode. 1 = read mode 1. Host reads data via color compare registers. 0 = read mode 0. Host reads data directly from display memory.
[4]	Odd/even mode. 1 = graphics controller in odd/even addressing mode. 0 = normal mode. This bit should be set opposite to 3C5.4 [2], "Memory mode," on page 131.
[7:5]	Reserved.

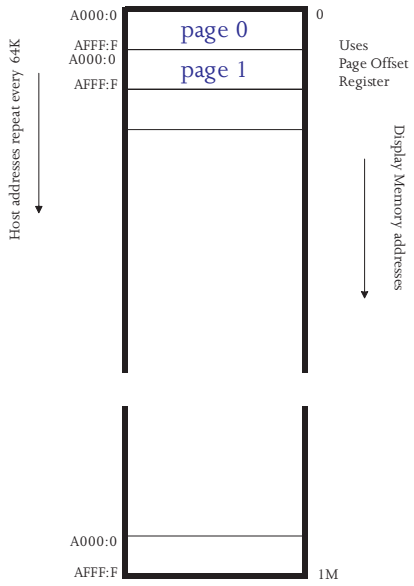
15.4.8. Miscellaneous

Read/write:	r/w	*Address:*	3CFh
Default:	0h	*Address index:*	06h

7	6	5	4	3	2	1	0
				memory aperture		*chain 2 select*	*graphics/alpha*

Bits	Description
[0]	Graphics/alpha mode select. 1 = graphics mode. 0 = text mode.
[1]	Chain 2 select. Used for MDA emulation.
[3:2]	Memory aperture. 11 = B800:0-BFFF:F (32KB: CGA). 10 = B000:0-B7FF:F (32KB). 01 = A000:0-AFFF:F (64KB: EGA-VGA). 00 = A000:0-BFFF:F (128KB: Extended).

Figure 15.4.8 Memory aperture



Display Memory

15.4.9. Color don't care

Use this register to determine which planes are involved in color compares. These bits mask against 3CF.2 [3:0], "Color compare," on page 133, during planar mode readback. Pixels that match the compare return 1, others return 0. Use 3CF.5 [3], "Graphics mode," on page 134, to select this comparison.

Read/write:	r/w	*Address:*	3CFh
Default:	Undefined.	*Address index:*	07h

7	6	5	4	3	2	1	0
				plane 3	plane 2	plane 1	plane 0

Bits	Description
[0]	Color don't care. 1 = plane 0 active. 0 = plane 0 inactive.
[1]	Color don't care. 1 = plane 1 active. 0 = plane 1 inactive.

Bits	Description
[3]	Color don't care. 1 = plane 3 active. 0 = plane 3 inactive.
[2]	Color don't care. 1 = plane 2 active. 0 = plane 2 inactive.

15.4.10. Bit mask

Use this register to enable writing to display memory in write modes 0, 2, and 3.

Read/write:	r/w	*Address:*	3CFh
Default:	Undefined.	*Address index:*	08h

7	6	5	4	3	2	1	0
bit mask							

Bits	Description
[7:0]	Bit mask. Enables writing to display memory on a bit basis. Any bit of this register set to 1 permits the corresponding bit to be written into display memory.

15.5 VGA CRTC registers

All VGA CRTC registers are supported. In addition, the horizontal and vertical timing, start, and offset have been extended at 3D5.19–1D, which are described starting on page 181.

A bit exists to lock VGA CRTC registers. When the lock bit is set, writes to the VGA portion of the CRTC registers (3D4 index 0–18) are ignored. When the lock bit is not set, writes to the VGA portion of any CRTC register cause the extended CRTC bits of all registers to be reset, so in order to load extended values into these registers, the VGA portions of all CRTC registers must be loaded first. Writing any VGA CRTC register (assumed to be a mode-switch) also disables cursor enable and motion video enable.

15.5.1. CRTC index

Use this register to specify the register accessed with the next read or write to address 3D5.

Read/write:	r/w	*Address:*	3D4h
Default:	Undefined.	*Address index:*	-

7	6	5	4	3	2	1	0
		CRTC index					

Bits	Description
[5:0]	CRTC index.

15.5.2. Horizontal total

Use this register to specify the number of character clocks per horizontal period.

Note that writing to this register automatically resets the following other register bits unless you set the Extended CRTC autoreset register to disable the autoreset feature. Refer to "Extended CRTC autoreset," on page 183, for a description of the autoreset feature.

Registers reset by writing 3D5.00, "Horizontal total."	
3D5.19 [7:0]	"Horizontal interlaced start," on page 181.
3D5.1A [7:0]	"Vertical extended overflow," on page 181.
3D5.1B [7:0]	"Horizontal overflow," on page 182.
3D5.1C [7:0]	"Serial overflow," on page 182.
3D5.1D [7:0]	"Character clock adjust," on page 183.
M080 [6:0]	"Serial control," on page 221.
M082 [0]	"vWindow group 0 control," on page 205.
M092 [0]	"vWindow group 1 control," on page 211.
M0C0 [9:0]	"Page offset," on page 222.
M0D2 [16]	"Monitor interlace control," on page 232.

Registers reset by writing 3D5.00, "Horizontal total."	
M0E0 [7:0]	"Color correction," on page 253.
M0E4 [3:0]	"DAC control," on page 254
M140 [1:0]	"Hardware cursor control," on page 237.

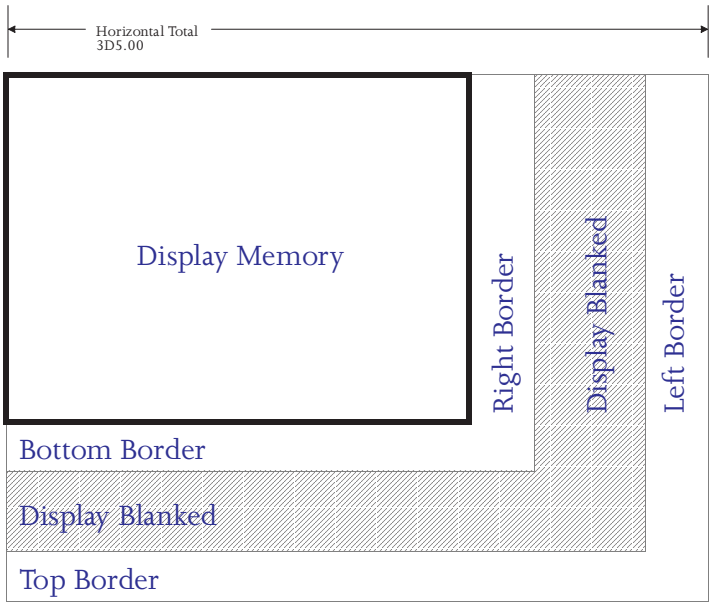
Note that unlike most other CRTC registers, where the actual value is "less 1," this register is "less 5." For example, a horizontal period of 128_{10} character clocks would be coded as FBh.

Read/write:	r/w	*Address:*	3D5h
Default:	Undefined.	*Address index:*	00h

7	6	5	4	3	2	1	0
horizontal total							

Bits	Description
[7:0]	Horizontal total [7:0] of [8:0] less 5_{10}. Horizontal total [8] is stored in 3D5.1B[0]. See "Horizontal overflow," on page 182.

Figure 15.5.2 Horizontal total



15.5.3. Horizontal display enable end

Use this register to specify the number of character clocks during horizontal display time.

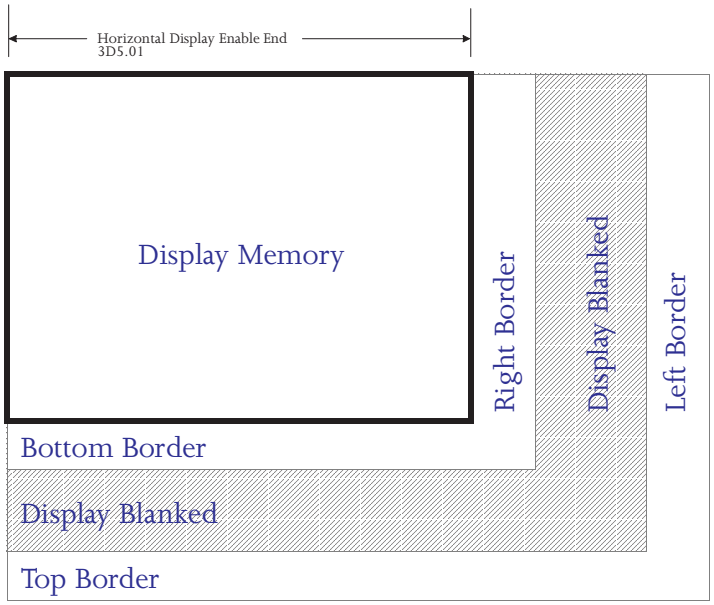
Any write to this register resets bit [0] of "vWindow group 0 control," on page 205, and "vWindow group 1 control," on page 211.

Read/write:	r/w	*Address:*	3D5h
Default:	Undefined.	*Address index:*	01h

7	6	5	4	3	2	1	0
horizontal display enable end							

Bits	Description
[7:0]	Horizontal display enable end [7:0] of [8:0], less 1. Horizontal display enable end [8] is stored in 3D5.1B[1]. See "Horizontal overflow," on page 182.

Figure 15.5.3 Horizontal display enable end



15.5.4. Horizontal blank start

Use this register to specify the character clock where horizontal blanking begins.

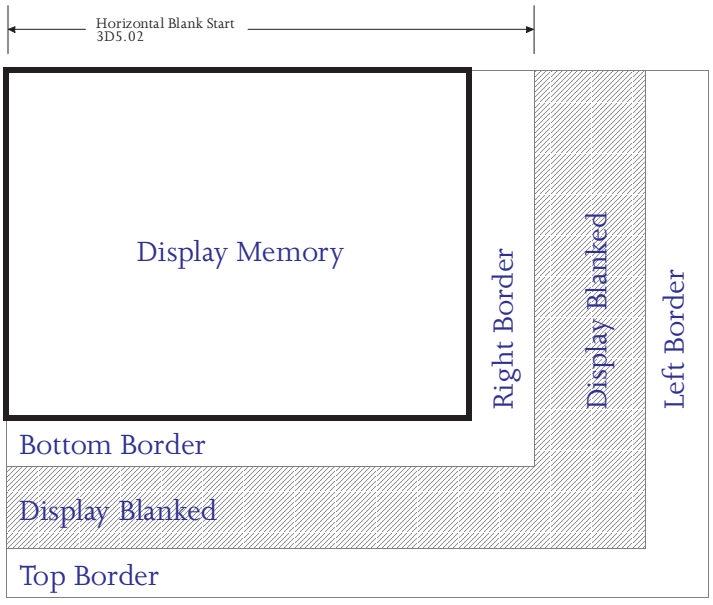
Any write to this register resets bit [0] of "vWindow group 0 control," on page 205, and "vWindow group 1 control," on page 211.

Read/write:	r/w	*Address:*	3D5h
Default:	Undefined.	*Address index:*	02h

7	6	5	4	3	2	1	0
horizontal blank start							

Bits	Description
[7:0]	Horizontal blank start [7:0] of [8:0], less 1. Horizontal blank start [8] is stored in 3D5.1B[2]. See "Horizontal overflow," on page 182.

Figure 15.5.4 Horizontal blank start



15.5.5. Horizontal blank end

Use this register to determine display enable skew and to specify the width of the horizontal blanking period, in character clocks.

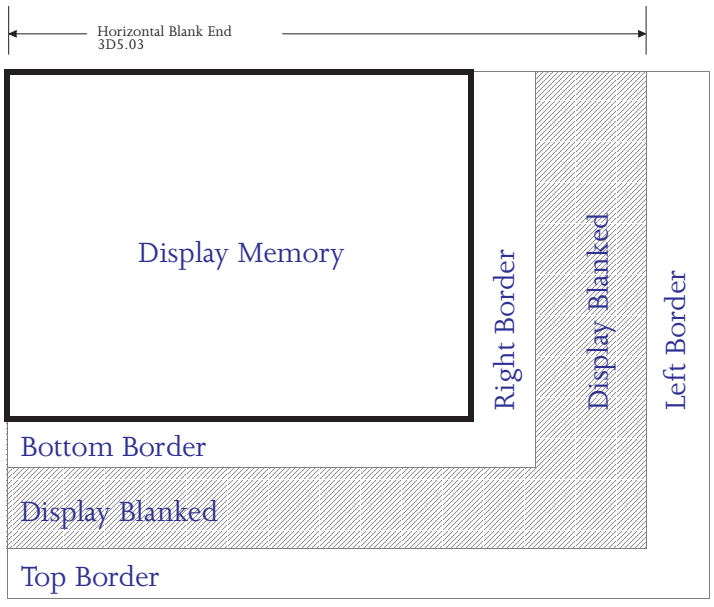
Any write to this register resets bit [0] of "vWindow group 0 control," on page 205, and "vWindow group 1 control," on page 211.

Read/write:	r/w	*Address:*	3D5h
Default:	Undefined.	*Address index:*	03h

7	6	5	4	3	2	1	0
	horizontal display enable skew		horizontal blank end				

Bits	Description
[4:0]	Horizontal blank end [4:0] of [5:0], less 1. Horizontal blank end [5] is stored in 3D5.5[7]. See "Horizontal retrace end," on page 144. ☞ Standard VGA and ProMotion use only the low order bits [5:0].
[6:5]	Horizontal display enable skew. This field delays the horizontal retrace (HSYNC) signal with respect to other signals by 0 to 3 character clocks (0, 8, 16, 0r 24 pixel clocks).

Figure 15.5.5 Horizontal blank end



15.5.6. Horizontal retrace start

Use this register to specify the character clock at which HSYNC goes active.

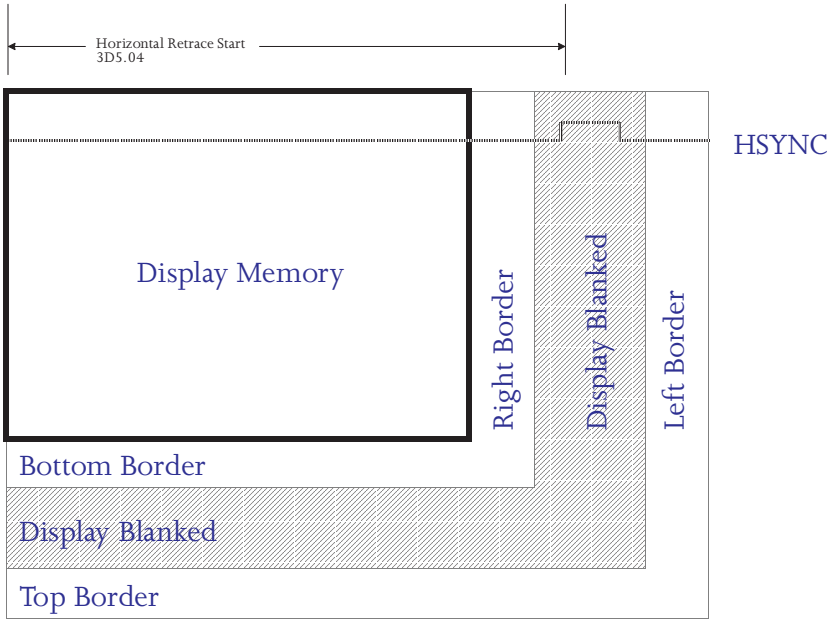
Any write to this register resets bit [0] of "vWindow group 0 control," on page 205, and "vWindow group 1 control," on page 211.

Read/write:	r/w	*Address:*	3D5h
Default:	Undefined.	*Address index:*	04h

7	6	5	4	3	2	1	0
horizontal retrace start							

Bits	Description
[7:0]	Horizontal retrace start [7:0] of [8:0], less 1. Horizontal retrace start [8] is stored in 3D5.1B[3]. See "Horizontal overflow," on page 182.

Figure 15.5.6 Horizontal retrace start



15.5.7. Horizontal retrace end

Use this register to specify the end of the HSYNC pulse, in character clocks.

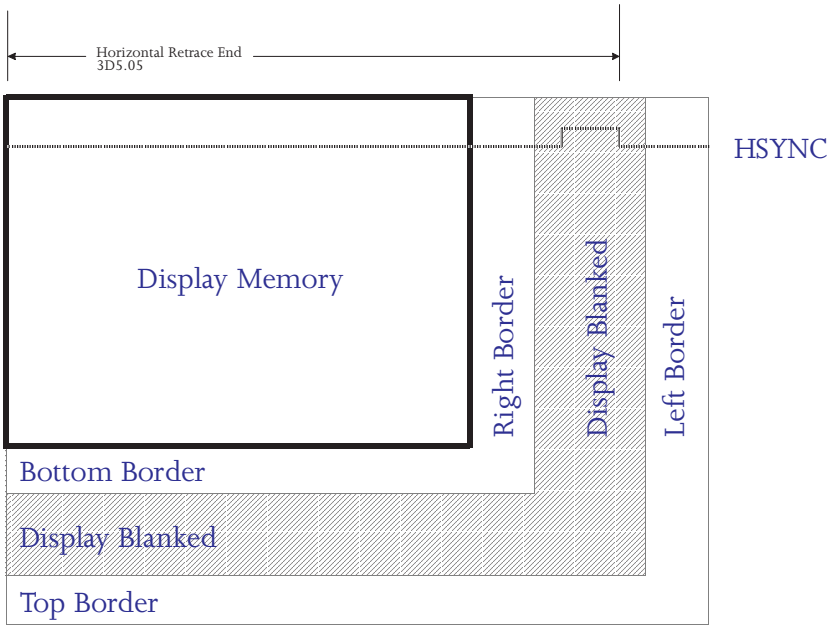
Any write to this register resets bit [0] of "vWindow group 0 control," on page 205, and "vWindow group 1 control," on page 211.

Read/write:	r/w	*Address:*	3D5h
Default:	Undefined.	*Address index:*	05h

7	6	5	4	3	2	1	0
horiz. blank end	horizontal retrace skew		horizontal retrace end				

Bits	Description
[4:0]	Horizontal retrace end. ☞ Standard VGA and ProMotion use only the low order bits [5:0].
[6:5]	Horizontal retrace skew, in character clocks. This bit delays the retrace pulse.
[7]	Horizontal blank end [5] of [5:0]. See "Horizontal blank end," on page 142.

Figure 15.5.7 Horizontal retrace end



15.5.8. Vertical total

Use this register to specify the number of scan lines per screen frame. This includes visible and non-visble lines.

Any write to this register resets bit [0] of "vWindow group 0 control," on page 205, and "vWindow group 1 control," on page 211.

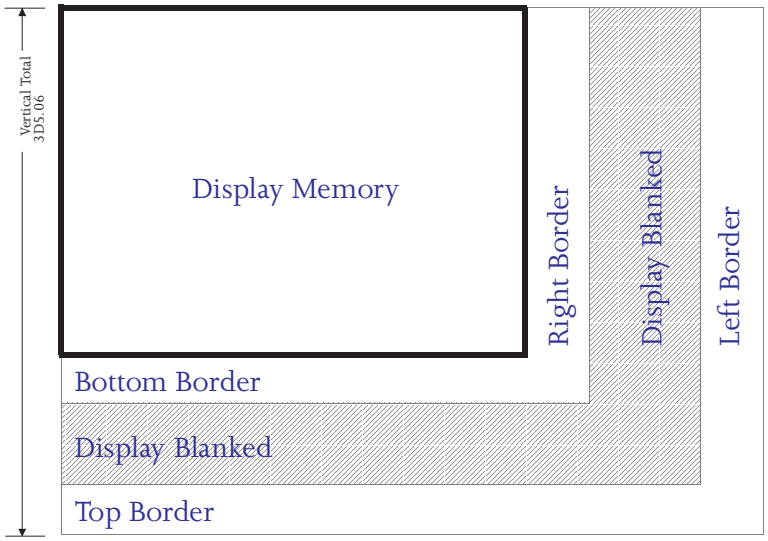
Note that this value is “less 2.”

Read/write:	r/w	*Address:*	3D5h
Default:	Undefined.	*Address index:*	06h

7	6	5	4	3	2	1	0
vertical total							

Bits	Description
[7:0]	Vertical total [7:0] of [10:0], less 2. Vertical total [8] is stored in 3D5.7[0], "Vertical overflow," on page 146. Vertical total [9] is stored in 3D5.7[5] "Vertical overflow," on page 146. Vertical total [10] is stored in 3D5.1A[0] "Vertical extended overflow," on page 181.

Figure 15.5.8 Vertical total



15.5.9. Vertical overflow

This register contains extended bits for other registers.

Any write to this register resets bit [0] of "vWindow group 0 control," on page 205, and "vWindow group 1 control," on page 211.

Read/write:	r/w	*Address:*	3D5h
Default:	Undefined.	*Address index:*	07h

7	6	5	4	3	2	1	0
vertical retrace start	vertical display enable end	vertical total	line compare	vertical blank start	vertical retrace start	vertical display enable end	vertical total

Bits	Description
[0]	Vertical total [8] of [10:0]. See "Vertical total," on page 145.
[1]	Vertical display enable end [9] of [10:0]. See "Vertical display enable end," on page 154.
[2]	Vertical retrace start [8] of [10:0]. See "Vertical retrace start," on page 152.
[3]	Vertical blank start [8] of [10:0]. See "Vertical blank start," on page 157.
[4]	Line compare [8] of [10:0]. See "Line compare," on page 160.
[5]	Vertical total [9] of [10:0]. See "Vertical total," on page 145.
[6]	Vertical display enable end [8] of [10:0]. See "Vertical display enable end," on page 154.
[7]	Vertical retrace start [9] of [10:0]. See "Vertical retrace start," on page 152.

15.5.10. Preset row scan

This register is relevant only in text modes.

Any write to this register resets bit [0] of "vWindow group 0 control," on page 205, and "vWindow group 1 control," on page 211.

Read/write:	r/w	*Address:*	3D5h
Default:	Undefined.	*Address index:*	08h

7	6	5	4	3	2	1	0
	byte panning		preset row scan				

Bits	Description
[4:0]	Preset row scan. Selects the scan line in the first character row corresponding to the top of the visible screen. This is useful for smooth scrolling, where a partial character row must be displayed at the top and bottom of the screen.
[6:5]	Byte panning, in bytes. Pans 0–3 characters in text modes (not intended for graphics modes).

Figure 15.5.10 Preset row scan

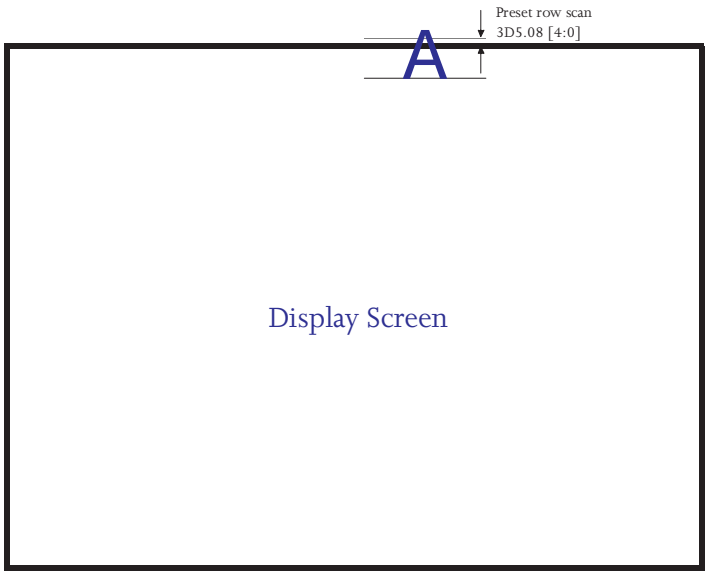


Diagram elements not to scale.

15.5.11. Maximum scan line

Use this register to specify the number of scan lines in a character row. This register also contains overflow bits.

Any write to this register resets bit [0] of "vWindow group 0 control," on page 205, and "vWindow group 1 control," on page 211.

Read/write:	r/w	*Address:*	3D5h
Default:	Undefined.	*Address index:*	09h

7	6	5	4	3	2	1	0
double scan	line compare	vertical blank start	maximum scan lines				

Bits	Description
[4:0]	Maximum scan lines, less one.
[5]	Vertical blank start [9] of [10:0]. See "Vertical blank start," on page 157.
[6]	Line compare [9] of [10:0]. See "Line compare," on page 160.
[7]	Double scan. 1 = enabled. 0 = disabled.

Figure 15.5.11 Maximum scan line

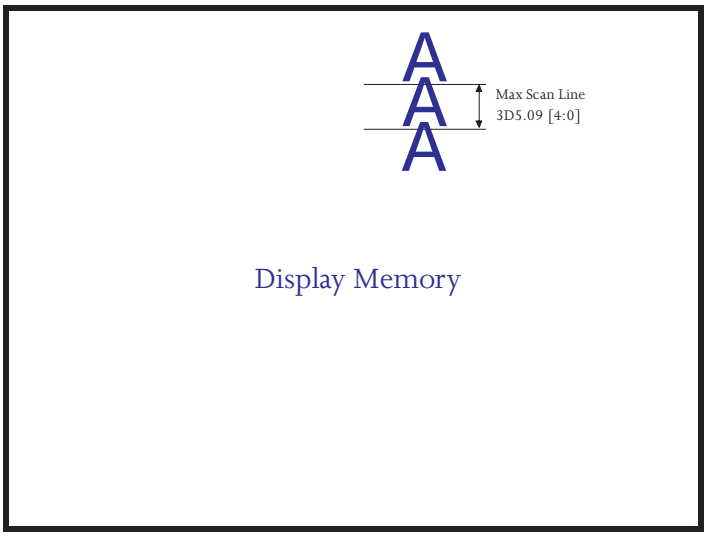


Diagram elements not to scale.

15.5.12. Block cursor start

Use this register to specify the scan line within a character for the top edge of the block cursor, or to disable the cursor.

Any write to this register resets bit [0] of "vWindow group 0 control," on page 205, and "vWindow group 1 control," on page 211.

Read/write:	r/w	*Address:*	3D5h
Default:	Undefined.	*Address index:*	0Ah

7	6	5	4	3	2	1	0
		b cursor off	block cursor start				

Bits	Description
[4:0]	Block cursor start.
[5]	Block cursor off. 1 = cursor disabled. 0 = cursor enabled.

Figure 15.5.12 Block cursor start

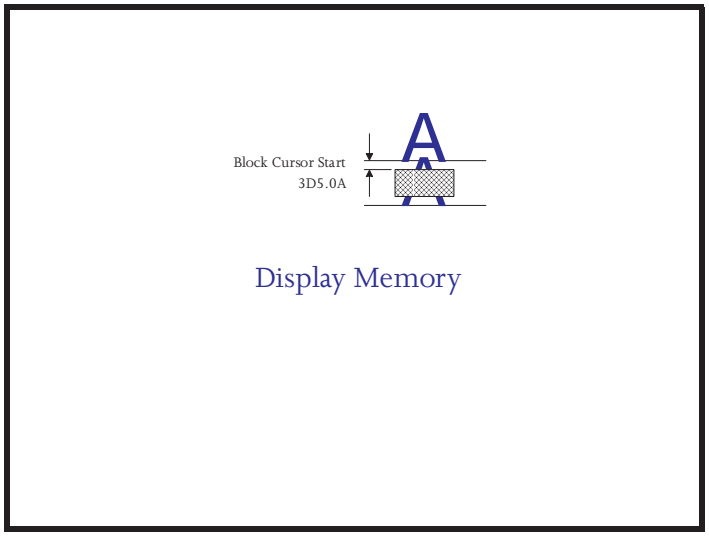


Diagram elements not to scale.

15.5.13. Block cursor end

Use this register to specify the scan line within a character for the lower edge of the block cursor, and the horizontal offset of the cursor from the current character position. These bits are relevant only in text modes.

Any write to this register resets bit [0] of "vWindow group 0 control," on page 205, and "vWindow group 1 control," on page 211.

Read/write:	r/w	*Address:*	3D5h
Default:	Undefined.	*Address index:*	0Bh

7	6	5	4	3	2	1	0
	block cursor skew		block cursor end				

Bits	Description
[4:0]	Block cursor end, less 1.
[6:5]	Block cursor skew.

Figure 15.5.13 Block cursor end

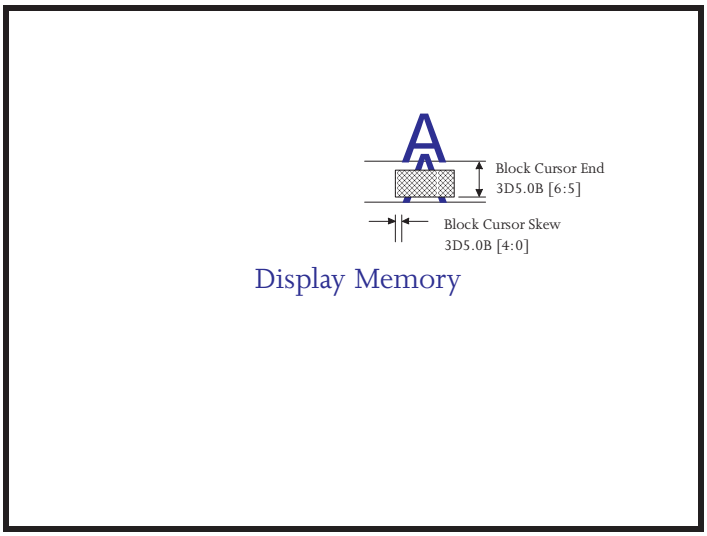


Diagram elements not to scale.

15.5.14. Serial start address

Use this register to specify the location in memory at which data representing the first pixel to be displayed begins. This register is usually 0.

This may be used to create split screens in conjunction with 3D5.18, "Line compare," on page 160. Serial start address is always used for the first line of the display even if Line compare is 0.

Serial start address should always be set to a dword boundary.

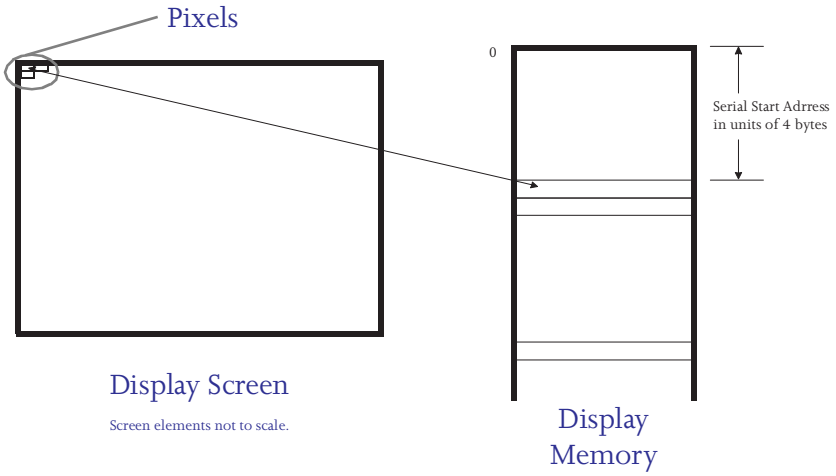
Any write to this register resets bit [0] of "vWindow group 0 control," on page 205, and "vWindow group 1 control," on page 211.

Read/write:	r/w	*Address:*	3D5h
Default:	Undefined.	*Address index:*	0C–0Dh

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
serial start address															

Bits	Description
[15:0]	Serial start address [15:0] of [19:0] in doublewords (4 bytes). Serial start address [19:16] are stored in 3D5.1C[3:0], "Serial overflow," on page 182.

Figure 15.5.14 Serial start address



15.5.15. Block cursor location

Use this register to specify the location in display memory of the character over which the block cursor is positioned. This register is relevant only in text modes; one specifies a graphics cursor in terms of screen coordinates.

Any write to this register resets bit [0] of "vWindow group 0 control," on page 205, and "vWindow group 1 control," on page 211.

Read/write:	r/w	*Address:*	3D5h
Default:	Undefined.	*Address index:*	0E–0Fh

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
block cursor location															

Bits	Description
[15:0]	Block cursor location, in character attribute pairs, where each character is comprised of two bytes: character byte and attribute byte.

15.5.16. Vertical retrace start

Use this register to specify the scan line at which the VSYNC becomes active.

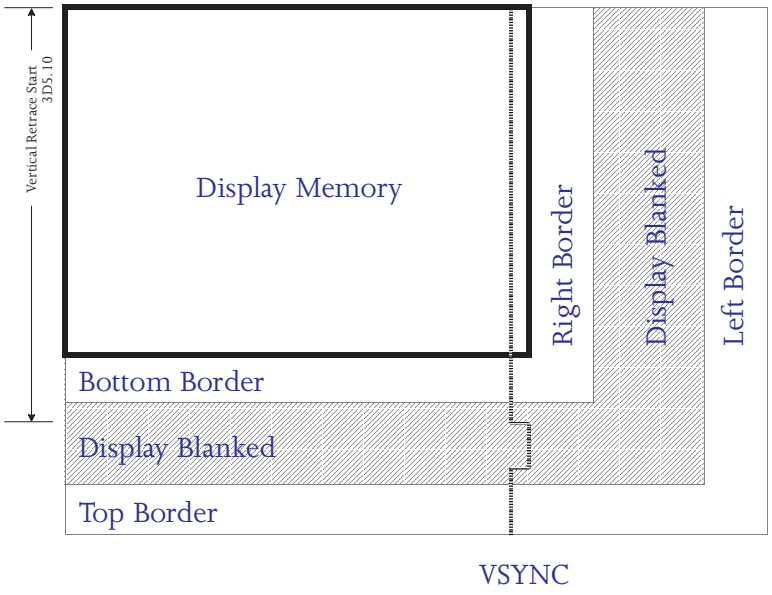
Any write to this register resets bit [0] of "vWindow group 0 control," on page 205, and "vWindow group 1 control," on page 211.

Read/write:	r/w	*Address:*	3D5h
Default:	Undefined.	*Address index:*	10h

7	6	5	4	3	2	1	0
vertical retrace start							

Bits	Description
[7:0]	Vertical retrace start [7:0] of [10:0], less 1. Vertical retrace start [8] is stored in 3D5.7[2], "Vertical overflow," on page 146. Vertical retrace start [9] is stored in 3D5.7[7], "Vertical overflow," on page 146. Vertical retrace start [10] is stored in 3D5.1A[3], "Vertical extended overflow," on page 181.

Figure 15.5.16 Vertical retrace start



15.5.17. Vertical retrace end

Use this register to determine the scan line at which VSYNC becomes inactive, and to specify other CRTC parameters.

Any write to this register resets bit [0] of "vWindow group 0 control," on page 205, and "vWindow group 1 control," on page 211.

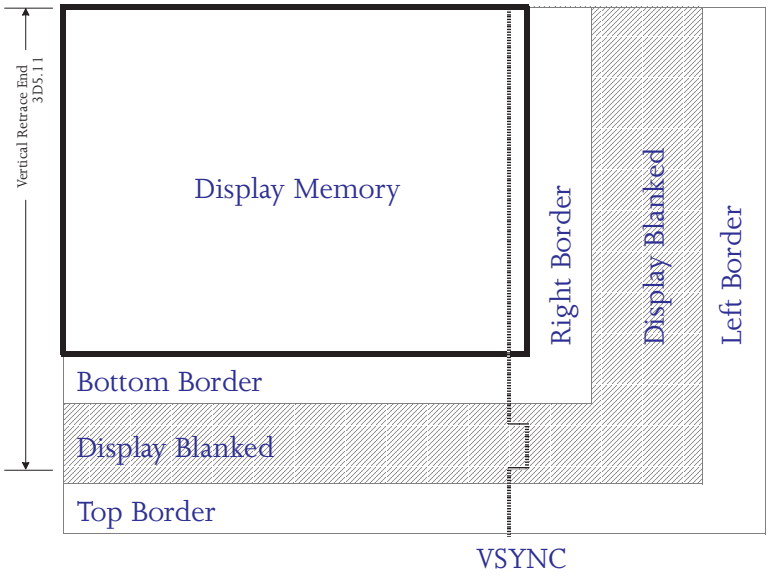
Read/write:	r/w	*Address:*	3D5h
Default:	0h	*Address index:*	11h

7	6	5	4	3	2	1	0
lock CRTC		disable v interrupt	clear v interrupt	vertical retrace end			

Bits	Description
[3:0]	Vertical retrace end, less 1. ☞ Standard VGA and ProMotion use only the low order bits [3:0].
[4]	Clear vertical interrupt. 1 = normal vertical retrace. 0 = clear vertical retrace interrupt.

Bits	Description
[5]	Disable vertical interrupt. 1 = disable interrupt when vertical retrace begins. 0 = normal retrace.
[6]	Reserved.
[7]	Lock other CRTC registers. 1 = locked. 0 = unlocked. This setting locks 3D5.0–3D5.7 except for 3D5.7[4], "Vertical overflow," on page 146. This bit is overridden by 0C8–0C9, "VGA override," on page 226.

Figure 15.5.17 Vertical retrace end



15.5.18. Vertical display enable end

Use this register to specify the scan line at which the display ends.

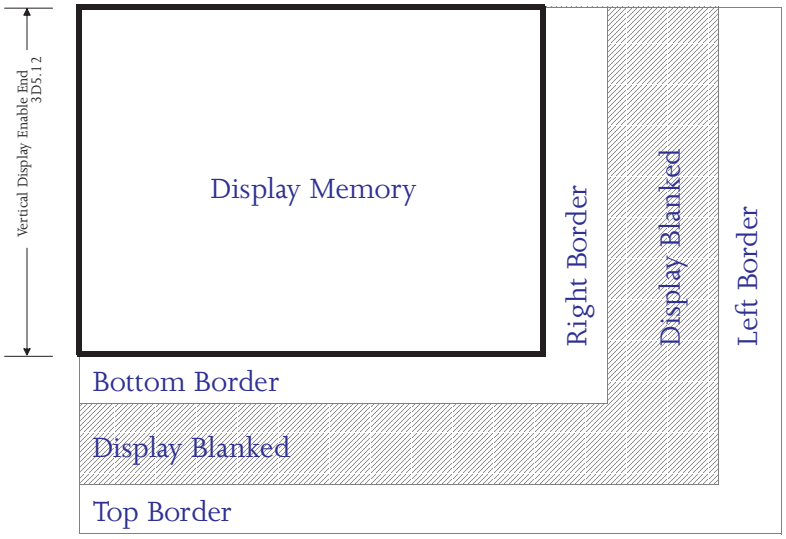
Any write to this register resets bit [0] of "vWindow group 0 control," on page 205, and "vWindow group 1 control," on page 211.

Read/write:	r/w	*Address:*	3D5h
Default:	Undefined.	*Address index:*	12h

7	6	5	4	3	2	1	0
vertical display enable end							

Bits	Description
[7:0]	Vertical display enable end [7:0] of [10:0], less 1. Vertical display enable end [8] is stored in 3D5.7[6], "Vertical overflow," on page 146. Vertical display enable end [9] is stored in 3D5.7[1], "Vertical overflow," on page 146. Vertical display enable end [10] is stored in 3D5.1A[1], "Vertical extended overflow," on page 181.

Figure 15.5.18 Vertical display enable end



15.5.19. Serial offset

Use this register to specify the offset in display memory from pixel row n to pixel row n+1. This is normally set to the width of the display, but may be set to a larger value.

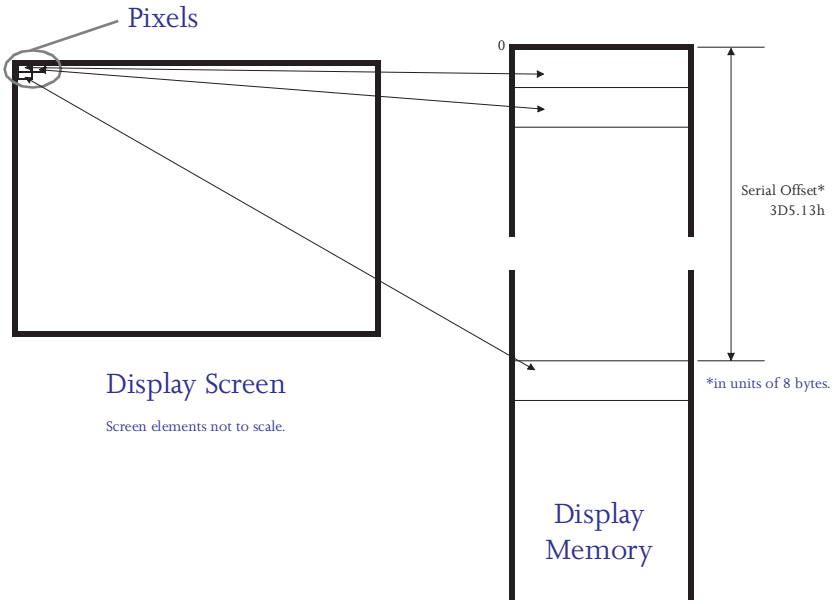
Any write to this register resets bit [0] of "vWindow group 0 control," on page 205, and "vWindow group 1 control," on page 211.

Read/write:	r/w	*Address:*	3D5h
Default:	Undefined.	*Address index:*	13h

7	6	5	4	3	2	1	0
serial offset							

Bits	Description
[7:0]	Serial offset [7:0] of [11:0]. Serial offset [11:8] are stored in 3D5.1C[7:4], "Serial overflow," on page 182.

Figure 15.5.19 Serial offset



15.5.20. Underline location/dword mode

Use bit [6] of this register in conjunction with 3D5.17[6], "CRTC mode control register," on page 159, to control byte or doubleword addressing. Use bits [4:0] of this register to specify the scan line within a character where the underline appears.

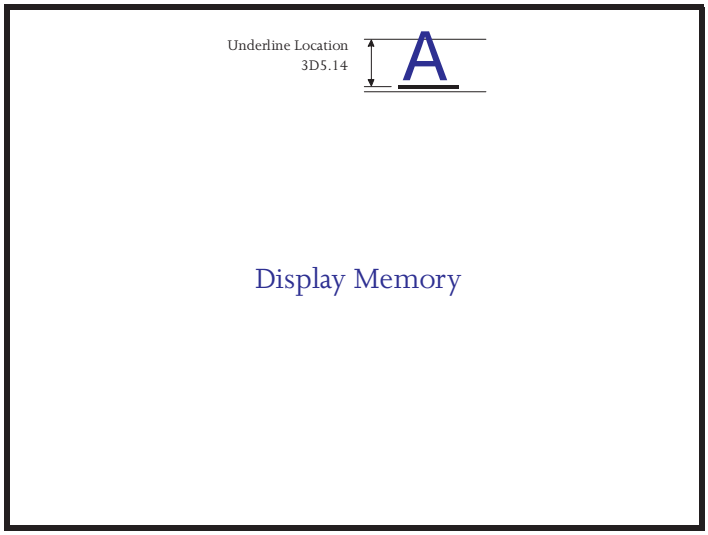
Any write to this register resets bit [0] of "vWindow group 0 control," on page 205, and "vWindow group 1 control," on page 211.

Read/write:	r/w	*Address:*	3D5h
Default:	Undefined.	*Address index:*	14h

7	6	5	4	3	2	1	0
	doubleword mode		underline location				

Bits	Description
[4:0]	Underline location.
[5]	Reserved.
[6]	Doubleword mode. 1 = CRTC display memory addresses incremented by 4. 0 = CRTC display memory addresses incremented by 1 or 2, as set by 3D5.17[6], "CRTC mode control register," on page 159.

Figure 15.5.20 Underline location



15.5.21. Vertical blank start

Use this register to specify the scan line at which blank begins.

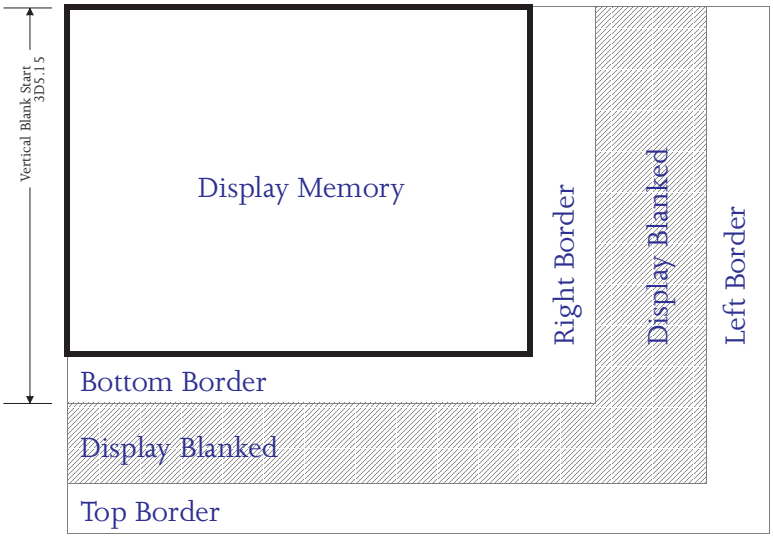
Any write to this register resets bit [0] of "vWindow group 0 control," on page 205, and "vWindow group 1 control," on page 211.

Read/write:	r/w	*Address:*	3D5h
Default:	Undefined.	*Address index:*	15h

7	6	5	4	3	2	1	0
vertical blank start							

Bits	Description
[7:0]	Vertical blank start [7:0] of [10:0], less 1. Vertical blank start [8] is stored in 3D5.7[3], "Vertical overflow," on page 146. Vertical blank start [9] is stored in 3D5.9[5], "Maximum scan line," on page 147. Vertical blank start [10] is stored in 3D5.1A[2], "Vertical extended overflow," on page 181.

Figure 15.5.21 Vertical blank start



15.5.22. Vertical blank end

Use this register to specify the scan line at which blank ends.

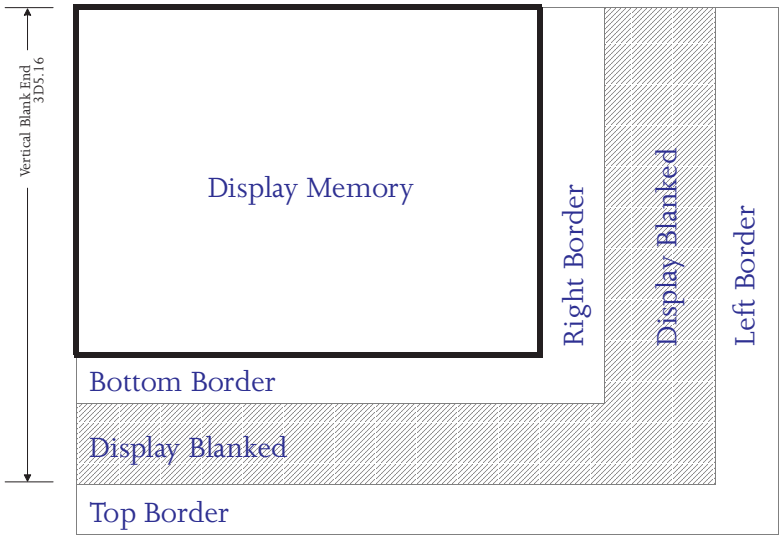
Any write to this register resets bit [0] of "vWindow group 0 control," on page 205, and "vWindow group 1 control," on page 211.

Read/write:	r/w	*Address:*	3D5h
Default:	Undefined.	*Address index:*	16h

7	6	5	4	3	2	1	0
vertical blank end							

Bits	Description
[7:0]	Vertical blank end, less 1.
	☞ Standard VGA and ProMotion use only the low order bits [7:0].

Figure 15.5.22 Vertical blank end



15.5.23. CRTC mode control register

Any write to this register resets bit [0] of "vWindow group 0 control," on page 205, and "vWindow group 1 control," on page 211.

Read/write:	r/w	*Address:*	3D5h
Default:	0h	*Address index:*	17h

7	6	5	4	3	2	1	0
sync reset	byte mode	address wrap			horizontal retrace start		CGA compatibility

Bits	Description
[0]	Compatibility mode. 1 = CGA compatibility disabled. 0 = CGA compatibility mode.
[1]	Reserved.
[2]	Horizontal retrace select. 1 = scan line clocked as HSYNC/2. 0 = scan line clocked as $\overline{\text{HSYNC}}$. This operation is intended for VGA very high resolution modes.
[4:3]	Reserved.
[5]	Address wrap. Ignored if 3C5.17[6] is set to 1. 1 = address wraps around bit 16 of the CRTC address counter. 0 = address wraps around bit 14 of the CRTC address counter.
[6]	Byte mode. 1 = CRTC display memory addresses incremented by 2. 0 = CRTC display memory addresses incremented by 1. This register is relevant only if 3D5.14[6] = 0. Refer to "Underline location/dword mode," on page 156.
[7]	Sync reset. 1 = $\overline{\text{HSYNC}}$ and $\overline{\text{VSYNC}}$ disabled. 0 = $\overline{\text{HSYNC}}$ and $\overline{\text{VSYNC}}$ enabled.

15.5.24. Line compare

Use this register to implement a VGA split screen. This register specifies the scan line at which the split screen occurs.

Read/write:	r/w	*Address:*	3D5h
Default:	Undefined.	*Address index:*	18h

7	6	5	4	3	2	1	0
line compare							

Bits	Description
[7:0]	Line compare [7:0] of [10:0], less 1. Line compare [8] is stored in 3D5.7[4], "Vertical overflow," on page 146. Line compare [9] is stored in 3D5.9[6], "Maximum scan line," on page 147. Line compare [10] is stored in 3D5.1A[4], "Vertical extended overflow," on page 181.

15.5.25. Readback latch data

Use this register to read the graphics controller data latch selected with 3CF.04[1:0], "Read map select," on page 134.

Read/write:	r	*Address:*	3D5h
Default:	Undefined.	*Address index:*	22h

7	6	5	4	3	2	1	0
readback latch data							

Bits	Description
[7:0]	Readback latch data.

15.5.26. Attribute index data

This read-only register returns attribute controller information.

Read/write:	r	*Address:*	3D5h
Default:	Undefined.	*Address index:*	24h, 26h

7	6	5	4	3	2	1	0
index/data		enable palette	attribute index				

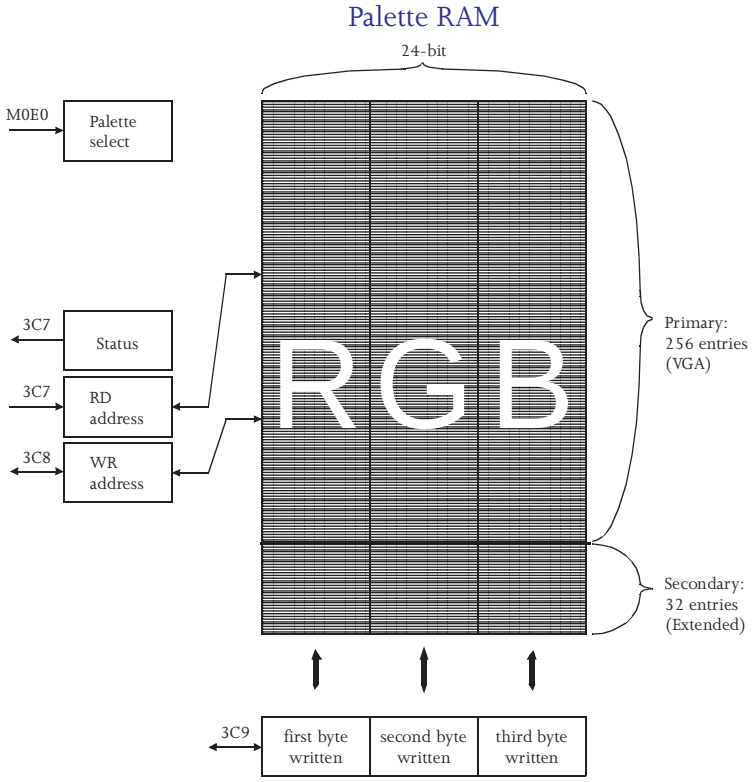
Bits	Description
[4:0]	Attribute index. This is the same as 3C0.4, "Index," on page 119.
[5]	Enable palette. 1 = enabled. 0 = palette disabled.
[7]	Index/data flip-flop. 1 = data. 0 = index.

15.6 VGA palette RAM registers

VGA Port Address	Register Name
3C6	Palette RAM pel mask.
3C7 (write)	Palette RAM read address.
3C7 (read)	Palette RAM state
3C8	Palette RAM write address.
3C9	Palette RAM port.

Three consecutive byte writes to 3C9 are required in order to load one entry in the palette RAM. Bytes are written in the following order: red, green, blue. Access to another palette RAM port before all three writes have taken place interrupts the sequence and causes the previous writes to be lost. Successful completion of the sequence of three writes increments the index. Register bit M0E0[5], of "Color correction," on page 253, selects whether the read/write goes to the primary or secondary palette RAM.

Figure 15.6.1 Palette RAM registers



15.6.2. Palette RAM pel mask

A write to this address specifies a mask which is ANDed with all pixel addresses to be translated by the palette RAM.

Read/write:	r/w	*Address:*	3C6h
Default:	Undefined.	*Address index:*	-

7	6	5	4	3	2	1	0
palette RAM pixel mask							

Bits	Description
[7:0]	Palette RAM pixel mask.

15.6.3. Palette RAM state/read address

Write to 3C7 to set the palette RAM read pointer. Read from 3C7 to obtain the palette RAM read/write status.

Note that this port does not return the value written to it.

Read/write:	r/w	*Address:*	3C7h
Default:	Undefined.	*Address index:*	-

7	6	5	4	3	2	1	0
palette RAM read address (WRITE)							

Bits	Description (write)
[7:0]	Palette RAM read address.

7	6	5	4	3	2	1	0
						palette RAM state (READ)	

Bits	Description (read)
[1:0]	Palette RAM state. 11 = 3C7 was written after 3C8 (last palette access was most likely a read). 00 = 3C8 was written after 3C7 (last palette access was most likely a write).

15.6.4. Palette RAM write address

Use this register to set the palette RAM write pointer.

Read/write:	r/w	*Address:*	3C8h
Default:	Undefined.	*Address index:*	-

7	6	5	4	3	2	1	0
palette RAM write address							

Bits	Description
[7:0]	Palette RAM write address.

15.6.5. Palette RAM data

Use this port to load the palette RAM, three bytes at a time.

Three writes to this port causes the three bytes written to be loaded into palette RAM at the address specified by 3C8. Values are not transferred to the lookup table until after the third (blue) value is written here. The palette RAM write address at 3C8 is incremented after the third write occurs.

Three reads from this register return the red, green, and blue values from the palette RAM location specified in the palette RAM read address. The palette RAM read address at 3C7 is incremented after the third write occurs, but it cannot be read.

Note that writes to 3C8h or 3C7h reset the next read/write pointer to red.

Read/write:	r/w	*Address:*	3C9h
Default:	Undefined.	*Address index:*	-

7	6	5	4	3	2	1	0
palette RAM data							

Bits	Description
[7:0]	Palette RAM data.

15.6.6. Primary palette registers 0–255

Use these registers to define colors to be displayed for 8-bit and 4-bit pixels on the desktop. The 9-bit index consists of Host palette select, bit [5] of M0E0, "Color correction," on page 253, followed by the 8-bit palette RAM read/write address 3C7/3C8.

Note that three consecutive 8-bit values (in order red/green/blue) are wrtiten to each 24-bit palette RAM location.

Read/write:	r/w	*Address:*	3C9h
Default:	Undefined.	*Address index:*	000–0FFh

23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
red								green								blue							

Bits	Description
[7:0]	Blue value.
[15:8]	Green value.
[23:16]	Red value.

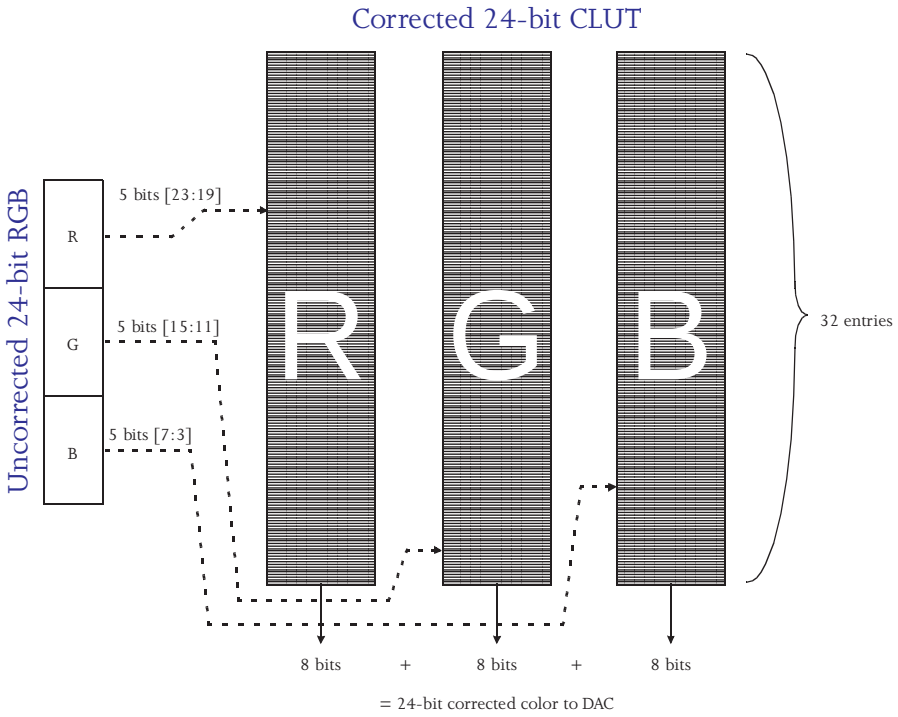
15.6.7. Secondary palette registers 0–31

Although the secondary palette is not standard VGA, it is included with VGA registers for convenience.

Use these registers to load the secondary palette into RAM. The secondary palette is used typically for gamma correction for the vWindow. The 9-bit index consists of Host palette select, bit [5] of M0E0, "Color correction," on page 253, followed by the 8-bit palette RAM read/write address 3C7/3C8.

The five high-order bits of each color of each RGB pixel determine corrected color. These high-order bits function as the index (0-31) into three 8-bit-wide lookup tables. The low-order bits of the uncorrected color are ignored.

Figure 15.6.7 Secondary palette color correction



Read/write:	r/w	*Address:*	3C9h
Default:	Undefined.	*Address index:*	100–11Fh

23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
red								green								blue							

Bits	*Description*
[7:0]	Blue value.
[15:8]	Green value.
[23:16]	Red value.

16. Extended register descriptions

To prevent unexpected operation, all reserved register bits should be written with 0s and masked off if read back. Future compatability is jeopardized if this procedure is not followed.

16.1 Extended setup registers

Writes to ProMotion registers M000–17Fh pass through command FIFO. Writes to ProMotion registers M180–1FFh do not pass through command FIFO.

16.1.1. Unlock extended registers

Use this simulated register to unlock ProMotion extended (non-VGA) I/O registers. ProMotion memory mapped registers are locked using 3C5.1B, "Remap control," on page 168.

Read/write:	w	*Address:*	3C5h
Default:	0h	*Address index:*	10h

7	6	5	4	3	2	1	0
unlock extended registers							

Bits	Description
[7:0]	Writing 12h, (1010b) unlocks the ProMotion I/O registers. Writing any other value locks the registers. 00001010 = ProMotion I/O registers unlocked.

16.1.2. Chip ID

Driver developers should use M188, "PCI revision ID," on page 246, for chip ID and revision level, or use BIOS calls. Refer to "ProMotion stepping information," on page 316 for more information on chip revision identification.

Read/write:	r	*Address:*	3C5h
Default:	-	*Address index:*	11–19h

71:64	63:56	55:48	47:40	39:32	31:24	23:16	15:8	7:0
ASCII string							fab code	revision ID

Bits	Description
[71:16]	ASCII string "PRO643D"
[15:8]	Fab code (implementation dependent).
[7:0]	Revision ID (implementation dependent).

16.1.3. Flat model base address

Use this register to specify the location of display memory within host memory space. This register is used in conjunction with 3C5.1C[0] "Flat model control," on page 169.

Because this register changes the access aperture to display memory and/or comand FIFO, Alliance recommends drivers perform a dummy read to a location independent of the aperture immediately after writing this register to ensure the command FIFO is flushed before the next command.

Read/write:	r/w	*Address:*	3C5h
Default:	0h	*Address index:*	1Ah

7	6	5	4	3	2	1	0
flat model base address							

Bits	Description
[7:0]	Flat model base address. This value is specified in megabytes, and must be aligned to the aperture. For instance, if the aperture is 2MB, the flat model base address may be set to 2, 4, 6, etc.

16.1.4. Remap control

Use this register to map ProMotion extended registers and ProMotion host BLT port. Refer to "Host BLT read/write," on page 88.

Because this register changes the access aperture to display memory and/or comand FIFO, Alliance recommends drivers perform a dummy read to a location independent of the aperture immediately after writing this register to ensure the command FIFO is flushed before the next command.

The remap ProMotion registers field [2:0] controls mapping of the ProMotion extended registers. When set to 000h, the ProMotion extended registers are mapped out (i.e.: locked) and cannot be accessed, but any values previously loaded in those registers remain valid and control their respective functions.

The remap host BLT port field [5:3] maps the ProMotion host BLT port into the host memory space. It works in a manner analogous to the remap ProMotion registers field.

Though the two fields may be set independently, it should be noted that the VGA display memory must be mapped into a region different from both ProMotion extended registers and the ProMotion host BLT port, using 3CE–F.6, therefore it is usually advisable to set the two fields in tandem.

Read/write:	r/w	*Address:*	3C5h
Default:	0h	*Address index:*	1Bh

7	6	5	4	3	2	1	0
		remap host BLT port			remap ProMotion registers		

Bits	Description
[5:3]	Remap host BLT port. 100 = Last 32K of flat space less final 2K 011 = B900:0 - BFFF:F. 010 = B100:0 - B7FF:F. 001 = A100:0 - A7FF:F. 000 = mapped out.
[2:0]	Remap ProMotion registers. 100 = Last 2K of flat space. 011 = B800:0 - B87F:F. 010 = B000:0 - B07F:F. 001 = A000:0 - A07F:F. 000 = mapped out. When set to 001b, 010b, or 011b, the ProMotion extended registers are mapped into the first 256 bytes at A000:0, B000:0, or B800:0 respectively. For example, the register at ProMotion offset 030 may be mapped into A000:30. In any of these three cases, the software **MUST** assure that the standard VGA display memory aperture (specified in the VGA Graphics Controller at port 3CE–F index 6) is set to a different aperture than the ProMotion registers. When set to 100b, the ProMotion extended registers are mapped into the last 2K of the flat model address space. In order to use this setting, the flat model address space must be defined, although the actual display memory does not have to be mapped into it. If the Flat Model Aperture is set to the actual display memory size, the ProMotion extended registers overlap the last 2K of display memory, preventing its access for host read or write. Alternatively, the driver may set the aperture to a size larger than the actual display memory, effectively overlapping the extended register space with an empty region.

16.1.5. Flat model control

Use this register to control flat model access, VGA memory, and I/O access.

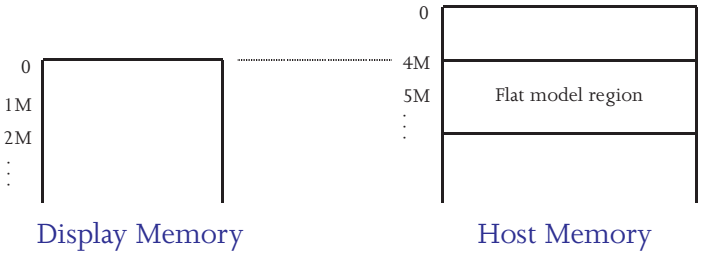
Because this register changes the access aperture to display memory and/or comand FIFO, Alliance recommends drivers perform a dummy read to a location independent of the aperture immediately after writing this register to ensure the command FIFO is flushed before the next command.

Read/write:	r/w	*Address:*	3C5h
Default:	0h	*Address index:*	1Ch

7	6	5	4	3	2	1	0
		simultaneous access	VGA aperture addressing	disable VGA memory access	flat model aperture		flat model access

Bits	Description
[0]	Flat model access. 1 = enabled 0 = disabled. This bit enables access through the flat model access space. The flat model aperture field specifies the size of the flat model region. It should generally be the same size as the display memory, though it may be larger if the ProMotion extended registers are also mapped into the flat model space. The bottom of the flat model region always corresponds to the bottom of display memory, as shown in the following diagram.
[2:1]	Flat model aperture. 11 = 8MB. 10 = 4MB. 01 = 2MB. 00 = 1MB.
[3]	VGA memory access. 1 = disabled. 0 = enabled. If this bit is set, access to display memory through the VGA address spaces (A000:0–BFFF:F) is disabled.
[4]	VGA aperture addressing. 1 = linear addressing within VGA aperture. 0 = addresses within the VGA aperture may undergo modifications related to the VGA chain mode, as set by VGA register "Memory Mode" 3C5.04[3:2].
[5]	Simultaneous linear/drawing engine access. 1 = enabled. 0 = disabled. If this bit is set, linear accesses to display memory can take place while the drawing engine is active. Otherwise, all host access to display memory is queued until the drawing engine completes. This bit should be set only if the software can guarantee that host accesses take place exclusively to display memory locations that are not read or updated by the current or queued drawing engine commands.

Figure 16.1.5 3C5.1C[0] Flat model access



16.1.6. Alternate access space pointer/decode registers

Any memory mapped ProMotion register may be accessed alternately through I/O space using the alternate access space pointer and decode registers. The pointer register 3C5.1D stores PMPOINTER, the memory offset of the memory mapped register to be accessed, shifted right by 2 bits. The decode register 3C5.1E stores PMDECODE, a 16-bit I/O byte address which must be a multiple of 4 bytes.

ProMotion decodes any of four I/O addresses defined by PMDECODE, PMDECODE+1, PMDECODE+2, and PMDECODE+3.

- an 8-bit read or write to the I/O port PMDECODE + n accesses the memory mapped byte register at (PMPOINTER<<2)+n, for n = 0, 1, 2, 3.
- a 16-bit read or write to I/O port PMDECODE + n accesses the memory mapped word register at (PMPOINTER<<2)+n, for n = 0, 2.
- a 32-bit read or write to I/O port PMDECODE accesses the memory mapped dword register at (PMPOINTER<<2).

Normally ProMotion VGA BIOS sets PMDECODE at boot time. Since Plug and Play motherboard BIOS allocates this I/O address, PMDECODE should remain unchanged after initialization, while PMPOINTER is normally changed as required to access different registers.

Although ProMotion defines extended registers in terms of byte addresses, the Alternate access space register is in terms of dword addresses. Therefore to write the cursor control register at 0x140, load 0x140 shifted right by two bits, i.e.0x150, into PMPOINTER. Don't load the byte PMPOINTER into this field, instead load all but the two low-order bits of it.

16.1.6.1 Alternate access space pointer LOW

Read/write:	r/w	*Address:*	3C5h
Default:	Undefined.	*Address index:*	1Dh

7	6	5	4	3	2	1	0
PMPOINTER [9:2]							

Bits	Description
[7:0]	PMPOINTER LOW. Bits [9:2] of memory mapped register offset. High bits [17:10] of this field are contained in 3C5.28, "Alternate access space pointer HIGH," described on page 177

16.1.6.2 Alternate access space decode

Read/write:	r/w	*Address:*	3C5h
Default:	0h.	*Address index:*	1E–1Fh

7	6	5	4	3	2	1	0
PMDECODE						! must be 0	! must be 0

Bits	Description
[15:0]	PMDECODE. ! Low order bits [1:0] must be 00b.

The following sample code reads and writes a memory mapped register using alternate access space.

```
;---------------
; Procedure: ReadIOMappedReg
;
; Desc: Read an extended register by means of the IO-mapped facility.
;       size of the transfer is one byte.
; Input: register offset in units of bytes, in BX
; Output: byte value in AL
;---------------
    public ReadIOMappedReg
ReadIOMappedReg proc    near
     push  bx
     push  cx
     push  dx

     mov   dx, 3C4h             ;write aligned offset to register 3C5.1Dh
     mov   al, 1Dh
     out   dx, al
     inc   dx
     mov   ax, bx
     mov   cl, 2
     shr   ax, cl
     out   dx, al

     call  ReadIoMapPort        ;get base port address of register space

     and   bx, 03h              ;add the aligned offset
     add   dx, bx
     in    al, dx

     pop   dx
     pop   cx
     pop   bx
     ret
ReadIOMappedReg endp
```

```
;---------------
; Procedure: WriteIOMappedReg
;
; Desc: Write an extended register by means of the IO-Mapped facility.
;           size of the transfer is one byte.
; Input: register offset in units of bytes, in BX
;      byte value in AL
;---------------
    public WriteIOMappedReg
WriteIOMappedReg proc    near
      push  ax
      push  bx
      push  cx
      push  dx

      mov   ch, al              ;save AL in CH

      mov   dx, 3C4h            ;write aligned offset to index register
      mov   al, 1Dh
      out   dx, al
      inc   dx
      mov   ax, bx
      mov   cl, 2
      shr   ax, cl
      out   dx, al

      call  ReadIoMapPort       ;get the base port address of register space

      and   bx, 03h
      add   dx, bx
      mov   al, ch
      out   dx, al

      pop   dx
      pop   cx
      pop   bx
      pop   ax
      ret
WriteIOMappedReg endp

;---------------
; Procedure : ReadIoMapPort
; Desc: Read values of PMDECODE registers
; Input: Nothing
; Output: DH - 3C5.1F
;       DL - 3C5.1E
;---------------
    public  ReadIoMapPort
ReadIoMapPort  proc    near
      push  ax
      push  bx
      mov   dx, 3C4h            ;read 3C5.1Eh
      mov   al, 1Eh
      out   dx, al
      inc   dx
      in    al, dx

      mov   bl, al              ;save in BL

      dec   dx                  ;read 3C5.1Fh
      mov   al, 1Fh
      out   dx, al
      inc   dx
      in    al, dx

      mov   dh, al
      mov   dl, bl
```



```
      pop   bx
      pop   ax
      ret
ReadIoMapPort  endp
```

16.1.7. Scratchpad

Use this register for temporary storage. This storage area has no on-chip function.

ProMotion-3210 has **one** 32-bit scratchpad register in **memory space**.
ProMotion-6410/6422 have **four** 8-bit scratchpad registers in **I/O space**.
ProMotion-AT3D/AT24 have **eight** 8-bit scratchpad registers in **I/O space**.

Read/write:	r/w	*Address:*	3C5h
Default:	Undefined.	*Address index:*	20–27h

63:8	7:0
scratchpad	scratchpad, GPIO
	cnf 0–7

Bits	Description
[63:0]	Scratchpad register.
	Default 3C5.20[7:0] = configuration straps MD[7:0], general purpose I/O.

General purpose I/O may be implemented for stereo glasses, DDC, VMI video/audio/MPEG hardware, and other devices.

16.1.7.1 Sample source code to identify ProMotion controller for scratchpad register use

ProMotion driver software which depends on scratchpad registers and is intended for both the ProMotion-3210 as well as later versions should verify the chip using a routine such as the following sample code.

```
;---------------
; Procedure: ReadScratch
; Desc: Read scratch register
; In: bx has index of scratch reg., = 0,1,2, ... 7.
; Out: read value in ax.
;---------------
    public  ReadScratch
ReadScratch  proc    near
ifdef 3210                        ; true if code must support 3210
      call  DetectIomappedScratch
      jnc   RSMemmapped
endif
      add   bx, 20h
```



```
      call    GetSequencerReg
      jmp   RSEnd

RSMemmapped:
      add     bx, 0F0h
      call    GetExtdReg

RSEnd:
      ret
ReadScratch  endp

;---------------
; Procedure : DetectIomappedScratch
; Desc : Determine if scratch regs are I/O mapped in this chip.
; In   : None.
; Out  : Sets carry if IOmapped, clears carry otherwise.
; Regs : All registers preserved
;---------------
    public  DetectIomappedScratch
DetectIomappedScratch  proc    near
      push  ax
      push  bx
      clc
      mov   bx, SCRATCHPD_IO_3
      call  GetSequencerReg
      push  ax                    ; save value to restore later.
                                  ; try writing the reg.
      mov   al,0AAh
      call  SetSequencerReg
      mov   al,0h
      call  GetSequencerReg
      cmp   al,0AAh
      jne   DISNotDec
      stc                         ; Reg. does exist.
      jmp   DISDone

DISNotDec:
      clc

DISDone:
      ; restore scratch reg.
      pop   ax
      call  SetSequencerReg

      pop   bx
      pop   ax
      ret
DetectIomappedScratch  endp

;---------------
; Procedure: GetSequencerReg
; Description: Read Sequencer Register
; Input:  BL - index
; Output: AL - data
;---------------
    public  GetSequencerReg
GetSequencerReg  proc    near
      push  cx
      push  dx
      mov   dx, 3C4h
      in    al, dx

      mov   cl, al                ; save original index

      mov   al, bl
      out   dx, al
      inc   dx
      in    al, dx
```



```
      mov   ch, al              ; save data in CH

      mov   al, cl              ; Restore index.
      dec   dx
      out   dx, al
      mov   al, ch              ; restore data
      pop   dx
      pop   cx
      ret
GetSequencerReg  endp

;---------------
; Procedure : GetExtdReg
; Desc : Returns an Extended Configuration Register
; Input: BX[13:0] = extended register address
;        BX[15:14]= byte(00)/word(01)/dword(10)
; Output:
; Note: AX, BX are used
;---------------
    public  GetExtdReg
GetExtdReg  proc    near
      PUSHR   ds, es, cx, dx

    ; Enable memory-mapped registers to read configuration registers
      mov     dx, XR_BASE_REG; dx = 3c4
      in      al, dx            ; Read current value of index
      mov     es, ax            ; save it

      mov     al, 1Ch
      out     dx, al            ; Set to index 1ch
      in      ax, dx            ; Read current value
      mov     cl, ah            ; Save value in cl
      or      ah, XR_VGADIS_BIT ; Disable VGA access to A000-BFFF
      out     dx, ax

      mov     dx, bx            ; Save extended memory address
      and     bh, 3fh           ; Clear upper two bits

      call  ReadIOMappedReg

      test    dh, 0c0h          ; Upper bits: 00=byte, 01=word, 10=dword
      jz      GERDone

      inc     bx
      xchg    ah, al
      call  ReadIOMappedReg
      xchg    ah, al

      test    dh, 080h          ; Upper bits: 00=byte, 01=word, 10=dword
      jz      GERDone

      push    ax
      inc     bx
      call  ReadIOMappedReg
      xchg    ah, al
      inc     bx
      call  ReadIOMappedReg
      xchg    ah, al
      pop     ax

GERDone:
      push    ax
      mov     dx, XR_BASE_REG   ; dx = 3c4
      mov     al, 1Ch
      mov     ah, cl            ; Restore value
      out     dx, ax
      mov     ax, es            ; Restore prev. value of index.
```

```
        out     dx, al
        pop     ax

        POPR    ds, es, cx, dx
        ret
GetExtdReg  endp

;---------------
SCRATCHPD_IO_3 EQU 23h
```

16.1.8. Alternate access space pointer HIGH

Refer to "Alternate access space pointer LOW," on page 171, for a discussion of PMPOINTER.

Read/write:	r/w	*Memory offset:*	3C5
Default:	Undefined.	*Address index:*	28

7	6	5	4	3	2	1	0
PMPOINTER [17:10]							

Bits	
[7:0]	PMPOINTER HIGH. Bits [17:10] of memory mapped register offset. Low bits [9:2] of this field are contained in 3C5.1D, "Alternate access space pointer LOW," described on page 171

16.1.9. BIOS Paging

Use this register to map a 32K pages of BIOS ROM to a region of host memory.

Read/write:	r/w	*Memory offset:*	3C5h
Default:	Undefined.	*Address index:*	30

7	6	5	4	3	2	1	0
local	BIOS page memory mapping		BIOS page				

Bits	Description	
[4:0]	BIOS page	
	n =	select 32K BIOS page “n”.
	00000 =	BIOS paging disabled

Bits	Description
[6:5]	BIOS page memory mapping 11 = B800:0–BFFF:F. 10 = B000:0–B7FF:F. 01 = reserved. 00 = A000:0–A7FF:F.
[7]	Local 1 = BIOS local to device. 0 = BIOS on motherboard.

16.1.10. Extended/DAC status

Use this read only register to poll status conditions. Unlike other ProMotion setup registers, this status register is memory mapped. Reads from this register are non-blocking and may execute before the command FIFO has fully drained.

Device drivers MUST poll 1FC and wait until bit [8] is high before writing to the host BLT port. Writing to the host BLT port while 1FC[8] is low may produce unexpected results.

Read/write:	r	*Memory offset:*	1FC–1FFh
Default:	Undefined.	*Address index:*	-

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Undefined for reads. For write, see "Abort," described on page 180								3D engine busy	swap pending on tiles	swap completed on desktop	LEFT output	XHREF input	XVREF input	SCL input	SDA input	feature connector	EXSYNC pin	EXPCLK pin	EXVID pin	vertical display active	drawing engine busy	host BLT read	host BLT in progress	signature analyzer busy	DAC threshold blue	DAC threshold green	DAC threshold red	command FIFO entries available		

Bits	Description
[3:0]	Command FIFO entries available. 1111 = fifteen or more entries available (command FIFO empty). ... 0001 = one entry available. 0000 = zero entries available (command FIFO full). The Command FIFO entries available field returns the number of entries available in the command FIFO. If the host attempts to write more 32-bit entries into the FIFO than are available the host is held off until the FIFO is no longer full. Only drivers that want to avoid bus hold off during a sequence of commands need check this field.
[4]	DAC threshold red. 1 = DAC output exceeds threshold voltage. 0 = DAC output below threshold.

Bits	Description
[5]	DAC threshold green. 1 = DAC output exceeds threshold voltage. 0 = DAC output below threshold.
[6]	DAC threshold blue. 1 = DAC output exceeds threshold voltage. 0 = DAC output below threshold.
[7]	Signature analyzer busy. 1 = Signature analyzer busy. 0 = Signature analysis complete. Refer to "Signature analyzer overview," on page 301, for a discussion of the signature analyzer registers.
[8]	Host BLT in progress. 1 = host BLT is in progress. 0 = host BLT inactive. This bit may be used to indicate a host BLT operation where the host has read or written too few dwords. ! Device drivers MUST poll this bit wait until it is high before writing to the host BLT port. Writing to the host BLT port while 1FC[8] is low may cause unexpected results.
[9]	Host BLT read data available. 1 = host BLT data available to be read by host. 0 = no host BLT data available to be read.
[10]	Drawing engine busy. 1 = drawing engine busy/commands pending. 0 = idle drawing engine/empty command FIFO.
[11]	Vertical display active. 1 = raster within vertical active region. 0 = raster not within vertical active region. This bit may be polled for palette animation programs or other programs that require frame synchronization or count information.
[12]	Input on $\overline{\text{EXVID}}$ pin. This bit returns the input on the feature connector pin. 1 = pin high. 0 = pin low.
[13]	Input on $\overline{\text{EXPCLK}}$ pin. This bit returns the input on the feature connector pin. 1 = pin high. 0 = pin low.
[14]	Input on EXSYNC pin. This bit returns the input on the feature connector pin. 1 = pin high. 0 = pin low.
[15]	Feature connector input [3].
[16]	SDA input. This bit returns the input on the SDA pin. 1 = pin high. 0 = pin low. This bit is equivalent to 0D0[4]; refer to "DPMS/sync control," on page 232.
[17]	SCL input. This bit returns the input on the SCL pin. 1 = pin high. 0 = pin low.

Bits	Description
[18]	XVREF input. This bit returns the logic level on the VREF pin. 1 = pin high. 0 = pin low.
[19]	XHREF input. This bit returns the logic level on the XHREF pin. 1 = pin high. 0 = pin low.
[20]	LEFT output. This bit returns the stsus of the LEFT pin. 1 = pin high. 0 = pin low.
[21]	Swap completed on desktop. 1 = true. 0 = false.
[22]	Swap pending on tiles. 1 = true. 0 = false.
[23]	SCL input. This bit returns the input on the SCL pin. 1 = pin high. 0 = pin low.
[31:24]	Reserved on read. For write see "Abort," described on page 180.

16.1.11. Abort

Use this register to cancel the current drawing engine operation. It is intended primarily for mode-set operations.

Writing this register aborts the drawing engine operation in progress. The act of writing triggers the abort; any data written is ignored. Writing this register also aborts a host-BLT operation that is waiting for additional input from the host.

Writes to this abort register do not pass through or flush the command FIFO.

Read/write:	w	*Memory offset:*	1FFh
Default:	Undefined.	*Address index:*	-

7	6	5	4	3	2	1	0
any write aborts							

Bits	Description
no bits	Any write aborts the current drawing operation in progress.

16.2 Extended CRTC registers

16.2.1. Horizontal interlaced start

Use this register to specify the location within a horizontal line at which VSYNC occurs between fields of a frame. The unit is character clocks.

Writing to VGA register 3D5.00, Horizontal Total, resets this register unless autoreset is disabled. Refer to 3D5.1E, "Extended CRTC autoreset," on page 183, for a description of the autoreset feature.

Read/write:	r/w	*Address:*	3D5h
Default:	0h	*Address index:*	19

7	6	5	4	3	2	1	0
horizontal interlaced start							

Bits	Description
[7:0]	Horizontal interlaced start [7:0] of [8:0]. Horizontal interlaced start [8] is at 3D5.1B[4], "Horizontal overflow," on page 182.

16.2.2. Vertical extended overflow

Use this register to specify high order bits for vertical CRTC registers.

Writing to VGA register 3D5.00, Horizontal Total, resets this register unless autoreset is disabled. Refer to 3D5.1E, "Extended CRTC autoreset," on page 183, for a description of the autoreset feature.

Read/write:	r/w	*Address:*	3D5h
Default:	0h	*Address index:*	1A

7	6	5	4	3	2	1	0
			line compare	vertical retrace start	vertical blank start	vertical display enable end	vertical total

Bits	Description
[0]	Vertical total [10]. Overflow from "Vertical total," on page 145.
[1]	Vertical display enable end [10]. Overflow from"Vertical display enable end," on page 154.

Bits	Description
[2]	Vertical blank start [10]. Overflow from "Vertical blank start," on page 157.
[3]	Vertical retrace start [10]. Overflow from "Vertical retrace start," on page 152.
[4]	Line compare [10], Overflow from "Line compare," on page 160.

16.2.3. Horizontal overflow

Use this register to specify high order bits for horizontal CRTC registers.

Writing to VGA register 3D5.00, Horizontal Total, resets this register unless autoreset is disabled. Refer to 3D5.1E, "Extended CRTC autoreset," on page 183, for a description of the autoreset feature.

Read/write:	r/w	*Address:*	3D5h
Default:	0h	*Address index:*	1B

7	6	5	4	3	2	1	0
			horizontal interlaced start	horizontal retrace start	horizontal blank start	horizontal display enable end	horizontal total

Bits	Description
[0]	Horizontal total [8]. Overflow from "Horizontal total," on page 138.
[1]	Horizontal display enable end [8]. Overflow from "Horizontal display enable end," on page 140.
[2]	Horizontal blank start [8]. Overflow from "Horizontal blank start," on page 141.
[3]	Horizontal retrace start [8]. Overflow from "Horizontal retrace start," on page 143.
[4]	Horizontal interlaced start [8]. Overflow from "Horizontal interlaced start," on page 181.

16.2.4. Serial overflow

Use this register to specify high order bits for two serializer registers.

Writing to VGA register 3D5.00, Horizontal Total, resets this register unless autoreset is disabled. Refer to 3D5.1E, "Extended CRTC autoreset," on page 183, for a description of the autoreset feature.

Read/write:	r/w	*Address:*	3D5
Default:	0h	*Address index:*	1C

7	6	5	4	3	2	1	0
serial offset				serial start address			

Bits	Description
[3:0]	Serial start address [19:16]. Overflow from "Serial start address," on page 150.
[7:4]	Serial offset [11:8]. Overflow from "Serial offset," on page 155.

16.2.5. Character clock adjust

Use this register to specify a number of pixel clocks (0-5) by which the last character clock on each scan line is shortened.

Writing to VGA register 3D5.00, Horizontal Total, resets this register unless autoreset is disabled. Refer to 3D5.1E, "Extended CRTC autoreset," on page 183, for a description of the autoreset feature.

Read/write:	r/w	*Address:*	3D5h
Default:	0h	*Address index:*	1D

7	6	5	4	3	2	1	0
					character clock adjust		

Bits	Description
[2:0]	Character clock adjust. 11x = not valid. 101 = five pixel clocks. 100 = four pixel clocks. 011 = three pixel clocks. 010 = two pixel clocks. 001 = one pixel clocks. 000 = zero pixel clocks.

16.2.6. Extended CRTC autoreset

Use this register to override the feature by which any write to VGA register 3D5.00, "Horizontal total," described on page 138, resets all extended CRTC control registers and extended mode registers.

Table 16.2.6 Extended registers reset by writing 3D5.00 "Horizontal Total"

Register	Name and page number
3D5.19 [15:0]	"Horizontal interlaced start," on page 181.
3D5.1A [12:0]	"Vertical extended overflow," on page 181.

Table 16.2.6 Extended registers reset by writing 3D5.00 "Horizontal Total"

Register	Name and page number
3D5.1B [15:0]	"Horizontal overflow," on page 182.
3D5.1C [15:0]	"Serial overflow," on page 182.
3D5.1D [10:0]	"Character clock adjust," on page 183.
M080 [6:0]	"Serial control," on page 221.
M082 [0]	"vWindow group 0 control," on page 205.
M092 [0]	"vWindow group 1 control," on page 211.
M0C0 [9:0]	"Page offset," on page 222.
M0D2 [16]	"Monitor interlace control," on page 232.
M0E0 [7:0]	"Color correction," on page 253.
M0E4 [3:0]	"DAC control," on page 254
M140 [1:0]	"Hardware cursor control," on page 237.

This autoreset feature is provided to ensure compatibility with VGA legacy software that is not aware of ProMotion extended CRTC register bits.

Well-behaved applications should modify CRTC parameters only through driver and BIOS calls. However, legacy applications may write directly to VGA standard registers to set CRTC parameters. If legacy applications write to 3D5.00 then, regardless of the previous state of extended CRTC bits, these extended registers are forced to default values without the legacy application having to explicitly clear ProMotion's extended registers.

Note for BIOS writers: to set extended modes which require these fields, either (a) disable autoreset before setting mode, or (b) write 3D5.00 first and then explicitly set values for all registers affected by autoreset.

Read/write:	r/w	*Address:*	3D5h
Default:	0h.	*Address index:*	1Eh

7	6	5	4	3	2	1	0
							autoreset

Bits	Description
[0]	Automatic CRTC bit reset feature. 1 = disabled. 0 = enabled.

16.2.7. Vertical current position

Use this register to determine which scan line is being redrawn currently.

Read/write:	r	*Address:*	M1FAh
Default:	0h	*Address index:*	-

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits	Description
[10:0]	Current scan line

16.3 2D Drawing engine registers

16.3.1. Clipping control

See also "Clipping," on page 87, for programming notes.

Clipping abort M030[2] should not be used when destination update M040[28:27] is in use, as the destination location registers may not contain reliable endpoint information when the operation terminates. Refer to "Drawing engine control," on page 188 for a description of M040[28:27].

Read/write:	r/w	*Memory offset:*	030h
Default:	[0] =0h others =undefined.	*Address index:*	-

7	6	5	4	3	2	1	0
					clipping abort	*clipping polarity*	*clipping enable*

Bits	Description
[0]	Clipping enable. 1 = clipping enabled. 0 = clipping disabled.
[1]	Clipping polarity. 1 = outclip: pixels inside rectangle not drawn. 0 = inclip: pixels outside rectangle not drawn. Clipping polarity may be inverted if desired, in which case only pixels outside the clipping rectangle are written. The one-pixel border defined by the clipping rectangle is inside the rectangle.
[2]	Clipping abort. 1 = abort at next inside-outside transition of destination pointer. 0 = normal operation. The clipping abort feature permits early completion of vector and BITBLT operations. When the clipping abort bit is set, a vector operation terminates as soon as the destination pointer makes an inside-to-outside transition of the confines of the clipping rectangle. The vector may safely start outside of the clipping rectangle, it is not aborted until it enters and then leaves the rectangle. BITBLT operations are clipped in a similar way, except that only the vertical extent of the clipping region is used to abort the operation, as a BITBLT may enter and leave the clipping region on each line.

16.3.2. Clipping boundary left

Use this register to specify the left edge of the clipping rectangle.

Read/write:	r/w	*Memory offset:*	038–039h
Default:	Undefined.	*Address index:*	-

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				clipping boundary left											

Bits	Description
[11:0]	Clipping boundary left. This pixel is inside the region.

16.3.3. Clipping boundary top

Use this register to specify the top of the clipping rectangle.

Read/write:	r/w	*Memory offset:*	03A–03Bh
Default:	Undefined.	*Address index:*	-

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				clipping boundary top											

Bits	Description
[11:0]	Clipping boundary top. This pixel is inside the region.

16.3.4. Clipping boundary right

Use this register to specify the right edge of the clipping rectangle.

Read/write:	r/w	*Memory offset:*	03C–03Dh
Default:	Undefined.	*Address index:*	-

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				clipping boundary right											

Bits	Description
[11:0]	Clipping boundary right. This pixel is inside the region.

16.3.5. Clipping boundary bottom

Use this register to specify the bottom edge of the clipping rectangle.

Read/write:	r/w	*Memory offset:*	03E–03Fh
Default:	Undefined.	*Address index:*	-

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				clipping boundary bottom											

Bits	Description
[11:0]	Clipping boundary bottom. This pixel is inside the region.

16.3.6. Drawing engine control

See "Drawing engine control," on page 88, for a discussion of programming the drawing engine.

Clipping abort M030[2] should not be used when destination update M040[28:27] is in use, as the destination location registers may not contain reliable endpoint information when the operation terminates. Refer to "Clipping control," on page 186 for a description of M030[2].

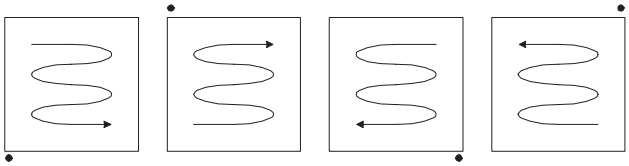
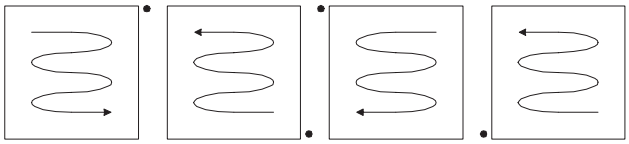
Read/write:	r/w	*Memory offset:*	040–043h
Default:	0h	*Address index:*	-

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
engine start	quick start		destination update		XY/linear access			pattern format		transparency	destination transp	contiguous/rectangle	destination address		pixel depth			source transparent	source color/mono	source rectangular/contiguous	Must be 0	source address XY/lineaer	major axis	direction y	direction X			drawing engine command			

Bits	Description
[3:0]	Drawing engine command. See notes on page 88.
	1101 = vector, don't draw endpoint.
	1100 = vector, draw endpoint.
	1001 = host BLT read–screen to memory.
	1000 = host BLT write–memory to screen.
	0100 = strip draw. Draws a single-pixel wide rectangular strip.
	0011 = reserved.
	0010 = rectangle.
	0001 = screen-screen BLT.
	0000 = NOP. Use to load register bits without starting an operation.
[5:4]	Reserved.

Bits	Description
[6]	Direction X. See notes on page 91. 1 = negative. 0 = positive.
[7]	Direction Y. See notes on page 91. 1 = negative. 0 = positive.
[8]	Major axis. See notes on page 91. 1 = X axis. 0 = Y axis.
[9]	Source address XY/linear. See notes on page 92. 1 = linear pixel count from the top-left corner of display memory. 0 = X,Y pair in pixels.
[10]	Reserved. Must be set to 0.
[11]	Source rectangular/contiguous. See notes on page 92. 1 = adjacent rows of source contiguous. 0 = adjacent rows of source are separated by a number of 64-bit regions, as determined by "Serial offset," described on page 155.
[12]	Source color/monochrome. See notes on page 92. 1 = source region monochrome. 0 = source region color, same depth as the display memory.
[13]	Source transparent. 1 = source pixels matching source transparency color not written. 0 = all source pixels written. When the source region is monochrome and the Transparent bit is set, any 0 bit in the source is transparent and is not written to the destination.
[16:14]	Bit depth. 100 = 24 bits per pixel 011 = 32 bits per pixel 010 = 16 bits per pixel 001 = 8 bits per pixel 000 = backwards compatability mode. Uses M080[2:1]; refer to "Serial control," on page 221.
[17]	E-Z linear mode. 1 = enabled. 0 = disabled.
[18]	Destination address XY/linear. See notes on page 93. 1 = linear pixel count from the top-left corner of display memory. 0 = (X,Y) pair in pixels.
[19]	Destination address rectangular. See notes on page 93. 1 = rectangular. 0 = other.
[20]	Destination transparent. 1 = destination transparency enabled. 0 = destination transparency not enabled.

Bits	Description
[21]	Destination transparency polarity. 1 = only pixels matching the transparency color are updated. 0 = pixels matching the transparency color are not updated. This bit applies to the destination region only, and only when destination transparency bit M040[20] is enabled.
[23:22]	Pattern format. 11 = 8×8×8b color pattern 10 = 8×8×1b monochrome. 01 = 4×4×4b color DitherFill™. 00 = none. Specifies a 4×4 16-color dither for the foreground color using the pattern register. In 256 color modes dither colors 0–7 into external palette colors 00–07h and dither colors 8–Fh into external palette colors F8–FFh. Set only this bit [22] or bit [10] (not both).
[26:24]	Address model: X/Y or linear. 111 = 1600 pixels per line. 110 = 1280 pixels per line. 101 = 1152 pixels per line. 100 = 1024 pixels. The two high order bits are not used. 011 = 512 pixels per line. 010 = 800 pixels per line. 001 = 640 pixels per line. 000 = linear (4096 pixels per line.). This specifies the number of pixels per line to be used when converting X/Y coordinates to linear display memory coordinates. This number may be larger than the width of the visible screen if off-screen display memory is available to the right of each row. For more information on X/Y vs. Linear addressing, see "Source formats, M040[13:10]," on page 92.

Bits	Description
[28:27]	Destination update. This feature permits the destination location to be updated automatically at the conclusion of each drawing engine operation in preparation for the next operation. 11 = destination location is set to the last pixel of the destination. This is useful for chained vector operations. This setting should not be used when clipping abort is enabled because the contents of the destination location registers may not match the last pixel drawn. 10 = the destination location is set to the pixel below the bottom-left corner of the destination. This is useful for host-assisted trapezoidal fills. As each strip of the trapezoid is drawn, the destination location is set to the next row down. When the left edge of the trapezoid is vertical or nearly vertical, this places the starting point correctly most of the time. When the left edge of the trapezoid is mostly horizontal, this places the Y coordinate of the destination location correctly, though once the destination location X coordinate must be set, the Y coordinate can generally be set at the same time. For right-to-left and/or bottom-to-top operation, the above operations are suitably mirrored, with destination location indicated by the dot in the following diagrams.  01 = destination location is set to the pixel to the right of the top-right corner of the destination. This is useful for both text operations and decoding RLE images. Right-to-left and/or bottom-to-top operations mirror the above as appropriate, with destination location indicated by the dot in the following diagrams.  00 = destination update disabled.
[30:29]	Quick start. See notes on page 93. 11 = start on destination. Causes the currently specified drawing engine operation to begin automatically when the Destination register is written, even if the write operation does not change the value in the register. 10 = start on source. Causes current drawing engine operation to begin automatically when the Source register is written, even if write operation does not change value in register. 01 = start on dimension X. Causes current drawing engine operation to begin automatically when Dimension X register is written, even if the write operation does not change value in register. 00 = quick start disabled.
[31]	Drawing engine start. Write bits M040[30:0] with this bit [31] = 0 in order to set up without starting an operation. 1 = start drawing operation. 0 = do not start drawing operation.

16.3.7. Raster operation

Use this register to specify the raster operation applied during a drawing engine operation. Refer to "Raster operation deltas," on page 100 for more information about programming this register.

This register is 8-bits in ProMotion-AT3D/AT24, and 4-bits in ProMotion-3210, 6410, and 6422. ProMotion-AT3D/AT24 have a three operand raster op, which is **NOT** backwards compatible with previous ProMotion family members.

You cannot use raster operations during a host BLT read or with a 4x4x4 DitherFill operation.

With PCI burst on, writing to the Raster Operation register M046 after writing to control register M040 may produce unexpected results. Refer to the sample source code below to avoid undesired operation.

Read/write:	r/w	*Memory offset:*	046h
Default:	Undefined.	*Address index:*	-

7	6	5	4	3	2	1	0
raster operation							

Bits	Description
[7:0]	Raster operation. Refer to "Raster operation deltas," on page 100, for a discussion of Raster ops.

16.3.7.1 Sample code

The "before code" example may produce unexpected results, with correction in the recommended solution.

Before (DO NOT USE!):

```
mov   eax, cs: base_command
or    eax, BLT_SS+SRC_MONO+SRC_LINEAR+SRC_XPARENT+BLT_START
mov   fs:BLT_CTRL, eax
mov   al, ROP_DSx
mov   fs:BLT_ROP,al
```

After (Recommended solution)

Use this sample code to prevent undesired results which may otherwise arise from writing M046 after M040.

```
mov   eax, cs: base_command
or    eax, BLT_SS+SRC_MONO+SRC_LINEAR+SRC_XPARENT+BLT_START
mov   fs:BLT_CTRL, eax
mov   ax, TEMP_Y+1
...
mov   fs:SRC_X,ax
mov   fs:SRC_Y,ax
```



```
mov   fs:DST_X,ax
mov   fs:DST_Y,ax
mov   al, ROP_DXx
mov   fs:BLT_ROP,al
...
```

16.3.8. Byte mask

Use this register to specify an 8-bit mask. The mask protects individual bytes in each aligned 32-bit region, as shown in the example below. This register is relevant only for drawing register operations, and has no effect in VGA mode.

ProMotion-AT24 has an 8-bit byte mask. ProMotion-3210, 6410, and 6422 controllers have 4-bit byte mask registers. Refer to page 101 for discussion of the changes.

This register must be enabled before writing with any drawing engine operation.

Read/write:	r/w	*Memory offset:*	047h
Default:	Undefined.	*Address index:*	-

7	6	5	4	3	2	1	0
byte mask							

Bits	Description	
[7:0]	Byte mask.	
	1 =	update byte.
	0 =	prevents the corresponding byte from being updated.

Figure 16.3.8 Byte mask

Example: mask = 00000111b

not written	not written	not written	not written	not written	R	G	B
63							0

16.3.9. Pattern

Use this register to specify type of pattern, anchored to the top-left corner of display memory. Apply the pattern in this register, instead of source pattern, to rectangle and strip draw operations using M040[10], described under "Drawing engine control," on page 188. The patterns available via M048 follow:

- 8×8×1 (monochrome) pattern
- 4×4×16-color DitherFill™ pattern
- 64 pixels of text character as a monochrome pattern, created from raster-font data

This register is accessed as doublewords only.

2×2×2 patterns, such as those used by OS/2, are not supported by this register.

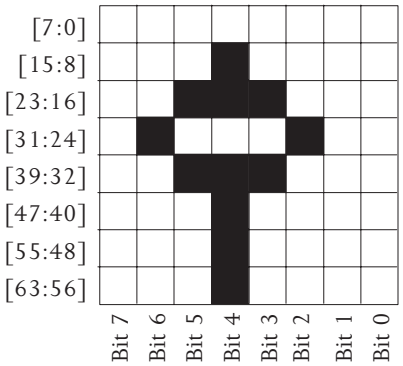
Write to this register prior to each use. The contents of M048–04F are not sustained across all operations.

Read/write:	r/w	*Memory offset:*	048–04Fh
Default:	Undefined.	*Address index:*	-

8×8×1 monochrome							
63:56	55:48	47:40	39:32	31:24	23:16	15:8	7:0
top row	r2	r3	r4	r5	r6	r7	bottom row

Bits	Description 8×8×1 (monochrome)
[7:0]	Top row of pattern.
[15:8]	Second row of pattern.
[23:16]	Third row of pattern.
[31:24]	Fourth row of pattern.
[39:32]	Fifth row of pattern.
[47:40]	Sixth row of pattern.
[55:48]	Seventh row of pattern.
[63:56]	Bottom row of pattern.

Figure 16.3.9a: 8×8×1 (monochrome) pattern



4×4×16 Ditherfill™															
63:60	59:56	55:52	51:48	47:44	43:40	39:36	35:32	31:28	27:24	23:20	19:16	15:12	11:8	7:4	3:0
r4c4	r4c3	r4c2	r4c1	r3c4	r3c3	r3c2	r3c1	r2c4	r2c3	r2c2	r2c1	r1c4	r1c3	r1c2	r1c1

Bits	Description 4×4×16-color DitherFill™
[3:0]	R1C1 color pixel of pattern
[7:4]	R1C2 color pixel of pattern
[11:8]	R1C3 color pixel of pattern
[15:12]	R1C4 color pixel of pattern
[19:16]	R2C1 color pixel of pattern
[23:20]	R2C2 color pixel of pattern
[27:24]	R2C3 color pixel of pattern
[31:28]	R2C4 color pixel of pattern
[35:32]	R3C1 color pixel of pattern
[39:36]	R3C2 color pixel of pattern
[43:40]	R3C3 color pixel of pattern
[47:44]	R3C4 color pixel of pattern
[51:48]	R4C1 color pixel of pattern
[55:52]	R4C2 color pixel of pattern
[59:56]	R4C3 color pixel of pattern
[63:60]	R4C4 color pixel of pattern

Figure 16.3.9b: 4×4×16-color Ditherfill™ pattern

	Col. 1	Col. 2	Col. 3	Col. 4
Row 1	[3:0]	[7:4]	[11:8]	[15:12]
Row 2	[19:16]	[23:20]	[27:24]	[31:28]
Row 3	[35:32]	[39:36]	[43:40]	[47:44]
Row 4	[51:48]	[55:52]	[59:56]	[63:60]

16.3.10. Source location X/low

Use this register to specify the corner of the source rectangle of a BITBLT operation. See "Source location registers," on page 94. This location is included in the region. The corner which this register specifies depends on direction bits.

Read/write:	r/w	*Memory offset:*	050–051h
Default:	Undefined.	*Address index:*	-

XY addressing															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				source location x											

Bits	Description (X/Y addressing mode)
[11:0]	X/Y addressing mode: source location X.

linear addressing															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				source linear pixel address (low)											

Bits	Description (Linear addressing mode)
[11:0]	Linear addressing mode: source linear pixel address [11:0] of [23:0].

16.3.11. Source location Y/high

Use this register to specify the corner of the source rectangle of a BITBLT operation. See "Source location registers," on page 94. This location is included in the region. The corner which this register specifies depends on direction bits.

Read/write:	r/w	*Memory offset:*	052–053h
Default:	Undefined.	*Address index:*	-

XY addressing															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				source location y											

Bits	Description (X/Y addressing mode)
[11:0]	X/Y addressing mode: source location Y.

linear addressing															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				source linear pixel address (high)											

Bits	Description (Linear addressing mode)
[11:0]	Linear addressing mode: source linear pixel address [23:12] of [23:0].

16.3.12. Destination location X/low

Use this register to specify the corner of the destination rectangle of a BITBLT operation. See "Destination location registers," on page 94, for notes. This location is included in the region. Which corner this specifies depends on direction bits.

This register may change automatically when destination update is enabled. Refer to M040[28:27], "Drawing engine control," on page 188 for information on destination update.

Read/write:	r/w	*Memory offset:*	054–055h
Default:	*See below.*	*Address index:*	-

XY addressing															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				destination location x											

Bits	Description (X/Y addressing mode)
[11:0]	Destination location X. Default is 1h.

linear addressing															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				destination linear pixel address (low)											

Bits	Description (Linear addressing mode)
[11:0]	Destination linear pixel address [11:0] of [23:0]. Default is 1h.

16.3.13. Destination location Y/high

Use this register to specify the corner of the destination rectangle of a BITBLT operation. See "Destination location registers," on page 94, for notes. This location is included in the region. Which corner this specifies depends on direction bits.

This register may change automatically when destination update is enabled. Refer to M040[28:27], "Drawing engine control," on page 188 for information on destination update.

Read/write:	r/w	*Memory offset:*	056–057h
Default:	*See below.*	*Address index:*	-

XY															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				destination location y											

Bits	Description (X/Y addressing mode)
[11:0]	Destination location Y. Default is 1h.

linear															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				destination linear pixel address (high)											

Bits	Description (Linear addressing mode)
[11:0]	Destination linear pixel address [23:12] of [23:0]. Default is 1h.

16.3.14. Source size X/vector pixel count

Read/write:	r/w	Memory offset:	058–059h
Default:	Undefined.	Address index:	-

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				dimension x											

Bits	Description
[11:0]	Dimension X pixel count. This is the width of the source, in pixels, for BITBLT operations, or major axis pixel count for vector operations.

16.3.15. Source size Y

This register has no effect for vector operations.

Read/write:	r/w	Memory offset:	05A–05Bh
Default:	Undefined.	Address index:	-

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				dimension y											

Bits	Description
[11:0]	Dimension Y pixel/line count. This is the height of the source, in pixels/lines, for BITBLT operations.

16.3.16. Destination row pitch

Use this register when operating in packed 24-bit mode to specify the number of bytes between vertically adjacent pixels in display memory.

In ProMotion-AT24/3D this register is active only for 24-bit packed modes. For ProMotion-AT3D this register applies to all modes.

Read/write:	r/w	*Memory offset:*	05C–05Dh
Default:	Undefined.	*Address index:*	-

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			destination row pitch												

Bits	Description
[12:0]	Destination row pitch.

16.3.17. Source row pitch

Use this register when operating in packed 24-bit mode to specify the number of bytes between vertically adjacent pixels in display memory.

In ProMotion-AT24 this register is reserved.

Read/write:	r/w	*Memory offset:*	05C–05Dh
Default:	Undefined.	*Address index:*	-

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			source row pitch												

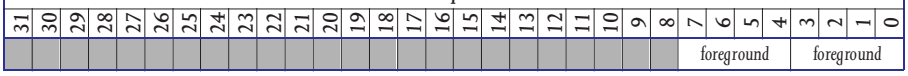
Bits	Description
[12:0]	Source row pitch.

16.3.18. Foreground color

Use this register to specify the color for rectangle fill, strip draw, vector draw, and the color of 1 bits after monochrome-to-color expansion.

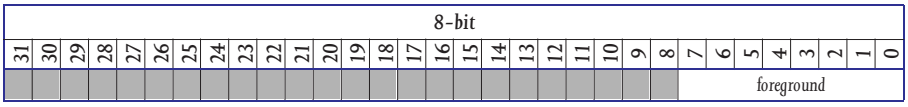
Read/write:	r/w	*Memory offset:*	060–063h
Default:	Undefined.	*Address index:*	-

4-bit packed



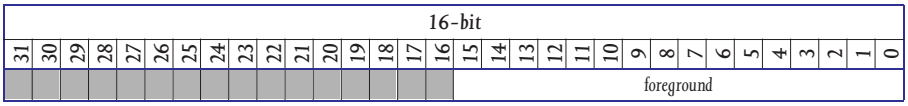
Bits	Description (4-bit packed mode)
[3:0]	Foreground color. Bits [7:4] must match [3:0].
[7:4]	Foreground color. Bits [7:4] must match [3:0].
[31:8]	Reserved.

8-bit



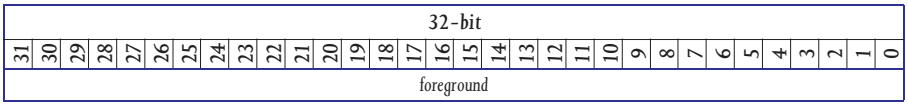
Bits	Description (8-bit mode)
[7:0]	Foreground color.
[31:8]	Reserved.

16-bit



Bits	Description (16-bit mode)
[31:16]	Reserved.
[15:0]	Foreground color.

32-bit



Bits	Description (32-bit mode)
[31:0]	Foreground color.

16.3.19. Background color/source transparency

Use this register to specify the color used to draw 0 bits during monochrome-to-color expansion.

Use the background color register also to specify the source transparency color when source transparency is enabled.

Read/write:	r/w	*Memory offset:*	064–067h
Default:	Undefined.	*Address index:*	-

4-bit packed

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																								background				background			

Bits	Description (4-bit packed mode)
[3:0]	background color. Bits [7:4] must match [3:0].
[7:4]	background color. Bits [7:4] must match [3:0].
[31:8]	Reserved.

8-bit

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																								background							

Bits	Description (8-bit mode)
[7:0]	background color.
[31:8]	Reserved.

16-bit

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																background															

Bits	Description (16-bit mode)
[31:16]	Reserved.
[15:0]	background color.

32-bit

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
background																															

Bits	Description (32-bit mode)
[31:0]	background color.

16.3.20. Destination transparency color

Use this register to specify which destination color is transparent or opaque. This register is valid only when destination transparency bit M040[20], "Drawing engine control," described on page 188, is enabled.

Read/write:	r/w	*Memory offset:*	06C–06Eh
Default:	Undefined.	*Address index:*	-

4-bit packed

23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																destination transparency				destination transparency			

Bits	Description (4-bit packed mode)
[3:0]	*Destination transparency* color. Bits [7:4] must match [3:0].
[7:4]	*Destination transparency* color. Bits [7:4] must match [3:0].
[23:8]	Reserved.

8-bit

23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																destination transparency							

Bits	Description (8-bit mode)
[7:0]	*Destination transparency* color.
[23:8]	Reserved.

16-bit

23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								destination transparency															

Bits	Description (16-bit mode)
[15:0]	*Destination transparency* color.
[23:16]	Reserved.

24-bit

23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
destination transparency																							

Bits	Description (32-bit mode)
[23:0]	*Destination transparency* color.

16.3.21. Destination transparency mask

Use this register to compare display memory against the color mask register.

Read/write:	r/w	*Memory offset:*	06F
Default:	Undefined.	*Address index:*	-

7	6	5	4	3	2	1	0
			compare [31:24]	compare [23:16]	compare [15]	compare [14:8]	compare [7:0]

Bits	Description
[0]	Compare bits [7:0] in display memory against bits [7:0] of color mask register.
[1]	Compare bits [14:8] in display memory against bits [14:8] of color mask register.
[2]	Compare bit [15] in display memory against bits [15] of color mask register.
[3]	Compare bits [23:16] in display memory against bits [23:16] of color mask register.
[4]	Compare bits [31:24] in display memory against bits [7:0] of color mask register.
[7:5]	Reserved.

16.3.22. DDA axial step constant

Use this digital differential analyzer register to specify the slope of a line. See "Vector line draw," on page 89, for programmer's notes.

Read/write:	r/w	*Memory offset:*	070–072h
Default:	Undefined.	*Address index:*	-

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
axial step constant															

Bits	Description
[15:0]	DDA axial step constant.

16.3.23. DDA diagonal step constant

Use this digital differential analyzer register to specify the slope of a line. See "Vector line draw," on page 89, for programmer's notes.

Read/write:	r/w	*Memory offset:*	072–073h
Default:	Undefined.	*Address index:*	-

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
diagonal step constant															

Bits	Description
[15:0]	DDA diagonal step constant.

16.3.24. DDA error term

Use this register to initialize the digital differential analyzer before drawing a line with the DDA step constant registers. See "Vector line draw," on page 89, for programmer's notes.

Read/write:	r/w	*Memory offset:*	074–075h
Default:	Undefined.	*Address index:*	-

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
error term															

Bits	Description
[15:0]	DDA error term.

16.4 Motion video registers

See "Motion video notes," on page 105, for additional information about motion video on the ProMotion-aT3D.

Note that there are two sets of video window registers, vWindow 0 and vWindow 1, each set corresponding to a rectangular video window region. The visible portion of each vWindow need not be rectangular, and is composed of a number of rectangular tiles.

There may be up to 12 "tiles" on the screen at one time. Each tile is attached to one of the two possible "windows" and thus takes on attributes of that window, such as scale factor and data format.

16.4.1. vWindow group 0 control

Use this register to control motion video window 0.

Writing to any CRTC register among 3D5.0–17 resets M082[0].

Writing to VGA register 3D5.00, Horizontal Total, resets this register unless autoreset is disabled. Refer to 3D5.1E, "Extended CRTC autoreset," on page 183, for a description of the autoreset feature.

Read/write:	r/w	*Memory offset:*	082–083h
Default:	0h	*Address index:*	-

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	enable chromakey	stretch minimum replication	enable smoothing filter	Enable vertical pixel interpolation	Enable horizontal pixel interpolation	Enable stretch	YUV → RGB		vWindow format			vWindow pixel depth			vWindow enable

Bits	Description
[0]	Enable vWindow 0. All tiles that form the visible window are enabled. There is no individual window control for tiles. ! Any write to any CRTC register among 3D5.0–17 resets this bit to 0.
[3:1]	vWindow pixel depth in display memory. This field specifies the pixel size of video data as stored in display memory. This may or may not be the same depth as graphics data store in display memory. 111 = 32 bits per pixel. 101 = 16 bits per pixel. 100 = 15 bits per pixel. 010 = 8 bits per pixel. Other settings are reserved.

Bits	Description
[6:4]	vWindow format in display memory. This is not necessarily the same format as graphics data store in display memory. 111 = YUV 4:0:0. 110 = YUV 4:2:2 (U = high byte). 101 = YUV 4:1:1. 100 = YUV 4:2:2 (U = low byte). 011 = YUV 4:2:2 (V = high byte). 010 = RGB. 001 = reserved. 000 = indexed. Other settings are reserved.
[7]	Reserved.
[8]	YUV-to-RGB conversion. Set whenever a YUV format is specified for the video window. 1 = enabled. 0 = disabled.
[9]	vWindow 0 stretch. 1 = enabled. 0 = disabled. When this bit is set, each pixel of vWindow 0 data is displayed as one or more pixels on the screen. Scale factors below 1.0 (shrinking) are not supported. Set this bit in conjunction with the video scale factor registers: • "vWindow group 0 scale factor horizontal," described on page 208; • "vWindow group 0 scale offset horizontal," described on page 208; • "vWindow group 0 scale factor vertical," described on page 209; and • "vWindow group 0 stretch offset vertical," described on page 209.
[10]	Horizontal stretch pixel interpolation. 1 = color blending enabled. 0 = pixel replication. When this bit is set, pixels are stretched using linear weighting. When this bit is not set, pixels are stretched using pixel replication. You may use pixel interpolation (color blending) with YUV or RGB video data formats, but should not be used with indexed video data.
[11]	Vertical stretch pixel interpolation. 1 = enabled. 0 = disabled. When this bit is set, pixels are stretched using linear weighting. When this bit is not set, pixels are stretched using pixel replication. You may use pixel interpolation with YUV or RGB video data formats, but should not be used with indexed video data. ! ProMotion-AT24/3D supports vertical interpolation only for vWindow 0.
[12]	Smoothing filter. 1 = enabled. 0 = disabled. This bit enables an lowpass filter. This filter is designed for only YUV data, and attempts to reconstruct YUV 4:4:4 data from the YUV 4:2:2 input.

Bits	Description
[13]	Reserved.
[14]	Chromakey. 1 = enable. 0 = disabled. This bit enables chromakey-based merging of desktop data with data from the feature connector

16.4.2. vWindow group 0 data pitch

Use this register to specify the byte address difference, in doublewords, between adjacent rows of the vWindow data in display memory.

When a video window is composed of multiple tiles, even when not all of the video window is visible, the data representing the video window is stored as a single packed rectangle in display memory, with a single pitch.

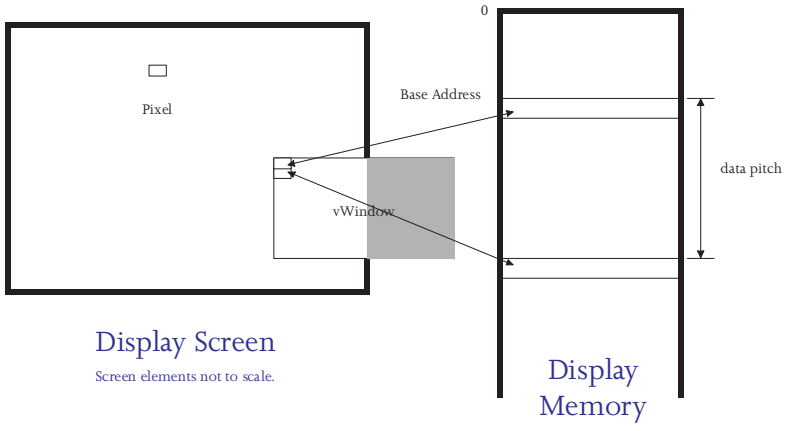
In cases of "in-place video windows," the video data pitch matches the VGA Offset register, adjusted for the fact that the VGA Offset register is specified in quadwords.

Read/write:	r/w	*Memory offset:*	084–085h
Default:	Undefined.	*Address index:*	-

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				data pitch											

Bits	Description
[11:0]	vWindow 0 data pitch, in dwords.

Figure 16.4.2 vWindow 0 data pitch



16.4.3. vWindow group 0 scale factor horizontal

Use this register to specify the horizontal stretch in combination with "vWindow group 0 scale offset horizontal." Horizontal and vertical stretch factors should be computed separately. It is not necessary for both factors to be the same. If set inaccurately, the right edge of the vWindow may not stop where expected and may extend to the right edge of the screen.

The value in this register specifies the number of horizontal video pixels corresponding to a screen pixel. Since there is an implied binary point to the left of the MSB of this register, the possible values in this register range from 0.000h to almost 0.FFFh. Formula:

Video Scale Factor Horizontal = 4096 * (SourceX_Dimension - 1) / DestinationX_Dimension-1)

Exception: A value of 0 in this register is used to represent a factor of 1.000.

Read/write:	r/w	*Memory offset:*	086–087h
Default:	Undefined.	*Address index:*	-

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				vWindow 0 scale factor horizontal											

Bits	Description
[11:0]	vWindow 0 scale factor horizontal. Loaded with the width of the unstretched image, in pixels, minus one.

16.4.4. vWindow group 0 scale offset horizontal

Use this register to specify the horizontal stretch in combination with "vWindow group 0 scale factor horizontal."

This register contains a value required internally by the horizontal interpolation circuitry. It is loaded according to the following formula:

Video Scale Offset Horizontal = FFFh - ((Video Scale Factor Horizontal) * (WindowLeftPosition - 1))&FFFh

Read/write:	r/w	*Memory offset:*	088–89h
Default:	Undefined.	*Address index:*	-

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				vWindow 0 scale offset horizontal											

Bits	Description
[11:0]	vWindow 0 scale offset horizontal. Loaded into the register as a 2's complement number, with the width of the unstretched image minus the width of the stretched image. Since only stretching is supported (and not shrinking) this result is always negative.

16.4.5. vWindow group 0 scale factor vertical

Use this register to specify the vertical stretch in combination with "vWindow group 0 stretch offset vertical." Horizontal and vertical stretch factors should be computed separately. It is not necessary for both factors to be the same. This is analogous to the Video Scale Factor Horizontal register. Formula:

Video Scale Factor Vertical = 4096 * (SourceY_Dimension - 1) / (DestinationY_Dimension - 1)

Exception: A value of 0 in this register is used to represent a factor of 1.000.

Read/write:	r/w	*Memory offset:*	08A–08Bh
Default:	Undefined.	*Address index:*	-

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				vWindow 0 scale factor vertical											

Bits	Description
[11:0]	vWindow 0 scale factor vertical. Loaded with the height of the unstretched image, in pixels, minus one.

16.4.6. vWindow group 0 stretch offset vertical

Use this register to specify the vertical stretch in combination with "vWindow group 0 scale factor vertical." This is analogous to the Video Scale Offset Horizontal register. It is loaded with a value calculated by the following formula:

Video Scale Offset Vertical = FFFh - ((Video Scale Factor Vertical) * (WindowTopPosition) &FFFh

Read/write:	r/w	*Memory offset:*	08C–08Dh
Default:	Undefined.	*Address index:*	-

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				vWindow 0 stretch offset vertical											

Bits	Description
[11:0]	vWindow 0 stretch offset vertical. Loaded into the register as a 2's complement number, with the height of the unstretched image minus the height of the stretched image. Since only stretching is supported (and not shrinking) this result is always negative.

16.4.7. Tile sequence control

Use this register to specify how the 12 video tiles are arranged into multiple visible buffers.

Read/write:	r/w	*Memory offset:*	08E–08Fh
Default:	Undefined.	*Address index:*	-

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits	Description
[3:0]	Tile sequence base.
[7:4]	Tile sequence length. 1111 = reserved. 1110 = reserved. 1101 = reserved. 1100 = reserved. 1011 = 11 tiles. 1010 = 10 tiles. 1001 = 9 tiles. 1000 = 8 tiles. 0111 = 7 tiles. 0110 = 6 tiles. 0101 = 5 tiles. 0100 = 4 tiles. 0011 = 3 tiles. 0010 = 2 tiles. 0001 = 1 tile. 0000 = 12 tiles. ☞ A value of zero is treated as a length of 12 tiles.
[8]	Buffer swap using TV input.
[9]	Buffer swap using SWAP input.
[11:10]	Buffer count 11 = reserved. 10 = triple buffering. 01 = double buffering. 00 = single buffering.
[12]	Stereo buffering enable The tile sequence for each frame begins with the Tile sequence base 08E[3:0] plus zero or more multiples of the Tile sequence length 08E[7:4]. Typically sequences differ only in Tile data location ~ register values.

16.4.8. Chromakey color

Use this register to specify the color for chromakeying when receiving video data through the feature connector. Chromakey is not available for video data stored in display memory.

Desktop pixels that do not match the chromakey color are displayed. When desktop pixels match the chromakey color, then the foreground color pixel is displayed instead.

Read/write:	r/w	*Memory offset:*	090–091h
Default:	Undefined.	*Address index:*	-

direct color															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
chromakey color															

Bits	Description (direct color)
[15:0]	Chromakey color.

indexed 8-bit color															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								chromakey color							

Bits	Description (indexed 8-bit color)
[7:0]	Chromakey color.

16.4.9. vWindow group 1 control

Use this register to control motion video window 1.

Writing to any CRTC register among 3D5.0–17 resets M092[0].

Writing to VGA register 3D5.00, Horizontal Total, resets this register unless autoreset is disabled. Refer to 3D5.1E, "Extended CRTC autoreset," on page 183, for a description of the autoreset feature.

Read/write:	r/w	*Memory offset:*	092–093h
Default:	0h	*Address index:*	-

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	enable chromakey	stretch minimum replication	enable smoothing filter	Enable vertical pixel interpolation	Enable horizontal pixel interpolation	Enable stretch	YUV → RGB		vWindow format			vWindow pixel depth			vWindow enable

Bits	Description
[0]	Enable vWindow 1. All tiles that form the visible window are enabled. There is no individual window control for tiles. ! Any write to any CRTC register among 3D5.0–17 resets this bit to 0.
[3:1]	vWindow 1 pixel depth in display memory. This field specifies the pixel size of video data as stored in display memory. This may or may not be the same depth as graphics data store in display memory. 111 = 32 bits per pixel. 101 = 16 bits per pixel. 100 = 15 bits per pixel. 010 = 8 bits per pixel. Other settings are reserved.
[6:4]	vWindow 1 format in display memory. This is not necessarily the same format as graphics data store in display memory. 111 = YUV 4:0:0. 101 = YUV 4:1:1. 100 = YUV 4:2:2. 010 = RGB. 000 = indexed. Other settings are reserved.
[7]	Reserved.
[8]	YUV-to-RGB conversion. Set whenever a YUV format is specified for the video window. 1 = enabled. 0 = disabled.
[9]	vWindow 1 stretch. 1 = enabled. 0 = disabled. When this bit is set, each pixel of vWindow 0 data is displayed as one or more pixels on the screen. Scale factors below 1.0 (shrinking) are not supported. Set this bit in conjunction with the video scale factor registers: • "vWindow group 1 scale factor horizontal," described on page 214; • "vWindow group 1 scale offset horizontal," described on page 215; • "vWindow group 1 scale factor vertical," described on page 215; and • "vWindow group 1 stretch offset vertical," described on page 216.
[10]	Horizontal stretch pixel interpolation. 1 = color blending enabled. 0 = pixel replication. When this bit is set, pixels are stretched using linear weighting. When this bit is not set, pixels are stretched using pixel replication. You may use pixel interpolation (color blending) with YUV or RGB video data formats, but should not be used with indexed video data.

Bits	Description
[11]	Vertical stretch pixel interpolation. 1 = enabled. 0 = disabled. When this bit is set, pixels are stretched using linear weighting. When this bit is not set, pixels are stretched using pixel replication. You may use pixel interpolation with YUV or RGB video data formats, but should not be used with indexed video data. ! ProMotion-AT24 supports vertical interpolation only for vWindow 0.
[12]	Smoothing filter. 1 = enabled. 0 = disabled. This bit enables an lowpass filter. This filter is designed for only YUV data, and attempts to reconstruct YUV 4:4:4 data from the YUV 4:2:2 input.
[13]	Reserved.
[14]	Chromakey. 1 = enable. 0 = disabled. This bit enables chromakey-based merging of desktop data with data from the feature connector

16.4.10. vWindow group 1 data pitch

Use this register to specify the byte address difference, in doublewords, between adjacent rows of the vWindow data in display memory.

When a video window is composed of multiple tiles, even when not all of the video window is visible, the data representing the video window is stored as a single packed rectangle in display memory, with a single pitch.

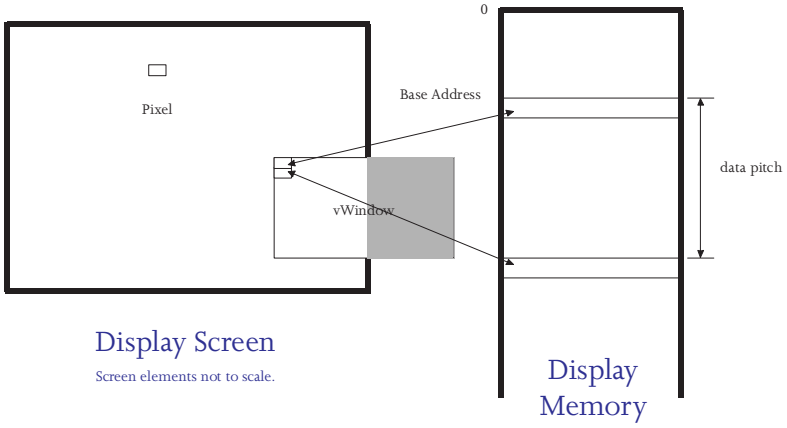
In cases of "in-place video windows," the video data pitch matches the VGA Offset register, adjusted for the fact that the VGA Offset register is specified in quadwords.

Read/write:	r/w	*Memory offset:*	094–095h
Default:	Undefined.	*Address index:*	-

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				data pitch											

Bits	Description
[11:0]	vWindow 1 data pitch, in dwords.

Figure 16.4.10 vWindow 1 data pitch



16.4.11. vWindow group 1 scale factor horizontal

Use this register to specify the horizontal stretch in combination with "vWindow group 1 scale offset horizontal." Horizontal and vertical stretch factors should be computed separately. It is not necessary for both factors to be the same. If set inaccurately, the right edge of the vWindow may not stop where expected and may extend to the right edge of the screen.

The value in this register specifies the number of horizontal video pixels corresponding to a screen pixel. Since there is an implied binary point to the left of the MSB of this register, the possible values in this register range from 0.000h to almost 0.FFFh. Formula:

Video Scale Factor Horizontal = 4096 * (SourceX_Dimension - 1) / DestinationX_Dimension - 1)

Exception: A value of 0 in this register is used to represent a factor of 1.000.

Read/write:	r/w	*Memory offset:*	096–097h
Default:	Undefined.	*Address index:*	-

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				vWindow 1 scale factor horizontal											

Bits	Description
[11:0]	vWindow 1 scale factor horizontal. Loaded with the width of the unstretched image, in pixels, minus one.

16.4.12. vWindow group 1 scale offset horizontal

Use this register to specify the horizontal stretch in combination with "vWindow group 1 scale factor horizontal," described on page 214.

This register contains a value required internally by the horizontal interpolation circuitry. It is loaded according to the following formula:

Video Scale Offset Horizontal = FFFh - ((Video Scale Factor Horizontal) * (WindowLeftPosition - 1))&FFFh

Read/write:	r/w	*Memory offset:*	098–99h
Default:	Undefined.	*Address index:*	-

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				vWindow 1 scale offset horizontal											

Bits	Description
[11:0]	vWindow 1 scale offset horizontal. Loaded into the register as a 2's complement number, with the width of the unstretched image minus the width of the stretched image. Since only stretching is supported (and not shrinking) this result is always negative.

16.4.13. vWindow group 1 scale factor vertical

Use this register to specify the vertical stretch in combination with "vWindow group 1 stretch offset vertical.". Horizontal and vertical stretch factors should be computed separately. It is not necessary for both factors to be the same. This is analogous to the Video Scale Factor Horizontal register. Formula:

```
Video Scale Factor Vertical=
4096 * (SourceY_Dimension - 1) / (DestinationY_Dimension - 1)
```

Exception: A value of 0 in this register is used to represent a factor of 1.000.

Read/write:	r/w	*Memory offset:*	09A–09Bh
Default:	Undefined.	*Address index:*	-

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				vWindow 1 scale factor vertical											

Bits	Description
[11:0]	vWindow 1 scale factor vertical. Loaded with the height of the unstretched image, in pixels, minus one.

16.4.14. vWindow group 1 stretch offset vertical

Use this register to specify the vertical stretch in combination with "vWindow group 1 scale factor vertical." This is analogous to the Video Scale Offset Horizontal register. It is loaded with a value calculated by the following formula:

Video Scale Offset Vertical = FFFh - ((Video Scale Factor Vertical) * (WindowTopPosition) &FFFh

Read/write:	r/w	*Memory offset:*	09C–09Dh
Default:	Undefined.	*Address index:*	-

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				vWindow 1 stretch offset vertical											

Bits	*Description*
[11:0]	vWindow 1 stretch offset vertical. Loaded into the register as a 2's complement number, with the height of the unstretched image minus the height of the stretched image. Since only stretching is supported (and not shrinking) this result is always negative.

16.5 Video tile buffer registers

ProMotion-AT3D has 12 sets of video tile registers.

Table 16.5 Video tile buffer register groups 0–11

Tile 0 registers 200–20F.	Tile 6 registers 260–26F.
Tile 1 registers 210–21F.	Tile 7 registers 270–27F.
Tile 2 registers 220–22F.	Tile 8 registers 280–28F.
Tile 3 registers 230–23F.	Tile 9 registers 290–290F.
Tile 4 registers 240–24F.	Tile 10 registers 2A0–2AF.
Tile 5 registers 250–25F.	Tile 11 registers 2B0–2BF.

Tile 0 registers are detailed below. Tile 2–11 buffer register groups have equivalent parallel structure.

16.5.1. Tile 0 control register

Use this register to specify control information for tile 0.

Read/write:	r/w	*Memory offset:*	200h
Default:	*Undefined.*	*Address index:*	-

7	6	5	4	3	2	1	0
			tile rightmost		*tile vWindow select*		

Bits	*Description*
[2:0]	Tile vWindow select. 1xx = reserved. 011 = reserved. 010 = vWindow 1. 001 = vWindow 0. 000 = reserved. This field specifies whether the tile is assigned to vWindow 0 or vWindow 1
[3]	Reserved.
[4]	Tile rightmost. 1 = enabled. 0 = default. Set this bit when the tile is the rightmost (or only) tile in a tile strip

16.5.2. Tile 0 display position left

Use this register to specify the left edge of the video tile, in pixels. The top-left corner of the screen is (0,0). The column specified by this register is **inside** the video tile.

Read/write:	r/w	*Memory offset:*	202–203h
Default:	Undefined.	*Address index:*	-

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					tile 0 display position left										

Bits	Description
[10:0]	Tile 0 display position left. This column is **inside** video tile 0.

16.5.3. Tile 0 display position right

Use this register to specify the right edge of the video tile, in pixels. The column specified by this register is just outside the video window .

Read/write:	r/w	*Memory offset:*	204–205h
Default:	Undefined.	*Address index:*	-

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					tile 0 display position right										

Bits	Description
[10:0]	Tile 0 display position right. This column is **outside** video tile 0.

16.5.4. Tile 0 display position bottom

Use this register to specify the bottom edge of the video tile, in pixels. The row specified by this register is **inside** the video tile.

ProMotion-AT24 has no register for Tile display position Top.

Read/write:	r/w	*Memory offset:*	206–207h
Default:	Undefined.	*Address index:*	-

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					tile 0 display position bottom										

Bits	Description
[10:0]	Tile 0 display position bottom. This row is inside video tile 0.

16.5.5. Tile 0 data width

Use this register to specify the width, in units of pixels, of the video data stored in the corresponding tile. This register does not specify the width of the video tile on the screen, which may be larger due to stretching.

Tile Data Width = TileRightUnstretched - TileLeftUnstretched

where:

TileRightUnstretched = ((TileRight - (WindowLeft - 1))*(Video Scale Factor Horizontal) + 0xFFF)/4096

TileLeftUnstretched = ((TileLeft - (WindowLeft))*(Video Scale Factor Horizontal) + 0xFFF)/4096

Read/write:	r/w	*Memory offset:*	208–209h
Default:	Undefined.	*Address index:*	-

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					tile 0 data width										

Bits	*Description*
[10:0]	Tile 0 data width.

16.5.6. Tile 0 data location

Use this register to specify the address, in bytes, of the top-left corner of the video tile data where it is stored in display memory, not the location of the window where the data is displayed.

TileDataLocation = (Beginning address of source data) + (TopOffset)*(Stride of source data) + (LeftOffset) * (Bytes per pixel of motion video window)

where

(TopOffset = ((TileTop - 1) - WindowTop) * (Video Scale Factor Vertical)/4096

LeftOffset = ((TileLeft) - WindowLeft) * (Video Scale Factor Horizontal)/4096

Read/write:	r/w	*Memory offset:*	20A–20C
Default:	Undefined.	*Address index:*	-

23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		tile 0 data location																					

Bits	*Description*
[21:0]	Tile 0 data location.

16.5.7. Tile 0 current data location

This register is for internal controller use, and is not intended for host access.

Read/write:	*auto*	*Memory offset:*	20D–20F
Default:	*Undefined.*	*Address index:*	-

23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		tile 0 current data location																					

Bits	Description
[21:0]	Tile 0 current data location in bytes.

16.6 Extended configuration registers

16.6.1. Serial control

Use this register to specify direct color and high-resolution modes.

ProMotion vWindow is not supported in double indexed modes.

Writing to VGA register 3D5.00, Horizontal Total, resets this register unless autoreset is disabled. Refer to 3D5.1E, "Extended CRTC autoreset," on page 183, for a description of the autoreset feature.

Read/write:	r/w	*Memory offset:*	080h
Default:	0h	*Address index:*	-

7	6	5	4	3	2	1	0
nibble swap	*VGA enable*	*double index*	*desktop pixel format*		*desktop bit depth*		

Bits	Description
[2:0]	Desktop pixel depth. 111 = 32 bits per pixel. 101 = 16 bits per pixel. 100 = 15 bits per pixel. 010 = 8 bits per pixel. 001 = 4 bits per pixel. 000 = VGA modes. Any write to CRTC registers 3D5.0–17 resets these bits to 0h.
[4:3]	Desktop pixel format. 1x = reserved. 01 = direct RGB. 00 = indexed. This selection is independent of the pixel format of the motion video window.
[5]	Double index. 1 = double index enabled. 0 = double index disabled. This bit enables two 8-bit pixels per clock to be transferred to the ProMotion-aT3D internal DAC. Use double indexing for high resolution modes which exceed maximum VCLK frequency while in single indexing mode. ☞ The vWindow is not available when double-index mode is enabled. When this bit = 1 then the motion video registers must be set as follows: M082[0] = 0; M080[4:3] = 00; M080[2:0] = 010.

Bits	Description
[6]	Enable extended VGA modes. 1 = Disables 256K address wrapping; disables pixel doubling circuitry in video refresh logic. 0 = Enables 256K address wrapping in video reftresh logic. Set this bit for VGA modes >13h.
[7]	Nibble swap mode. 1 = low nibble of each byte on left. 0 = high nibble of each byte on left. This bit is relevant only to 4-bit packed modes.

16.6.2. Page offset

Use this register to position the 64K aperture within display memory. This offset is in 4KB units. For example, when page offset is 0 then the offset is 0 x 4K, and A000:0–AFFF:F access display memory locations 0K–64K. When page offset is 1 then the offset is 1 x 4K, and A000:0–AFFF:F access display memory locations 4K–68K.

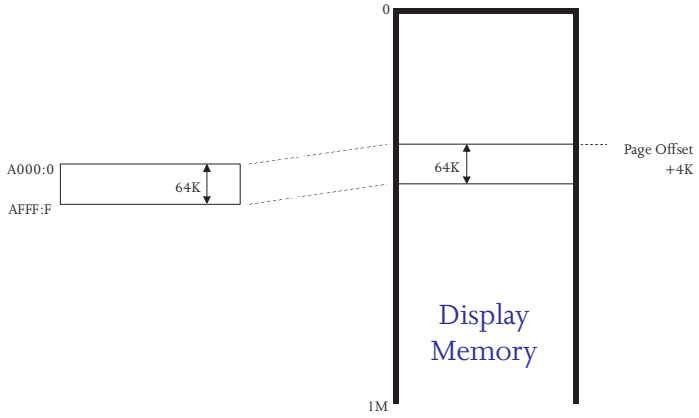
Writing to VGA register 3D5.00, Horizontal Total, resets this register unless autoreset is disabled. Refer to 3D5.1E, "Extended CRTC autoreset," on page 183, for a description of the autoreset feature.

Read/write:	r/w	*Memory offset:*	0C0–0C1h
Default:	0h	*Address index:*	-

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						page offset									

Bits	Description
[9:0]	Page offset, in 4K increments.

Figure 16.6.2 Page offset



16.6.3. Aperture control

Read/write:	r/w	*Memory offset:*	0C2h
Default:	*See below.*	*Address index:*	-

7	6	5	4	3	2	1	0
host XY addressing	*palette access*		*ROM enable*	*ROM access*			*enable alternate PCI ID*
				cnf31			*cnf24*

Bits	Description
[0]	Enable alternate PCI ID. Default =MD[24], where pulldown = 1, open = 0.
[1]	Reserved.
[3:2]	ROM access. 1X = map out ROM. 01 = C000:0 - CFFF:F (64KB). 00 = C000:0 - C7FF:F (32KB). Default: bit [3] = MD[31], where pulldown = 1, open = 0.
[4]	Flash ROM enable. 1 = Writes to C000:0 cause $\overline{\text{ROMWR}}$ and $\overline{\text{ROMEN}}$ to be asserted. 0 = Flash ROM disabled. Default = 0.

Bits	Description
[6:5]	Palette access. 10 = map out palette. Neither writes nor reads take place. 01 = do not shadow palette. Writes and reads are normal. 00 = shadow palette writes. Writes to RAMDAC addresses 3C6:9 are not acknowledged by $\overline{\text{LDEV}}$ (VL) or $\overline{\text{DEVSEL}}$ (PCI) but write does take place. Reads are handled normally. Default = 00.
[7]	Enable host XY addressing. 1 = enabled. 0 = disabled. Default = 0.

16.6.4. Display memory configuration

0C4 bit [5] default = 0 although this bit MUST be set to 1.

Read/write:	r/w	Memory offset:	0C4–0C5h
Default:	See below.	Address index:	-

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				slow DRAM refresh	single cycle page mode	EDO disable	mem 64			! must be 1	dual $\overline{WE}$	dual bank drive		fast $\overline{RAS}$ disable	inter-leaved mem-ory
					cnf12	cnf13	cnf16	cnf17	cnf23		cnf29			cnf22	cnf30

Bits	Description
[0]	Interleaved memory. 1 = ensabled. 0 = disabled. Default = configuration strap MD[30], where pulldown = 1, open = 0.
[1]	Fast $\overline{\text{RAS}}$ disable. Refer to "Page mode DRAM: read/write," on page 72. 1 = fast $\overline{\text{RAS}}$ disabled. 0 = fast $\overline{\text{RAS}}$ enabled. Default = configuration strap MD[22], where pulldown = 1, open = 0.
[2]	Reserved
[3]	Dual bank drive. Set this bit to enable 2MB modes in 3MB BetterHalf configuration. 1 = enable. 0 = diable.

Bits	Description
[4]	Dual WE DRAM select. 1 = multiple WE. 0 = multiple $\overline{CAS}$. Default = configuration strap MD[29], where pulldown = 1, open = 0.
[5]	Enable 128-bit graphics engine access. Default = 0. ! This bit MUST be set to 1.
[6]	Reserved. Default = configuration strap MD[23], where pulldown = 1, open = 0.
[7]	Reserved. Default =configuration strap MD[17], where pulldown = 1, open = 0.
[8]	64-bit memory bus. 1 = 64-bit memory enabled. 0 = 64-bit memory disabled. Default = configuration strap MD[16], where pulldown = 1, open = 0. Normally set by software: 0 = 1MB; 1 = 2MB.
[9]	EDO DRAM disable. 1 = EDO DRAM disabled. 0 = EDO DRAM enabled. Default = configuration strap MD[13], where pulldown = 1, open = 0.
[10]	Enable single cycle page mode. 1 = Single cycle page mode enabled. 0 = default. Default = configuration strap MD[12], where pulldown = 1, open = 0.
[11]	Slow DRAM refresh rate. 1 = Refresh rate: 512 rows in 64 ms. 0 = Refresh rate: 512 rows in 8 ms.

16.6.5. DRAM timing adjust

Use this register to delay sampling time for reads from 64-bit display memory.

This register is relevant only for 64-bit reads. Do not use this in 32-bit mode.

Read/write:		*Memory offset:*	0C7h
Default:		*Address index:*	-

7	6	5	4	3	2	1	0
				timing delay factor			*timing adjust enable*

Bits	Description
[0]	DRAM timing adjust enable 1 = enabled 0 = disabled
[3:1]	DRAM timing delay factor. Nominal delay is 1.2 ns + (0.4 ns * this field)

16.6.6. VGA override

Use this register to force various VGA settings.

Read/write:	r/w	*Memory offset:*	0C8–0C9h
Default:	0h	*Address index:*	-

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			diable VGA I/O	*DRAM refresh disable*	*cursor blink*	*force 3C2*	*force DCLK*	*force graph-ics mode*	*force 8-dot*	*CRTC unlock*	*lock VGA general*	*lock VGA attri-bute con-troller*	*lock VGA graph-ics con-troller*	*lock VGA CRTC*	*lock VGA sequen-cer*

Bits	Description
[0]	Lock VGA sequencer registers 3C5. 1 = locked. 0 = unlocked.
[1]	Lock VGA CRTC registers 3D5.00–24h. 1 = locked. 0 = unlocked.
[2]	Lock VGA graphics controller registers 3CF.00–08h. 1 = locked. 0 = unlocked.
[3]	Lock VGA attribute controller registers 3C0.00–14h. 1 = locked. 0 = unlocked.
[4]	Lock VGA general registers 3BA, 3C2, 3CA, 3CC. 1 = locked. 0 = unlocked.
[5]	Force CRTC 0-7 unlock. 1 = unlocked. 0 = disabled, unlock depends on 3D5.11[7]. This bit overrides VGA register 3D5.11[7], Vertical retrace end.

Bits	Description
[6]	Force 8-dot clock. 1 = 8-dot clock. 0 = disabled, clock depends on 8/9 bit. Clock depends on VGA register 3C5.1[0], Clocking mode.
[7]	Force graphics mode. 1 = graphics mode. 0 = disabled, graphics-text mode depends on VGA settings. Mode depends on VGA registers 3C0.10[0], Mode Control; and 3CF.6[0], Miscellaneous.
[8]	Force DCLK = VCLK. Ignores setting which permits VCLK = 1/2 DCLK. 1 = DCLK = VCLK. 0 = disabled, VCLK depends on VGA settings. DCLK depends on VGA register 3C5.1[3], Clocking Mode.
[9]	Force 3C2[3:2] = 11. If this bit is enabled then 3C2[3:2] treated as 11b regardless of contents. 3C3 is "Item select/miscellaneous output," described on page 124. 1 = 3C2[3:2] = 11b. 0 = disabled.
[10]	Cursor blink. This bit is relevant only in text modes. 1 = cursor blink disabled. 0 = cursor blink enabled.
[11]	DRAM refresh disable. 1 = refresh disabled. 0 = refresh enabled.
[12]	Disable VGA I/O access. 1 = VGA I/O disabled. 0 = VGA I/O enabled.

16.6.7. Host interface

This register detects VL bus or PCI bus host operation and tri-state $\overline{\text{LDEV}}$ operation. This register reflects values latched from configuration straps when reset is deasserted.

Read/write:	r	*Memory offset:*	0CAh
Default:	*See below.*	*Address index:*	-

7	6	5	4	3	2	1	0
				PCI 66/$\overline{33}$	Tri-state $\overline{\text{LDEV}}$		PCI/$\overline{\text{VL}}$
				cnf10	cnf25	cnf26	cnf27

Bits	Description
[0]	Host interface. 1 = PCI bus. 0 = VESA VL-bus. Default = configuration strap MD[27], where pulldown = 1, open = 0.
[1]	Reserved. Default =configuration strap MD[26], where pulldown = 1, open = 0.
[2]	Tri-state $\overline{\text{LDEV}}$. 1 = $\overline{\text{LDEV}}$ output is a synchronous signal which is tri-stated when not actively asserted (normal for PCI configuration). 0 = $\overline{\text{LDEV}}$ is a combinational signal which is always actively driven (normal for VL-bus configurations). Default = configuration strap MD[25], where pulldown = 1, open = 0.
[3]	PCI 66/$\overline{33}$. 1 = PCI 33 MHz clock. 0 = PCI 66 MHz clock. Default = configuration strap MD[10], where pulldown = 1, open = 0.

16.6.8. PCI $\overline{\text{STOP}}$ latency

Use this register to specify the maximum number of clock cycles per data phase before $\overline{\text{STOP}}$ is asserted.

Read/write:	r/w	*Memory offset:*	0CBh
Default:	0h	*Address index:*	-

7	6	5	4	3	2	1	0
addditonal cycles: transadactions after first transaction				*addditonal cycles: first transaction*			

Bits	Description
[3:0]	Additional clock cycles (0–15).over and above the 16 allowed by PCI 2.1 specification, used for the first data transaction.
[7:4]	Additional clock cycles (0–15).over and above the 8 allowed by PCI 2.1 specification, used for subsequent data transactions.

16.6.9. Feature connector control

Read/write:	r/w	*Memory offset:*	0CCh
Default:	*See below.*	*Address index:*	-

7	6	5	4	3	2	1	0
generic FC	*genlock interlace*		*genlock reset*	*genlock enable*	*FC disable*	*FC direction*	*VAFC/$\overline{\text{VSVPC}}$*

Bits	Description
[0]	Feature connector select. 1 = VAFC. 0 = VSVPC. Default = configuration strap MD[15], where pulldown = 1, open = 0.
[1]	Feature connector direction. 1 = in. 0 = out. Default = 0h.
[2]	Feature connector disable. 1 = disabled. 0 = enabled. Default = 0h. Set this bit for TV input.
[3]	Genlock enable. 1 = enabled. 0 = disabled. Default = 0h.
[4]	Genlock reset. 1 = reset V counter only. 0 = reset H and V counters. Default = Undefined.
[6:5]	Genlock interlaced control. 1x = reserved. 0x = auto field detect. Default = Undefined.
[7]	Generic feature connector enable. 1 = enabled. 0 = disabled Default = 0h.

16.6.10. Generic feature connector control

These bits are preserved if mode changes from generic mode to another and back.

This register is relevant only if feature connector mode = generic as set by M0CC[7], "Feature connector control," on page 228.

Read/write:	r/w	*Memory offset:*	0CDh
Default:	Undefined.	*Address index:*	-

7	6	5	4	3	2	1	0
P[7]/VID[7]	P[6]/VID[6]	P[5]/VID[5]	P[4]/VID[4]	P[3]/VID[3]	P[2]/VID[2]	P[1]/VID[1]	P[0]/VID[0]
pin 73	pin 74	pin 75	pin 76	pin 77	pin 78	pin 81	pin 82

Bits	Description
[7:0]	Generic feature connector outputs. These bits reflect pins P[7:0]/VID[7:0], where pulldown = 1, open = 0.

16.6.11. VAFC control

These bits are preserved if mode changes from VAFC to another and back.

This register is relevant only if feature connector mode = VAFC (M0CC[1:0] = 10 or 01).

Read/write:	r/w	*Memory offset:*	0CEh
Default:	Undefined.	*Address index:*	-

7	6	5	4	3	2	1	0
			16-bit FC	FC format	Chromakey	GRDY	DCLK

Bits	Description
[0]	DCLK control. 1 = {DCLK = 1/2 PCLK}. 0 = {DCLK = PCLK}.
[1]	GRDY control. 1 = GRDY matches video window. 0 = GRDY matches $\overline{\text{BLANK}}$.
[2]	Chromakey enable. 1 = enabled. 0 = disabled.
[3]	Feature connector format direct. 1 = enabled. 0 = disabled.
[4]	16-bit feature connector. 1 = enabled. 0 = disabled.

16.6.12. Genlock control

Use this register to enable genlock skew.

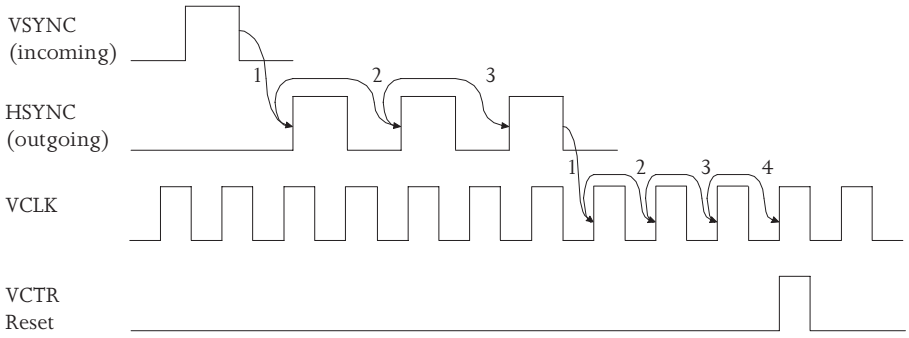
Read/write:	r/w	*Memory offset:*	0CFh
Default:	Undefined.	*Address index:*	-

7	6	5	4	3	2	1	0
genlock skew horizontal				*genlock skew vertical*			

Bits	Description
[3:0]	Genlock skew vertical. 1111 = fifteen lines. ... 0002 = two lines. 0001 = one lines. 0000 = no lines.
[7:4]	Genlock skew horizontal. 1111 = fifteen pixels. ... 0002 = two pixels. 0001 = one pixel. 0000 = no skew.

Figure 16.6.12 Genlock skew

Example: vertical skew = 3, horizontal skew = 4



16.6.13. DPMS/sync control

Use this register to suspend sync signals to the monitor, and to control HSYNC for DDC implementation with VL bus configurations.

Results from reading M0D0[3:6] return pin values, not register contents.

Read/write:	r/w	*Memory offset:*	0D0h
Default:	0h	*Address index:*	-

7	6	5	4	3	2	1	0
	SCL[1]	SDA[1:0]		SCL[0]	tri-state HSYNC	VSYNC suspend	HSYNC suspend

Bits	Description
[0]	DPMS $\overline{\text{HSYNC}}$ suspend. 1 = $\overline{\text{HSYNC}}$ disabled. 0 = $\overline{\text{HSYNC}}$ enabled.
[1]	DPMS $\overline{\text{VSYNC}}$ suspend. 1 = $\overline{\text{VSYNC}}$ disabled. 0 = $\overline{\text{VSYNC}}$ enabled.
[2]	DDC tri-state HSYNC. 1 = enabled. 0 = disabled. This bit is used for VL bus configurations, where dedicated DDC pins are not available.
[3]	SCL control, bit 0 of [1:0]. Bit 1 is 0D0[6]. 11 = drive SCL pin high. 10 = drive SCL pin low. 0x = input/tri-state.
[5:4]	SDA control. 11 = drive SDA pin high. 10 = drive SDA pin low. 0x = input/tri-state. Bit [4] is equivalent of 1FC[16]; refer to "Extended/DAC status," on page 178.
[6]	SCL control, bit 1 of [1:0]. Bit 0 is 0D0[3]. Refer to 0D0[3] for the description of SCL control.

16.6.14. Monitor interlace control

Refer to ProMotion application notes for programming information on interlace.

Writing to VGA register 3D5.00, Horizontal Total, resets this register unless autoreset is disabled. Refer to 3D5.1E, "Extended CRTC autoreset," on page 183, for a description of the autoreset feature.

Read/write:	*special*	*Memory offset:*	0D2h
Default:	0h	*Address index:*	-

7	6	5	4	3	2	1	0
					drive XODD		interlace

Bits	Description
[0]	Interlaced video signal. 1 = interlaced. 0 = non-interlaced.
[1]	Reserved
[2]	Drive XODD pin 1 = drive high. 0 = drive low.

16.6.15. Pixel FIFO request point

Use this register to specify the low water mark where pixel FIFO must read pixels. If set too low, underflow may result. If set too high, performance suffers. Contact Alliance for recommended settings.

Read/write:	r/w	*Memory offset:*	0D4–0D6h
Default:	*See below.*	*Address index:*	-

23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			low priority request point								no page break request point								page break request point				

Bits	Description
[4:0]	High priority request point—page break. Default = 14h.
[7:5]	Reserved
[12:8]	High priority request point—no page break. Default = 14h.
[15:13]	Reserved
[20:16]	Low priority request point. Default = 14h.
[23:21]	Reserved

16.6.16. FIFO underflow

Use this register to determine whether or not the Video FIFO has underflowed since this register was last reset.

Read/write:	r/w	*Memory offset:*	0D8h
Default:	undefined.	*Address index:*	-

7	6	5	4	3	2	1	0
							underflow

Bits	Description
[0]	FIFO underflow. 1 = FIFO underflow. 0 = no underflow since reset.

16.6.17. External signal timing

Use this register to disable two PCI signals, to specify the external VCLK frequency, and the number of $\overline{\text{LDEV}}$ wait states.

Read/write:	r/w	*Memory offset:*	0D9h
Default:	See below.	*Address index:*	-

7	6	5	4	3	2	1	0
PCI lock disable	PCI stop disable	LDEV wait		EPROM_WAIT			

Bits	Description
[3:0]	EPROM_WAIT. EPROM access timing, in MCLKs. $\overline{\text{ROMEN}}$ and $\overline{\text{ROM}}$WR pulse width are determined as follows: If EPROM_WAIT = 0 then pulse width = 4. If EPROM_WAIT ≥1 then pulse width = [5+(EPROM_WAIT–1)∞2]. Default pulse width is [5+(8–1)∞2], or 19_{10} MCLK. For ROM read, data is strobed into ProMotion-aT3D one MCLK cycle before $\overline{\text{ROMEN}}$ is released. Valid data must be presented to the ProMotion controller at [4 + (2 ∞ (EPROM_WAIT – 1))] ∞ MCLK ns after $\overline{\text{ROMEN}}$, minus setup time of 3 ns. Default = 1000b. Alliance recommends this value remain unchanged.
[5:4]	$\overline{\text{LDEV}}$ wait states. This register bit applies only to VL bus applications, and is unused for PCI. Default = 10b.

Bits	Description
[6]	Disable PCI STOP signal. 1 = STOP disabled. 0 = STOP enabled.
[7]	Disable PCI LOCK signal. 1 = LOCK disabled. 0 = LOCK enabled.

16.6.18. Enable extended registers

Use this register to enable additional mappings of existing registers.

Read/write:	r/w	*Memory offset:*	0DBh
Default:	undefined.	*Address index:*	-

7	6	5	4	3	2	1	0
				second linear aperture	coprocessor apertures	linear space	DOS space

Bits	Description
[0]	Enable extended register - DOS space. 1 = enabled. 0 = default.
[1]	Enable extended registers - linear space. 1 = enabled. 0 = default.
[2]	Enable coprocessor apertures. 1 = enabled. 0 = default.
[3]	Enable second lineaer aperture. 1 = enabled. 0 = default.

16.6.19. Bi-endian control

Use this register enable bi-endian operations in ProMotion-aT3D modules.

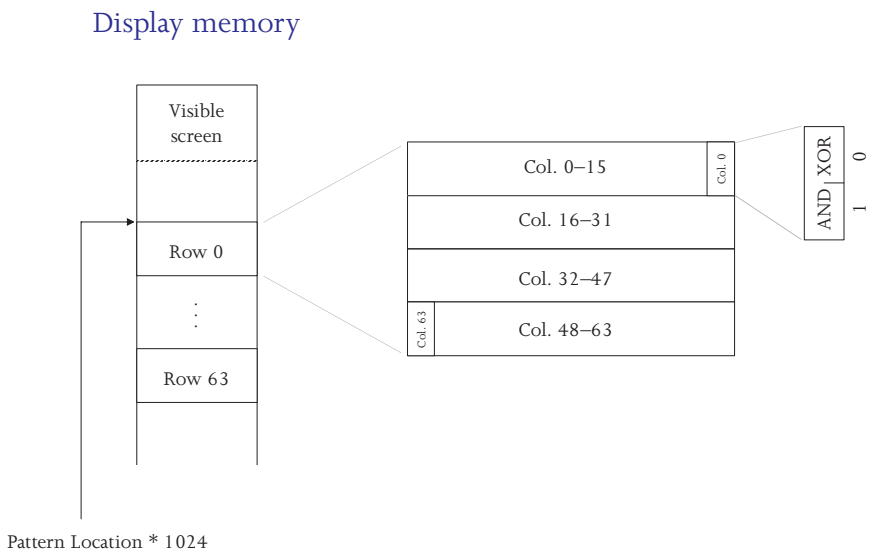
Read/write:	r/w	*Address:*	M0DC–0DDh
Default:	0h	*Address index:*	-

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				TV module		3d module		2d module		Pixel data		host aperture 1		host aperture 0	

Bits	Description
[1:0]	Host aperture 0 11 = reserved. 10 = 32-bit transform. 01 = 16-bit transform. 00 = no transform.
[3:2]	Host aperture 1 11 = reserved. 10 = 32-bit transform. 01 = 16-bit transform. 00 = no transform.
[5:4]	Pixel data module transform control x1 = transform possible. x0 = transform disabled.
[7:6]	2D graphics engine module transform control x1 = transform possible. x0 = transform disabled.
[9:8]	3D graphics engine transform control x1 = transform possible. x0 = transform disabled.
[11:10]	TV module transform control x1 = transform possible. x0 = transform disabled.
[12:15]	Reserved

16.7 Hardware cursor registers

The hardware cursor is a 64 ∞ 64 cursor at 2 bits per pixel. It is stored at any kilobyte aligned address in off-screen display memory. The cursor pattern is stored as a linear strip; the first 16 bytes represent the top row of the pattern.



Within each 16-byte row, the first 32-bit dword represents the leftmost 16 pixels. Within each dword, the low-order 2 bits represent the leftmost pixel. Of these two pixels, the low-order bit represents the XOR plane and the higher bit represents the AND plane.

Hardware cursor colors are 8-bit registers. In direct color modes they represent 3:3:2 RGB as with the ProMotion-aT3D 8-bit direct model. Full white is available.

The hardware cursor pattern display position and origin registers unit is pixels. The hardware cursor pattern base address register unit is kilobytes (KB). The pattern must be KB aligned.

16.7.1. Hardware cursor control

Use this register to enable the hardware cursor.

Writing to VGA register 3D5.00, Horizontal Total, resets this register unless autoreset is disabled. Refer to 3D5.1E, "Extended CRTC autoreset," on page 183, for a description of the autoreset feature.

The ProMotion cursor bit pattern of 00 corresponds to all 0s for the software cursor defined by Microsoft Windows 3.x. A pattern of 10 is always transparent, and a pattern of 11 inverts the background unless the hardware cursor 3-color mode bit is set, in which case cursor color 3 is used.

Figure 16.7.1 Hardware cursor pattern

2-bit pixel cursor	Software	Hardware 2-color	Hardware 3-color
11	inverse		color 3
10	transparent		
01	all 1s	color 2	
00	all 0s	color 1	

inverse is not available when the cursor is over the vWindow; 3-color mode is always active. Any cursor pixel with a cursor pattern of 11 uses cursor color 3 when over the vWindow.

For direct color modes each of the 8-bit cursor color registers specifies a direct color in 3:3:2 mode, with the 2 low-order register bits specifying blue.

All cursor pattern, location, and position registers are internally synchronized to the next VSYNC, so there is no need to wait for retrace to update any register. Cursor color registers are not synchronized.

Read/write:	r/w	*Memory offset:*	140h
Default:	*See below.*	*Address index:*	-

7	6	5	4	3	2	1	0
					full color	3 color	cursor enable

Bits	Description
[0]	Hardware cursor enable. When this bit is set the cursor pattern is displayed. Any write to the VGA CRTC registers index 0-17 automatically resets this bit to 0. (VGA mode set does not begin with a cursor enabled.)
[1]	Hardware cursor 3-color mode. When this bit is set a cursor pattern of 11 is displayed using cursor color 3 instead of inversion. Any write to the VGA CRTC registers index 0-17 automatically resets this bit to 0. (VGA mode set does not begin with a cursor enabled.)
[2]	Hardware cursor full color enable.

16.7.2. Hardware cursor color 1

Use hardware cursor color registers 1–3 to specify the color of the hardware cursor.

Read/write:	r/w	*Memory offset:*	141h
Default:	*Undefined.*	*Address index:*	-

7	6	5	4	3	2	1	0
cursor color 1							

Bits	Description
[7:0]	Hardware cursor color 1

16.7.3. Hardware cursor color 2

Use hardware cursor color registers 1–3 to specify the color of the hardware cursor.

Read/write:	r/w	*Memory offset:*	142h
Default:	Undefined.	*Address index:*	-

7	6	5	4	3	2	1	0
cursor color 2							

Bits	Description
[7:0]	Hardware cursor color 2

16.7.4. Hardware cursor color 3

Use hardware cursor color registers 1–3 to specify the color of the hardware cursor.

Read/write:	r/w	*Memory offset:*	143h
Default:	Undefined.	*Address index:*	-

7	6	5	4	3	2	1	0
cursor color 3							

Bits	Description
[7:0]	Hardware cursor color

16.7.5. Hardware cursor pattern base address

Use this register to specify the offset in display memory of the cursor pattern data. This location is in linear kilobytes. Therefore, a register value of 001h represents display memory locations 00400–007FFh.

This register points to the top-left corner of the pattern regardless of the cursor position. Cursor patterns may be changed either by changing the pattern in display memory or simply pointing to a new pattern.

To avoid sparkle when changing cursor patterns, write the new pattern elsewhere in memory and set the base address pointer to the new pattern location. This register is synchronized internally to the new video frame.

Read/write:	r/w	*Memory offset:*	144–145h
Default:	Undefined.	*Address index:*	-

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				pattern location											

Bits	Description
[11:0]	Hardware cursor pattern location, in KB, and must be Kbyte aligned.

16.7.6. Hardware cursor display position X

Use this register to specify the left edge of the rectangle where the cursor pattern is displayed. Where a cursor has a non-zero hotspot, the driver must adjust the display position registers accordingly for the desired display.

These values are always non-negative; use "Hardware cursor display offset Y," on page 242, to specify cursor straddling the edge(s) of the screen.

Read/write:	r/w	*Memory offset:*	148–149h
Default:	Undefined.	*Address index:*	-

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				display position x											

Bits	Description
[11:0]	Hardware cursor display position X, in pixels. This value should be set to zero when the cursor box straddles the left edge of the screen.

16.7.7. Hardware cursor display position Y

Use this register to specify the top edge of the rectangle where the cursor pattern is displayed

Read/write:	r/w	*Memory offset:*	14A–14Bh
Default:	Undefined.	*Address index:*	-

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				display position y											

Bits	Description
[11:0]	Hardware cursor display position Y, in pixels. This value should be set to zero when the cursor box straddles the top edge of the screen.

16.7.8. Hardware cursor display offset X

Use this register to specify a starting point within the pattern where display is to begin. This register is zero unless the cursor straddles the left edge of the screen.

Read/write:	r/w	*Memory offset:*	14Ch
Default:	Undefined.	*Address index:*	-

7	6	5	4	3	2	1	0
		cursor offset x					

Bits	Description
[5:0]	Hardware cursor offset X.

Use the following sample code to display a cursor with hotspot (xc,yc) at screen position (xs,ys):

```
xp = xs - xc
yp = ys - yc
if (xp>=0) then {DispPosX=xp; DispOffX=0}
      else {DispPosX=0; DispOffX=-xp}
if (yp>=0) then {DispPosY=yp; DispOffY=0}
      else {DispPosY=0; DispOffY=-yp}
```

Figure 16.7.8 Hardware cursor display offset

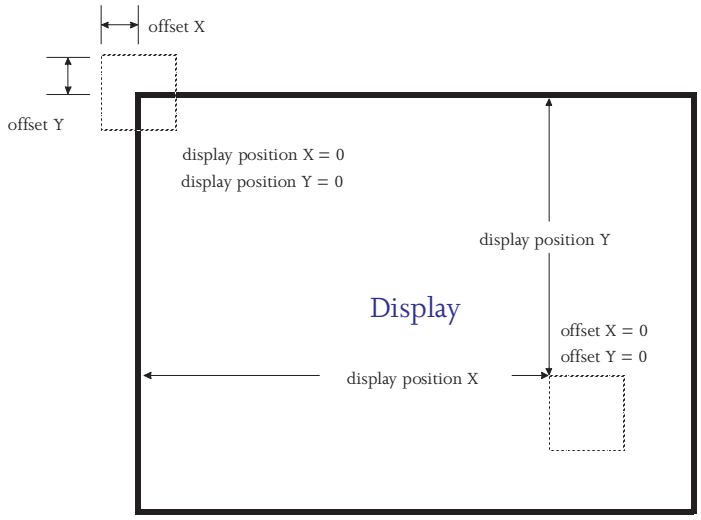


Diagram elements not to scale.

16.7.9. Hardware cursor display offset Y

Use this register to specify a starting point within the pattern where display is to begin. This register is zero unless the cursor straddles the top edge of the screen.

Read/write:	r/w	*Memory offset:*	14Dh
Default:	Undefined.	*Address index:*	-

7	6	5	4	3	2	1	0
		cursor offset y					

Bits	Description
[5:0]	Hardware cursor offset Y.

Use the following sample code to display a cursor with hotspot (xc,yc) at screen position (xs,ys):

```
xp = xs - xc
yp = ys - yc
if (xp>=0) then {DispPosX=xp; DispOffX=0}
      else {DispPosX=0; DispOffX=-xp}
if (yp>=0) then {DispPosY=yp; DispOffY=0}
      else {DispPosY=0; DispOffY=-yp}
```

Figure 16.7.9 Hardware cursor display offset

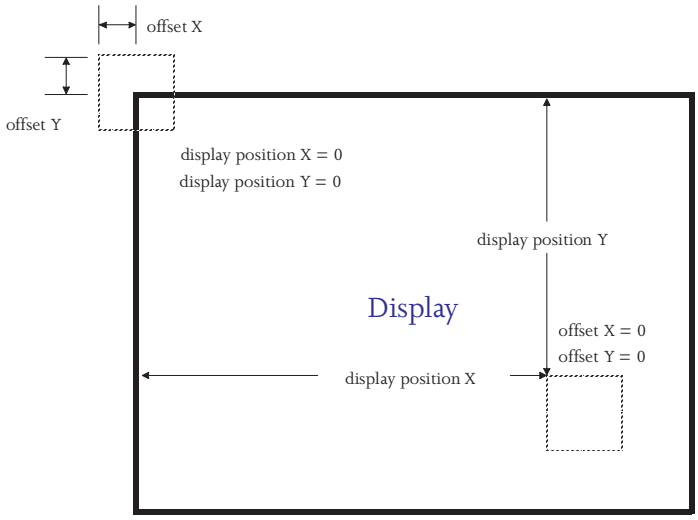


Diagram elements not to scale.

16.8 PCI configuration registers

See also extended register M0CBh, "PCI STOP latency," described on page 228.

16.8.1. PCI vendor ID

Read/write:	r	*Memory offset:*	180–181h
Default:	1142h.	*PCI I/O:*	00–01h

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
vendor ID															

Bits	Description
[15:0]	Vendor ID (1142h = Alliance Semiconductor Corporation).

16.8.2. PCI device ID

The ProMotion-aT3D device ID is the same for all versions of the AT3D. Differentiation may be made using revision number, "PCI revision ID," described on page 246.

Read/write:	r	*Memory offset:*	182–183h
Default:	643Dh.	*PCI I/O:*	02–03h

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
device ID															

Bits	Description
[15:0]	Device ID (643Dh).

16.8.3. PCI command

Read/write:	r/w	*Memory offset:*	184–185h
Default:	0h	*PCI I/O*	04–05h

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
										snoop				mem space	I/O space

Bits	Description
[0]	I/O space. 1 = enabled. 0 = disabled. Normally this bit is set.
[1]	Memory space. 1 = enabled. 0 = disabled. Normally this bit is set.
[4:2]	Reserved.
[5]	VGA palette snooping. 1 = enabled. 0 = disabled.

16.8.4. PCI status

Use this register to specify the detection of parity errors, and to determine DEVSEL timing.

Read/write:	r	*Memory offset:*	186–187h
Default:	40h.	PCI I/O	06–07h

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
parity error					DEVSEL timing										

Bits	Description
[8:0]	Reserved.
[10:9]	DEVSEL timing (read only). Value = binary 01. This corresponds to medium speed address decode in PCI. Refer to the PCI specification for more information on DEVSEL timing.
[14:11]	Reserved.
[15]	Detected parity error. 1 = error detected. 0 = no parity error.

16.8.5. PCI revision ID

Read/write:	r	*Memory offset:*	188h
Default:	*See below.*	*PCI I/O:*	08h

7	6	5	4	3	2	1	0
revision ID							

Bits	Description
[7:0]	Revision ID (implementation dependant). Refer to "ProMotion stepping information," on page 316 for information on detecting ProMotion controllers.

16.8.6. Class code

PCI motherboard BIOS reads this register to determine the type of device. The read-only value returned is 300h for a graphics controller. Refer to the PCI specification for more information on class code.

Read/write:	r	*Memory offset:*	189–18Bh
Default:	300h	*PCI I/O:*	09–0Bh

23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																class code							

Bits	Description
[7:0]	Class code. (300h)
[24:8]	Reserved

16.8.7. Cache line size

This PCI functionality is not supported by ProMotion-aT3D.

Read/write:	r	*Memory offset:*	18Ch
Default:	0	*PCI I/O:*	0Ch

7	6	5	4	3	2	1	0

Bits	Description
[7:0]	cache line size.

16.8.8. Latency timer

This PCI functionality is not supported by ProMotion-aT3D.

Read/write:	r	*Memory offset:*	18Dh
Default:	0h	*PCI I/O:*	0Dh

7	6	5	4	3	2	1	0

Bits	Description
[7:0]	latency timer.

16.8.9. Header type

This PCI functionality is not supported by ProMotion-aT3D.

Read/write:	r	*Memory offset:*	18Eh
Default:	0h	*PCI I/O:*	0Eh

7	6	5	4	3	2	1	0

Bits	Description
[7:0]	header type.

16.8.10. BIST

This PCI functionality is not supported by ProMotion-aT3D.

Read/write:	r	*Memory offset:*	18Fh
Default:	0	*PCI I/O:*	0Fh

7	6	5	4	3	2	1	0

Bits	Description
[7:0]	BIST.

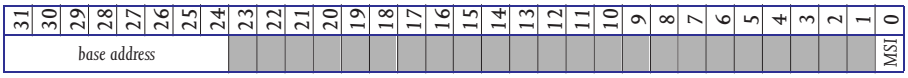
16.8.11. PCI memory base address

Use this register to specify the base address of the linear frame buffer.

If ProMotion memory mapped registers are mapped into linear (flat) memory space (rather than a VGA aperture), this base address must be set before setting 3C5.1B, "Remap control," on page 168.

This register can also be set via an extended I/O address, 3C5.1A.

Read/write:	r/w	*Memory offset:*	190–193h
Default:	0h	*PCI I/O:*	10h



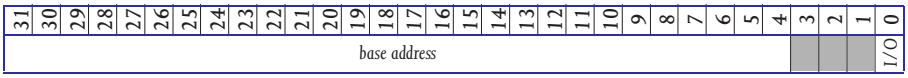
Bits	Description
[0]	Memory space indicator (read only). This bit always returns 0, indicating ProMotion requests a reserved area in memory space.
[23:1]	Reserved.
[31:24]	Base address.

16.8.12. PCI I/O base address

Use this register to allocate area in I/O space for use by ProMotion.

ProMotion xx10 controllers did not reserve I/O space.

Read/write:	*See below.*	*Memory offset:*	194–197h
Default:	*See below.*	*PCI I/O:*	14h



Bits	Description
[0]	I/O space indicator (read only). This bit always returns 1, indicating ProMotion requests a reserved area in I/O space.
[3:1]	Reserved (read only). Default = 000b.
[31:4]	Base address (r/w). Typically written by the system (at bootup) and read by BIOS/ driver software. Default = undefined.

16.8.13. Subsystem vendor ID

Read/write:	r	*Memory offset:*	1AC–1ADh
Default:	0	*PCI I/O:*	2C–2D

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
subsystem ID															

Bits	Description
[15:0]	Subsystem ID 0h.

16.8.14. Subsystem ID

Read/write:	r	*Memory offset:*	1AE–1AFh
Default:	0h	*PCI I/O:*	2E–2Fh

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
subsystem vendor ID															

Bits	Description
[15:0]	Subsystem vendor ID 0h.

16.8.15. Expansion ROM base address

Use this register to enable and to specify the address for on-board ROM. Refer to the PCI specification for more information.

Read/write:	r/w	*Memory offset:*	1B0–1B3h
Default:	*See below.*	*PCI I/O:*	30–33h

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ROM base address																															*ROM*

Bits	Description
[0]	ROM address enable. 1 = ProMotion-aT3D responds to ROM at the address specified in [31:16]. 0 = ProMotion-aT3D does not respond to ROM at any address. Default = 0.

Bits	Description
[15:1]	Reserved.
[31:16]	ROM base address, bits [31:16]. Set this for on-board ROM. Default = Ch.

16.8.16. Interrupt line

Use this register to specify which input of the system interrupt controller(s) the device interrupt pin is connected to. This register is usually used to determine priority and vector information. Refer to the PCI specification for more information on PCI interrupts.

Read/write:	r/w	**Memory offset:**	1BCh
Default:	0h.	**PCI I/O:**	3Ch

7	6	5	4	3	2	1	0
interrupt line							

Bits	Description
[7:0]	Interrupt line.

16.8.17. Interrupt pin

This read-only register pre-loads at power-up/reset with the status of configuration strap MD[11], INTPIN.

Board vendors may configure the chip to request a PCI interrupt, which is appropriate for a fully Plug and Play system. However, this may cause conflicts in systems that are not PnP compliant. Board vendors may choose to disable INTPIN and claim no PCI interrupt level, since many VGA systems do not use VSYNC interrupt.

Read/write:	r	**Memory offset:**	1BDh
Default:	1h.	**PCI I/O:**	3Dh

7	6	5	4	3	2	1	0
							interrupt
							cnf11

Bits	Description	
[0]	Status of configuration strap MD[11].	
	1 =	PCI interrupt requested (pin pulled down).
	0 =	no interrupt requested (pin open).

16.8.18. Minimum grant

This read only register indicates the duration of the burst period ProMotion needs to gain access to the PCI bus. The unit is 1/4 of a microsecond.

Read/write:	r	*Memory offset:*	1BEh
Default:	0h.	*PCI I/O:*	3Eh

7	6	5	4	3	2	1	0
minimum grant							

Bits	Description
[7:0]	Minimum grant. Default = 0h (no requirement).

16.8.19. Maximum grant

This read only register indicates how often ProMotion needs to gain access to the PCI bus.

Read/write:	r	*Memory offset:*	1BFh
Default:	0h.	*PCI I/O:*	3Fh

7	6	5	4	3	2	1	0
maximum grant							

Bits	Description
[7:0]	Maximum grant. Default = 0h (no requirement).

16.8.20. Enable write subsystem ID

Use this register enable overwriting of the normally read-only PCI subsystem ID register.

Read/write:	see below	*Address:*	M1C0h
Default:	see below	*PCI I/O:*	

7	6	5	4	3	2	1	0
					dual PCI ID	subsystem device ID	subsystem vendor ID
					MD[24]		

Bits	Description
[0]	Subsystem vendor ID. Writable, reset = 0. 1 = enabled. 0 = disabled.
[1]	Subsystem device ID Writable, reset = 0. 1 = enabled. 0 = disabled.
[2]	Enable dual PCI device IDs. 1 = (pin pulled down). 0 = (pin open). Default = is configuration strap MD[24] where pulldown = 1, open = 0.

16.9 DAC registers

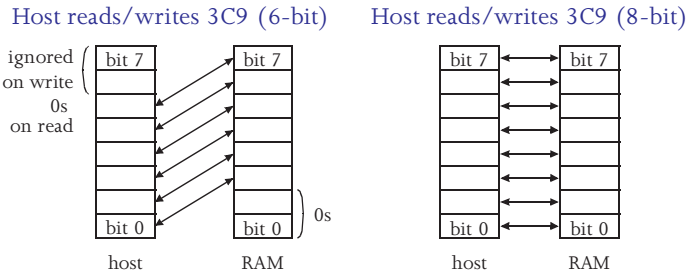
16.9.1. Color correction

Use this register to specify color (gamma) correction and palette RAM settiings.

Writing to VGA register 3D5.00, Horizontal Total, resets this register unless autoreset is disabled. Refer to 3D5.1E, "Extended CRTC autoreset," on page 183, for a description of the autoreset feature.

Read/write:	r/w	*Memory offset:*	0E0h
Default:	Undefined.	*Address index:*	-

7	6	5	4	3	2	1	0
		host palette	host RAM	vWindow palette		desktop palette	

Bits	Description
[1:0]	Desktop color correction. 11 = reserved. 10 = use "Secondary palette registers 0–31," on page 165, for correction. 01 = use "Primary palette registers 0–255," on page 165, for correction. 00 = no correction.
[3:2]	vWindow color correction. 11 = reserved. 10 = use "Secondary palette registers 0–31," on page 165, for correction. 01 = use "Primary palette registers 0–255," on page 165, for correction. 00 = no correction.
[4]	Host RAM data width. 1 = 8-bit. Host reads all 8 bits of VGA ragister 3C9, Palette RAM data. 0 = 6-bit. Host reads 6 low-order bits of 3C9 as 6 high-order bits. 
[5]	Host palette select. 1 = use "Secondary palette registers 0–31," on page 165, for correction. 0 = use "Primary palette registers 0–255," on page 165, for correction. ☞ This bit determines the palette for host I/O and not for display.
[6]	Reserved.

16.9.2. DAC control

Use this register to specify ProMotion-aT3D internal DAC settings.

Writing to VGA register 3D5.00, Horizontal Total, resets this register unless autoreset is disabled. Refer to 3D5.1E, "Extended CRTC autoreset," on page 183, for a description of the autoreset feature.

Read/write:	r/w	*Memory offset:*	0E4h
Default:	Undefined.	*Address index:*	-

7	6	5	4	3	2	1	0
				DAC power	boost		blanking

Bits	Description
[0]	Blanking pedestal enable. 1 = enabled. 0 = disabled.
[1]	Reserved.
[2]	Overcurrent boost. 1 = minimum 1. 0 = minimum 0. This bit determines whether the minimimum overcurrent increment is 0 or 1. Refer to "Overcurrent red," on page 254, for a description of how to use this bit.
[3]	DAC power off. 1 = power off. 0 = power on.

16.9.3. Overcurrent red

Use overcurrent registers 0E5, 0E6, and 0E7h to specify brightness and tint within the vWindow. Overcurrent registers offer separate control of red, green, and blue tints, respectively, within the vWindow, without altering the color of the desktop.

Use identical R = G = B overcurrent levels for altering overall brightness; use dissimilar red, green, and blue levels for altering tint.

Each tint value determines extra current added to that color's brightness in the DAC, in increments of 8 LSBs. The default range is 0–7 increments. ProMotion's *overcurrent boost* feature adds 8 LSBs to all three colors, for a boosted range of 1–8 increments. Enable overcurrent boost with 0E4[2], "DAC control," on page 254.

Maximum tint (with boost enabled) produces an overall 25% boost to brightness: 8 increments multiplied by 8 LSBs equals 64 LSBs added to a base brightness value of 255 LSBs.

Linear current change does not necessarily give the perception of linear color change from the display.

Alliance recommends using only lower increments of boost (0–4) to prevent overdriving monitors which lack protective circuitry.

Read/write:	r/w	*Memory offset:*	0E5h
Default:	Undefined.	*Address index:*	-

7	6	5	4	3	2	1	0
		red boost					

Bits	Description
[5:3]	Red vWindow boost.
[2:0]	Reserved.

16.9.4. Overcurrent green

Use overcurrent registers to specify brightness and tint within the vWindow. Refer to "Overcurrent red," on page 254. for a discuusion of overcurrent registers.

Alliance recommends using only lower increments of boost (0–4) to prevent overdriving monitors which lack protective circuitry.

Read/write:	r/w	*Memory offset:*	0E6h
Default:	Undefined.	*Address index:*	-

7	6	5	4	3	2	1	0
		green boost					

Bits	Description
[5:3]	Green vWindow boost.
[2:0]	Reserved.

16.9.5. Overcurrent blue

Use overcurrent registers to specify brightness and tint within the vWindow. Refer to "Overcurrent red," on page 254. for a discuusion of overcurrent registers.

Alliance recommends using only lower increments of boost (0–4) to prevent overdriving monitors which lack protective circuitry.

Read/write:	r/w	Memory offset:	0E7h
Default:	Undefined.	Address index:	-

7	6	5	4	3	2	1	0
		blue boost					

Bits	Description
[5:3]	Blue vWindow boost.
[2:0]	Reserved.

16.10 Clock registers and formulas

Generate MCLK and VCLK rates using formulas a–e below with the values in the following register bits as variables.

Variable	*Range*	*MCLK*	*Programmable VCLK*	*VCLK default 0*	*VCLK default 1*
Numerator (N)	8† to 127	0E8 [22:16]	0EC [22:16]	0F0 [22:16]	0F4 [22:16]
Denominator (M)	1 to 5	0E8 [14:8]	0EC [14:8]	0F0 [14:8]	0F4 [14:8]
Postscaler (L)	0 to 3	0E8 [3:2]	0EC [3:2]	0F0 [3:2]	0F4 [3:2]

† = Alliance recommends values from 8–127 although the full range is 0–127.

Alliance recommends using the lowest feasable value of M. Higher values decrease output clock stability.

a. Clock frequency: $F_{OUT} = \frac{(N+1)(F_{REF})}{(M+1)(2^L)}$

b. F_{REF} range: 8 MHz–20 MHz (Alliance strongly recommends 14.318 MHz.)

c. VCO range: 185 MHz–370 MHz

d. $F_{VCO} = (F_{OUT})(2^L)$

e. $\frac{F_{VCO}}{(N+1)}$ and $\frac{F_{REF}}{(M+1)}$ range > 2mHz

Setting VCO frequency:

1. Determine the integer value of L which yields an acceptable VCO frequency:

$$185\ MHz < (F_{OUT})(2^L) < 370\ MHz$$

$$\log_2(185\ MHz) - \log_2(F_{OUT}) < L < \log_2(370\ MHz) - \log_2(F_{OUT})$$

2. Determine values of M and N which generate F_{VCO} within 0.5% of desired frequency. Any values which yield the correct ratio and which satisfy (e) above are acceptable.

$$F_{VCO} = \frac{(N+1)}{(M+1)}(F_{REF})$$

$$\frac{(N+1)}{(M+1)} = \frac{F_{VCO}}{F_{REF}} = \frac{(F_{OUT})(2^L)}{F_{REF}}$$

3. Visually test output PLL parameters at room, cold, and hot termperature to verify stability.

16.10.1. MCLK control

Use this register to specify MCLK settings.

Read/write:	r/w.	*Memory offset:*	0E8h
Default:	*See below.*	*Address index:*	-

7	6	5	4	3	2	1	0
speed	F			L		*power off*	*bypass*
						cnf21	*cnf21*

Bits	Description
[0]	MCLK bypass. 1 = bypass MCLK. 0 = enable MCLK. Default = configuration strap MD[21], where pulldown = 1, open = 0. Strap MD[21] determines four bits of the ProMotion controller: 0E8[1:0] and 0EC[1:0].
[1]	MCLK power off. 1 = power off. 0 = power on. Default = configuration strap MD[21], where pulldown = 1, open = 0. Strap MD[21] determines four bits of the ProMotion controller: 0E8[1:0] and 0EC[1:0].
[3:2]	MCLK postscaler (L). MCLK divided by 2^L. 11 = Post divide MCLK VCO frequency by 8. 10 = Post divide MCLK VCO frequency by 4. 01 = Post divide MCLK VCO frequency by 2. 00 = Post divide MCLK VCO frequency by 1. Default = 00b.
[6:4]	MCLK frequency range (F). Contact Alliance for proper values for F. 111 = MCLK range 7. 110 = MCLK range 6. 101 = MCLK range 5. 100 = MCLK range 4. 011 = MCLK range 3. 010 = MCLK range 2. 001 = MCLK range 1. 000 = MCLK range 0. Default = 000b.
[7]	MCLK high speed. 1 = Speed programmed by 0E8 [6:4], below. 0 = 40 MHz. Default = 0.

16.10.2. MCLK denominator

Use this register to specify MCLK denominator.

Read/write:	r/w.	*Memory offset:*	0E9h
Default:	undefined	*Address index:*	-

7	6	5	4	3	2	1	0
	MCLK denominator (M).						

Bits	Description
[6:0]	MCLK denominator (M).
[7]	reserved

16.10.3. MCLK numerator

Use this register to specify MCLK numerator.

Read/write:	r/w.	*Memory offset:*	0EAh
Default:	undefined	*Address index:*	-

7	6	5	4	3	2	1	0
	MCLK numerator (N).						

Bits	Description
[6:0]	MCLK numerator (N).
[7]	reserved

16.10.4. VCLK control

Use this register to specify programmable VLCK settings.

Read/write:	r/w	*Memory offset:*	0ECh
Default:	Undefined.	*Address index:*	-

7	6	5	4	3	2	1	0
resync	F			L		power off cnf21	bypass cnf21

Bits	Description
[0]	VCLK bypass. 1 = bypass VCLK. 0 = enable VCLK. Default = configuration strap MD[21], where pulldown = 1, open = 0. Strap MD[21] determines four bits of the ProMotion controller: 0E8[1:0] and 0EC[1:0].
[1]	VCLK power off. 1 = power off. 0 = power on. Default = configuration strap MD[21], where pulldown = 1, open = 0. Strap MD[21] determines four bits of the ProMotion controller: 0E8[1:0] and 0EC[1:0].
[3:2]	VCLK postscaler (L). 11 = Post divide VCLK VCO by 8. 10 = Post divide VCLK VCO by 4. 01 = Post divide VCLK VCO by 2. 00 = Post divide VCLK VCO by 1. Default = undefined.
[6:4]	VCLK frequency range (F). Contact Alliance for proper values for F. 111 = VCLK range 7. 110 = VCLK range 6. 101 = VCLK range 5. 100 = VCLK range 4. 011 = VCLK range 3. 010 = VCLK range 2. 001 = VCLK range 1. 000 = VCLK range 0. Default = 100b.
[7]	VCLK PLL resync. ! Upon mode change, after other settings are complete write 0 then write 1. Default = 0.

16.10.5. VCLK denominator

Use this register to specify VCLK denominator.

Read/write:	r/w	*Memory offset:*	0EDh
Default:	Undefined.	*Address index:*	-

7	6	5	4	3	2	1	0
	M						

Bits	Description
[6:0]	VCLK denominator (M).
[7]	Reserved.

16.10.6. VCLK numerator

Use this register to specify VCLK numerator.

Read/write:	r/w	*Memory offset:*	0EEh
Default:	*Undefined.*	*Address index:*	-

7	6	5	4	3	2	1	0
	N						

Bits	Description
[6:0]	MCLK numerator (N).
[7]	reserved

16.10.7. VCLK default 0 control

Use this register to specify default 0 settings for VCLK. Specify use of VCLK default 0 with VGA register 3C2 [3:2], Item select/miscellaneous output.

Read/write:	r/w	*Memory offset:*	0F0h
Default:	*See below.*	*Address index:*	-

7	6	5	4	3	2	1	0
	F			L		0	0

Bits	Description
[0]	Reserved. ! This bit must be set to 0.
[1]	Reserved. ! This bit must be set to 0.
[3:2]	VCLK default 0 postscaler (L). 11 = Post divide VCLK VCO by 8. 10 = Post divide VCLK VCO by 4. 01 = Post divide VCLK VCO by 2. 00 = Post divide VCLK VCO by 1. Default = Undefined.

Bits	Description
[6:4]	VCLK default 0 frequency range (F). Contact Alliance for proper values for F. 111 = VCLK range 7. 110 = VCLK range 6. 101 = VCLK range 5. 100 = VCLK range 4. 011 = VCLK range 3. 010 = VCLK range 2. 001 = VCLK range 1. 000 = VCLK range 0. Default = 100b.
[7]	Reserved.

16.10.8. VCLK default 0 denominator

Use this register to specify default 0 settings for VCLK. Specify use of VCLK default 0 with VGA register 3C2 [3:2], Item select/miscellaneous output.

Read/write:	r/w	*Memory offset:*	0F1h
Default:	*See below.*	*Address index:*	-

7	6	5	4	3	2	1	0
	M						

Bits	Description
[6:0]	VCLK default 0 denominator (M).
[7]	Reserved.

16.10.9. VCLK default 0 numerator

Use this register to specify default 0 settings for VCLK. Specify use of VCLK default 0 with VGA register 3C2 [3:2], Item select/miscellaneous output.

Read/write:	r/w	*Memory offset:*	0F2h
Default:	*See below.*	*Address index:*	-

7	6	5	4	3	2	1	0
	N						

Bits	Description
[6:0]	VCLK default 0 numerator (N).
[7]	Reserved.

16.10.10. VCLK default 1 control

Use this register to specify default 1 settings for VCLK. Specify use of VCLK default 1 with VGA register 3C2 [3:2], Item select/miscellaneous output.

Read/write:	r/w	*Memory offset:*	0F4–0F7h
Default:	Undefined.	*Address index:*	-

7	6	5	4	3	2	1	0
	F			L			

Bits	Description
[0]	Reserved. ! This bit must be set to 0.
[1]	Reserved. ! This bit must be set to 0.
[3:2]	VCLK default 1 postscaler (L). 11 = Post divide VCLK VCO by 8. 10 = Post divide VCLK VCO by 4. 01 = Post divide VCLK VCO by 2. 00 = Post divide VCLK VCO by 1. Default = Undefined.
[6:4]	VCLK default 1 frequency range (F). Contact Alliance for proper values for F. 111 = VCLK range 7. 110 = VCLK range 6. 101 = VCLK range 5. 100 = VCLK range 4. 011 = VCLK range 3. 010 = VCLK range 2. 001 = VCLK range 1. 000 = VCLK range 0. Default = 100b.
[7]	Reserved.

16.10.11. VCLK default 1 denominator

Use this register to specify default 1 settings for VCLK. Specify use of VCLK default 1 with VGA register 3C2 [3:2], Item select/miscellaneous output.

Read/write:	r/w	*Memory offset:*	0F4–0F7h
Default:	Undefined.	*Address index:*	-

7	6	5	4	3	2	1	0
	M						

Bits	Description
[6:0]	VCLK default 1 denominator (M).
[7]	Reserved.

16.10.12. VCLK default 1 numerator

Use this register to specify default 1 settings for VCLK. Specify use of VCLK default 1 with VGA register 3C2 [3:2], Item select/miscellaneous output.

Read/write:	r/w	*Memory offset:*	0F4–0F7h
Default:	Undefined.	*Address index:*	-

7	6	5	4	3	2	1	0
	N						

Bits	Description
[6:0]	VCLK default 1 numerator (N).
[7]	Reserved.

16.11 General purpose I/O registers

16.11.1. GPIO control

Use this register to override normal function of GPIO pins.

Read/write:	r/w	*Address:*	M1F0h
Default:	Undefined.	*Address index:*	-

7	6	5	4	3	2	1	0
GPIO 7	GPIO 6	GPIO 5	GPIO 4	GPIO 3	GPIO 2	GPIO 1	GPIO 0

Bits	Description
[0]	GPIO pin 0. 1 = enabled 0 = disabled
[1]	GPIO pin 1. 1 = enabled 0 = disabled
[2]	GPIO pin 2. 1 = enabled 0 = disabled
[3]	GPIO pin 3. 1 = enabled 0 = disabled
[4]	GPIO pin 4. 1 = enabled 0 = disabled
[5]	GPIO pin 5. 1 = enabled 0 = disabled
[6]	GPIO pin 6. 1 = enabled 0 = disabled
[7]	GPIO pin 7. 1 = enabled 0 = disabled

16.11.2. GPIO direction

Use this register to enable output on the GPIO pins. This register is relevant only if you have set GPIO control.

Read/write:	r/w	*Address:*	M1F1h
Default:	Undefined.	*Address index:*	-

7	6	5	4	3	2	1	0
GPIO 7	GPIO 6	GPIO 5	GPIO 4	GPIO 3	GPIO 2	GPIO 1	GPIO 0

Bits	Description
[0]	GPIO pin 0 direction. 1 = output 0 = input
[1]	GPIO pin 1 direction. 1 = output 0 = input
[2]	GPIO pin 2 direction. 1 = output 0 = input
[3]	GPIO pin 3 direction. 1 = output 0 = input
[4]	GPIO pin 4 direction. 1 = output 0 = input
[5]	GPIO pin 5 direction. 1 = output 0 = input
[6]	GPIO pin 6 direction. 1 = output 0 = input
[7]	GPIO pin 7 direction. 1 = output 0 = input

16.11.3. GPIO level

Use this register to set drive level for GPIO pins that are configured as ouputs via "GPIO direction," on page 265..

Read/write:	r/w	*Address:*	M1F2h
Default:	Undefined.	*Address index:*	-

7	6	5	4	3	2	1	0
GPIO 7	GPIO 6	GPIO 5	GPIO 4	GPIO 3	GPIO 2	GPIO 1	GPIO 0

Bits	Description
[0]	GPIO pin 0 level. 1 = drive pin high 0 = drive pin low
[1]	GPIO pin 1 level. 1 = drive pin high 0 = drive pin low
[2]	GPIO pin 2 level. 1 = drive pin high 0 = drive pin low
[3]	GPIO pin 3 level. 1 = drive pin high 0 = drive pin low
[4]	GPIO pin 4 level. 1 = drive pin high 0 = drive pin low
[5]	GPIO pin 5 level. 1 = drive pin high 0 = drive pin low
[6]	GPIO pin 6 level. 1 = drive pin high 0 = drive pin low
[7]	GPIO pin 7 level. 1 = drive pin high 0 = drive pin low

16.11.4. GPIO readback

Use this register to read the real-world drive level on GPIO pins, regardless of the function or bit settings.

Read/write:	r	*Address:*	M1F3h
Default:	Undefined.	*Address index:*	-

7	6	5	4	3	2	1	0
GPIO 7	GPIO 6	GPIO 5	GPIO 4	GPIO 3	GPIO 2	GPIO 1	GPIO 0

Bits	Description
[0]	GPIO pin 0 status. 1 = driven high 0 = driven low
[1]	GPIO pin 1 status. 1 = driven high 0 = driven low
[2]	GPIO pin 2 status. 1 = driven high 0 = driven low
[3]	GPIO pin 3 status. 1 = driven high 0 = driven low
[4]	GPIO pin 4 status. 1 = driven high 0 = driven low
[5]	GPIO pin 5 status. 1 = driven high 0 = driven low
[6]	GPIO pin 6 status. 1 = driven high 0 = driven low
[7]	GPIO pin 7 status. 1 = driven high 0 = driven low

16.12 VMI+ host port registers

For a description of the VMI+ physical connector, refer to "Recommended VMI+ interface," described on page 319.

The VMI+ host port appears at the ProMotion register locations specified by "Remap control," described on page 168.

16.12.1. VMI+ host port 0 control

Use this register to specify host port 0 configuration.

Read/write:	r/w	*Address:*	M100h
Default:	Undefined.	*Address index:*	-

7	6	5	4	3	2	1	0
		port 0 width		port 0 retry	port 0 access type	port 0 repeat	port 0 host control

Bits	Description
[0]	Port 0 enable. 1 = port 0 enabled 0 = port 0 disabled
[1]	Port 0 repeat 1 = all of VMI 1h address space is ,mapped to the single VMI device address specified by "VMI+ host port 0 index offset," described on page 270. 0 = VMI 1K address space is mapped to the 64K VMI device space according to bits [15:0] of "VMI+ host port 0 index offset," described on page 270.
[2]	Port 0 access type 1 = Motorola ($\overline{DS}$, R/$\overline{W}$) 0 = Intel (AEN, $\overline{IOR}$, $\overline{IOW}$)
[3]	Port 0 retry. 1 = enabled 0 = disabled Setting this bit frees up ProMotion to do memory cycles after timeout, for slow VMI devices. ☞ This functionality requires additional glue logic.
[5:4]	Port 0 width 1x = reserved 01 = 16-bit data port width 00 = 8-bit data port width

16.12.2. VMI+ host port 0 timing

Use this register to specify command pulse width and timeout parameters for host port 0.

Read/write:	r/w	*Address:*	M101h
Default:	Undefined.	*Address index:*	-

7	6	5	4	3	2	1	0
port 0 time-out				port 0 pulse width			

Bits	Description
[3:0]	Port command pulse width.
[7:4]	Port 0 time-out. Set this in conjunction with M100[3], "VMI+ host port 0 control," described on page 269.

16.12.3. VMI+ host port 0 index offset

Use this register to specify locations within the VMI device to be mapped to the VMI port.

Setting M100[1] causes any access to the VMI+ host 0 port to generate an access to the VMI+ device 0 at the address specified by this register. When M100[1] is not set then bits [15:14] of this register are concatenated with the actual offset within the VMI+ port 0 to generate an address to the VMI+ device 0.

Read/write:	r/w	*Address:*	M102-103h
Default:	Undefined.	*Address index:*	-

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
host port 0 index offset															

Bits	Description
[15:0]	Index offset.

16.12.4. VMI+ host port 1 control

Use this register to specify port 1 configuration.

Read/write:	r/w	*Address:*	M104h
Default:	Undefined.	*Address index:*	-

7	6	5	4	3	2	1	0
		port 1 width		port 1 retry	port 1 access type	port 1 repeat	port 1 host control

Bits	Description
[0]	Port 1 enable. 1 = port 1 enabled 0 = port 1 disabled
[1]	Port 1 repeat 1 = all of VMI 1h address space is ,mapped to the single VMI device address specified by "VMI+ host port 1 index offset," described on page 272. 0 = VMI 1K address space is mapped to the 64K VMI device space according to bits [15:0] of "VMI+ host port 1 index offset," described on page 272.
[2]	Port 1 access type 1 = Motorola ($\overline{DS}$, R/$\overline{W}$) 0 = Intel (AEN, $\overline{IOR}$, $\overline{IOW}$)
[3]	Port 01 retry. 1 = enabled 0 = disabled Setting this bit frees up ProMotion to do memory cycles after timeout, for slow VMI devices. ☞ This functionality requires additional glue logic.
[5:4]	Port 1 width 1x = reserved 01 = 16-bit data port width 00 = 8-bit data port width

16.12.5. VMI+ host port 1 timing

Use this register to specify command pulse width and timeout parameters for host port 1.

Read/write:	r/w	*Address:*	M105h
Default:	Undefined.	*Address index:*	-

7	6	5	4	3	2	1	0
port 1 time-out				port 1 pulse width			

Bits	Description
[3:0]	Port command pulse width.
[7:4]	Port 1 time-out. Set this in conjunction with M100[3], "VMI+ host port 1 control," described on page 270.

16.12.6. VMI+ host port 1 index offset

Use this register to specify locations within the VMI device to be mapped to the VMI port.

Setting M104[1] causes any access to the VMI+ host 1 port to generate an access to the VMI+ device 1 at the address specified by this register. When M104[1] is not set then bits [15:14] of this register are concatenated with the actual offset within the VMI+ port 1 to generate an address to the VMI+ device 1.

Read/write:	r/w	*Address:*	M106-107h
Default:	Undefined.	*Address index:*	-

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
host port 1 index offset															

Bits	Description
[15:0]	Index offset.

16.13 THP interface registers

For a description of the THP connector, refer to "Recommended 3Dfx THP interface," described on page 317.

16.13.1. THP control

Use this register enable THP mode and other shared-memory modes.

Read/write:	r/w	*Address:*	M110h
Default:	Undefined.	*Address index:*	-

7	6	5	4	3	2	1	0
						THP control	
						MD[9]	MD[8]

Bits	Description	
[1:0]	THP control	
	11 =	3Dfx (PUMA)
	10 =	unified memory architecture (UMA)
	01 =	shared frame buffer (SFB)
	00 =	none
[15:2]	Reserved.	

16.13.2. Slave request high timing

Use this register specify how long AT3D waits to convert a low-priority request to high-priority in UMA mode.

Read/write:	r/w	*Address:*	M112h
Default:	Undefined.	*Address index:*	-

7	6	5	4	3	2	1	0
slave request high timing							

Bits	Description
[7:0]	Slave request high timing, in MCLKs.

16.13.3. Slave grant high timing

Use this register specify how long AT3D holds the bus after a grant is taken away, in 3Dfx and UMA modes.

Read/write:	r/w	*Address:*	M113h
Default:	Undefined.	*Address index:*	-

7	6	5	4	3	2	1	0
slave grant high timing							

Bits	Description
[7:0]	Slave grant high timing, in MCLKs.

16.13.4. Serial input

Use this register to determine the latest serial word captured by the serial input port.

Read/write:	r	*Address:*	M1F4–1F5h
Default:	Undefined.	*Address index:*	-

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
serial word															

Bits	Description
[15:0]	Serial word

16.14 VMI+ video port registers

For a description of the physical VMI+ connector, refer to "Recommended VMI+ interface," described on page 319.

16.14.1. VMI+ video port control

Use this register to configure the VMI+ port.

Read/write:	r/w	*Address:*	M120–123h
Default:	Undefined.	*Address index:*	-

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
invert TVCLK	internal active	swap odd&even	swap U&V	reset cl VSYNC	increment lc HSYNC	reset pix HSYNC	sample ODD VSYNC	internal vertical blank	internal horizontal blank	internal odd field	XVSYNC pin positive	XHSYNC pin positive	XODD pin positive	source interlace	divide clock by 2	invert pixel qualifier			FIFO trip oint				decimation vertical			decimation horizontal			input averaging	double buffering	VMI+ video port enable

Bits	Description
[0]	VMI+ video port enable. 1 = VMI+ video port enabled. 0 = VMI+ disabled.
[1]	Double frame buffers. Use this bit to enable double-buffered operation in conjunction with "VMI+ video port base address 1," on page 279. 1 = double buffering. 0 = single buffering.
[2]	Horizontal filtering. Alliance recommends this bit always be set. 1 = horizontal filter on. 0 = no filter.
[5:3]	Decimation horizontal. 101–111 = reserved. 100 = 16X decimation. 011 = 8X decimation. 010 = 4X decimation. 001 = 2X decimation. 000 = pass through.
[8:6]	Decimation vertical. 101–111 = reserved. 011 = 8X decimation. 010 = 4X decimation. 001 = 2X decimation. 000 = pass through.
[12:9]	FIFO trip point, in quad-words. Valid values are 0000–0111b. Use this field to set the FIFO count (number of FIFO entries: $0\text{-}8_{10}$) which causes FIFO contents to be written to memory.

Bits	Description
[14:13]	Reserved.
[15]	Invert pixel qualifier. Set this bit if pixel qualifier is active low. 1 = inverted. 0 = normal.
[16]	Divide clock. Set this bit when using VMI 1 = accept data every clock cycle. 0 = accept data qualified by bit 15, above.
[17]	Source interlace. Set this bit if source data is interlaced. 1 = input signal interlaced. 0 = input signal non-interlaced.
[18]	Invert XODD pin. Set to 1 if input signal is active LOW 1 = input signal active LOW. 0 = input signal active HIGH.
[19]	Invert XHSYNC pin. Set to 1 if input signal is active LOW 1 = input signal active LOW. 0 = input signal active HIGH.
[20]	Invert XVSYNC pin. Set to 1 if input signal is active LOW 1 = input signal active LOW. 0 = input signal active HIGH.
[21]	Odd field. Set this bit to use internally-generated odd field. 1 = internally generate odd field. 0 = use external XODD pin.
[22]	Internal horizontal blank. 1 = internally generate signal. 0 = use signal from feature conector pin.
[23]	Internal vertical blank. 1 = internally generate signal. 0 = use signal from feature conector pin.
[24]	Sample HSYNC. 1 = on falling VSYNC. 0 = on rising VSYNC.
[25]	Reset pixel counter. 1 = on falling HSYNC. 0 = on rising HSYNC.
[26]	Increment line counter. 1 = on rising HSYNC. 0 = on falling HSYNC.
[27]	Reset line counter. 1 = on falling HSYNC. 0 = on rising VSYNC.
[28]	Swap U & V data in memory. 1 = U & V data memory locations swapped. 0 = normal U and V.

Bits	Description
[29]	Swap odd and even lines in memory. 1 = top line VIP buffer even. 0 = top line VIP buffer odd.
[30]	Internal active data. Set this bit when decoder delivers active signal for frame buffer. 1 = active signal. 0 = BLANK signal.
[31]	Invert TVCLK input. 1 = invert TVCLK. 0 = normal TVCLK.

16.14.2. VMI+ video input port pitch

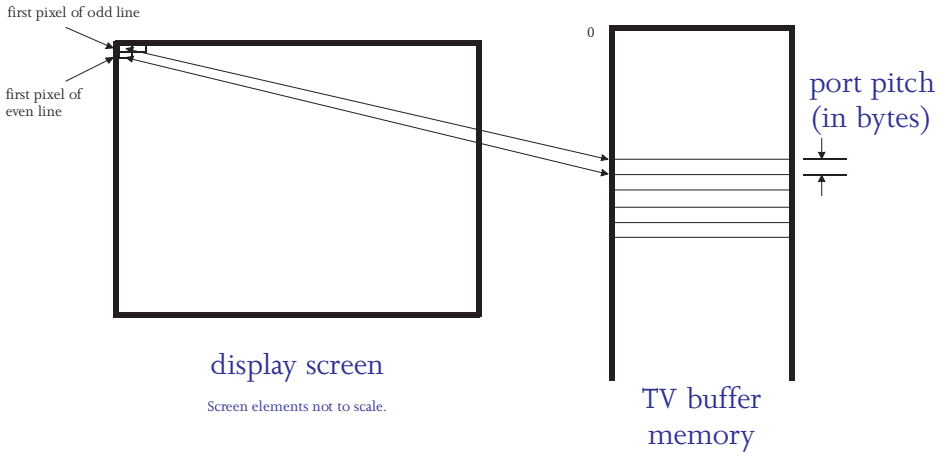
Use this register to set the number of bytes between rows of interlaced lines in TV buffer.

Read/write:	r/w	*Address:*	M124-125h
Default:	Undefined.	*Address index:*	-

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	VMI+ video port pitch ☞ 2:0 must be 0														

Bits	Description
[14:0]	VMI+ video port pitch (bytes). ☞ Bits 2:0 must be 0
[15]	Reserved

Figure 16.14.2 VMI video port pitch



16.14.3. VMI+ FIFO status

Use this register to detect whether the FIFO overflow condition has ocurred.

This register is reset when read.

Read/write:	r	*Address:*	M127h
Default:	0h	*Address index:*	-

7	6	5	4	3	2	1	0
overflow status							

Bits	Description
[6:0]	Reserved
[7]	FIFO overflow status 1 = FIFO overflow 0 = no overflow since reset/power-up

16.14.4. VMI+ video port base address 0

Use this register to set the memory location for the start of the first line in the primary VIP buffer

Read/write:	r/w	*Address:*	M128–12Ah
Default:	Undefined.	*Address index:*	-

23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					Base address 0																		

Bits	Description
[18:0]	VMI+ base address 0. ☞ Bits 2:0 must be 0
[23:19]	Reserved.

16.14.5. VMI+ video port base address 1

Use this register to set the memory location for the start of the first line in the second VIP buffer. To use double buffering you must set bit M120[15],"VMI+ video port control," described on page 275.

Read/write:	r/w	*Address:*	M12C–12Eh
Default:	Undefined.	*Address index:*	-

23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					Base address 1																		

Bits	Description
[18:0]	VMI+ base address1. ☞ Bits 2:0 must be 0
[23:19]	Reserved.

16.14.6. Video input cropping boundary left

Use this register to set the left boundary of the video input active area. For example, this register may be used to crop the overscan area of a television signal, leaving only an active area of meaningful data.

Figure 16.14.6, "Video input cropping boundaries," on page 280, shows the relationship of the four cropping boundaries.

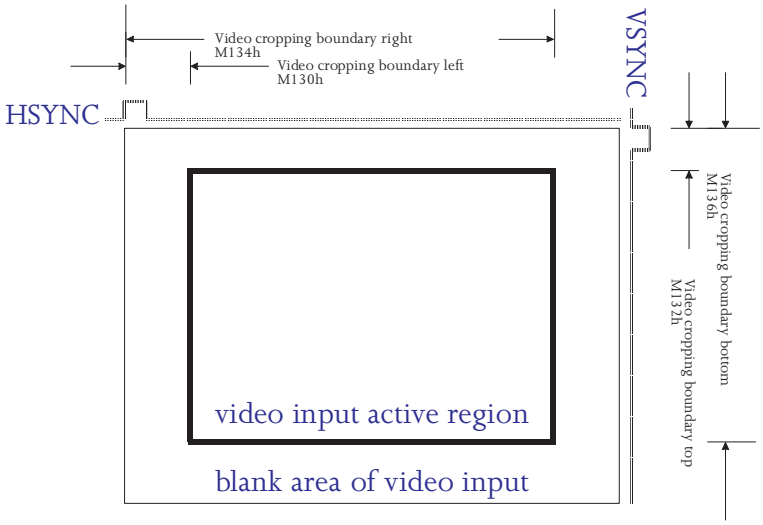
The cropped area may extend into the active signal, allowing isolation of any rectangular area from the input signal for processing by the ProMotion controller.

Read/write:	r/w	*Address:*	M130–131h
Default:	Undefined.	*Address index:*	-

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				cropping boundary left											

Bits	Description
[11:0]	Cropping boundary left, in TVCLKs.
[15:12]	Reserved.

Figure 16.14.6 Video input cropping boundaries



16.14.7. Video input cropping boundary top

Use this register to set the top boundary of the video input active area. Refer to the figure and discussion under "Video input cropping boundary left," on page 279.

Read/write:	r/w	*Address:*	M132–133h
Default:	Undefined.	*Address index:*	-

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				cropping boundary top											

Bits	Description
[11:0]	Cropping boundary top, in scan lines.
[15:12]	Reserved.

16.14.8. Video input cropping boundary right

Use this register to set the right boundary of the video input active area. Refer to the figure and discussion under "Video input cropping boundary left," on page 279.

Read/write:	r/w	*Address:*	M134–135h
Default:	Undefined.	*Address index:*	-

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				cropping boundary right											

Bits	Description
[9:0]	Cropping boundary right, in TVCLKs.
[15:10]	Reserved.

16.14.9. Video input cropping boundary bottom

Use this register to set the bottom boundary of the video input active area. Refer to the figure and discussion under "Video input cropping boundary left," on page 279.

Read/write:	r/w	*Address:*	M136–137h
Default:	Undefined.	*Address index:*	-

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				cropping boundary bottom											

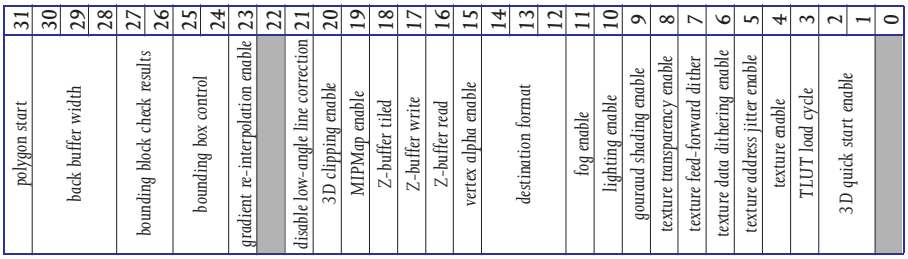
Bits	Description
[9:0]	VMI+ vertical active total
[15:10]	Reserved.

16.15 3D rendering engine registers

16.15.1. Polygon engine control 0

Use this register enable various 3D control settings.

Read/write:	r/w	*Address:*	M300–303h
Default:	[27:26] =0 others undefined.	*Address index:*	-



Bits	Description
[0]	Reserved
[2:1]	3D quick start. 11 = enabled, polygon strips. 10 = enabled, polygon lists. 0x = disabled. Polygon lists start automatically every 3 vertices, polygon strips start automatically on the 3rd and every subsequent vertex
[3]	TLUT load cycle. 1 = enabled. 0 = disabled.
[4]	Texture mapping. 1 = enabled. 0 = disabled.
[5]	Texture address jitter. 1 = enabled. 0 = disabled.
[6]	Texture data dithering. 1 = enabled. 0 = disabled.
[7]	Texture data dithering mode. 1 = feed forward. 0 = random.
[8]	Texture transparency. 1 = enabled. 0 = disabled.

Bits	Description
[9]	Gouraud shading. 1 = enabled. 0 = disabled.
[10]	Lighting. 1 = enabled. 0 = disabled.
[11]	Fog. 1 = enabled. 0 = disabled.
[14:12]	Destination format 111 = 32 bpp. 100 = 16 bpp. 100 = 15 bpp. 011 = 8 bpp direct. 010 = 8 bpp indexed. Other settings are reserved.
[15]	Vertex alpha 1 = enabled. 0 = disabled.
[16]	Z-buffer read 1 = enabled. 0 = disabled. If the Z-buffer read bit is set without the Z-buffer write bit, only pixels closer to the viewer than the existing pixel will be written, but no Z-buffer values will be updated in any event. If the Z-buffer write bit is set without the Z-buffer read bit, all pixels will be written along with their corresponding Z values.
[17]	Z-buffer write 1 = enabled. 0 = disabled. See note under bit 16, above.
[18]	Z-buffer tiled 1 = enabled. 0 = disabled. If the Z-buffer tiled bit is set, the Z-buffer is assumed to be 64x64 pixels. Otherwise, the Z-buffer is assumed to be the same size as the back and front buffers.
[19]	MIPMap 1 = enabled. 0 = disabled.
[20]	3D clipping 1 = enabled. 0 = disabled.
[21]	Low-angle line correction 1 = disabled. 0 = enabled.
[22]	Reserved

Bits	Description
[23]	Gradient re-interpolation 1 = enabled. 0 = disabled.
[25:24]	Bounding box control 11 = abort if both visible and invisible pixels found. 10 = abort if any pixel disabled. 01 = reserved. 00 = bounding box disabled.
[27:26]	Bounding box check status 11 = some pixels visible (mixed) 10 = all pixels visible (front) 01 = all pixels invisible (back) 00 = ☞ Reading this field resets contents to 00. This field may accumulate over many polygons.
[30:28]	Back buffer width 111 = 1600 pixels. 110 = 1280 pixels. 101 = 1152 pixels. 100 = 1024 pixels. 011 = 512 pixels. 010 = 800 pixels. 001 = 640 pixels. 000 = 320 pixels.
[31]	Polygon start 1 = enabled. 0 = disabled.

16.15.2. Polygon engine control 1

Use this register enable various 3D control settings.

Read/write:	r/w	*Address:*	M304–307h
Default:	Undefined.	*Address index:*	-

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
overlap Gouraud interpolate and write	3D FIFO watermark			U/V monotonicity clamp	hidden spanlet skip	Gouraud overlap timing 32bpp				Gouraud overlap timong 16bpp			Gouraud overlap timong 8bpp			Z compare mode			128-bit access	vertex stack	alpha polarity	texture source alpha	texture address rounding	texture clamp	texture mirror	programming gradient					

Bits	Description
[4:0]	Reserved.
[5]	Programmable gradient re-interpolation 1 = enabled. 0 = disabled.
[6]	Texture mirror 1 = enabled. 0 = disabled.
[7]	Texture clamp 1 = enabled. 0 = disabled.
[8]	Texture address rounding 1 = disabled. 0 = enabled..
[9]	Texture source alpha 1 = enabled. 0 = disabled. ☞ Set both this bit and 30C[3], "Texture format," described on page 286, to enable source texture alpha
[10]	Alpha polarity 1 = transparent = FF; opaque = 00. 0 = transparent = 00; opaque = FF.
[11]	Vertical stack 1 = disabled. 0 = enabled.
[12]	128-bit access 1 = disabled. 0 = enabled.
[15:13]	Z compare mode 111 = always write new pixel. 110 = write if new Z >= old Z. 101 = write if new Z <> old Z. 100 = write if new Z > old Z. 011 = write if new Z <= old Z. 010 = write if new Z = old Z. 001 = write if new Z < old Z. 000 = necver write new pixel.
[18:16]	Gouraud overlap timing - 8 pp nnn = number of entries in 3D FIFO before Gouraud interpolation restarts when destination is 8 bpp.
[21:19]	Gouraud overlap timing - 16 bpp nnn = number of entries in 3D FIFO before Gouraud interpolation restarts when destination is 16 bpp.
[25:23]	Gouraud overlap timing - 32 bpp nnn = number of entries in 3D FIFO before Gouraud interpolation restarts when destination is 32 bpp.

Bits	Description
[26]	Hidden spanlet skip 1 = disable. 0 = enable.
[27]	U/V monotonicity clamp 1 = disable. 0 = enable.
[30:28]	3D FIFO watermark nnn = number of entries in 3D FIFO before output to display memory begins.
[31]	Overlap Gouraud interpolate & write 1 = enabled. 0 = disabled. ☞ This bit shoudl be 0 when textures are enabled.

16.15.3. Texture map base address

Use this register to specify the address for texture map.

Read/write:	r/w	*Address:*	M308–30Ah
Default:	Undefined.	*Address index:*	-

23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
texture map base address																							

Bits	Description
[23:0]	Texture map base address

16.15.4. Texture format

Use this register specify size and bit-depth of the texture map.

Read/write:	r/w	*Address:*	M30C–30Dh
Default:	Undefined.	*Address index:*	-

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					texture wrap height			texture wrap width				source texture alpha	texel format		

Bits	Description
[2;0]	Texel format 111 = 24 bits per pixel. 110 = reserved. 101 = 16 bits per pixel. 100 = 15 bits per pixel. 011 = 12 bits per pixel. 010 = 8 bits per pixel. 001 = 4 bits per pixel. 000 = reserved.
[3]	Source texture alpha 1 = enabled. 0 = disabled. ☞ Set both this bit and 304[9], "Polygon engine control 1," described on page 284, to enable source texture alpha
[4]	Reserved
[7:5]	Texture wrap width 11x = reserved. 101 = 256 pixels. 100 = 128 pixels. 011 = 64 pixels. 010 = 32 pixels. 001 = 16 pixels. 000 = 8 pixels.
[10:8]	Texture wrap height 11x = reserved. 101 = 256 pixels. 100 = 128 pixels. 011 = 64 pixels. 010 = 32 pixels. 001 = 16 pixels. 000 = 8 pixels.

16.15.5. Texel index offset

Use this register specify an offset to be added to texels of 8 bits or less to create an indexed into the texture look-up table (TLUT). The application is responsible for assuring that the resultant sum will be within the existing TLUT (whose size is not necessarily constrained to be a power of 2). ProMotion-AT3D TLUT is 256.

Read/write:	r/w	*Address:*	M30Eh
Default:	Undefined.	*Address index:*	-

7	6	5	4	3	2	1	0
Texel index offset							

Bits	Description
[7:0]	Texel index offset

16.15.6. 3D internal register index

Use this register to read or write internal 3D engine registers. First write the corresponding internal register index into this register and then read or write the register value from M314h, "3D internal register data," described on page 290.

The following combinations of major and minor registers do not exist: Y registers: current register (000000) only exists. X registers: dX (001100) does not exist.

Read/write:	r/w	*Address:*	M310h
Default:	Undefined.	*Address index:*	-

7	6	5	4	3	2	1	0
major index					minor index		

Bits	Description	
[2:0]	Minor index	
	111 =	reserved.
	110 =	"d*r" (delta on right side).
	101 =	"d*l" (delta on left edge).
	100 =	"*d" prefix (delta on span).
	011 =	"*r" suffix (right).
	010 =	"*l" suffix (left).
	001 =	"*s" suffix (span).
	000 =	"*" current.

Bits	Description
[3]	Reserved.
[7:4]	Major index 1001–1111 = reserved. 1000 = A registers. 0111 = VW registers. 0110 = UW registers. 0101 = F registers. 0100 = L registers. 0011 = W registers. 0010 = Z registers. 0001 = X registers. 0000 = Y registers.

16.15.7. Disable span delta calculation

Use this register disable internal recalculation of various span delta values.

These bits should be used only if the driver individually loads disabled register(s).

Read/write:	r/w	*Address:*	M312h
Default:	Undefined.	*Address index:*	-

7	6	5	4	3	2	1	0
dA/dx	dVW/dx	dUW/dx	dF/dx	dL/dx	dW/dx	dZ/dx	

Bits	Description
[0]	Reserved
[1]	dZ/dx 1 = do not recalculate. 0 = recalculate.
[2]	dW/dx 1 = do not recalculate. 0 = recalculate.
[3]	dL/dx 1 = do not recalculate. 0 = recalculate.
[4]	dF/dx 1 = do not recalculate. 0 = recalculate.
[5]	dUW/dx 1 = do not recalculate. 0 = recalculate.

Bits	Description	
[6]	dVW/dx	
	1 =	do not recalculate.
	0 =	recalculate.
[7]	dA/dx	
	1 =	do not recalculate.
	0 =	recalculate.

16.15.8. 3D internal register data

Use this register to read or write internal 3D engine registers. First write the corresponding internal register index into M310h, "3D internal register index," described on page 288, and then read or write the register value from this register.

Read/write:	r/w	*Address:*	M314–317h
Default:	Undefined.	*Address index:*	-

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
3d internal register data																															

Bits	Description	
[7:0]	field	
	1 =	enabled.
	0 =	disabled.

16.15.9. Z buffer base pointer

Use this register to specify the base address of the Z-buffer

Read/write:	r/w	*Address:*	M318–31Ah
Default:	Undefined.	*Address index:*	-

23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Z-bufferbase pointer																							

Bits	Description
[23:0]	Z-buffer base pointer.

16.15.10. Z buffer front clipping plane

Use this register specify a plane for which pixels closer to the viewer are not drawn.

Read/write:	r/w	*Address:*	M31C–31Dh
Default:	0h	*Address index:*	-

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Z-buffer front clipping plane															

Bits	Description
[15:0]	Z-buffer front clipping plane

16.15.11. Z–buffer back clipping plane

Use this register specify a plane for which pixels farther from the viewer are not drawn.

Read/write:	r/w	*Address:*	M31E–31Fh
Default:	FFFFh	*Address index:*	-

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Z-buffer back clipping plane															

Bits	Description
[15:0]	Z-buffer back clipping plane

16.15.12. Texel transparency color

Use this register to specify which color in the current texture is rendered transparent.

Read/write:	r/w	*Address:*	M320–322h
Default:	Undefined.	*Address index:*	-

23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																				texel transparency color			

Bits	Description (4-bit textures)
[23:0]	Texel transparency color

23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																texel transparency color							

Bits	Description (8-bit textrues)
[23:0]	Texel transparency color

23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
texel transparency color																							

Bits	Description (direct color)
[23:0]	Texel transparency color

16.15.13. Fog color

Use this register to specify the 24-bit color of fog,

Read/write:	r/w	*Address:*	M324–326h
Default:	Undefined.	*Address index:*	-

23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
fog color																							

Bits	Description
[23:0]	Fog color

16.15.14. Back buffer base address

Use this register to specify the base address of the buffer currently being rendered to.

Read/write:	r/w	*Address:*	M328–32Ah
Default:	Undefined.	*Address index:*	-

23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Back buffer base address																							

Bits	Description
[23:0]	Back buffer base address

16.15.15. 3D clipping left

Use this register determine the left boundary of the clipping region.

Read/write:	r/w	*Address:*	M330–331h
Default:	Undefined.	*Address index:*	-

23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
											clipping boundary												

Bits	Description
[11:0]	Clipping boundary left

16.15.16. 3D clipping top

Use this register determine the top boundary of the clipping region.

Read/write:	r/w	*Address:*	M332–333h
Default:	Undefined.	*Address index:*	-

23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												clipping boundary											

Bits	Description
[11:0]	Clipping boundary top

16.15.17. 3D clipping right

Use this register determine the right boundary of the clipping region.

Read/write:	r/w	*Address:*	M334–335h
Default:	Undefined.	*Address index:*	-

23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												clipping boundary											

Bits	Description
[11:0]	Clipping boundary right

16.15.18. 3D clipping bottom

Use this register determine the clipping region.

Read/write:	r/w	*Address:*	M336–337h
Default:	Undefined.	*Address index:*	-

23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												clipping boundary											

Bits	*Description*
[11:0]	Clipping boundary.

16.16 Polygon vertex stack registers

ProMotion-AT3D has three groups of polygon stack registers. Stack 0 registers are detailed below. Stack 1–2 register groups have equivalent parallel structure.

Table 16.16 Polygon stack register groups 0–2

Register	Stack 0	Stack 1	Stack 2	Bits	r/w
Destination vertex X	342–343	362–363	382–383	12	r/w
Destination vertex Y	346–347	366–367	386–387	12	r/w
Destination vertex Z	34A–34B	36A–36B	38A–38B	16	r/w
Destination vertex W	34D	36D	38D	8	r/w
Destination vertex L	350	370	390	8	r/w
Destination vertex A	353	373	393	8	r/w
Destination vertex F	354	374	394	8	r/w
Source vertex U	35A	37A	39A	16	r/w
Source vertex V	35E	37E	39E	16	r/w

Operation of the polygon vertex stack

The three vertices required to generate each polygon may be written to their respective addresses, or the vertex stack may be used. Writing any register in the top of the vertex stack (group 0) causes the two previously written values to be pushed down into the stack and the value prior to those two to be lost.

For example writing register X in the Polygon Vertex Stack 0 causes the previous X Stack 0 value to be pushed to Polygon vertex X stack 1 register, and Polygon vertex X stack 1 values to be pushed to vertex 'n' stack 2).

To write the first triangle in a triangle strip, the register stack must be written three times. For subsequent triangles in the same strip, only one write to the stack should be performed.

16.16.1. Destination vertex X stack 0

Use this register to specify the X coordinate of the vertex relative to the currently defined Back Buffer.

Read/write:	r/w	*Address:*	M342–343h
Default:	Undefined.	*Address index:*	-

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				destination vertex X											

Bits	Description
[11:0]	Destination vertex X [27:16] (12.0).
[15:12]	Rserved.

16.16.2. Destination vertex Y stack 0

Use this register to specify the Y coordinate of the vertex relative to the currently defined Back Buffer.

Read/write:	r/w	*Address:*	M346–347h
Default:	Undefined.	*Address index:*	-

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				destination vertex Y											

Bits	Description
[11:0]	Destination vertex Y [27:16] (12.0).
[15:12]	Rserved.

16.16.3. Destination vertex Z stack 0

Use this register specify the depth from the viewer of the vertex.

Read/write:	r/w	*Address:*	M34A–34Bh
Default:	Undefined.	*Address index:*	-

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
destination vertex Z															

Bits	Description
[15:0]	Destination vertex Z [31:16] (16.0).

16.16.4. Destination vertex W stack 0

Use this register to specify a w coordinate used in perspective correction. This value is generally proportional to 1/Z.

Alliance recommends all vertices in a given polygon have W value in the range 0.5 to 1.0.

Read/write:	r/w	*Address:*	M34Dh
Default:	Undefined.	*Address index:*	-

7	6	5	4	3	2	1	0
destination vertex W							

Bits	Description
[7:0]	Destination vertex W [15:8] (0.8). Used for Gouraud: destination vertex B.

16.16.5. Destination vertex L (lighting) stack 0

Use this register to specify the lighting value of the vertex. An L value of FFh represents maximum lighting.

Read/write:	r/w	*Address:*	M350h
Default:	Undefined.	*Address index:*	-

7	6	5	4	3	2	1	0
destination vertex L							

Bits	Description
[7:0]	Destination vertex L (0.8).

16.16.6. Destination vertex A (alpha) stack 0

Use this register to specify the alpha value of the vertex. An A value of FFh indicates full opacity.

Read/write:	r/w	*Address:*	M353h
Default:	Undefined.	*Address index:*	-

7	6	5	4	3	2	1	0
destination vertex A							

Bits	Description
[7:0]	Destination vertex A (0.8).

16.16.7. Destination vertex F (fog) stack 0

Use this register to specify the fog value for the vertex. A value of FFh represents no fog.

Read/write:	r/w	*Address:*	M354h
Default:	Undefined.	*Address index:*	-

7	6	5	4	3	2	1	0
destination vertex F							

Bits	Description
[7:0]	Destination vertex F (0.8).

16.16.8. Source vertex U stack 0

Use this register to specify the U coordinate within the texture.for the vertex.

Read/write:	r/w	*Address:*	M35Ah
Default:	Undefined.	*Address index:*	-

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
source vertex U															

Bits	Description
[7:0]	Source vertex U (0.8). Used for Gouraud: destination vertex R.

16.16.9. Source vertex V stack 0

Use this register to specify the V coordinate within the texture.for the vertex.

Read/write:	r/w	*Address:*	M35Eh
Default:	Undefined.	*Address index:*	-

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
source vertex V															

Bits	Description
[7:0]	Source vertex V (0.8). Used for Gouraud: destination vertex G.

16.17 Texture scale registers

Use these registers to map the (U,V) coordinates from Source vertex stack registers into the actual (U,V) space, which can be much larger or much smaller. These values should ideally be spread apart into the range 0 to 1.

16.17.1. U factor

Read/write:	r/w	*Address:*	M3C0–3C1h
Default:	Undefined.	*Address index:*	-

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
U factor															

Bits	Description
[15:0]	U factor [15:0] (8.8).

16.17.2. U offset

Read/write:	r/w	*Address:*	M3C2–3C3h
Default:	Undefined.	*Address index:*	-

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits	Description
[9:0]	U offset [8:0] (1.8).

16.17.3. V factor

Read/write:	r/w	*Address:*	M3C4–3C5h
Default:	Undefined.	*Address index:*	-

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
U factor															

Bits	Description
[15:0]	V factor [15:0] (8.8).

16.17.4. V offset

Read/write:	r/w	*Address:*	M3C6–3C7h
Default:	Undefined.	*Address index:*	-

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							V offset								

Bits	Description
[7:0]	V offset [8:0] (1.8).

16.17.5. Gradient re-interpolation count

Use this register to specify a number of scan lines after which the span gradients are longer recalculated for each span line. Often this is set to reach the inflection point of the triangle, where the span is largest, and therefore most accurate.

Read/write:	r/w	*Address:*	M3C8h
Default:	Fh.	*Address index:*	-

7	6	5	4	3	2	1	0

Bits	Description
[3:0]	Gradient re-interpolation count.
[7:4]	Reserved

16.18 Test Registers

16.18.1. Signature analyzer overview

The ProMotion Signature Analyzer permits efficient checking of the device at high speeds. The analyzer generates a signature (polynomial) based on the input to the on-board RAMDAC.

All frame timing is included in this signature, as such, the analyzer tests the video path as well as the CRTC timing circuitry.

When combined with two DRAMs on the load board, the Signature Analyzer permits testing of virtually the entire device. This is accomplished by using the host interface and on-board accelerator functions to generate one or more "visible" screens of information within the DRAMs and having the Analyzer perform signature analysis on those screens during video refresh. Any error in generating the screens within the DRAM will result in an incorrect signature.

The Signature Analyzer implements the polynomial with a seed value of 0001h.

$$x^{23}+x^{22}+x^{20}+x^{18}+x^{16}+x^{14}+x^{12}+x^{10}+x^{8}+x^{6}+x^{4}+1$$

While you can calculate a signature using simulation, this method is time-consuming. To generate a test-control signature swiftly, run the various tests on ProMotion controllers and read the contents of the Analyzer. Generate signature from more than one sample controller to guard against the possibility of your control sample being defective.

16.18.1.1 Use of the signature analyzer

1) Create patterns in display memory exercising various internal functions of the device.
2) Set bit 0B4[0], "Signature analyzer control," described on page 301. Capture begins automatically at the beginning of the next frame and lasts for exactly one frame. In interlaced mode, a frame consists of two fields; this is handled automatically by the Analyzer. When the analysis is complete, the Analyzer shuts down and resets Signature analyzer busy bit 1FF[7], of "Extended/DAC status," described on page 178.
3) Depending on the particular tester in use, 1FF[7] can either be polled, or the correct number of clock cycles can simply be waited out, after which the 24-bit signature is read from 0B5–0B7"Signature value," described on page 302.
4) Resetting start bit M0B4[0] resets the Analyzer for the next capture.

16.18.2. Signature analyzer control

Refer to "Signature analyzer overview," described on page 301 for a discussion of this register.

Read/write:	r/w	*Address:*	M0B4h
Default:	0h	*Address index:*	-

7	6	5	4	3	2	1	0
							start/clear

Bits	Description
[0]	Signature start/clear.
[7:1]	Reserved.

16.18.3. Signature value

Refer to "Signature analyzer overview," described on page 301 for a discussion of this register.

Read/write:	r/w	*Address:*	M0B5h
Default:	0h	*Address index:*	-

23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
signature value																							

Bits	Description
[23:0]	Signature value

17. Data formats

17.1 vWindow data formats in display memory

17.1.1. Indexed color

vWindow data may be stored in display memory in 8-bit indexed color format. This format is passed to the DAC as indexed-color data. This format may be stretched but should not be blended. It is the responsibility of software to load the DAC palette with correct color values for the video data.

17.1.2. Direct color

vWindow data may be stored in display memory in 8-bit, 15-bit, 16-bit, or 24-bit direct color formats. These formats are passed to the DAC as direct-color data.

17.1.3. YUV

Video data may be stored in display memory in 4:2:2, 4:1:1, or 4:0:0 YUV formats. These formats are converted to direct color and passed to the DAC. The 4:1:1 format requires that the host replicate UV bytes to provide pseudo-4:2:2; byte order is appropriate to perform this conversion efficiently.

The 4:0:0 format generates greyscale images.

17.1.4. In–place video data

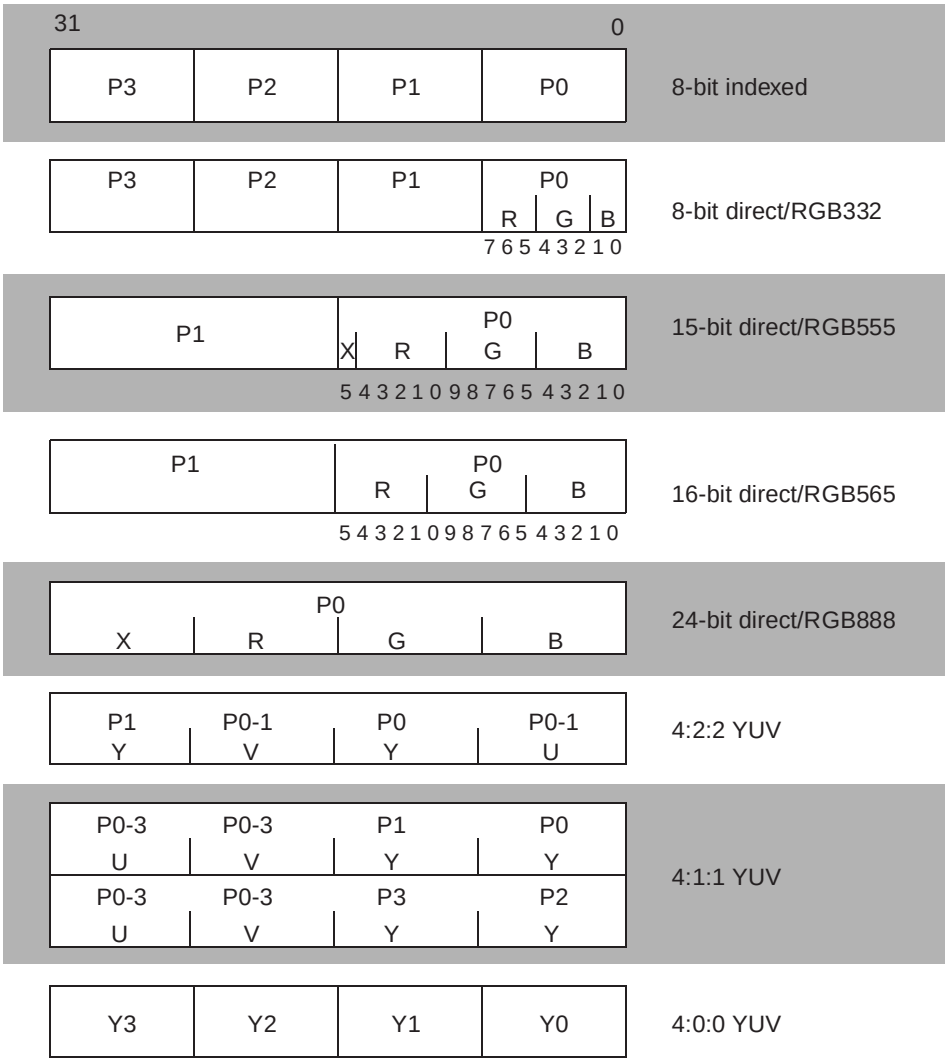
Normally, video data is stored in off-screen memory in a contiguous region. Where this is not possible or desired, it is possible to store video data "in place" by writing it into the on-screen display memory region that will be occluded by the motion video window.

In this case, each row of video data should be stored in a different row of on-screen memory. The Video Data Offset register then will continue to specify the offset between adjacent rows of video data, however in this case the value will match the VGA Offset register rather than the Video Data Width (as adjusted for pixel depth).

This technique will generally work only if the image is stretched by at least a factor of 2.0, or if the desktop is 16 bits per pixel, as the amount of data being stored in for each line of the motion video window will require 2 bytes per pixel.

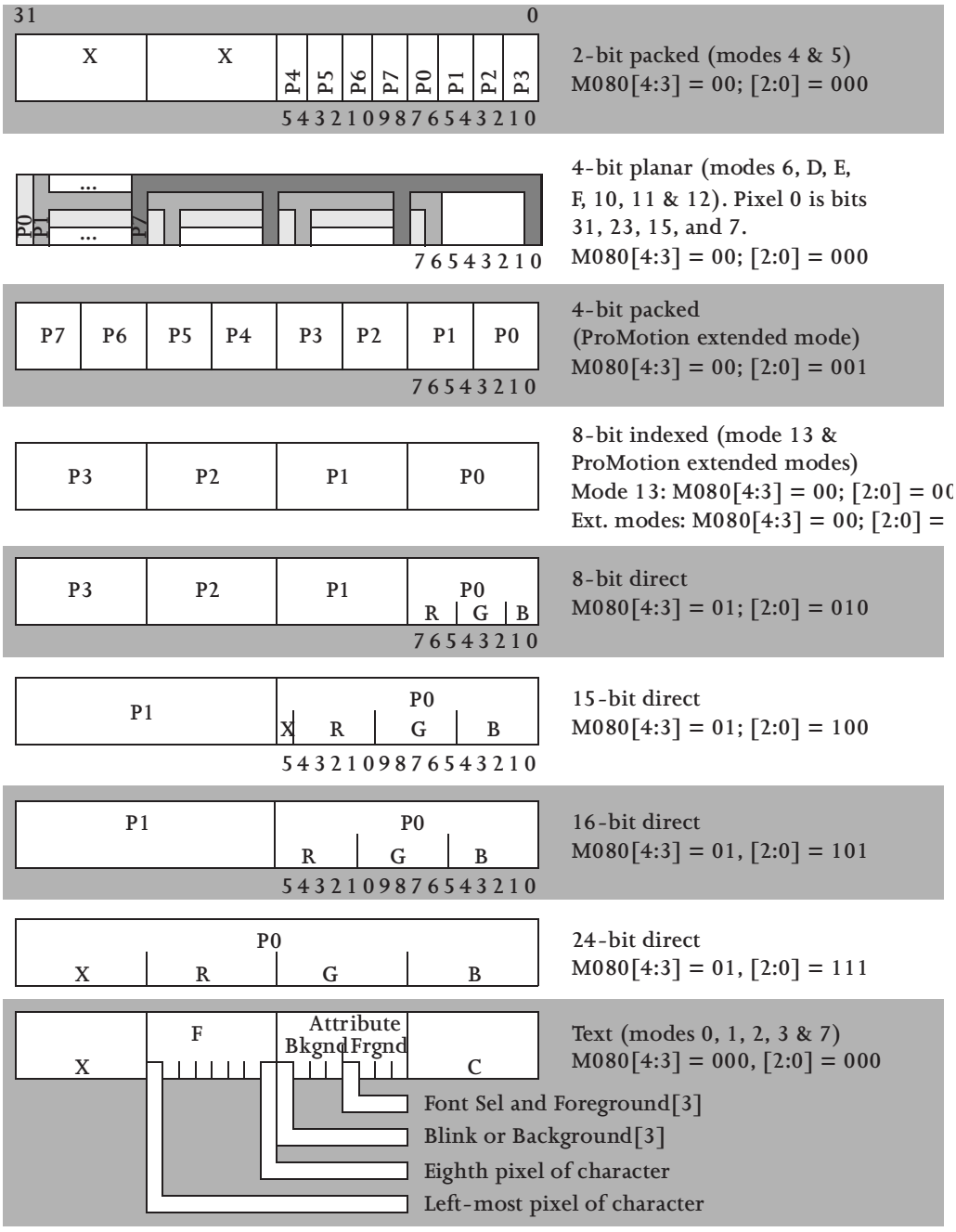
When the motion video window is partly occluded (not fully visible), it may be extremely difficult to implement the above procedure. Therefore, it is recommended in this case to find the single largest visible rectangular area within the motion video window, and to store the entire motion video data stream at that location.

17.2 Memory video formats



17.3 Memory graphics formats

For strict VGA mode set all M0C8–0C9 (VGA Override) bits to 0.



18. Appendices

18.1 Glossary

aperture	Subregion of display memory currently accessible to the host through a fixed access area. Sometimes refers to the size of the aforementioned region: in VGA usually 64K or 128K.
banked	A memory arrangement by which two or more "banks" of memory chips are addressed simultaneously, and one bank is then selected to be read or written. Banked memories usually fall into one of two arrangements: interleaved and non-interleaved.
character clock	Pixel clock divided by 8 or 9 (PCLK divided by the pixel width of characters).
chroma correction	Complete color control of motion video window(s), independent of the graphics desktop. Chroma correction is an additional hardware LUT in the ProMotion DAC designed specifically for motion video.
Cinepak	Motion video compression owned by SuperMac, competing with MPEG.
CLUT	Color Look Up Table.
CLUT8	An 8-bit per pixel indexed mode Color Look Up Table
codec	A contraction of "coder-decoder," a piece of hardware or software that can encode data to some other format (for instance JPEG) on the way out, and decode JPEG data on the way in. The term is sometimes used loosely in place of "decoder."
color space conversion	The act of converting a pixel between RGB and YUV color spaces. In graphics controllers, generally refers to hardware translation of YCrCb to RGB.
DCI	Display Control Interface. A joint Microsoft/Intel specification for an interface between Video for Windows and other rmotion video applications, and specific video hardware.
DCT	Discreet Cosine Transform. A mathematical operation at the heart of JPEG and MPEG compression, converting a group of pixels from the color domain to the frequency domain.
DDA	Digital Differential Analyzer. A mathematical operation used for determining vectors for line draw operations.
DDC	Display Data Channel, a VESA standard by which PCs comminicate with their monitors. At a minimum, a monitor sends information about itself and its capabilities to the host. Some monitors can also be control led through DDC (ie. the host can adjust the monitor brightness, etc.)
direct color	A scheme by which the pixel value directly specifies red, green, and blue components of the color to be displayed without an intervening CLUT. Exaple: a 16-bit direct color pixel may specify 5 bits of red intensity, 6 bits of green intensity, and 5 bits of blue intensity.
dithering	A technique by which the appearance of an unavailable color is created by using a number of similar but available colors in a cluster of adjacent pixels.
DitherFill™	A 4×4×4 pattern, limited to Windows' standard colors, which can be applied to drawing operations. Refer to "Pattern," on page 193.
DPMS	Display Power Management Signaling.
filtering	In motion video stretching, refers to interpolation.

gamma correction	Gamma Correction is a technique used in professional-quality graphics and imaging systems to correct for the non-linear characteristics of display devices and the human eye. Linear color values are adjusted in a non-linear way to compensate for the fact that monitors themselves have non-linear response. Color values 00 and FF s till translate to the minimum and maximum brightness levels, but a pixel with a value of 80h (halfway between 00 and FF) would appear on the screen as considerably less than half brightness, so it's adjusted with a gamma corrected value that would produce half brightness on the screen.
hi color	Generally 16-bit per pixel direct color, often associated with the Sierra RAMDACs.
Indeo	Intel format for compressing motion video.
indexed	In a conventional VGA, a 4-bit or 8-bit pixel is looked up in a 16-entry or 256-entry palette (or CLUT) which specifies the RGB color to be displayed for that piuxel. There is no visible relationship between similar pixel values; for exanple, color 10 might be blue and color 11 might be pink.
interleaved memory	contiguous memory is accessed by reading a location within one bank, then reading the same location from another bank. This is often done for performance reasons: switching between banks is sometimes faster than fetching the data in the first place, so the "slow" process of fetching data is done just once to all banks in parallel.
interpolation	A method of stretching by which the pixels needed to fill a stretched image are calculated from nearby original pixels.
linear access	A non-VGA manner of accessing display memory, in which the entire display memory is available to the host at the same time, as one contiguous memory region.
linear addressing	A straight pixel count from the top-left corner of display memory, used as an alternative to X/Y addressing. Linear Addressing is particularly useful for off-screen memory operations.
MPEG	Motion Picture Experts group. A standar means of motion video compression.
non-interleaved memory	each memory bank is a contiguous region of data. Banking was done simply because there is more memory than there are address pins available.
occlusion	The act of one window "occluding", or making part of another window invisible by being "on top" of it.
packed	A display memory architecture in which the bits which comprise a single pixel are packed into one or more bytes such that the entire pixel can be accessed at one time.
pixel replication	A method of stretching by which original pixels are repeated a certain number of times to fill a larger area.
planar	A display memory architecture in which the bits that comprise a single pixel are accessed at the same bit position but each at different memory locations. In the specific case of VGA: at the same memory location but in different maps which must be accessed separately.
RGB	A color space in which each pixel is described by its Red, green, and Blue components.
RGB332	An 8-bit per pixel direct color mode. D[7:5]=R, D[4:2]=G, D[1:0]=B.
RGB555	A 16-bit per pixel direct color mode. D[15]=X, D[14:10]=R, D[9:5]=G, D[4:0]=B.
RGB565	A 16-bit per pixel direct color mode. D[15:11]=R, D[10:5]=G, D[4:0]=B.

RGB888	A 32-bit per pixel direct color mode. D[31:24]=X, D[23:16]=R, D[15:8]=G, D[7:0]=B.
RLE	Run Length Encoding. A simplistic form of still image compression.
scaling	In motion video, the act of enlarging a motion video image to an arbitrary size. Same as stretching
smooth scaling	Scaling with interpolation, as opposed to pixel replication.
stretch	In motion video, the act of enlarging a motion video image to an arbitrary size. Same as scaling.
stride	The number of bytes (or some other unit of measure) between adjacent rows in memory. For instance, in an 800x600 16-bit per pixel system, the display memory stride would be at least 1600 bytes, but if software chose to "pad out" each row to 1024 (by adding 224 invisible pixels on the right side) the stride would be 2K bytes.
tiling	A technique by which a partially occluded window, in which the visibel portion cannot be described as a rectangle, is decomposed into a series of ajoining windows, or tiles, each of which is fully visible and rectangular.
true color	Generally a 24-bit per pixel direct color with 8 bits of intensity for each R, G, B.
VAFC	VESA Advanced Feature Connector.
VSVPC	VESA Standard VGA Pass-through Connector.
vWindow™	Hardware motion video window, optionally scaled and color-space converted, displayed anywhere on the graphics desktop.
watermark	A generic term for a location in memory above which memory is treated differently in some way. Also could refer to a location in a FIFO, when you reach this location, something happens.
X/Y addressing	Coordinate addressing system, used as an alternative to linear addressing. X/Y addressing is particularly useful for on-screen memory operations.
YCrCb	A color space similar to YUV and occasionally used to refer to YUV.
YUV400	An 8-bit per pixel mode. D[7:0] = greyscale, where R=G=B=Y. Each byte is a pixel.
YUV411	A 16-bit per pixel mode. D[31:24]=U, D[23:16]=V, D[15:8]=Y1, D[7:0]=Y0. U and V are shared between pixels 0 and 1.
YUV422	A 16-bit per pixel mode. D[31:24]=Y1, D[23:16]=V, D[15:8]=Y0, D[7:0]=U. U and V are shared between pixels 0 and 1.

18.2 Windows 3.11 driver configuration

Include the following entties in the SYSTEM.INI file under a heading of [ProMotion]. These entries allow users to disable certain features of the ProMotion controller. ProMotion driver software features are enabled in software if it finds a 0 (zero) or no entry after the entry, ProMotion driver software disables a particular feature if there is a 1 after the entry.

SYSTEM.INI entry	*Feature*
DisableBltPort=	Disable BLT port.
DisCdCd=	Disable color screen to color screen.
DisCmCd=	Disable color memory to color screen.
DisCdCm=	Disable color screen to color memory.
DisCd1m=	Disable color screen to mono memory.
Dis1mCd=	Disable mono memory to color screen.
DisPolyLine=	Disable poly line.
DisScanFill=	Disable scan fill.
DisScanline=	Disable scan line.
DisInitScanline=	Disable initialization scan line.
DisBuildString=	Disable build string.

18.3 Windows95 Driver configuration

18.3.1. INTRODUCTION

This document describes the various switches that control the functionality of the Windows 95 driver and the Alliance ProMotion Multimedia Accelerators. These features are functional beginning with Windows driver version 4.03.00.1101

The driver uses switch settings from the file PROMTN.INI.

18.3.2. PROMTN.INI description

The paragraph headings in PROMTN.INI:

[CRTC Registers]

[Current Refresh Rates]

[FIFO Overrides]

[HW Acceleration Settings]

[Max Refresh Rates]

[Override Refresh Rates]

[ProMotion]

[Resolution Preferences]

18.3.3. CRTC Registers

The entry is mainly use to center the screen by modifying certain CRTC registers as follows.. Blank CRTC entries or a missing entry altogether for a particular screen resolution indicates that the driver uses the default settings. Each register value assigned is entered in hexadecimal byte. The entry format is as follows: [x resolution]x[y resolution]x[refresh rate]=[Horizontal Retrace Start(3D5.4)],[Horizontal Retrace End(3D5.5)],[Vertical Overflow(3D5.7)],[Vertical Retrace Start(3D5.10)],[Vertical Retrace End(3D5.11)].

Example:

[CRTC Registers]

1024x768x75=a2,14,52,00,03

18.3.4. Current Refresh Rates

The entry sets the refresh rate for a specific resolution. A blank refresh rate entry or a missing entry altogether indicates that the driver uses the recommended refresh rate specific to a monitor type. The entry format is as follows: Mode_[x resolution]x[y resolution]=[refresh rate in decimal].

Example:

[Current Refresh Rates]

Mode_1152x864=60

18.3.5. FIFO Overrides

The driver is configured to support FIFO settings when MCLK is 50.0 MHz. If the OEM sees the need to adjust these settings for a given user or users, they can override the driver FIFO settings by placing the appropriate values in this section. Use extreme caution when modifying these values, as Alliance assumes no responsibility what-so-ever if OEM's change these values with out first consulting with Alliance. Each FIFO value is entered in decimal. The entry format is as follows: [x resolution]x[y resolution]x[refresh rate]=[8bpp high priority request point, page break],[8bpp high priority request point, no page break],[8bpp low priority request point], [16bpp high priority request point, page break],[16bpp high priority request point, no page break],[16bpp low priority request point], [32bpp high priority request point, page break],[32bpp high priority request point, no page break],[32bpp low priority request point].

Example:

[FIFO Overrides]

640x480x60=1,0,16,3,2,17,10,6,21

18.3.6. HW Acceleration Settings

(WARNING: This is not properly supported in the current driver.)

These entries are for debug purposes ONLY. The entries disable HW acceleration for the following graphics engine functions. Zero(0) or no entry for a particular command indicates that the command is performed by hardware. An entry of one(1) for a particular command indicates the command is performed by software (DIB engine.)

[HW Acceleration Settings]

DisableBltPort= Disable Blt port

DisCdCd=Disable color screen to color screen Blt

DisCmCd=Disable color memory to color screen Blt

DisCdCm=Disable color screen to color memory Blt

DisCd1m=Disable color screen to mono memory Blt

Dis1mCd=Disable mono memory to color screen Blt

DisPolyLine= Disable poly line

DisScanFill= Disable scan fill

DisScanline=Disable scan line

DisInitScanline=Disable initialization scan line

DisBuildString=Disable build string

18.3.7. Max Refresh Rates

Tentatively; to be removed in later driver versions.

The entry allows the OEM to override the maximum refresh rates per resolution, per color depth. Each refresh rate value is entered in decimal. The entry format is as follows: mode_[x resolution]x[y resolution]=[8bpp maximum resolution], [16bpp maximum resolution], [32bpp maximum resolution]. The following entries are recognized:

[Max Refresh Rates]

mode_640x480= [xx,][yy,][zz]

mode_800x600=[xx,][yy,][zz]

mode_1024x768=[xx,][yy,][zz]

mode_1152x864=[xx,][yy,][zz]

mode_1280x1024=[xx,][yy,][zz]

mode_1600x1200=[xx,][yy,][zz]

To skip a color depth, a comma must be supplied. For example to set the refresh rates for 640x480 at 8bpp to 72 Hz and 32bpp to 75 Hz the entry would look like this:

mode_640x480=72,,75

18.3.8. Override Refresh Rates

The entry determines whether or not to override the list of available refresh rates specific to a monitor setting. The override enables the use of all valid refresh rates in the driver table regardless of the monitor setting. The entry format is as follows: Override_[x resolution]x[y resolution]x[color depth in bits per pixel]=[1 or 0/blank]. Setting the entry to one (1) enables the override of refresh rates. A blank or zero (0) or a missing entry altogether disables the override of refresh rates for the particular display mode.

Example:

[Override Refresh Rates]

Override_1024x768x8=1

18.3.9. ProMotion

The paragraph contains miscellaneous entries. They are described as follows:

[ProMotion]

Entry	Feature
HorInterp	If set = 0, Horizontal Interpolation is disabled for AT24.If set = 1 or blank entry, Horizontal Interpolation is enabled for AT24. Default setting: 1
NumberOfOverlays	Sets up to 2 video overlays; default setting: 1
VerInterp	If set = 0, Vertical Interpolation is DISABLED for AT24. If set = 1, Vertical Interpolation is ENABLED for AT24 If set = 2 (or blank entry), Vertical Interpolation is ENABLED for VClk > 60MHz, Vertical Interpolation is DISABLED for VClk < 60MHz. Default setting: 2
VESATimings	Enables the VESA timings table (for driver versions <= 4.1.3403, ☞ This value should always be set to 1 by the user for the AT24.) Set to 1 to enable; set to 0 to disable. Default setting: VESA timings are disabled, except for AT24 and later controllers.
VisibleResolution	Sets the visible resolution (view port.) This switch is intended to support virtual desktop, where the desktop size is greater than the view port size. Virtual desktop is ONLY supported in driver version 4.03.0.1097 and later versions. The following are valid visible resolutions to use: 640x480, 800x600, 1024x768, 1152x864 and 1280x1024. Default setting: view port size = desktop size. Example: VisibleResolution=800x600
BaseAddress	To be used ONLY to debug 3210; sets PCI base address for 3210. Valid entry is a 32-bit hexadecimal value. Default setting: set by system BIOS. Example: BaseAddress=fe000000.
CursorColor1	To be used ONLY to debug cursor color; valid entries are 3-byte decimal values for cursor color 1, 2 and 3 (see memory-mapped registers 141h, 142h and 143h.) Default setting: 255,255,255 Example: CursorColor1=255,255,255
HWOptionB	To be used ONLY for debugging; sets bursts capability for AT24 Rev. C. Set to 1 to enable; set to 0 to disable. Default setting: burst disabled.
IgnoreVsync	To be used ONLY for debugging; set to 1 to enable (for example, ignore vertical sync during a display page flip;) set to 0 to disable.

18.3.10. Resolution Preferences

The entry determines whether or not a specific resolution is included as part of available resolutions to set. The entry format is as follows: Res_[x resolution]x[y resolution]=[0 or 1 or blank]. A zero (0) entry disables the resolution. A blank or one (1) entry or a missing entry altogether enables the resolution. Exception: Mode 1600x1200x8 on a 2 MB board is disabled by default; this mode may be enabled by setting Res_1600x1200=1.

Example:

[Resolution Preferences]

Res_1152x864=0

18.3.11. Dynamic MCLK setting for VG96

The Win95/Direct Draw driver now has the ability to switch between two settings of the MCLK frequency. One will be used while the VG96 is active, and the other while it is inactive. Normally the setting for VG96 inactive will be a higher clock frequency. These fast and slow mclk settings are defined either in the registry, if the driver is compiled for that, or in the PROMTN.INI file.

For example in promtn.ini:

[HW Acceleration Settings]

MCLKFrequency=65

VG96_MCLKFrequency=50

At Win95 boot time, the MCLK will be changed to the value defined by MCLKFrequency, in this example 65 Mhz, from whatever it was initialized to by the BIOS. Then, when the VG96 becomes active on the PUMA by means of the invocation of a Direct 3D application, the frequency will be reset to that defined by VG96_MCLKFrequency, in this example 50 Mhz. When the VG96 is no longer active on the PUMA, the frequency will be once again set to that defined by MCLKFrequency.

The available MCLK frequency settings in either mode are: 25, 40, 45, 50, 55, 60, 62, 65, 67, 70, 72, 75, 80, 85, 90, 95, 100 Hz.

18.4 ProMotion stepping information

This information is provided to aid developers in detecting the revision levels of ProMotion controllers.

The Alliance ProMotion Family Plug-and-Play ID is "ALL". However, this PnP ID is not needed for PCI cards, which use PCI ID to identify vendor.

Table 18.4 Registers used to identify controller revisions

	Register		
Controller-revision	3C5.11-19h	M188h (PCI 08)	M0D8h†
3210-b*	0000 0000	0000 0001	
6410-c		0000 0010	
6410-d		0000 0011	
6422-f			0000 0000
6422-g			0000 0001
6422-h			0000 0011
AT24-c	0000 0001		
AT24-e	0000 0010		
AT3D-ES3	0000 0000	0000 0000	
AT3D-a	0000 0001	0000 0001	
AT3D-b	0000 0002	0000 0002	

* Both 3C5.11-19h and M188h must be read to identify this controller revision.

† This register is used to detect revision levels for only the 6422.

Distinguish between AT24-e and AT24-f by lot number via markings on every package. Rev E lots have these markings: M9F60, MAB67, MAC62, MAG53, MAG54, MAN76, MAN80, or MAN81. All other AT24 lots with 0000 0010 in registers 3C5.11-19h are Rev F.

18.5 Recommended 3Dfx THP interface

Recommended connector:

- simple pin headers
- 0.1" spacing
- two-row for left and three-row for right
- shrouded or unshrouded.
- typically through-hole rather than surface mount

For memory-only upgrade, both daughtercard connectors should be two-row. A two-row female right connector fits between rows of three-row male.

Figure 18.5. THP connector diagram

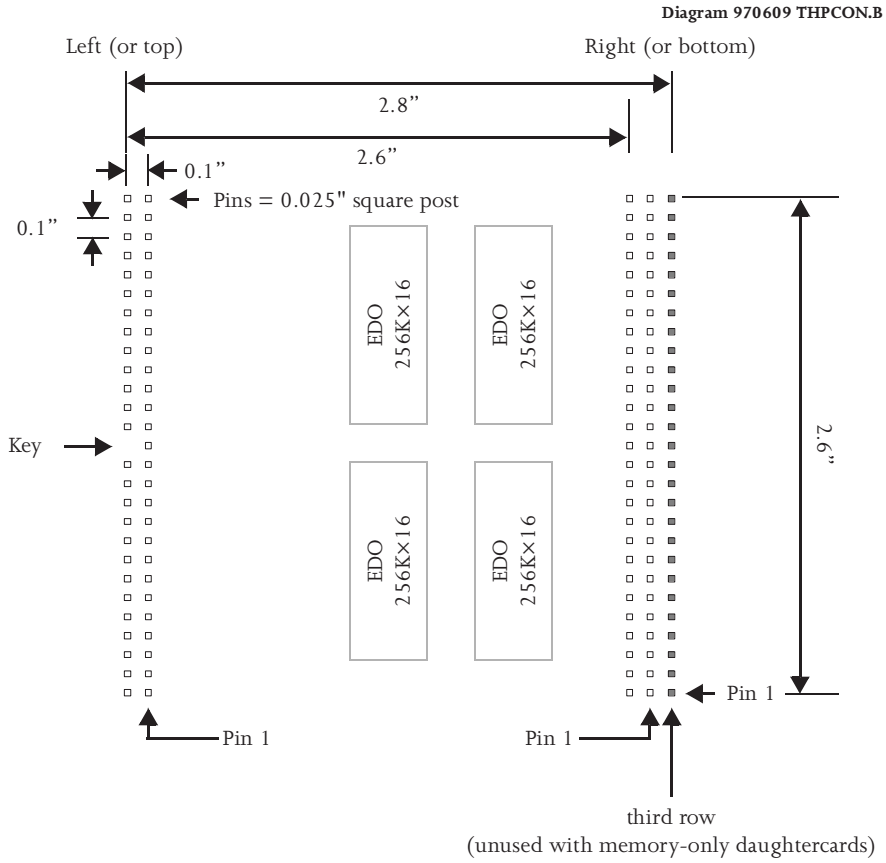


Table 18.5 Recommended pinout THP connector (Rev. 3)

Left connector				Right connector				5th rail	
54	MD[09]	VCC	53	54	VCC	MD[32]	53	GND	27
52	GND	MD[08]	51	52	MD[33]	GND	51	VCC	26
50	MD[11]	MD[10]	49	50	MD[35]	MD[34]	49	NC	25
48	MD[13]	MD[12]	47	48	MD[37]	MD[36]	47	NC	24
46	MD[15]	MD[14]	45	46	MD[39]	MD[38]	45	GND	23
44	MD[17]	MD[16]	43	44	MD[56]	MD[58]	43	GND	22
42	GND	MD[18]	41	42	MD[57]	GND	41	GND	21
40	MD[19]	MD[20]	39	40	MD[59]	MD[60]	39	$\overline{WE}$[0]	20
38	MD[21]	MD[22]	37	38	MD[61]	MD[62]	37	GND	19
36	MD[23]	reserved	35	36	MD[63]	NC/BS/A9†	35	$\overline{OE}$[0]	18
34	$\overline{WE}$[1]/CAS†	$\overline{CAS}$[4]/ DQM.B4†	33	34	A8	A0	33	GND	17
32	GND	$\overline{CAS}$[7]/ DQM.B7†	31	32	A6	GND	31	NC	16
30	$\overline{CAS}$[2]/ DQM.B2†	$\overline{CAS}$[1]/ DQM.B1†	29	30	VCC	A5	29	VCC	15
28	KEY	$\overline{CAS}$[5]/ DQM.B5†	27	28	$\overline{OE}$[1]	A4	27	VCC	14
26	$\overline{CAS}$[6]/ DQM.B6†	$\overline{CAS}$/ DQM.B3†[3]	25	26	A3	A2	25	VCC	13
24	$\overline{CAS}$[0]/ DQM.B0†	VCC	23	24	A1	A7	23	VCC	12
22	GND	MD[06]	21	22	$\overline{RAS}$[1]	GND	21	VSYNC	11
20	MD[07]	MD[04]	19	20	$\overline{RAS}$[0]	MD[40]	19	HRESET	10
18	MD[05]	MD[02]	17	18	MD[41]	MD[42]	17	SERIAL_IN	9
16	MD[03]	MD[00]	15	16	MD[43]	MD[44]	15	SRESET	8
14	MD[01]	MD[24]	13	14	MD[45]	MD[46]	13	SWAP	7
12	GND	MD[25]	11	12	MD[47]	GND	11	NC	6
10	MD[27]	MD[26]	9	10	MD[49]	MD[48]	9	3REQ	5
8	MD[29]	MD[28]	7	8	MD[51]	MD[50]	7	3GNT	4
6	MD[31]	MD[30]	5	6	MD[53]	MD[52]	5	GND	3
4	NC/WE†	VCC	3	4	VCC	MD[54]	3	3CLK	2
2	GND	NC/CLK	1	2	MD[55]	GND	1	GND	1

† = SGRAM signals, provided for future compatibility only, not for current controllers. SGRAM CS connected to GND, CKE to Vcc.

Updated 961212

18.6 Recommended VMI+ interface

Table 18.6.1VMI+ input port pin description

Pin	Feature Connector	VMI+	AT3D	AT3D pin	Pin	Feature Connector	VMI+	AT3D	AT3D pin
Z1	GND	GND	-	-	Y1	P0	VID[0]	P0	82
Z2	GND	GND	-	-	Y2	P1	VID[1]	P1	81
Z3	GND	GND	-	-	Y3	P2	VID[2]	P2	78
Z4	$\overline{\text{EVIDEO}}$	VACTIVE	$\overline{\text{EVIDEO}}$	66	Y4	P3	VID[3]	P3	77
Z5	$\overline{\text{ESYNC}}$	user	XODD	68	Y5	P4	VID[4]	P4	76
Z6	EDCLK	VREF	VREF	67	Y6	P5	VID[5]	P5	75
Z7	NC	I^2CCLK	SCL	99	Y7	P6	VID[6]	P6	74
Z8	GND	GND	-	-	Y8	P7	VID[7]	P7	73
Z9	GND	GND	-	-	Y9	DCLK	-	PIXCLK	70
Z10	GND	GND	-	-	Y10	BLANK	HREF	HREF	72
Z11	GND	GND	-	-	Y11	HSYNC	NC	$\overline{\text{HSYNC}}$	64
Z12	NC	user	-	-	Y12	VSYNC	NC	VSYNC	65
Z13	NC	I^2CDAT	SDA	98	Y13	GND	GND	-	-

Connector A: 26-pin dual-row receptacle; 0.100 inch centers

Table 18.6.2VMI+ host port pin description

Pin	VMI+	AT3D	AT3D pin	Pin	VMI+	AT3D	AT3D pin
Z1	+12V	-	-	Y1	HD[0]	MD[0]	18
Z2	HD[1]	MD[1]	17	Y2	GND	-	-
Z3	GND	-	-	Y3	HD[2]	MD[2]	16
Z4	HD[3]	MD[3]	15	Y4	HD[4]	MD[4]	14
Z5	+5V	-	-	Y5	HD[5]	MD[5]	13
Z6	HD[6]	MD[6]	12	Y6	HD[7]	MD[7]	11
Z7	OSC	-	-	Y7	HA[0]	MD[32]	192
Z8	HA[1]	MD[33]	191	Y8	HA[2]	MD[34]	190
Z9	HA[3]	MD[35]	189	Y9	+5V	-	-
Z10	GND	-	-	Y10	$\overline{\text{RESET}}$	RESET	62
Z11	$\overline{\text{CS}}$	$\overline{\text{CS}}$	100	Y11	GND	-	-
Z12	$\overline{\text{RD}}$	$\overline{\text{RD}}$	101	Y12	$\overline{\text{WR}}$	-	102
Z13	+3.3V	-	-	Y13	READY	READY	103
Z14	SCLK	-	-	Y14	$\overline{\text{INTREQ}}$	$\overline{\text{IRQA}}$	84
Z15	LRCK	-	-	Y15	PCMDATA	-	-
Z16	+5V	-	-	Y16	+3.3V	-	-
Z17	user	-	-	Y17	user	-	-

Connector B: 40-pin dual-row receptacle; 0.100 inch centers

Table 18.6.2VMI+ host port pin description

Pin	VMI+	AT3D	AT3D pin	Pin	VMI+	AT3D	AT3D pin
Z18	user	-	-	Y18	key	-	-
Z19	INSERT	-	-	Y19	AUDIOL	-	-
Z20	AUDGND	-	-	Y20	AUDIOR	-	-

Connector B: 40-pin dual-row receptacle; 0.100 inch centers

18.7 Promotion-AT3D extended memory map

This appendix specifies the new areas of PCI address space created for access to new functionality, within the 16MB of space allocated by the system to the AT3D. There is also provision for the new functionality to appear in the 0xA0000 VGA aperture when enabled to do so, which is also described here.

18.7.1. Linear space

Offset from frame buffer base address	Size	Functionality	Notes
16MB - 1K (0xFFFC00)	1K	PISA1	1, 8
16MB - 2K (0xFFF800)	1K	PISA0	1, 8
16MB - 3K (0xFFF400)	1K	TLUT - 3D texture lookup table RAM.	1, 8
16MB - 4K (0xFFF000)	1K	MMVGA - memory mapped access to VGA registers which are normally in I/O space.	5, 9
16MB -5K (0xFFEC00)	1K	ATxx extended register space 1	
12MB	4MB-5K	co-processor memory space aperture 2	6, 8, 10
8MB	4MB	linear frame buffer, aperture 2	6
4MB	4MB	coprocessor memory space , aperture 1	7,8,10
nMB-2K	2K	ATxx extended register space 1	2,3
nMB-32K	30K	Host BLT port	2,4
0	nMB-32K	linear frame buffer	

1 Enabled when ExtendedMemoryEnable[1] = 1
2 n = 1,2 or 4, depending on whether 3C5.1C[2:1] = 0, 1 or 2 respectively.
3 This is the original memory-mapped register area up to and including AT24. Enabled when 3C5.1B[2:0] = 4
4 Enabled when 3C5.1B[5:3] = 4
5 Example: io space 0x304 => mem space (linear base address + 0xFFF3C4)
6 Enabled when ExtendedMemoryEnable[3] = 1
7 Enabled when ExtendedMemoryEnable[2] = 1
8 AT3D only
9 LDEV wait states register field (0xD9[5:4]) must be programmed to value 2 in order to access this space.
10 Flat Model Aperture field (0x3C4,1C[2:1]) must be set equal to binary 11 in order to access this space.

18.7.2. DOS memory space base 0xA0000, size 64K

Offset from 0xA0000	functionality	Size	Notes
63K (0xFC00)	PISA1	1K	11
62K (0xF800)	PISA0	1K	11
61K (0xF400)	TLUT	1K	11
60K (0xF000)	MMVGA	1K	11
4K	Host BLT port	28K	12
0	ATxx extended register space 1	4K	13

11 Enabled when ExtendedMemoryEnable[0] = 1
12 Enabled when 3C5.1B[5:3] = 1
13 Enabled when 3C5.1B[2:0] = 1

18.7.3. DOS memory space base 0xB0000 , size 32K

Offset from 0xB0000	Functionality	Size	Notes
4K	Host BLT port	28K	14
0	ATxx extended register space 1	4K	15

14 Enabled when 3C5.1B[5:3] = 2
15 Enabled when 3C5.1B[2:0] = 2

18.7.4. DOS memory space base 0xB8000 , size 32K

Offset from 0xB8000	Functionality	Size	Notes
4K	Host BLT port	28K	16
0	ATxx extended register space 1	4K	17

16 Enabled when 3C5.1B[5:3] = 3
17 Enabled when 3C5.1B[2:0] = 3

18.7.5. "Enable extended registers," described on page 235

Bit	Function
0	Enable Extended registers - DOS space
1	Enable Extended registers - linear space
2	Enable coprocessor apertures
3	Enable second linear aperture

Offset: 0xDB in ATxx extended register space 1

INDEX

Numerics

A

B

C

D

E

F

G

H

I

L

M

O

P

Q

R

S

T

U

V

W

X

Y