

# P9/P10

## *Architecture Overview*

PROPRIETARY AND CONFIDENTIAL  
INFORMATION



---

Issue 2

---



---



---

## Proprietary Notice

---



---

The material in this document is the intellectual property of 3Dlabs®. It is provided solely for information. You may not reproduce this document in whole or in part by any means. While every care has been taken in the preparation of this document, 3Dlabs accepts no liability for any consequences of its use. Our products are under continual improvement and we reserve the right to change their specification without notice. 3Dlabs may not produce printed versions of each issue of this document. The latest version will be available from the 3Dlabs web site.

3Dlabs products and technology are protected by a number of worldwide patents. Unlicensed use of any information contained herein may infringe one or more of these patents and may violate the appropriate patent laws and conventions.

3Dlabs ® is the worldwide trading name of 3Dlabs Inc. Ltd., a division of Creative Technologies Ltd.

3Dlabs, GLINT, GLINT Gamma, Permedia, Oxygen and Wildcat are trademarks or registered trademarks of 3Dlabs Ltd., 3Dlabs Inc. Ltd or 3Dlabs Inc.

Microsoft, Windows and Direct3D are either registered trademarks or trademarks of Microsoft Corp. in the United States and/or other countries. OpenGL is a registered trademark of Silicon Graphics, Inc. All other trademarks are acknowledged and recognized.

© Copyright 3Dlabs Inc. Ltd. 2003. All rights reserved worldwide.

Email: [info@3dlabs.com](mailto:info@3dlabs.com)  
 Web: <http://www.3dlabs.com>



### European Headquarters

3Dlabs Ltd.  
 Meadlake Place  
 Thorpe Lea Road, Egham  
 Surrey, TW20 8HE  
 United Kingdom  
 Tel: +44 (0) 1784 470555  
 Fax: +44 (0) 1784 470699

### Japan Office

Level 16.  
 Shiroyama Hills  
 4-3-1 Toranomom  
 Minato-ku, Tokyo, 105, Japan  
 Tel: +81-3-5403-4653  
 Fax: +91-3-5403-4646

### Corporate Headquarters

Creative Labs  
 1901 McCarthy Blvd.  
 Milpitas, CA 95035  
 Tel: +1 408-432-6700  
 Fax: +1 408-432-6701

### 3Dlabs US

9668 Madison Blvd.  
 Madison,  
 Alabama 35758  
 Tel: 877 286 1185 (Freephone)  
 Tel: +1 256 319 1100

---

---

## Change History

---

---

Document	Issue	Date	Change
174.1.1 01	1	25 Jun 2001	Creation
174.1.1 01	2	06 Nov 2003	Progressive updates

---



---

## Table of Contents

---



---

<b>1</b>	<b>INTRODUCTION</b>	<b>5</b>
1.1	Introduction	5
1.2	Target Markets	6
1.3	Design Characteristics and Performance	6
1.4	Embedded Application Support Program	8
1.5	Changes from Earlier Architectures	8
1.5.1	Tile-based working	8
1.5.2	Multitasking and extended programmability	9
1.5.3	Command input and Real time rendering	9
1.6	Flexible memory implementation	10
1.6.1	Virtual Memory	10
1.6.2	Physical Characteristics	11
1.7	I/O and Bus Support	11
1.8	Chip Level Block Diagram (P10)	12
<b>2</b>	<b>CORE FEATURE OVERVIEW</b>	<b>2-1</b>
2.1	3D Graphics	2-1
2.2	2D Graphics	2-2
2.3	MPEG2	2-2
<b>3</b>	<b>ARCHITECTURAL ENHANCEMENTS</b>	<b>3-2</b>
3.1	Host Interfaces - AGP/PCI	3-2
3.1.1	PCI Interface	3-2
3.1.2	AGPBus	3-2
3.2	Transform and Lighting System	3-3
3.3	FIFO and Memory Interface Enhancements	3-4
3.3.1	Primitive Setup system	3-4
3.4	Rasterizer	3-5
3.4.1	Video Operations Without Dedicated Hardware	3-5
3.4.2	Geometry Rasterizer	3-5
3.5	Cacheing Enhancements	3-6
3.6	Routing, Depth and GID	3-6
3.7	Texture and Depth Processing	3-6
3.7.1	Texture Indexing and Filtering	3-6
3.8	Shading Unit	3-6
3.9	Pixel Processing	3-7
<b>4</b>	<b>VIDEO UNIT AND RAMDAC</b>	<b>4-1</b>
4.1	Overview	4-1
4.1.1	Pixel Formats	4-1
4.1.2	Scaling	4-2
4.1.3	Synchronization and Genlock	4-2
4.1.4	Clocks and PLLs	4-2
4.1.5	Digital Port Control	4-3
4.2	Software Drivers	4-3
4.2.1	ROM support and SVGA BIOS	4-4

4.2.2	Display Resolutions	4-4
4.2.3	Video Overlay	4-5
4.3	Video Input Port (VIP)	4-5
4.3.1	Video Stream Formats	4-5
<b>5</b>	<b>POWER AND THERMAL MANAGEMENT</b>	<b>5-6</b>
5.1	Power Consumption [P10]	5-6
5.2	Power Consumption [P9]	5-7
5.3	Power Management Features	5-7
5.4	Thermal Performance	5-7
<b>6</b>	<b>DATA SHEET</b>	<b>6-1</b>

## 1

## Introduction

## 1.1 Introduction

The P10 family of graphics parts (P9, P10 and P20) break with tradition and implement an innovative new architecture design concept – the Visual Processing Unit (VPU).

The P10 VPU<sup>1</sup> leads the industry in anticipating demanding new APIs such as DX9 and OpenGL 2.0; and hardware capabilities such as multi data-rate memories, high-resolution cinematic monitors, very large virtual textures, increasingly demanding lighting techniques and fast context switching for Longhorn-style multiple virtual VPU windows.

Conventional fixed-function registers and cycle-per-fragment pipeline designs were unable to deliver the flexibility and pipeline performance required. P10 solves these problems with fully programmable T&L and pixel shaders in conjunction with highly optimized fixed-function units to achieve a clean, fast and versatile design.

Significant architectural changes include:

- **Programmability:** Programmable registers allow dynamic reconfiguration of the number of vertex shaders, the number of texture pipes and the number of rasterizers per chip to deliver the greatest possible throughput under changing task conditions.
- **Scalar Array Vertex Processor:** Use of a scalar array of 16 32-bit SIMD processors as opposed to the traditional Vec4 vertex handlers allows multiple vector instructions per cycle, improves efficiency for non Vec4 operations and provides a compiler-friendly and natively parallel compilation.
- **Distributed Programmability:** throughout the VPU pipeline to support high-level shading languages, convolutions and other user-defined applications.
- **Optimisation:** Fixed-function registers for specialised tasks have been optimised for simplicity and speed with hand-polished main routines and the removal of legacy code.
- **Memory Bandwidth:** Memory bandwidth and DMA performance have been enhanced with support for high-density multi-data rate memory configurations up to 512MB (with virtual addressing up to 16GB) via a 256-bit bus and low-overhead circular buffers to provide up to 17Gbytes/second peak throughput.
- **Fast Context Switching:** The first graphics industry chip to support hardware multi-threading and complete context switch in 15 microseconds or less, with real-time video switching<sup>2</sup> at 3.5 microseconds (200MHz) for tear-free blitting.

3Dlabs has achieved this without compromising its long-standing commitment to quality 3D rendering. P10 delivers accuracy, stability and full OpenGL compliance while providing a feature-rich device with unparalleled real-world single-chip graphics performance.



**Figure 1 - P10 820-ball thermally-enhanced 37.5mm HSBGA package.**  
(Not shown: P9 644-ball thermally-enhanced HSBGA in 31mm package.)

<sup>1</sup> Except as indicated, "P10" is used to refer to the P9 and P10 VPUs.

<sup>2</sup> Typically on Vblank

## 1.2 Target Markets

P10's programmability and flexibility allow it to address an unusually wide range of market segments. The following application areas are fully supported:

- CAD/CAM/CAE
- Avionics
- Video Design and Editing
- Custom embedding and IP options

## 1.3 Design Characteristics and Performance

Performance data are based on silicon test results for actual parts.

Category	Performance / Capability	P9 <sup>3</sup>	P10 <sup>4</sup>
<b>Supported APIs</b>	OpenGL 1.3 & 2.0, DX8 & DX9, DXVA, XWindows	✓	✓
<b>OpenGL (GLPerf)<sup>5</sup></b>	Lines/sec (Open GL Disjoint Begin/End)	<b>6.35M</b>	<b>6.65M</b>
	Lines/sec (Open GL Strip Begin/End)	<b>12.6M</b>	<b>17.0M</b>
	Triangles/sec (OpenGL Strip, 1 light)	<b>6.0M</b>	<b>8.2M</b>
	Quads/sec (Open GL, 450 quads/Begin-End)	<b>2.26M</b>	<b>2.58M</b>
	CopyPixels/sec (32x32)	<b>66.7M</b>	<b>66.0M</b>
	DrawPixels/sec (RGBA, 512x512)	<b>60.0M</b>	<b>62.5M</b>
	Pixel Fill/sec (flat or smooth, depth test)	<b>1.22G</b>	<b>3.13G</b>
	Pixel Clear/sec (CallList, RGB, 3D, flat, with Z on P10)	<b>1.24G</b>	<b>2.25G</b>
<b>Programmability</b>	Texture SIMDs for programmable texture coord generation and shaders	<b>32x32bit</b>	<b>64x32bit</b>
	Vertex/Geometry SIMDs for accumulation buffering and convolution	<b>4x32bit</b>	<b>8x32bit</b>
	Simultaneous textures/pass		<b>8</b>
	coefficient memory (Vec4s)	<b>128</b>	<b>256</b>
	Vertex Shaders	<b>8</b>	<b>16</b>
	Pixel SIMDs	<b>16x32bit</b>	<b>32x32bit</b>
<b>Antialiasing</b>	Sample rate	<b>18G/sec</b>	<b>42G/sec</b>
	T-buffer FSAA, Quincunx	✓	✓
<b>Context Switch</b>	Full 3D context	<b>&gt;20μs</b>	<b>&gt;20μs</b>
	Real time interrupt	<b>3μs</b>	<b>3μs</b>
<b>Transform and Lighting</b>	Vertex rendering (no depth, texture or lighting), vertices/sec	<b>100M</b>	<b>225M</b>
	Max Textures/primitive <sup>6</sup>	<b>8</b>	<b>8</b>
	Max Accelerated Lights	<b>16</b>	<b>16</b>
	Triangles/sec (1 to 8 infinite lights, Open GL strip, immediate mode)	<b>6.05M to 6.01M</b>	<b>8.19M to 8.16M</b>
	Texels/Sec (Open GL TexImage, Immediate, RGB, ubyte, 64x64)	<b>102M</b>	<b>137M</b>
	Texture Pipes	<b>2</b>	<b>4</b>
<b>Memory</b>	Bandwidth (peak)	<b>6.4GB/s</b>	<b>9.6 GB/S</b>

<sup>3</sup> Based on VP560 PCB

<sup>4</sup> Based on VP970 PCB

<sup>5</sup> Peak performance except where indicated

<sup>6</sup> With any combination of trilinear, 3D, anisotropic filtering, bump mapping or cube mapping.

Category	Performance / Capability		
	Max memory	<b>256</b>	<b>512 MB</b>
	Addressable Memory	<b>4GB<sup>7</sup></b>	<b>16GB</b>
	Memory Bus	<b>128-bit</b>	<b>256-bit</b>
<b>System Bus</b>	AGP 1X, 2X, 4X, 8X	✓	✓
	PCI 33 capable	✓	✓
<b>Video</b>	Dual RAMDACs	<b>370MHz</b> <b>10bit</b>	<b>370MHz</b> <b>10bit</b>
	Stereo Sync	✓	✓
	VIP2 Video Input Port	✓	✓
	I <sup>2</sup> C bus support	✓	✓
	Interlace Mode support	✓	✓
	Max display resolution (analog 60Hz x 32bpp) on each channel	<b>2048 x</b> <b>1536 x</b>	<b>2048 x</b> <b>2048 x</b>
	Max display resolution (digital 60Hz)	<b>1920 x</b> <b>1440</b>	<b>1920 x</b> <b>1200</b>
	Genlock to external sync	✓ <sup>8</sup>	✓
<b>Power Range</b>	Full desktop 3D environment, high-end nominal speed and parallelism	<b>18.5W</b>	
	Custom low-power thermally-sensitive environment	<b>4.56W</b>	
<b>Signal Voltages</b>	AGP 1X, 2X, 4X	<b>1.5V</b>	<b>1.5V</b>
	AGP 1X, 2X	<b>3.3V</b>	<b>3.3V</b>
	PCI 33	<b>5Vtolerant</b>	<b>5V tolerant</b>
<b>API Compatibility</b>	OpenGL (including release 2.0) DX8/DX9 and DXVA XWindows Windows NT4, ME, 9X and 2000 3 <sup>rd</sup> -party support for MacOS, Linux.	✓	✓

Table 1.1 - P9/P10 Capability Overview

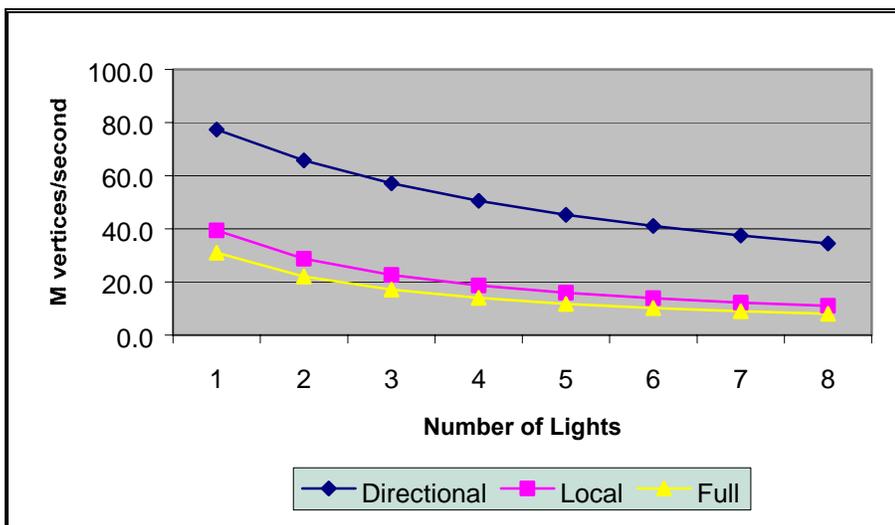


Table 1.3 P10 Lighting Performance

<sup>7</sup> With Virtual Memory enabled, 2GB without VM.

<sup>8</sup> Vertical Sync only

## 1.4 Embedded Application Support Program

The P10 family's highly flexible and compact design encourages embedded use for board, chip and IP solutions ranging from control and monitoring applications to real-time simulation, from medical imaging to test and training equipment, and more. The extensive programmability gives the ability to, for example, perform convolutions, radial gradient fills, even run the "Game of Life" on-chip.

To assist customers wishing to embed P9, P10 or P20 in a proprietary environment, the 3DLabs Embedded Support Program provides different IP and embedding options for our Development partners:

- technical documentation and support
- reference designs
- diagnostic applications
- BIOS ROM configuration tools
- microcode assembler/disassembler manuals and tools
- code samples
- access to 3DLabs driver source code and on-chip microcode (subject to the appropriate licensing)

For high-level API development there are translators for DX8 and OpenGL to P9/P10 source instructions, which include dead code removal, unused variable elimination, stall management, register coloring and other compiler techniques. Programs are assembled with a Dynamic Link Loader and downloaded to chip.

## 1.5 Changes from Earlier Architectures

Because of the extent of P9/P10's paradigm shift a complete list of changes is pointless. However the table below illustrates the areas where developers will find the most extensive innovation.

Previous Rasterizer Chips (P4/R4, MX)	Visual Programming Unit
Scanline Framebuffer	Tiled framebuffer
DDA based interpolators	Plane equations
Edge-walking rasterization	Tile-seeking rasterization
Multiple cycles per primitive	Multiple primitives per cycle
Fixed function units	Fixed/Programmable hybrid
FIFO-based memory	Cache-based memory
Asynchronous pipeline	Parallel pipes with pre-emption
Command and control data visits every unit	Command and control independent routing
Memory-mapped registers	GPIO interprets tag values in DMA stream

**Table 1.1 Evolutionary Changes**

### 1.5.1 Tile-based working

The VPU adopts the tile as its sole unit of internal work. All operations are performed on square screen-aligned planar byte pixel tiles<sup>9</sup> similar to the 64x1 pixel spans used in earlier chips. All data types are stored the same way, so for example anything (e.g. the Depth buffer) can be a texture, and it is possible to render to a texture.

Two or more accesses are used for pixel depths greater than 8 bits, which allows unusual formats such as 24, 40 and 48 bpp. All memory accesses are virtual and page faults are handled with a CPU-like page swap.

This uniformity results in tile scalability and substantial performance improvements, particularly in 3D and small 2D primitives (e.g. characters) where the improved scanline coherence and memory device efficiencies are most noticeable. Performance is further enhanced by the use of DDR memories.

<sup>9</sup> P10 uses 8x8 core tiles, P9 uses 4x4 core tiles.

## 1.5.2 Multitasking and extended programmability

Architecture innovations include the Context unit, which implements pre-emptive multitasking to support time-critical operations such as render during frame blank. The Context unit caches context data and keeps a copy in local memory. A small cache handles frequently updated values such as mode registers. When a context switch is needed the cache is flushed, the new context record is read from memory and the data converted into a message stream to update downstream units. Because only a small amount of cache data needs to be saved this process can be very fast – typically ¼ scanline.

P9/P10 are also the most comprehensively programmable graphics parts in the commercial market today. With over 200 32-bit SIMD processors P10 supports not only the new generation of high-level shading languages (OpenGL2.0, Direct3DX) but multipass convolutions, mathematical routines etc. in a compiler-friendly scalar array environment.

Multitasking when combined with P9/P10's extensive programmability provides powerful new abilities including, for example, reprogramming idle SIMDs on the fly as additional rasterizers to further improve fill and small primitive rates.

## 1.5.3 Command input and Real time rendering

There are two independent Command Units - one servicing the GP stream (for 3D and general 2D commands), the other servicing an Isochronous or 'real time' stream. Both command units manage the Circular Buffers and Input DMA. The GP Command unit also manages Vertex Arrays.

The Isochronous command stream is used for processing images for display on the video unit's overlay channel. Commands sent through the isochronous stream can either be processed immediately (interrupting the user command streams) or scheduled to be processed between display of specific scanlines (for example, during vblank), thus allowing images to be updated without visible tearing.

The isochronous stream does not go through the normal T&L pipeline and rasteriser, but instead has a simple dedicated rectangle rasteriser unit. This supports operations such as scaling, filtering, rotation and colour-space conversion which can all be performed in the texture subsystem.

The resultant surface is normally associated with it's own channel in the video unit, where blending and colour-keying operations are used to combine it with the main image. If non-rectangular regions need to be overlaid, this can be done by defining the bounding rectangle and using colour-keying to key in the desired shape.

The Isochronous stream is initiated with a **TimeStamp** command which controls when the isochronous stream is switched in. This has three fields: *StartScanline*, *EndScanline* and *Head*. If both *StartScanline* and *EndScanline* are zero, then the isochronous stream is switched in immediately. Otherwise it waits until the scanline being output by the video unit on the selected head lies between the two values.

After sending the Timestamp, all other commands can be sent as per the usual command streams. After rendering the rectangle (and using **CacheControl** to flush and invalidate the caches), the isochronous stream is switched back out by sending a **ChangePort** command.

Unlike earlier graphics processors, P9/P10 command and control data (register updates, mode changes etc.) are largely independant of the pixel stream. This improves flexibility and bandwidth between units.

### 1.5.3.1 Real time support features

The real time channel includes diagnostic support features such as:

### 1.5.3.2 Circular Buffers for more efficient DMA transfers

The VPU supports a comprehensive set of DMA engines and uses Circular Buffer input stream handling to reduce Command DMA setup overhead and latencies. Input streams can be from host or on-card memory with two levels of nesting. Output DMA returns data to host or local memory, performs image uploads and state return.

New in P9/P10, the circular buffers transfer small packets of work rapidly without the delays and overhead of setting up DMA buffers, making escape calls to the O/S, monitoring buffer status etc.<sup>10</sup> Circular buffers process the command stream identically to input DMA and can even call DMA buffers.

Circular buffers are usually stored in local memory and mapped into the ICD. As commands and data are added to the circular buffers, chip-resident write pointer registers are updated automatically, without any O/S intervention. When the current circular buffer goes empty the hardware automatically searches the pool of 16 circular buffers for more work and instigates a context switch if necessary.

### 1.5.3.3 Compact Vertex Arrays and Vertex Caching for Indexed Arrays

P9/P10 offers a compact and flexible vertex array strategy to support both OpenGL and DX8/DX9. An array element can hold up to 16 parameters, stored consecutively in memory or held in arrays. Vertex elements can be accessed in sequence or using array indices. As a further enhancement, the most recent 16 array indices are cached for comparison with the current index to check for vertex meshing and avoid duplicate vertex data, which in turn allows substantial savings in memory reads and Shader processing.

### 1.5.3.4 Load Smoothing

Pipeline buffer depths are carefully modelled and simulated for optimum FIFO depth on both P9 and P10. P9 also introduces two software-controlled FIFOs, one with depth-first filtering. This delivers additional load smoothing and performance efficiencies at key points in the pipeline by spilling extra buffer data into cache memory.

## 1.6 Flexible memory implementation

P10 memory architecture allows unusual flexibility in adapting performance to specific applications and markets. There are two independent memory controllers and groups of tiles alternate between controllers. This is more efficient than a single 256-bit or 128-bit controller and allows half-width bus configurations to suit cost/performance part selection tradeoffs.

P10 uses a 256-bit interface in two 128-bit controllers with replicated address and control lines capable of handling an 8x8x8 tile in one cache line. P9 uses a 128-bit bus in two 64-bit controllers, or one clock for two 4x4x8 tiles. Each 64-bit controller can operate as a single TQFP or CSP interface.

Memory parts already in use include 512Kx4x32, 1Mbx4x32, 2Mbx4x16 and 4Mbx4x16, to a maximum local memory size of 512Mb (P10) or 256Mb (P9). The 8Mbx16 and 16Mbx16 parts are the same as conventional PC SDRAM DDRs. Both AP8 and AP10<sup>11</sup> are supported.

### 1.6.1 Virtual Memory

P10 continues 3Dlabs' industry-leading virtual memory design – up to 16GB address space on P10 or 4GB on P9. With AGP4X and 256 or 512MB of onboard DDR memories, P10 in particular allows ultra-high resolution 9 Mpixel displays without cramping textures or full-scene antialiasing using page-fault DMA to incrementally access host memory, with current data cached in on-board memory.

P10 provides the basic tools to implement a memory management system, including a page table to map logical to physical addresses and determine the validity of pages; a page fault interrupt and a dedicated DMA controller to facilitate the transfer of pages between system memory and graphics memory under software control.

The page table mechanism allows a level of indirection which can be used to improve the efficiency of non-contiguous physical address use and to progressively move large database subsets such as Navaid maps or DTED terrain representations through the current viewport. Virtual memory allows the physical memory to be treated as a fast cache, which can be used in combination with host memory and even host disk space to support memory requirements much greater than the onboard cache size. Page faulting includes optimisations for page size and data type.

For applications which would not benefit from this kind of memory management, for example video port input to a texture map, it is also possible to disable page faulting while continuing to translate physical to logical addresses and vice versa under host control.

<sup>10</sup> Because DMA transfers take time to initiate they are normally optimized for large bursts of data.. This can result in graphics system latency while work accumulates in the DMA buffer waiting to trigger a burst

<sup>11</sup> AP10 with 9-column addressing – CAS must be contiguous.

## 1.6.2 Physical Characteristics

Because of the radical redesign many earlier subsystems and registers became irrelevant. This allowed designers to remove up to 40% of the total code lines - a substantial reduction in gate count and chip complexity which, together with the 15 Micron wafer process, delivers a small, clean design with significant efficiency improvements.

### 1.6.2.1 High-reliability, high-yield package

P10 and P9 use the well-known ASE HSBGA package<sup>12</sup> enhanced with a metal heat slug and thermal balls to improve the thermal path to air and chassis. This wire-bond package offers proven reliability and yields for Quality-sensitive application environments together with excellent thermal characteristics. Board-level MTBFs are typically estimated to range from 18.27 years (P9 *VP560*) to 18.45 years (P10 *VP970*).<sup>13</sup> Package and wafer Quality Audit data is available to our Embedded Support development partners on request.

## 1.7 I/O and Bus Support

I/O interfacing includes a full range of on- and off-board devices:

- Analogue VGA
- Dual and Stereo heads
- DVI-I single link DFP and TV Encoder
- TV Out
- VIP2
- DVO
- I<sup>2</sup>C bus

P9/P10 fully support Intel's AGP 4X Accelerated Graphics Port standard, including:

- AGP4X, AGP1X, AGP2X
- 33/66MHz PCI<sup>14</sup>
- DMA and execute mode support, Sideband addressing
- 3.3v/1.5v tolerant

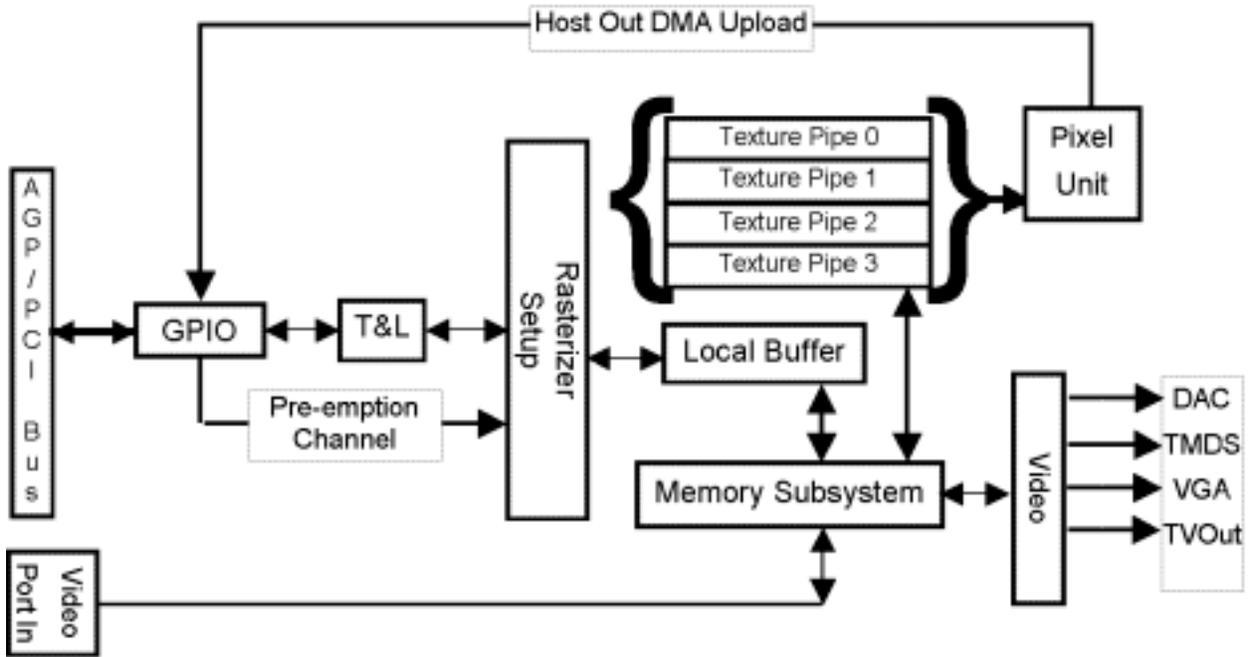
---

<sup>12</sup> P9 is a 644-ball thermally-enhanced HSBGA 31mm<sup>2</sup> package with 100 thermal balls. P10 is an 820-ball thermally-enhanced 37.5mm<sup>2</sup> HSBGA package

<sup>13</sup> Based on nominal clock speeds and thermal solutions implemented on those products. Reduced clock speeds and junction temperatures can increase life expectancy significantly.

<sup>14</sup> 66MHz PCI supports AGP timings – it is technically non-compliant with some PCI-66 timing specifications.

### 1.8 Chip Level Block Diagram (P10)



## 2

## Core Feature Overview

## 2.1 3D Graphics

P9 and P10 continue 3Dlabs' tradition of offering innovative, fully-featured and powerful geometry, lighting and rendering capabilities.

<b>Supported Function</b>	<b>Description</b>
Full primitive support	Full primitive support: triangle lists, fans and strips. Line lists and strips. Point lists. All either aliased or anti-aliased.
Efficient processing of small primitives	Integrated set-up, backface cull calculation, low latency
High fill rate	Wide data paths, high performance memory
Programmable Shaders, programmable texture co-ordinate and pixel units.	64 x 32bit Floating point texture coordinate processors, 64 x 32bit integer shader processors.
<b>Textures</b>	
Efficient texture storage	Fully flexible formats, internal 256 entry LUT
AGP textures	Textures directly from AGP memory
Dual/multi texture	Single-pass multi-textures, up to 8 textures per primitive
3D textures	3D volumetric textures; trilinear, anisotropic filtered, bump, cube and displacement maps; tessellation
High quality rendering	Sub-pixel and sub-texel accurate
High quality textures	Accurate perspective correction and trilinear filtering with per pixel MIP-Mapping with true level of detail calculation.
<b>Lighting/Optical</b>	
High quality lighting	Interpolated diffuse and specular components
Extremely realistic special effects	Interpolated colored fog, fog table and depth-cueing
Translucent objects and sprites	Blending/transparency on any primitive. Full dual texture blending. Interpolated alpha with direct support for all DirectX 6, 7 and OpenGL blend modes
High quality texture cut-outs	Color key with bilinear filter does not leave edge effects
Anti-aliasing	Edge anti-aliasing for zoomed sprites, full-scene T-buffer anti-aliasing
Fast hidden surface elimination	Depth (Z) buffering and non-linear Depth (Z) buffering. GID test for per pixel window clipping
Fast shadow, fog and transparency effects	Area stippling: vertex rendering with fog and texture at 106Mvertices/sec.
Integrated Geometry and Lighting	6 local lights at 20Mvertices/sec.
High quality output at any color depth	Dithering, programmable pixel formats
Fast sprite handling	Color key, scale, stretch, rotate, mirror
Seamless integration of video and 3D	Color key with depth test and perspective correction
Minimize update area, target selection	Hardware extent checking and picking

Improved image quality at lower resolutions	Full screen sort independent anti-aliasing
Use of rendered images as textures	Unified memory read and write to any buffer
Full range of double buffer techniques	Full screen flip, fast BLT, stereo buffers
Virtual texture map management	All memory is virtual/logical planar tiles, with cache-based page swapping.

**Table 2.1 3D Hardware Function Descriptions**

## 2.2 2D Graphics

<b>Supported Function</b>	<b>Description</b>
Full primitive support	Points, lines, spans, rectangles, polygons
Efficient processing of small primitives	Integrated set-up calculation, low latency, low-overhead circular DMA
Window clip	Hardware rectangle clipping
High speed color brushes	Internal pattern RAM
High speed monochrome brushes	Internal stipple table
Raster operations	Logic op unit
Fast BLTS	512 bit internal data path
Fast upload and download	Run-length encoded data
High speed monochrome download	
Flexible font caching support	Byte aligned monochrome bitmaps in local memory
Color translation	Through internal LUT
High speed stretch BLT	Using texture operations
Overlays	Per-pixel main image/overlay selection with color key and alpha blending
Statistic collection	Via dedicated StatisticMode register
Border color	Standard
Context save and restore	Cache-based context switch typically 3.5 $\mu$ s

**Table 2.2 2D Hardware Function Descriptions**

## 2.3 MPEG2

<b>Supported Function</b>	<b>Description</b>
MPEG motion compensation	Motion compensation calculations performed in hardware: user-programmable DXVA
Support for software decoders	DMA from system or write directly to local memory
High speed color space conversion	
Flexible YUV data formats	4:4:4, 4:2:2, 4:1:1 as standard and user-programmable additions.
Fast arbitrary stretch/shrink with filter	Bilinear filter at any zoom/shrink factor
Full featured video effects	Scale, shrink, stretch, rotate, mirror

**Table 2.3 MPEG2 Functions**

# 3

## Architectural Characteristics

The P10 and P9 architecture family consists of an integrated geometry and rasterization pipeline with unique features, capabilities and enhancements.

### 3.1 Host Interfaces - AGP/PCI

The Bus Interface design includes a PCI Target, PCI Master, AGP Master, PCI Configuration Space registers, local Control and Status registers, and a DMA Arbiter to handle bus master requests from the various controllers within the P10 device. The interface conforms to the *PCI Local Bus Specification* Revision 2.2. and AGP Interface Specification Revision 2.0. Dual signal voltages (1v5 and 3v3) are supported.

#### 3.1.1 PCI Interface

P10 is fully PCI 33 compliant (and also supports a non-compliant PCI 66, since PCI66 control signals may be up to 3 ns later than AGP 66 or PCI 33).

##### 3.1.1.1 PCI Target features

- PCI Config Space transactions
- PCI Memory Space transactions
- PCI Fast Writes (2X and 4X)
- PCI I/O Space transactions
- VGA palette write snooping
- 32-bit and 64-bit addressing (dual address cycles)
- PCI multi-function operation

##### 3.1.1.2 PCI Master features

- PCI Memory Space transactions
- 32-bit read and write data transfers
- 32-bit and 64-bit addressing (dual address cycles)

#### 3.1.2 AGPBus

AGP 4X is Intel's high performance, component level interconnect targeted at 3D display applications, which uses a 66MHz PCI specification as an operational baseline and provides significant performance extensions to the PCI specification.

Implementing these features enables P10 to achieve better than 1 GByte per second bandwidth from the host for instructions, textures and video data (limited by the host system throughput).

The add-in slot for AGP uses a connector body which is not compatible with the PCI connector. Boards designed for use in an AGP slot are not mechanically interchangeable with PCI boards. P10 supports AGP2x, AGP4x and PCI at signal voltages from 1.5vdc to 3.3vdc.<sup>15</sup>

##### 3.1.2.1 AGP Master features

- AGP low-priority Read transactions
- AGP low-priority Write transactions

---

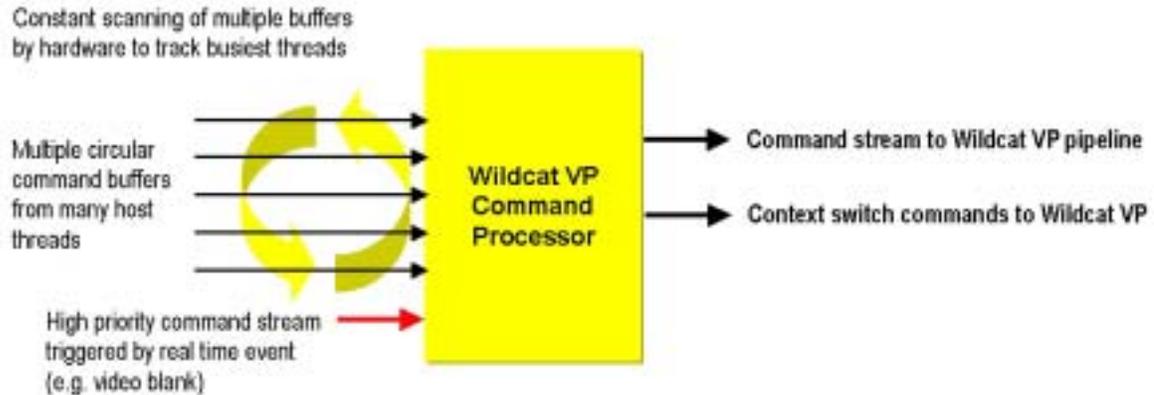
<sup>15</sup> Legacy 5vdc PCI logic may severely damage the chip.

- AGP Fence and Flush transactions
- Operation at 1X, 2X, and 4X data rates
- Sideband and pipe operation
- 48-bit addressing using sideband
- 64-bit addressing using pipe and dual address cycles

## 3.2 Transform and Lighting System

### **Command Processor**

The first thing in the pipeline is the 'Command Processor'. This unique 3Dlabs feature allows P9/P10 to be the first genuinely multithreaded family of 3D graphics devices for the PC.



Normally, switching from one host command stream to another is a cumbersome process for graphics chips which requires progress monitoring, flushing residual fragments, saving a context state, initialising a new graphics process etc. with all the associated host negotiation. P10, uniquely, uses a command processor to set up virtual processing for each thread and to arbitrate among them, i.e. hardware multithreading. The command stream is largely separate from the data stream.

### **Fast Context Switching**

Multithreading can only work efficiently with the Video Timing Generator if the time required to switch among context states is short enough. In addition, the configuration of the chip can be changed dynamically thanks to a context state cache. Together with an isochronous rectangle rasterizer P10 can respond to VTG event pre-emption in real time, typically as little as 3us for isochronous events or >20us for a full context switch.

This enhanced hardware context switching also ensures that the P10 family of devices are fully capable of supporting next-generation multithreading "Longhorn" Windows® implementations.

### **Transform and Lighting (T&L)**

The T&L pipeline features a number of design enhancements to improve flow control, vertex handling for advanced APIs, load smoothing, parallelism and application diversity.

P9/P10 include Transformation and Lighting, Graphics Core, Context Switching and I/O support for a wide range of hardware configurations, all of which are tightly integrated by the discrete core command, isochronous command and pixel streams.

T&L functionality includes vertex setup, transforms, lighting and culling. The pipeline uses a hybrid mixture of programmable and dedicated units which allow the chip to support both brute force highly-parallel fragment processing and complex multi-pass texture algorithms or effects. Precomputed convolutions, tessellations, any form of high order surface that can be represented by a mathematical model in hardware (e.g. NURBS, N-Patches, surface subdivision, vertex blending, static and dynamic displacement mapping) are all possible.

P10's high order surface implementation is unique – unlike most hardwired HOS tessellation solutions available to date which sit in front of the vertex processors and constrain throughput, P10's is integral and highly parallel.

The microcode instruction set, sequencer commands etc. are described in the extensive reference documentation. Assemblers/disassemblers and other microprogramming support tools are also available to developers.

### **Current Parameter Unit**

To avoid passing all 16 parameters for each vertex to the Vertex Shader Unit, P10 counts how many times each parameter has been sent and stops sending when each recipient vertex store holds a full complement.

Each parameter is typeless, so "VertexData" can actually be whatever the Vertex Shader defines it to be. The program running in the Vertex Shader Unit assigns meaning to the parameters, although conventional meanings are used in our documentation. This allows the use of the Vertex Shader for much more varied applications. Nor is it necessary to track vertex parameter values in software. A specific command dumps current values on request so that they appear in the Host Out FIFO.

### **Vertex Shading Unit**

The Vertex Shading Unit is implemented as an 8 element SIMD array, with each element (Virtual Processor) working on a separate vertex. The floating point ALU in each VP is a scalar multiplier accumulator which also supports multi cycle vector instructions. Each processor in the array acts as a mini-DSP core, with a 32-bit RISC instruction set, mathematical functions (*Move, Add, Mul, MAdd, Min, Max, IntFloat, Fract, Trunc, Dot, Div, RSqrt, Log, Clipping*), registers and register counters, temporary storage, and program storage.

The flow control of the processor array includes conditional jumps, subroutines and loops, which is a superset of DirectX9 vertex flow control. The chip is capable of loading a program and context-switching it to run it multiple times<sup>16</sup>. Programmability also allows the use of up to 200 lights and multi-pass operations. Because the vertex array can access the framebuffer it can store intermediate results, so operations too large to fit in the caches can still be processed.

## **3.3 FIFO and Memory Interface Enhancements**

FIFO placements in the Geometry pipeline have been optimised for depth and width, including the use of strategically placed caches to improve parallelism and clock independence in the memory interface.

P9/P10 memory is cache-based and all data types are stored as 8bit per pixel 'stackable' planar tiles<sup>17</sup>. All memory access is logical/virtual and page faults cause CPU-like page swaps.

Memory is preferably 256 bit wide DDR devices running at up to 266MHz. From 32MB to 256MB of x32 devices are supported, or alternatively up to 512MB of x16 devices.<sup>18</sup> SDR devices are not supported.

There are two independent 128-bit controllers (64-bit in P9) which hold alternating groups of tiles. Memory is divided into regions corresponding to the internal banks of a DDR device. Local memory is used to store color, depth, stencil, and texture data. These are largely interchangeable depending on the microcode application context.

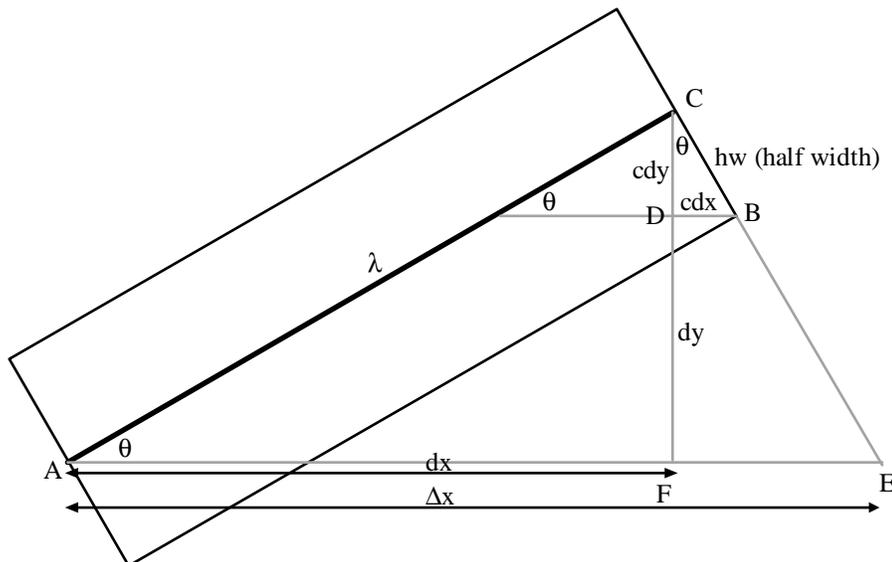
### **3.3.1 Primitive Setup system**

The Primitive Setup system takes coordinates, colors, texture coordinates etc. per vertex and predigests them for rasterization. This includes calculating triangle areas, splitting stippled lines into line segments, converting lines into quads, points into screen-aligned areas and windows-relative coordinates into fixed point screen coordinates. Finally, it calculates x and y gradients and depth gradients for all primitives and supports Run Length decoding for downloads.

<sup>16</sup> OpenGL2.0 prototype does not support this, but it is planned for the first release version. The aim is to take a 3MB Renderman shader (similar to the size used for 'Toy Story') and have it compiled and running on P10 hardware.

<sup>17</sup> 8x8 for P10, 4x4 for P9.

<sup>18</sup> The additional address lines can affect performance with x16 memories.



**Figure 3.3-1 Antialiased Line Construction**

Although the Primitive Setup functionality is not new, it must be able to provide a smooth data path from the T&L system to the Rasterizer system. P10 uses robust FIFO buffering, but P9 goes a step further, by introducing dynamic cacheing at both ends. This can operate as a typical 32-bit hardware FIFO; as a on-demand extended FIFO (to allow T&L to continue even when the Rasterizer is busy); or as a two-pass Binning unit which avoids calculating color values for overwritten fragments.

## 3.4 Rasterizer

P10 contains both a general purpose Rasterizer and a dedicated Rectangle Rasterizer.

### 3.4.1 Video Operations Without Dedicated Hardware

The Isochronous channel Rectangle Rasterizer delivers a new set of high-speed graphics tools for time-critical video applications. Capable of being context-switched into the rasterisation pipeline in 700 cycles or less, this represents only  $\frac{1}{4}$  scanline at 200MHz.

A **Timestamp** command allows the Rectangle rasterizer to be switched in at a specified VTG# and scanline for a set number of lines even while the geometry rasterizer is rendering an individual primitive. This degree of control allows non-tear blitting, Microsoft GDI+ hot button application support and other time-critical video functions.

### 3.4.2 Geometry Rasterizer

The geometry rasterizer identifies primitive edge functions and produces culled, scissored, masked and clipped tiles in an order which minimizes memory page swapping. AA sampling uses 64 parallel fragment samples per cycle

When trigger conditions are met the rasterizer outputs Tiles which control the rest of the core. Each Tile holds coordinates and a tile mask. The tiles are always screen relative and are aligned to tile boundaries.

Antialiasing uses up to 16 sample points per pixel. The sample points are normally positioned at the center of the pixels<sup>19</sup>, but a user programmable table allows the subpixel sample points to be irregularly positioned so that any edge moving across a pixel will cover (or uncover) the sample points gradually. This emulates stochastic (or jittered) sampling and gives better antialiasing results as, in general, more intensity levels are used. Coverage can be accumulated as a percentage (OpenGL) or mask (T-buffer).

<sup>19</sup> D3D expects the sample point to be at the origin of the pixel and this is allowed for when the appropriate mode bit is set.

### 3.5 Cacheing Enhancements

The LB and Pixel caches hold up to 16 tiles at various depths to smooth latencies to the memory system and improve small primitive handling. The Pixel cache can also receive non-aligned source tiles, which are then tile-aligned at the Destination, and stack fonts in bit planes.

### 3.6 Routing, Depth and GID

Depth is normally tested before texture ops to reduce unnecessary processing, but the existing 3Dlabs Router facility has been optimised for P10 to preserve OpenGL sequencing requirements when Alpha Testing is enabled. Depths can be 16 to 32 bit int or float with plane equation evaluation in floating point.

The new GSD unit supports per pixel ownership and stencil testing and early exit testing at load rate. 16-bit Z test requires 2 cycles, or 32 fragments per cycle.

### 3.7 Texture and Depth Processing

P10 introduces texture pipe parallelism, with 4 pipes<sup>20</sup> which can be enabled and disabled dynamically to meet texture load requirements. Any number of texture pipes can be supported, and a texture switch distributes tiles or small primitives to individual pipes according to round-robin arbitration results. At the far end, a MUX collects and re-orders the output for delivery to the Pixel Unit.

Sixty-four 32-bit SIMD processors drive Texture Coordinate routines capable of massive parallelism over a wide range of functionality, particularly when using the powerful native microcode flow-control features:

- **Perspectively correct plane equation evaluation**
- **LOD calculation (1D, 2D, 3D)**  
Max or Pythagoras
- **Feedback**  
Any filtered texture data can be feed back into the texture coordinate calculations (not just for bump mapping)  
Image download for colour mapping (24 bit lookups, piece wise linear interpolation)
- **Pass Through**  
Any calculated value can be passed directly to the Shading Unit without causing a texture lookup first  
Perspectively correct colour interpolation  
Wider dynamic range fog  
Phong shading
- **Cube and Bump Mapping**  
Per fragment matrix generation  
DX7 habits (e.g. bump env. mapping)  
High order filters and procedural textures supported

Up to 8 simultaneous textures can be supported in each pass with current APIs – more with APIs capable of using the additional feature set such as OpenGL 2.0

Texture pipelines use hardwired subsystems where these will be more efficient. Plane equation parameters and popular filters (bilinear and trilinear MIPmap) are hand-polished for efficiency. Other less common filtering schemes (e.g. non-standard Anisotropic) can be programmed, as can virtually any filter that can be expressed as a shader (wavelets, ray casting into volumetric textures etc.)

Texture programmability can also be used to deliver multi-tap Video filters such as DXVA acceleration including MPEG decoding, motion compensation, video scaling etc.

#### 3.7.1 Texture Indexing and Filtering

The texture pipes contain specialised high-efficiency fixed function units to perform these functions on up to 8 texels per pass. These include calculating interpolation coefficients for bi- and tri-linear filtering, coordinate wrapping, LOD clamping, cacheing etc. Trilinear filtering on 4-color components requires one cycle.

### 3.8 Shading Unit

This programmable unit uses a 4x4 SIMD Array to apply Gouraud and flat shades (16 fragments in 4 cycles), texture combinations, specular highlights, fog, and YUV conversion to 4x4 subtiles.

---

<sup>20</sup> 2 pipes on P9

Multi-pass programmes include program sequencing using specialised flow-control facilities:

- **32 plane equations**  
Not just limited to texture coordinates but perspective correct fog, etc.
- **32 global registers**  
Holds bias values, matrices  
**128 instructions**  
Loops, subroutines, conditional jumps, watchdog timer, programme sequencing  
Floating point ALU
- **8-bit ALU**

The sequencer's main role is to run a program when all the data for a subtile has been received. Program execution is deferred until the associated texture data has been received. One of four programs are run depending on the prog field accompanying the tile's data and the Texture Coordinate Unit can cause multiple different programs to run for a subtile. These combine to produce capabilities well beyond those offered in DX8 or 9.

### 3.9 Pixel Processing

The Pixel Unit combines each primitive's color from the Shading Unit with the framebuffer contents via alpha blending and/or logical ops, formatting the colour and finally updating the framebuffer. This unit replaces the functions previously carried out by the following units in earlier rasteriser chips:

- Alpha Blend Unit
- Logical Ops Unit
- Dither Unit

It does all the clearing and blitting of pixel data in the local buffer using a programmable SIMD array. The array holds 4x4 fragment processors to match up with the tile size. The basic unit of operation is a byte or a colour component, hence it takes multiple cycles to evaluate a four component colour value. For example it will do common alpha blending on 16 fragments in 5 cycles, or 3 fragments per cycle.

The operation of the SIMD array and sequencer are controlled by a short (<128 instructions) microcode program. This program is tailored to the exact sequence of operations needed to implement the current rendering state and there is an automatic watchdog mechanism to prevent an erroneous program from locking up the chip.

There is storage for 16 fragment data registers (32 bits wide) and 32 global byte registers. Each colour component takes one global register. The fragment data registers provide 4 bytes of unique data (loaded from the message stream) per fragment processor and they will hold (at different times) dither matrix values, coverage values, downloaded image data. The global registers, on the other hand, hold values common to all fragment processors for use during processing. This could be a pixel write mask, a foreground, a background colour, etc..

The power of the Pixel Unit is enhanced tremendously by its multi-pass mode of operation. This allows the same tile to be processed many times, potentially with different programs and different pixel data read or written to memory (the shading data remains constant for all passes). This allows multi-buffer operations, accumulation buffer processing (i.e. 64 bit pixels), convolution, etc. to be easily done. Each pass is launched and controlled by the Pixel Address Unit. Normally the first, middle and last passes would each have their own programs.

## 4

## Video Unit and RAMDAC

## 4.1 Overview

P9 and P10 support both digital and analog displays, with the capacity to handle very high resolution monitors, fast digital monitors, dual heads with and without stereo and interlacing, mixed digital/analog heads and a full VIP2 video input port. Specialised interfaces for genlock and video editing are also supported. P9/P10 use high-speed 10-bit 350MHz DACs or the 260MHz Digital Output port for Video Output.

P9/P10 support streamed digital video output designed to work with common PAL/NTSC encoders and flat panel controllers. DVO can be single- or double-edged, 12 or 24 bits wide depending on how the two channels are deployed. RGB 888 and other formats are supported, as is RGBA using 24bit double-edge.

P9/P10 support typical screen resolutions up to 1600x1200 with refresh rates of 96Hz or 1920x1080 with refresh rates of 90Hz, or 2048x1536 at 60Hz. Packed pixel formats with color depths of 8, 16, 24, 32 and 40 bits per pixel are supported. Both parts have dot-clock phase locked loops (PLLs) and triple 8-bit D/A converters. The RAMDAC contains a 64x64x2 bit cursor array to support a 2, 4, or 16 color hardware cursor with cursor shapes cache.

Stereo is supported on the main and overlay channels (left and right buffers). Dual head capability is built-in with two discrete video channels and Genlock to an external sync source (Hsync or Vsync<sup>21</sup>). An external clock can be used as an external reference source for the PLLs.

## 4.1.1 Pixel Formats

P10's planar tile structure and video bus support up to 64bpp in a wide variety of formats. Each 8x8 pixel screen-aligned tile is handled in one-byte increments up to 8 bytes per tile. Each memory access returns one tile, with multiple reads for 16, 32, 40 etc. bit depths. Each tile can be defined as a color, texture, depth or alpha as required, so an unusually wide range of pixel formats can be supported. 32 bit colour and 565 colour formats are handled directly, other formats such 555, 4444, etc. are configured In the Pixel Unit.

The table shows the bit positions in the input data used to represent different color components.

Format	Name	RGB	Bits/pixel	R	G	B	A	Index
0	CI8	-	8	-	-	-	-	0-7
1	3:3:2	0	8	0-2	3-5	6-7	-	-
1	3:3:2	1	8	5-7	2-4	0-1	-	-
2	5:5:5:1	0	16	0-4	5-9	10-14	15	-
2	5:5:5:1	1	16	10-14	5-9	0-4	15	-
3	5:6:5	0	16	0-4	5-10	11-15	-	-
3	5:6:5	1	16	11-15	5-10	0-4	-	-
4	8:8:8:8	0	32	0-7	8-15	16-23	24-31	-
4	8:8:8:8	1	32	16-23	8-15	0-7	24-31	-
5	10:10:10:2	0	32	0-9	10-19	20-29	30-31	-
5	10:10:10:2	1	32	20-29	10-19	0-9	30-31	-
6	CI4	-	4	-	-	-	-	0-3, 4-7

Table 3.1.1 Pixel formats

The pixel size is independent of the color format, so it is possible to have an 8 bit pixel with a 32 bit stride. The bitmask format is different because it uses 4 bits per pixel regardless of pixel size; this format must be used with a one byte pixel size. The pipeline maintains 16 bits per component, but various operations use

<sup>21</sup> On P9, HSync is derived from Vsync rather than fully-independent.

different numbers of bits. Color key uses 8 bits, blends use 8 bits, LUTs use 8 bits for input but output 10 bits.

#### 4.1.1.1 Pixel Channel Key

Each pixel to be displayed may have contributions from any of the four channels. The pixel color is determined by working through the channels in the order underlay, main, overlay, cursor:

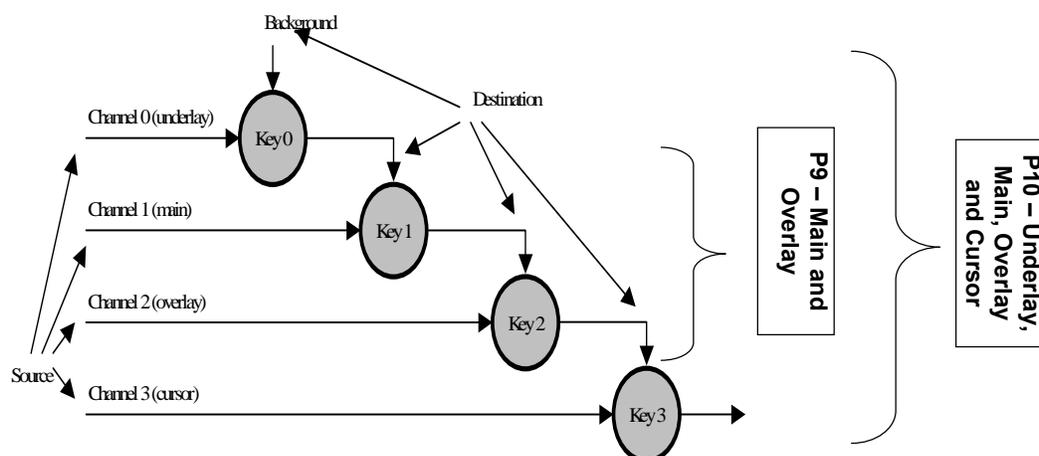


Figure 3.1 Pixel Channel Keys

On P9, only the main and overlay channels are supported on both heads. An interleave mode is available which allows a third channel to be 'stolen' from one head to support an additional cursor or overlay channel on the other.

#### 4.1.1.2 LUTs

Two lookup tables are used to remap the pixel color.<sup>22</sup> Typical applications include using one table to dereference index data while another gamma-corrects RGB data, or to support two different gammas (perhaps one for video, the other for 3D).

### 4.1.2 Scaling

P9/P10 handle general video overlay scaling (where the data needs to be up- or down-converted with high quality scaling) through the graphics processor. The video sub-system is also able to upscale in X and Y by a limited amount which is suitable for displaying small framebuffers on fixed resolution displays.

For example, in a two-head system, one head may be used to drive a projector with a fixed resolution of 800x600, while the other head displays the same data on a flat panel display at 1024x768. To get good quality projection the framebuffer is set to 800x600, but this will not fill the flat panel display so hardware scaling can be used to increase the effective size of the framebuffer.

### 4.1.3 Synchronization and Genlock

There are two lock bits which may be used to synchronize different channels within a head, or different heads. The lock registers hold a mask of which channels take part in the lock, and there are two lock registers per head. All heads have access to all lock pins so they can be used to synchronize two heads in the same chip; the pins can also be shared by separate chips.

Both P9 and P10 support Genlock to an external source. P10 provides two pins for external Vsync and HSync, P9 provides one Vsync pin and the ability to adjust HSync relative to Vsync line end events.

### 4.1.4 Clocks and PLLs

Clock/PLL configurations are highly flexible and support external clock reference for e.g. genlock. There is one clock for the graphics processor (KClk), one for the memory clock (MClk) and one for each display head (DCIk0..DCIkN).

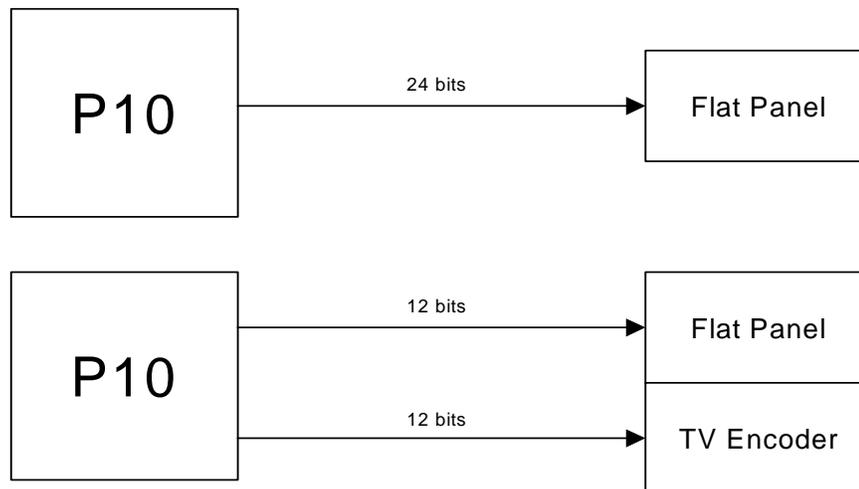
<sup>22</sup> On P9, one LUT is available

There are 4 PLLs which can be individually programmed to different frequencies; PLL0 has 4 sets of registers to allow switching between different frequencies (required for VGA). The PLLs can use the internal 14MHz oscillator as a reference clock, or an external source for genlocking. Each clock specifies its source which can be the PCI clock or one of the PLLs; any PLL can drive any clock.

One of the standard sources (PCIk or the PLLs) can be output to a pin; the frequency of this clock can be divided by 1, 2, or 4, and optionally inverted.

### 4.1.5 Digital Port Control

Both display heads share a single digital port which can be used to output or input digital video. Input video is only used when 2 P10s share the same display (other types of video input should use the video input port). Output video may be used to drive a flat panel controller or a TV encoder.



**Figure 3.2 Digital Port Configuration**

There are 24 data pins to which devices may be attached. The way the digital port pins are configured depends on how external devices have been connected. Some examples are:

Usage	Mode <sup>23</sup>	C 0 <sup>24</sup>	C 1 <sup>25</sup>	DE <sup>26</sup>	M 0 <sup>27</sup>	M 1 <sup>28</sup>	Notes
Single flat panel	Out0	X	X	No	SinglePixel	Off	Single edge 24 bit data
Fast flat panel	Out 0	X	X	Yes	DoublePixel	Off	Dual edge 24 bit data.
Dual flat panel	Shared	Out	Out	Yes	SinglePixel	Single Pixel	Dual edge 12 bit data (x2)
Video editing	Out0	X	X	Yes	AlphaPixel	Off	Dual edge 48 bit data.

## 4.2 Software Drivers

3Dlabs have extensive experience and a proven track record in delivering high performance, high quality, ready-to-ship WHQL certified software drivers that extract the maximum performance from both the Miranda P10 3D processor and the entire system.

### 2D

- Windows NT version 4

<sup>23</sup> Mode = VideoDigitalPortControl.Mode

<sup>24</sup> C0 = VideoDigitalPortControl.Channel0

<sup>25</sup> C1 = VideoDigitalPortControl.Channel1

<sup>26</sup> DE = VideoDigitalPortControl.DoubleEdge

<sup>27</sup> M0 = VideoDPMMode.Mode (head 0)

<sup>28</sup> M1 = VideoDPMMode.Mode (head 1)

- Windows 2000
- Windows ME

Other software drivers may be made available depending on current market requirements.

### 3D

P10 has been designed to accelerate the key consumer focused 3D APIs and drivers. 3Dlabs' processors have historically been the reference port for many 3D drivers including Microsoft's OpenGL DDK.

P10 high performance 3D drivers support:

- Direct3D 7 and 8
- OpenGL 1.3
- OpenGL 2.0 Beta
- Autodesk's Heidi for 3D Studio MAX support, including all D3D and OpenGL Depth and Stencil modes.

## 4.2.1 ROM support and SVGA BIOS

P9/P10 support Flash ROM. The ROM stores code needed for device-specific initialization and the SVGA BIOS. The SVGA BIOS is based on the proven, industry-standard Phoenix Technologies BIOS core. The on-chip SVGA unit is register-level compatible with standard VGA devices and requires no software emulation. It natively supports all standard VGA modes and many VESA VBE extended modes.

## 4.2.2 Display Resolutions

The following display resolutions are supported:

**Table 4-2 VESA and GP Graphics Modes**

Pixels	Colors	Windowed	Linear	Analog Refresh (Hz)	Analog Refresh (Dual head)	Digital Refresh (Hz)	Supported in SVGA	Supported in GP
640x400	256	✓	✓				✓	✓
640x480	256	✓	✓	200		60/75/85	✓	✓
800x600	256	✓	✓	200		60/75/85	X	✓
1024x768	256	✓	✓	200		60/75/85	X	✓
1280x1024	256	✓	✓	120		60/75/85	X	✓
320x200	32K (5:5:5:1)	✓	✓				X	✓
320x200	64K (5:6:5)	✓	✓				X	✓
320x200	16.8M (8:8:8)	✓	✓				X	✓
640x480	32K (5:5:5:1)	✓	✓	200		60/75/85	X	✓
640x480	64K (5:6:5)	✓	✓	200		60/75/85	X	✓
640x480	16.8M (8:8:8)	✓	✓	200		60/75/85	X	✓
800x600	32K (5:5:5:1)	✓	✓	200		60/75/85	X	✓
800x600	64K (5:6:5)	✓	✓	200		60/75/85	X	✓
800x600	16.8M (8:8:8)	✓	✓	200		60/75/85	X	✓
1024x768	32K (5:5:5:1)	✓	✓	200		60/75/85	X	✓
1024x768	64K (5:6:5)	✓	✓	200		60/75/85	X	✓
1024x768	16.8M (8:8:8)	✓	✓	200		60/75/85	X	✓
1280x960	16.8M (8:8:8)	✓	✓	120		60/75/85	X	✓
1280x1024	32K (5:5:5:1)	✓	✓	120		60/75/85	X	✓
1280x1024	64K (5:6:5)	✓	✓	120		60/75/85	X	✓
1280x1024	16.8M (8:8:8)	✓	✓	120		60/75/85	X	✓
1600x1200	16.8M (8:8:8)	✓	✓	120		60	X	✓
1920x1200		✓	✓	100		60	X	✓
1920x1440		✓	✓	90			X	✓
2048x1536		✓	✓	80			X	✓
2048x2048		✓	✓	60			X	✓

The following VESA VBE text modes are supportable in the SVGA:

**Table 4-3 VESA VBE Text Modes**

Mode (hex)	Characters (col/row)
0x108	80x60
0x109	132x25
0x10A	132x43
0x10B	132x50
0x10C	132x60

P9/P10 allow VESA bank switching to be done through the bypass to enable additional VESA mode support.

### 4.2.3 Video Overlay

The video overlay is used to display incoming video data on screen. The overlay selection is based on a transparent color, the overlay key, which can be any RGB color or alpha value. Optionally, the overlay can be blended with the main image by using a 2-bit blend factor. A filter process supports zooming and shrinking at any rate. It combines four pixels into one by using bilinear filtering to achieve best results. Furthermore the filtered output is optionally converted from YUV to RGB color space format.

## 4.3 Video Input Port (VIP)

The Video Port Unit implements a VESA Video Interface Port (VIP) Version 2 Level II video port master which supports:

- ITU-R BT.656 video stream – 8-bit @ 27MHz
- VIP1.1 video port – 8-bit @ 27Mhz
- VIP2 Level I video port – 8-bit @ 75MHz
- VIP2 Level II video port – 16-bit @ 75MHz
- Proprietary VIP2 video port – 16-bit @ 150MHz

The unit is controlled by PCI slave register writes and reads, which are transported through the PCI slave write (PciVpuWr) and PCI slave read (VpuPciRd) FIFOs respectively.

The P Clock half of the unit – VPUPClk – maintains the registers. Control registers drive the P clock to I clock signals. Status registers are driven by the I clock to P clock signals. The I Clock half of the unit – VPUIClk – transports the video stream to the memory controller.

### 4.3.1 Video Stream Formats

Active video is formatted as 4:2:2 YCrCb, and is transmitted as a byte stream of Cb-Y-Cr-Y. The conversion of 4:2:2 YCrCb to RGB is described in ITU-R BT.601. The VIP2 Level I video port provides 8-bit samples, and the VIP2 Level II video port provides 16-bit samples. In 8-bit video, each byte is sent across VID[7:0]. In 16-bit video, the first byte is sent across VID[7:0] and the second byte is sent across XVID[7:0], except control codes (SAV, EAV, and any ANC headers) which are always sent across VID[7:0]. The VPU transports the samples to the memory controller, 16 bits at a time, unmodified except as described below.

#### 4.3.1.1 Empty Cycles

In VIP2, skip data (“00”) during active video is used to mark an empty cycle. In 8-bit video, the “00” bytes appear on VID[7:0]. In 16-bit video, the “00” bytes appear on both VID[7:0] and XVID[7:0]. This is an extension to ITU-R BT.656.

In the port unit, empty cycles are optionally discarded. If the video stream is known to contain empty cycles, “00” bytes should be discarded. If the video stream conforms to ITU-R BT.656, or is known to contain out-of-range values, “00” bytes should be kept.

### 4.3.1.2 Fields and Frames

The VIP port supports both interlaced and non-interlaced frames. Frames are stored in individual buffers in memory. For non-interlaced video, the EAV Field (F) bit is ignored. For interlaced video, the EAV Field (F) bit is matched against a start field to determine the 1st field in the frame.

An interlaced video source can be stored as non-interlaced frames. This might be used to de-interlace video. The VPU can store frames in 1, 2 or 3 buffers per task. Host software provides up to 3 buffer addresses per task, and the VPU cycles through the buffers in turn. Triple buffering allows the VPU to cope with mismatched input and output frame rates.

# 5

## Power and Thermal Management

### 5.1 Power Consumption [P10]

P10 power consumption is a function of clock speeds and voltage. The sample range below assumes a constant MClk of 150/300 MHz, but further power reductions would be expected if both K and MClks were further reduced. The memory timings with DDR parts yield a clean window from 100/200 to 200/400MHz. KClk however can be reduced to 66MHz and possibly as low as 50MHz. Performance was stable running Viewperf when measured on a VP990Pro board at the frequencies shown for voltages down to 1.21vdc core.

Measured Power Consumption Range					
Core Voltage	Core Clock (KClk)		I/O W	Core W	Total
VDD 1.55vdc	K=240 MHz M=150/300 MHz	3D	2.47W (950mA)	18.64W (11.8A)	21.01W
		2D	1.82W (700mA)	17.22W (10.9A)	19.04W
VDD 1.4vdc (1.41 vdc)	K=100 MHz M=150/300 MHz	3D	1.15W (425mA)	7.68W (5.58A)	8.83W
		2D	1.03W (383mA)	7.05W (5.01A)	8.08W
VDD 1.3vdc (1.32 vdc)	K=100 MHz M=150/300 MHz	3D	900mW (333mA)	6.60W (5.00A)	7.50W
		2D	831mW (308mA)	6.00W (4.54A)	6.83W

**Table 5.1 P10 Power Consumption Range**

*Note: AGP/PCI, DAC and PLLs may not draw more than 1.0A per connector 'finger'  
VDD ratings are +/-10%.*

## 5.2 Power Consumption [P9]

P9 power consumption is a function of clock speeds and voltage. The sample range below assumes a constant MCik of 215/430 MHz, but further power reductions would be expected if both K and MCik were further reduced.

Measured Power Consumption Range					
Core Voltage	Core Clock (KClk)	I/O W	Core W	Total	
VDD 1.5vdc	K=270 MHz M=215 MHz	5.47W (2.19A)	13.05W (8.7A)	18.5W	
VDD 1.4vdc	K=100 MHz M=215 MHz	3D	2.16W (est.)	3.60W (2.60A)	5.76W (est.)
		2D	2.06W (est.)	3.38W (2.45A)	5.44W (est.)
VDD 1.3vdc	K=100 MHz M=215 MHz	1.92W	2.64W	4.56W	

**Table 5.2 P9 Power Consumption Range**

Note: AGP/PCI, DAC and PLLs may not draw more than 1.0A per connector 'finger' VDD ratings are +/-10%.

## 5.3 Power Management Features

Supported Function	Description
Clocks can be individually stopped	Separate clocks for: geometry processor, graphics processor, memory sub-system, video sub-system, video output and AGP
Automatic frequency reduction	Reduces average power consumption when idle
Memory power down mode	Low power while maintaining refresh and screen update
DPMS	Power management for monitors

**Table 5.3 Power Management Functions**

## 5.4 Thermal Performance

The P9 and P10 packages use thermal balls to improve the thermal path to the PCB. Although the ambient operational range depends on the chosen cooling solution and  $P_n$ , parts are qualified for 0°C to 125°C  $T_j(\max)$ .

820L HSBGA [P10]	Thermal Resistance				
	(deg.C/Watt) = $\theta_{ja}$			$\psi_{jt}$ (C/W)	$\theta_{jc}$ C/W
	0 m/s	1 m/s	2 m/s		
No heatsink, 4L PCB	10.3	9.2	7.9	1.2	2.1
No heatsink, 4L PCB with 2oz. Copper plane	9.9	8.8	7.5	1.1	2.0
Heatsink (37mm sq.x 6mm such as AAVID 373324)	9.3	6.7	5.4		
644L HSBGA [P9]	Thermal Resistance				
	(deg.C/Watt) = $\theta_{ja}$			$\psi_{jt}$ (C/W)	$\theta_{jc}$ C/W
	0 m/s	1 m/s	2 m/s		
No heatsink, 4L PCB	12.9	11.2	9.7	1.9	3.1

**Table 5-4 P9 / P10 Thermal Performance**



# 6

## Data sheet

### Texture Mapping

- True perspective correction
- Multiple texture engine (8+)
- Trilinear filtering with per-pixel MIP-mapping
- Palletized and RGB textures
- Bump Mapping, Convolutions, Displacement Mapping
- Transparency Maps
- Local texture buffer
- Specular, diffuse, ambient multiple lights
- Fast texture paging/loading
- AGP execute mode for remote texturing
- Color keying

### 3D Rendering

- Points, lines, triangles & bitmaps
- Gouraud and flat shading
- 8-, 16-, 24-, 32- and 40-bit RGB/A
- Depth (z), GID buffering
- Fogging & depth-cueing
- Alpha blending (flat and Gouraud)
- H/W full screen anti-aliasing (FSAA)
- Dithering
- Area stippling
- Stencil test and stencil buffer
- Scissors test and logic operations

### Display Features

- 8-, 16-, 24-, 32- and 40-bit RGB/A
- 8-bit color index
- Double and triple-buffering
- Hardware dithering
- Hardware pan
- Overlays

### Fast Video Playback

- MPEG2 playback acceleration
- YUV color space conversion
- Scaling and shrink (bilinear filtered)
- Dithering
- Color keying (blue-screen)
- Alpha overlay blending

### GUI Acceleration

- BitBit with ROPs
- Points, lines, polygons
- Fills and text primitives
- Fast linear framebuffer
- On chip SVGA
- Windows

### PCI/AGP Interface

- 32-bit glueless PCI V2.1
- 33/66 MHz PCI / 266MHz AGP 4X
- Dual 2.5/3.3VDC 4X and 2X compatible
- Target and master support
- DMA mastering
- 256 entry command FIFO
- Big-endian apertures on bus
- Interrupts

### Memory Architecture

- 128-bit DDRAM interface
- Single multi-function memory
- Optimal memory usage
- 8 to 256 Mbytes

### Display Resolutions

- 320x200 to 2048x2048
- Ergonomic refresh rates

### TV/Video Output

- 350 MHz RAMDAC interface
- LCD flat panel support
- 240MHz Digital Video output

### Power Management

- VESA DPMS
- VESA DDC support
- Separate clocks for all sub-systems
- Automatic frequency reduction when idle
- RAM power down mode

### HPBGA Package

- 2.5/3.3 V

### Driver Support

- Direct3D, DirectX and OpenGL
- Windows 95/98, Windows NT/Windows 2000, Windows ME.
- Heidi for 3D Studio MAX